CS 49/149: Approximation Algorithms

Problem set 5. Due: 13th May, 6:59pm

General small print: Please submit all homework electronically in PDF format ideally typeset using LaTeX. You need to submit only the problems above the line. Please try to be concise – as a rule of thumb do not take more than 1 (LaTeX-ed) page for a solution. We highly encourage students to also do the problems below the line for a better understanding of the course material.

Topics in this HW: Iterative Rounding, Ellipsoid Algorithm, Separation Oracles

Problem 1. In this problem we look at the *multi-dimensional* knapsack problem. The input is a knapsack which has d-dimensions with capacities $(B_1, \ldots, B_d)$ on each dimension. We are also given n items; each item j has a d-dimensional "weight" $(W_{1j}, \ldots, W_{dj})$, and a profit p_j. The objective is to select a subset S of items with maximum profit which fit in the knapsack in every dimension. That is, for each $1 \le i \le d$, $\sum_{j \in S} W_{ij} \le B_i$.

Design and analyze a PTAS for the multi-dimensional knapsack problem for constant d. The answer to this problem should be a complete description and analysis. You may want to go down the route suggested below (and also for partial credit) .

1. Guess opt, the value of the optimum. Call an item j *big* if $p_j > \frac{\varepsilon}{d} \cdot \texttt{opt}$. Let J_B be the set of big items and let J_S be the set of small jobs. How many big items can there be in the optimum solution? How much time does it take to "guess" the set of big items in the optimum solution? (**1 point.**)
2. Suppose you know $I \subseteq J_B$, the set of big items in the optimum solution. We now need to figure out the subset of small jobs to be selected. Write an LP relaxation for the problem. (**1 point.**)
3. Let x be a basic feasible solution for this LP and let $F := \{j : 0 < x_j < 1\}$. How big can $|F|$ be? (**1 point.**)

Problem 2. Recall the spanning tree polytope we did in class.

$$\{x \in \mathbb{R}^E : \sum_{e \in E} x_e = |V| - 1; \quad \forall S \subseteq V,\ x(E[S]) \le |S| - 1, \quad \forall e \in E,\ 0 \le x_e \le 1\}$$

Design a *separation oracle* for the problem. That is, given x_e's on the edges, either prove that x is feasible or find a constraint that x doesn't satisfy. Your algorithm should run in polynomial time.

Problem 3. (This problem is equivalent to 2 normal problems and is worth 14 points)

Consider the following problem. We are given an *undirected* graph $G = (V, E)$. For each pair of vertices $i, j \in V \times V$, we are given a non-negative integer r_{ij}. Each edge $e \in E$ has a cost $c(e)$. The output needs to be a subgraph $H = (V, F)$ of G such that between any two vertices $i, j \in V \times V$, there exist r_{ij}-edge disjoint paths between i and j in H. The objective is to minimize the cost of the edges in H. Note that when all the r_{ij}'s are 1, this is precisely the minimum spanning tree problem.

a. (**2 points**) Consider the function $r : 2^V \to \mathbb{Z}_{\ge 0}$ defined as $r(S) := \max_{i \in S, j \notin S} r_{ij}$. Prove that for any $S, T \subseteq V$, we either have

$$(1) \quad r(S) + r(T) \le r(S \cup T) + r(S \cap T)$$

or,

$$(2)\quad r(S) + r(T) \leq r(S \setminus T) + r(T \setminus S)$$

b (**1 point**) Write an LP relaxation algorithm for the problem with x_e as a variable and a constraint for every $S \subseteq V$. Use the r-function defined above.

c. (**2 points**) Given any positive weights w_e on the edges of an *undirected* graph, and any two subsets $S, T \subseteq V$, prove that it satisfies

$$w(\delta S) + w(\delta T) \geq w(\delta(S \setminus T)) + w(\delta(T \setminus S))$$

d. Let x be any bfs of the LP relaxation you described in part (b).

   i (**3 points**) Consider the tight sets induced by x. Describe an uncrossing procedure to establish that there exists a basis which corresponds to laminar family of sets.

   ii (**4 points**) Prove there muse exist an edge e with $x_e \geq 1/2$. Be careful that the sets in your laminar family in part (d.i) above can contain sets with only one element.

e. (**2 points**) Describe and analyze a 2-approximation to the problem described above.

Problem 4. Consider the following *min-max* cut problem. Given input an undirected graph $G = (V, E)$ the output is to place weights on edges such that the total weight on all the edges is *at least* 1 with the objective weight of the maximum weight cut is as small as possible. Design and analyze an approximation algorithm for the problem.