PCP and Limits of Approximation Algorithms - II 2 Jun, 2015

Problem Set 1

- Due Date: **18 Jun (Thurs), 2010**
- It is recommended that you try to solve all the exercises and problems, but you need to submit the writeup for only the problems.
- Collaboration is encouraged, but all writeups must be done individually.
- Indicate names of all collaborators.
- Refering sources other than the lecture notes is discouraged, since for some of the problems a Google search will reveal the solution. But if you do use an outside source (text books, lecture notes, any material available online), do mention the same in your writeup.

---

1. **[TSP with weights 1 and 2]** (10)

   Let G be an undirected complete graph in which all the edge weights are either 1 or 2 (clearly, G satisfies the triangle inequality). Give a 4/3-approximation for TSP in this special class of graphs.

   [Hint: Start by finding a minimum 2-matching in G. A 2-matching is a subset S of edges so that every vertex has exactly 2 edges of S incident at it.]

2. **[Primal-dual analysis for SETCOVER]** (15)

   The SETCOVER problem is defined as follows.

   Input: A collection of n sets $S_1, \dots, S_n$ such that $\bigcup_{i=1}^{n} S_i = U = \{1, 2, \dots, m\}$.
   Output: A subcollection $I \subseteq [n]$ of sets such that $\bigcup_{i \in I} S_i = U$. Such an I is called a cover of U.
   Objective: Minimize the size of the subcollection $|I|$.

   There is a natural greedy algorithm for this problem along the following lines.

   (a) Add the largest set S_i to the cover.
   (b) Delete S_i from the collection of sets and all elements of S_i from the universe U to obtain a smaller sized SETCOVER instance.
   (c) Repeat the above procedure as long as the universe U is non-empty.

   In this problem, we will give a primal-dual analysis to demonstrate that this greedy algorithm gives a $\ln(\max_i |S_i|)$-approximation.

   Consider the following LP-relaxation of the SETCOVER problem. Here, we associate each set S_i with a indicator variable x_i which is 1 iff S_i is in the cover.

$$
\begin{aligned}
\text{Minimize} \qquad & \sum_{i=1}^{n} x_i \\
\text{subject to} \\
\sum_{i: j \in S_i} x_i \quad & \geq \quad 1, \qquad \forall j \in \{1, \ldots, m\} \\
0 \leq \quad x_i \quad & \leq 1, \qquad \forall i \in \{1, \ldots, n\}
\end{aligned}
$$

The dual of this LP-relaxation is as follows:

$$
\begin{aligned}
\text{Maximize} \qquad & \sum_{j=1}^{m} y_j \\
\text{subject to} \\
\sum_{j \in S_i} y_j \quad & \leq \quad 1, \qquad \forall i \in \{1, \ldots, n\} \\
y_j \quad & \geq \quad 0, \qquad \forall j \in \{1, \ldots, m\}
\end{aligned}
$$

This dual can be interpreted as the following packing problem: we wish to assign a non-negative weight y_j to each of the elements in the universe so that the sum of weights is as large as possible, with the constraint that the sum of weights of any set is at most 1.

We will now view each step of the greedy algorithm as updating a pair of solutions, one for the primal LP and one for the dual LP. All the dual solutions will be dual feasible solutions while all but the final primal solution will be primal infeasible solutions. Initially all the variables x_i and y_j are set to zero. Each time the greedy algorithm adds a set S_i to the cover, the variables are updates as follows:

- Primal: Set $x_i \leftarrow 1$.
- Dual: For all elements $j \in S_i$, set $y_j \leftarrow 1/(|S_i| \cdot B)$.

where B is a mysterious constant whose value will be set later. B will be chosen large enough that the dual constraints will always be satisfied.

(a) Show that each run of the greedy algorithm increments the primal solution by 1 and the dual solution by $1/B$. Hence, show using weak duality that the greedy algorithm is a B-approximation. (Recall that at termination both the primal and dual solutions are feasible.)

(b) Show that at the end of the greedy algorithm, for every set S_i

$$\sum_{j \in S_i} y_j \leq \frac{1}{B} + \frac{1}{2B} + \frac{1}{3B} + \cdots + \frac{1}{|S_i| \cdot B} = \frac{H_{|S_i|}}{B},$$

where H_k denotes the k-th harmonic number, $1 + 1/2 + 1/3 + \cdots + 1/k \sim \ln k$.

(c) Argue using parts 2a and 2b, that we can set $B = \ln(\max_i |S_i|)$ and thus, greedy gives a $\ln(\max_i |S_i|)$-approximation algorithm.

3. **[Greedy 2-approximation for KNAPSACK]** (10)

   Recall the KNAPSACK problem

   Input: n items with sizes $s_1, \ldots, s_n$ and values $v_1, \ldots, v_n$ and a knapsack of total size B. Furthermore, assume $s_i \leq B$ for all i.
   Output: A subset of items $I \subset [n]$ such that $\sum_{i \in I} s_i \leq B$.
   Objective: Maximize $\sum_{i \in I} v_i$.

   Consider the following greedy algorithm for KNAPSACK. Sort the items by decreasing ratio of value to size. Let the sorted order of objects be $a_1, \ldots, a_n$. Find the lowest k such that the total size of the first $(k+1)$ elements exceeds B. Now, pick the more valuable of the sets $\{a_1, \ldots, a_k\}$ and $\{a_{k+1}\}$. Show that this algorithm achieves a 2-approximation.

4. **[weighted version of vertex cover]** (15)

   Consider the following weighted version of Vertex Cover (w-VC).

   Input: Undirected graph $G = (V, E)$ with weights $w : V \to \mathbb{Z}^{\geq 0}$ on the vertices.
   Output: A cover $C \subseteq V$ of the vertices such that for every edge $(u, v) \in E$ either $u \in C$ or $v \in C$.
   Objective: Minimize the weight of the cover (i.e., $\sum_{v \in C} w(v)$).

   Observe that the 2-approximation algorithm for the unweighted version discussed in lecture does not extend to this weighted version. Design an alternate deterministic 2-approximation algorithm for w-VC.

   [Hint: First design a LP relaxation of the problem with variables for each vertex in the graph and then deterministically round the LP to obtain a 2-approximate solution.]

5. **[gap preserving reductions]** (5)

   A reduction from one gap problem gap-A$_\alpha$ to gap-B$_\beta$ (for some $0 < \alpha, \beta < 1$) is said to be a gap preserving reduction if it maps YES instances of gap-A$_\alpha$ to YES instances of gap-B$_\beta$ and NO instances of gap-A$_\alpha$ to NO instances of gap-B$_\beta$. The existence of a gap preserving reduction from gap-A$_\alpha$ to gap-B$_\beta$ implies that if it is NP-hard to approximate problem A to within α, then it is NP-hard to approximate problem B to within β.

   For every $\alpha > 0$, show that there exists a and ε, β and a gap preserving reduction from gap-3SAT$_\alpha$ to gap-2SAT$_{1-\varepsilon,\beta}$. Hence, conclude that there exists a $\beta \in (0, 1)$ such that approximating MAX2SAT to within β is NP-hard.

   Note:The gap problems gap-3SAT$_\alpha$ and gap-2SAT$_{1-\varepsilon,\beta}$ are defined as follows.
   gap-3SAT$_\alpha$:

$$\begin{aligned}
\textsf{YES} &= \{\varphi | \varphi \text{ is a satisfiable 3CNF formula}\} \\
\textsf{NO} &= \{\varphi | \varphi \text{ is a 3CNF formula such that no assignment satisfies more} \\
&\quad\ \text{than } \alpha \text{ fraction of the clauses}\}
\end{aligned}$$

gap-2SAT$_{1-\varepsilon,\beta}$:

$$\begin{aligned}\mathsf{YES} &= \{\varphi|\varphi \text{ is a 2CNF formula with an assignment that satsifies} \\ &\quad \text{at least } (1-\varepsilon)\text{-fraction of the clauses}\} \\ \mathsf{NO} &= \{\varphi|\varphi \text{ is a 2CNF formula such that no assignment satisfies more} \\ &\quad \text{than } \beta \text{ fraction of the clauses}\}\end{aligned}$$

6. **[three vs. two queries]** (10)

   In class, we stated that Håstad proved the following strengthening of the PCP Theorem which shows that every language in NP has a PCP with 3 queries and soundness error almost 1/2.

   $$[\text{Håstad}]\ \forall\varepsilon > 0, \textsf{Circuit-SAT} \in PCP_{1-\varepsilon,1/2+\varepsilon}[O(\log n), 3].$$

   Suppose we were able to further strengthen the above result to prove that Circuit-SAT has a 2 query PCP (i.e., Circuit-SAT $\in PCP_{1,s}[O(\log n), 2]$ for some $0 < s < 1$), then show that then $NP = P$!

   Thus, Håstad's PCP is optimal with respect to the number of queries till the status of the P vs. NP question is resolved.

7. **[inapproximability of clique via graph products]** (15)

   In class, we proved the following theorem showing the inapproximability of clique. 3-COLOR $\in PCP_{c,s}[r, q]$ implies it is NP-hard to approximate MAXCLIQUE to within a factor s/c as long as $2^{r+q} = \text{poly}(\cdot)$. This resulted in the following inapproximability result for MAXCLIQUE assuming the PCP Theorem (3-COLOR $\in PCP_{1,1/2}[O(\log n), O(1)]$).

   $$\exists\alpha \in (0, 1), \text{ it is NP-hard to approximate CLIQUE to within } \alpha \tag{1}$$

   We then applied sequential repetition on the PCP (i.e., $PCP_{c,s}[r, q] \subseteq PCP_{c^k,s^k}[kr, kq]$ for all $k \in \mathbb{Z}^{\geq 0}$) to obtain the following strengthening of the above result.

   $$\forall\alpha \in (0, 1), \text{ it is NP-hard to approximate CLIQUE to within } \alpha \tag{2}$$

   In this problem, we will discuss an alternative approach to prove this result using graph products. For a graph $G = (V, E)$ we define the square of G, $G^2 = (V', E')$, as follows: The vertex set V' equals V^2, the set of pairs of vertices of G. Two distinct vertices (u_1, u_2) and (v_1, v_2) are adjacent in E' if and only if $(u_1, v_1) \in E$ and $(u_2, v_2) \in E$.

   (a) Prove that the squaring operation defined above satisfies $\omega(G^2) = (\omega(G))^2$ where $\omega(G)$ denotes the size of the largest clique in G.

   (b) Use (a) to given an alternate proof of (2) from (1).

8. **[recycling randomness via random walks on an expander]** (20)

In lecture, we showed that by sequential repetition of PCPs (i.e., $PCP_{c,s}[r,q] \subseteq PCP_{c^k,s^k}[kr,kq]$ for all $k \in \mathbb{Z}^{\geq 0}$) can be used to improve the hardness factor of approximating clique (also see earlier problem). In this problem, we will discuss a more efficient way to perform repetition by recycling randomness using expander walks.

Let $G = (V, E)$ be an (n, d, λ)-expander with $\lambda < d$ i.e, G is a d-regular graph on n vertices such that the second largest eigenvalue (in absolute value) has absolute value at most λ.. Let $B \subseteq V$ be a set of vertices with $|B| = \mu n$, where $0 < \mu < 1$. Suppose we pick a uniformly random vertex in G and then perform a t-step random walk in G starting from this vertex. We wish to upper-bound the probability p that all vertices encountered along this random walk are in the set B.

(a) Let A denote the normalized adjacency matrix of G, and let P denote the matrix corresponding to *projection* onto B; in other words, P is the $n \times n$ diagonal matrix with 1's in the positions corresponding to B. Show that $p = \|P(AP)^t \pi\|_1$, where π is the vector $(1/n, \ldots |V| \text{times} \ldots, 1/n)$ (ie., the probability distribution of a random vertex in V), and $\|z\|_1$ denotes the l_1-norm of z (i.e., $\|z\|_1 = \sum_{i=1}^n |z_i|$).

(b) The matrix 2-norm of a matrix C is defined to be $\|C\|_2 = \max_{y \neq 0} \|Cy\|_2 / \|y\|_2$. Show that $p \leq \mu \|PAPAP \ldots AP\|_2 \leq \|AP\|_2^t$.

(c) Show that $\|AP\|_2 \leq \sqrt{\mu + (\lambda/d)^2}$, and conclude $p \leq (\mu + (\lambda/d)^2)^{t/2}$.

[Hint: given arbitrary $y \neq 0$, write $z = Py$ and decompose $z = z_\parallel + z_\perp$ where $z_\parallel$ is the component of z along the all one's vector and $z_\perp$ the orthogonal component.]

Extra Credit: show that in fact $\|PAP\|_2 \leq (\lambda/d) + \mu(1 - \lambda/d)$ and show how this can be used to conclude the sharper upper bound $p \leq \mu(\lambda/d) + \mu(1 - \lambda/d)^t$.

(d) Use the earlier part (c) to conclude that $\mathrm{PCP}_{1,1-s}[r,q] \subseteq \mathrm{PCP}_{1,2^{-k}}[r + O(k), O(kq)]$ for all $k \in \mathbb{Z}^{\geq 0}$.

(e) Conclude from (d) (setting $k = \log n$) that it is NP-hard to approximate to within $n^{-\delta}$ for some $\delta \in (0,1)$.