Spring 2016, CS 583: Approximation Algorithms

Homework 2
Due: 3/2/2016

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.

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 We saw randomized rounding plus alternation for packing problems. The idea is also useful for covering problems. Recall the LP relaxation for Set Cover. There is a variable x_j for each set S_j in the given instance and the constraints ensure that $\sum_{j:i \in S_j} x_j \geq 1$ for each element $i \in \mathcal{U}$.

Given a feasible solution $\bar{x}$ define a new solution $\bar{y}$ where $y_j = \min\{1, \alpha \cdot x_i\}$ where α will be chosen as a parameter.. We then independently pick each S_j with probability y_j. The chosen sets may not form a set cover in that some elements may not be covered. To obtain a feasible set cover we do the following: for each uncovered element i we simply pick the cheapest set that contains it and add it to the solution. Clearly, the algorithm outputs a feasible set cover.

- Choosing α appropriately prove that the expected cost of the set cover output by this algorithm is $O(\ln n) \sum_j c_j x_j$.
- Improve the bound to $O(\ln d) \sum_j c_j x_j$ by choosing α appropriately. Here d is the maximum set size.

Problem 2 We saw a $\Delta(G)$-approximation for weighted maximum independent set problem (MIS) in a general graph. Obtain a fixed constant factor approximation for *weighted* MIS in planar graphs. *Hint:* Use the fact that a planar graph always has a vertex of degree at most five.

Problem 3 In the uniform-capacity Resource/Bandwidth Allocation Problem, the input is a path $P = \{v_1, v_2, \ldots v_n\}$, where v_i is adjacent to v_{i+1}; an integer capacity c; and a set of demand requests $\mathcal{R} = \{R_1, \ldots, R_m\}$. Each request R_h consists of a pair of vertices v_i, v_j, and an integer demand d_h; this is to be interpreted as a request for d_h units of capacity from v_i to v_j. Note that there can be multiple requests between the same pair of nodes. The goal is to find a largest subset of requests, $\mathcal{R}$, that can be satisfied simultaneously; that is, the total demand of satisfied requests going through any edge v_i, v_{i+1} should not exceed the capacity c.

(Note that when the path P is a single edge, this problem is equivalent to KNAPSACK.)

1. Assume $c = 1$ and all requests are for one unit of demand. In this case we are asking for the largest independent set in an interval graph. Consider the algorithm that orders the requests in increasing order of *length* and greedily selects them while maintaining feasibility. Show that this algorithm is a 1/2-approximation using the technique of dual-fitting. Write an LP and find a feasible dual to the LP and relate the solution output by the greedy algorithm to the dual value.

2. Consider the weighted version, where each request R_h also has a profit/weight p_h, and the goal is to find a maximum-profit set of requests that can be satisfied simulataneously. (Note that an optimal solution may have overlapping requests since the demands are now varying.) Write a Linear Program for this problem, and show that the LP has constant integrality gap:

   Hint 1: If you randomly round each request independently, with probability proportional to "how much" the request is selected by the LP, show that the expected profit of the integral "solution" is large, though the solution obtained may not be feasible.

   Hint 2: Scale down all probabilities by a constant factor (say 10), and round independently. Let S be the set of selected requests. Now, initialize set S' to be empty, and order requests in S by their left endpoint, from left to right. In this order, select $s \in S$ for S' if it can be added to S' without violating feasibility. Show that the probability a request $s \in S$ is selected for S' conditioned on it being in S is a constant. The analysis is similar to the scheme for k-sparse PIPs.

Problem 4 Multi-processor scheduling: given n jobs $J_1, \ldots, J_n$ with processing times $p_1, p_2, \ldots, p_n$ and m machines $M_1, M_2, \ldots, M_m$. For identical machines greedy list scheduling that orders the jobs in non-increasing sizes has an approximation ratio of 4/3. See Section 2.3 in the Shmoys-Williamson book.

Now consider the problem where the machines are not identical. Machines M_j has a speed s_j. Job J_i with processing time p_i takes p_i/s_j time to complete on machine M_j. Give a constant factor approximation for scheduling in this setting to minimize makespan (the maximum completion time). (Hint: consider jobs in decreasing sizes. Assuming $p_1 \geq p_2 \geq \ldots \geq p_n$ and $s_1 \geq s_2 \geq \ldots s_m$, show that $OPT \geq \max_{i \leq m}(\sum_{j \leq i} p_j / \sum_{j \leq i} s_j.)$

Problem 5 In the Generalized Assigment problem, you are given n jobs, and m machines/bins. For each job i and machine j, there is a size s_{ij} that job i occupies on machine

j. (Note that the s_{ij}s may be completely unrelated to each other.) A feasible assignment is one in which each jobs is assigned to some machine.

The *makespan* of an assignment is the maximum, over all machines i, of the total size (on i) of jobs assigned to it. Give a PTAS for the problem of minimizing makespan when the number of machines m is a constant. Use the following scheme.

- Guess all the "big" items and their assignments.
- Write an Linear Program for assigning the residual "small" items.
- Show that a basic feasible solution (a vertex solution) for the linear program has at most m fractionally assigned jobs.

Problem 6 In this problem, we solve Maximum Independent Set (MIS) in another family of graphs, the intersection graphs of disks in the Euclidean plane: Given a set of disks in the plane, construct a graph by creating a vertex for each disk, and connecting two vertices by an edge if the corresponding disks intersect. Give a PTAS for MIS problem in these graphs, assuming all disks have unit radius.

Hint 1: Consider a grid of lines spaced $\frac{1}{\epsilon}$ units apart. If no disks of an optimal solution intersect these grid lines, can you find an *exact* algorithm with running time polynomial in n for any fixed ϵ?

Hint 2: Consider a grid with *random* offset: Take a grid of lines spaced $\frac{1}{\epsilon}$ apart, such that the origin is at the intersection of a horizontal and vertical grid line. Pick a shift/offset L uniformly at random from $[0, \frac{1}{\epsilon})$, and shift the grid vertically and horizontally by an distance L. (Equivalently, consider the grid of spacing $\frac{1}{\epsilon}$ such that the point (L, L) is at the intersection of two grid lines.) What is the probability that a disk is intersected by a grid line? Can you give a deterministic approximation scheme?

Note: There is a PTAS for the problem, even if the disks are allowed to have different sizes. Do you see how to obtain a PTAS? For more information about geometric approximation, see the Chapter 11 in Vazirani book or Chapter 10 in Shmoys-Williamson book or the upcoming book of our own faculty member Prof. Har-Peled which is available on his website.