CS 49/149: Approximation Algorithms

Problem set 2. Due: 14th April, 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: Local Search, Approximation Schemes

Problem 1. (Local Search for Directed Max-Cut.) Given a *directed* graph $G = (V, E)$, the dicut induced by $S \subseteq V$ is the set of arcs $\delta^{\mathsf{out}} S := \{(u, v) \in E\}$. The directed max-cut problem is to find the subset S maximizing $|\delta^{\mathsf{out}} S|$. Consider the local search algorithm which adds/removes a vertex if it strictly increases the cut-size. Let A be the locally optimum cut returned by the algorithm.
First convince yourself that $|\delta^{\mathsf{out}} A|$ can be really bad compared to OPT.
Prove that $\max(|\delta^{\mathsf{out}} A|, |\delta^{\mathsf{out}} \overline{A}|)$ is a 3-approximation, where $\overline{A}$ is the complement of the set A.
For part credit prove it is a 4-approximation.
Hint: For the undirected cut we compared to the number of edges. Compare to the optimum cut instead.

BONUS PROBLEM[1]**:** Give a 3-approximation algorithm for the problem of maximizing a non-negative submodular function; note that this function needn't be monotone. Give a 2-approximation for maximizing non-negative, *symmetric* submodular functions. Symmetric functions satisfy $f(S) = f(\overline{S})$ for all $S \subseteq E$. You may find the following property of submodular functions useful (indeed, this is often thought of as the definition):

$$\text{For any two subsets } S, T \subseteq E, \quad f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$$

Problem 2 (Local Search is 5-approximation for k-median)**. Prove** that the local-search algorithm for k-median is a 5-approximation algorithm for the k-median problem.

In the analysis of the local search algorithm for k-median, we introduced the "nearest" map $\phi : O \to A$ where for any $i^* \in O$ the facility $\phi(i^*)$ was the nearest facility in A to O. We considered the swaps $(i^*, \phi(i^*))$. We proved that the algorithm was a 3-approximation if ϕ was one-to-one. There were *two* places where we used this property.

1. When we swapped i^* and i, for every client $j \in \Gamma_i \setminus \Gamma^*_{i^*}$, we were guaranteed that $\phi(O(j))$ was in $A - i + i^*$.
2. When we summed the final inequalities for all $i^* \in O$, every client $j \in \Gamma_i \setminus \Gamma^*_{i^*}$ was counted only once.

When ϕ is not one-to-one, how will you construct the swaps? Think of swaps which satisfy property 1 above for sure, and maybe property 2 with every $i \in A$ counted at most twice.

[1] Not mandatory, but extra points if you solve it.

Problem 3. In class, the running time of the PTAS for $P||C_{max}$ was $m^{2^{\frac{1}{\varepsilon}\log^2\left(\frac{1}{\varepsilon}\right)}}$ and thus doubly exponential in $1/\varepsilon$. In this problem you need to come up with a PTAS whose running time is singly exponential in $1/\varepsilon$, that is of the form $m^{O(1/\varepsilon \log(1/\varepsilon))}$.

Recall the definition of feasible profiles $\mathbf{v} := (v_1, \ldots, v_N)$ with $N \leq \frac{1}{\varepsilon}\log(1/\varepsilon)$. Let $\mathbb{V} := \{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(M)}\}$ be the collection of all feasible profiles, with $M \leq (1/\varepsilon)^N \leq 2^{\frac{1}{\varepsilon}\log^2(\frac{1}{\varepsilon})}$. Finally, recall the definition of histograms $(h_1, \ldots, h_M)$ where h_i is the number of machines getting the feasible profile $\mathbf{v}^{(i)}$.

Prove that if we find $(h_1, \ldots, h_M)$ which satisfies the following two conditions, then we can assign all the big jobs into m identical machines such that the total load from big jobs on any machine is at most OPT.

a. $\sum_{i=1}^{M} h_i = m$.
b. $\sum_{i=1}^{M} h_i v_t^{(i)} = |B_t|, \ \forall 1 \leq t \leq N$. Recall B_t is the subcollection of big jobs with the same cost.

In class we brute forced our way to get an m^M running time algorithm to find such an h – there were at most m^M many such histograms.

Design and Analyze a more efficient algorithm to find a feasible histogram if one exists. You should shoot for a running time of at most $O(M) \cdot (m/\varepsilon)^N$ which is a singly-exponential PTAS.

Problem 4 (Tight Examples)**.**

1. For the Local Search algorithm for Max-k-coverage problem done in class, find a local optimum solution which is factor $2 - o(1)$ away from the optimum solution.
2. For the k-median problem, find a local optimum solution which is factor $5 - o(1)$ away from the optimum solution.

Problem 5. Consider the GREEDY algorithm for $P||C_{max}$ when the jobs are considered in the non-decreasing order of p_j's. What is the approximation factor of this algorithm? This is described in the Williamson-Shmoys textbook. Try to do this yourself first.

Problem 6. In the k-means problem, the input is the same as the k-median problem but the objective is to minimize $\sum_{j \in C} d^2(j, \sigma(j))$. What factor do you think the local-search algorithm for the k-means problem gives? Hint: The following inequality may be useful: for any two a, b, we have $(a+b)^2 \leq 2a^2 + 2b^2$.

Problem 7. In the k-center problem, the input is the same as the k-median problem except the objective is to minimize $\max_{j \in C} d(j, \sigma(j))$. What algorithm can you design for the problem, and what approximation factor can you prove?