CO759: Approximation and Randomized Algorithms, Spring 2013

Instructor: Chaitanya Swamy

Assignment 1

Due: By June 20, 2013

You may use anything proved in class directly. I will maintain a FAQ about the assignment on the course webpage. *Acknowledge all collaborators and any external sources of help or reference.* To get full credit for the bonus problems, you should not refer to non-course literature (i.e., papers, books, external sources on the Internet etc.), but you may consult the reference books (Williamson-Shmoys and Vazirani) listed on the course webpage. All questions carry equal weightage.

Quick Primer on Linear Programming: A *linear program* (LP) is a problem of the following form(s)

$$\begin{array}{lll} \max & c^T x & \text{(P)} \\ \text{s.t.} & Ax \le b & \\ & x \ge 0 & \end{array} \qquad \begin{array}{lll} \min & b^T y & \text{(D)} \\ \text{s.t.} & A^T y \ge c & \\ & y \ge 0 & \end{array}$$

where we seek to optimize a linear function of a finite number of variables subject to a finite number of linear constraints. Any such LP may either be infeasible, or have an optimal solution, or be unbounded. Given a maximization primal linear program of the form (P) above, one can construct a *dual* linear program (D) that provides a tight upper bound on the optimal value of the primal LP. The rationale behind the construction of the dual is as follows. Let $A = (a_{ij})$ above be an $m \times n$ matrix, with a_i^T denoting the i-th row of A and A_j denoting the j-th column of A. Each primal constraint is associated with a nonnegative dual variable y_i. Now if we multiply each primal constraint with its corresponding dual variable y_i and sum the resulting inequalities, we obtain the compound inequality $\sum_{j=1}^{n}\big(\sum_{i=1}^{m} y_i a_{ij}\big)x_j \le b^T y$. Thus, if we enforce the constraints $\sum_{i=1}^{m} y_i a_{ij} \ge c_j$ for each variable x_j, then since $x \ge 0$, we obtain that $c^T x \le y^T A x \le b^T y$. Notice that these constraints are precisely the constraints of the dual LP, and so we have proved that the value of *any feasible solution* y to (D) provides an upper bound on the optimal value of the primal LP. This statement is often known as *weak duality*. (Also, observe that the primal LP (P) corresponds to the dual of the minimization problem (D).) The central theorem of linear programming is a much stronger theorem, often called *strong duality*.

(Strong Duality): Let (P) be a pair of primal and dual LPs, with (P) being a maximization LP (as above). Then, the following hold.

(i) (P) has an optimal solution iff (D) has an optimal solution;

(ii) The optimal values of (P) and (D) (if they exist) are equal;

(iii) If x^* and y^* are respectively optimal solutions to (P) and (D) respectively, then (part (ii) implies that) they must satisfy the following *complementary slackness* conditions: (a) $x_j^* > 0 \implies A_j^T y^* = c_j$; and (b) $y_i^* > 0 \implies a_i^T x^* = b_i$.

Q1: Set-cover I

(a) **(Do not hand this in)** Show that the integrality gap of the set-cover LP is $\Omega(B)$, where B is the maximum-frequency of an element. (That is, construct an instance where the ratio of the optimum values of the integer and linear programs is $\Omega(B)$.)

(b) In class, we constructed a set-cover instance with sets having different weights showing that the approximation ratio of the greedy algorithm is $\Omega(\ln n)$. Show that the approximation ratio of the greedy algorithm remains $\Omega(\ln n)$ even on *unweighted* set-cover instances, i.e., where all sets have unit weight.

(c) Show that for unweighted set-cover instances, the analysis of the greedy algorithm can be improved slightly to show that it returns a solution with at most $\big(1 + \ln(\frac{n}{OPT})\big)\,OPT$ sets.

(**Hint:** In the unweighted setting, the cost of the sets used by the greedy algorithm (or any reasonable algorithm) to cover k currently-uncovered elements is trivially bounded by k.)

(d) **(Bonus part)** Show the integrality gap of the set-cover LP is $\Omega(\ln n)$.

Q2: Set cover II

(a) (**Vazirani, Ex. 2.14**) Given a directed graph $G = (V, E)$, a *feedback vertex set* is a subset $V' \subseteq V$ whose removal makes the graph acyclic. Given node costs $\{c_v\}$, the feedback vertex set problem is to find a feedback vertex set of minimum cost.

A *tournament* is a directed graph $G = (V, E)$ where for every pair $u, v \in V$, exactly one of the edges (u, v) or (v, u) is in E. In the remainder of this part, G denotes a tournament.

Show that V' is a feedback vertex set iff the graph $G' = (V \setminus V', E[V \setminus V'])$ contains no directed triangles (cycles of length 3), where $E[S] := \{(u, v) \in E : u, v \in S\}$. Hence, give a 3-approximation algorithm for the feedback vertex set problem on tournaments.

(b) In class, we gave a randomized-rounding algorithm that returned a collection of sets that form a set cover with high probability and have expected cost $O(\ln n) \cdot OPT_{\text{LP}}$, where OPT_{LP} is the optimal value of the set-cover LP. We now describe how to *derandomize* this algorithm to obtain a deterministic algorithm that returns a set cover of cost at most $O(\ln n) \cdot OPT_{\text{LP}}$.

Instead of the randomized algorithm described in class, it will be slightly more convenient to consider the following randomized algorithm. Let x^* be an optimal solution to the set cover LP. Independently, for every set S, we pick S with probability $1 - e^{-2 \ln n \cdot x^*_S}$.

To derandomize this algorithm, we consider sets one by one and deterministically decide whether or not to pick a set, in such a way that the deterministic algorithm is always "ahead" of the randomized algorithm. Let $S_1, \ldots, S_m$ be the sets in the set-cover instance. Suppose that we have already picked the collection $\mathcal{S}'$ of sets from $\{S_1, \ldots, S_{i-1}\}$, and we pick sets from $S_i, \ldots, S_m$ according to the randomized algorithm. Let $\mathcal{R}$ be the random collection of sets picked from $S_i, \ldots, S_m$. Define

$$\Phi(\mathcal{S}', i) := \sum_{S \in \mathcal{S}'} w_S + \mathrm{E}\big[\sum_{S \in \mathcal{R}} w_S\big] + n \cdot OPT_{\text{LP}} \cdot \mathrm{E}\big[\text{no. of elements not covered by } \mathcal{S}' \cup \mathcal{R}\big].$$

Also define $\Phi(\mathcal{S}', m+1)$ as above, where $\mathcal{R}$ is now taken to be the empty set. Prove that (i) $\Phi(\emptyset, 1) \leq (2 \ln n + 1) \cdot OPT_{\text{LP}}$; and (ii) for any collection $\mathcal{S}' \subseteq \{S_1, \ldots, S_{i-1}\}$, we have

$\Phi(\mathcal{S}', i) \geq \min\{\Phi(\mathcal{S}', i+1), \Phi(\mathcal{S}' \cup \{S_i\}, i+1)\}$. Deduce that when we consider each set S_i, we can deterministically decide whether or not to pick S_i in our collection, so as obtain a set cover $\mathcal{S}'$ of cost at most $\Phi(\emptyset, 1) \leq (2 \ln n + 1) \cdot OPT_{\text{LP}}$.

Q3: Vertex cover and set cover; see Williamson-Shmoys Ex. 1.5

(a) Consider the following set-cover style LP-relaxation for the vertex cover problem.

$$\begin{aligned} \min \quad & \sum_v w_v x_v & \text{(VC-P)} \\ \text{s.t.} \quad & x_u + x_v \geq 1 \qquad \text{for all } (u,v) \in E \\ & x \geq 0 \end{aligned}$$

where v indexes the nodes of the underlying graph. Show that every extreme point $\hat{x}$ of (VC-P) is half-integral, that is, $\hat{x}_v \in \{0, \frac{1}{2}, 1\}$ for all v.

(**Hint:** One way to show this is to show that a feasible solution x to (VC-P) with $x_v \leq 1$ for all v can be expressed as a convex combination of half-integral solutions.)

(b) Devise a $\frac{3}{2}$-approximation algorithm for the vertex cover problem on planar graphs. You may use the fact that every planar graph is 4-colorable: that is, its nodes can be colored using 4 colors so that the endpoints of every edge receive distinct colors. You may use the result of part (a) even if you did not manage to solve part (a).

(c) (Bonus part) Generalize the result of part (b) to set cover as follows. Say that a set-cover instance $(U, \mathcal{S}, \{w_S\})$ is k-colorable if the set-collection $\mathcal{S}$ can be partitioned into k disjoint subcollections $\mathcal{S}_1, \ldots, \mathcal{S}_k$ such that every element is covered by at most one set in each $\mathcal{S}_j$. Give a $B(1 - \frac{1}{k})$-approximation algorithm for k-colorable set-cover instances, where B is the maximum frequency of an element.

Q4: Facility location

(a) Say that an algorithm is an LP-relative ρ-approximation algorithm for set cover if it returns a solution of cost at most $\rho OPT_{\text{LP}}(\mathcal{I})$ for every instance $\mathcal{I}$, where $OPT_{\text{LP}}(\mathcal{I})$ is the optimal value of the set-cover LP for instance $\mathcal{I}$. Similarly, an LP-relative ρ-approximation for uncapacitated facility location (UFL) means that the solution cost is at most ρ times the optimal value of the natural LP-relaxation for UFL described in class.

Use an LP-relative ρ-approximation algorithm for set cover to devise an LP-relative $(\rho + 1)$-approximation algorithm for non-metric UFL. Conclude that non-metric UFL admits an LP-relative $O(\ln n)$-approximation algorithm.

In the remaining parts, we consider metric UFL, and describe and analyze a $(1 + \frac{2}{e})$-approximation algorithm based on clustering and randomized rounding. Recall that we have a set $\mathcal{F}$ of facilities with facility-opening costs $\{f_i\}$, a set $\mathcal{D}$ of clients, and we incur an assignment cost c_{ij} for assigning client j to facility i, where the c_{ij}s form a metric. Let (x^*, y^*) be an optimal solution to the facility-location LP, and (α^*, β^*) be an optimal solution to the dual LP. Let $F^* = \sum_i f_i y_i^*$, and define $C_j^* = \sum_i c_{ij} x_{ij}^*$ and $F_j = \{i : x_{ij}^* > 0\}$ for a client j. It will be convenient to assume that if $x_{ij}^* > 0$

then $x^*_{ij} = y^*_i$ (note that we always have $x^*_{ij} \le y^*_i$). This can be arranged as follows. For a facility i, let $0 < \gamma_1 < \gamma_2 < \ldots < \gamma_k < y^*_i$ be the distinct positive x^*_{ij}-values that are strictly smaller than y^*_i. We replace facility i by $k+1$ "clones" $i_1, \ldots, i_{k+1}$ that are all co-located located at i (i.e., $c_{i_\ell j} = c_{ij}$ for all j and $\ell = 1, \ldots, k+1$). Define $\gamma_0 = 0,\ \gamma_{k+1} = y^*_i$. We set $y^*_{i_\ell} = \gamma_\ell - \gamma_{\ell-1}$ for all $\ell = 1, \ldots, k+1$. If $x^*_{ij} = \gamma_r$, we set $x^*_{i_\ell j} = y^*_{i_\ell}$ for all $\ell = 1, \ldots, r$. Clearly, any solution to the new instance translates to a solution to the original instance with the same cost, and vice versa.

The improved rounding algorithm is as follows. We form clusters as in the algorithm described in class, except that we now pick the client j (among the remaining candidate cluster centers) with smallest $C^*_j + \alpha^*_j$ value. Let $\mathcal{D}' \subseteq \mathcal{D}$ denote the cluster centers. We set $\mathsf{nbr}(k) = j$ if client k was removed from the list of candidate cluster centers due to client j (so $F_j \cap F_k \neq \emptyset$ and $C^*_j + \alpha^*_j \le C^*_k + \alpha^*_k$); also $\mathsf{nbr}(j) = j$ for $j \in \mathcal{D}'$. In each cluster F_j, where $j \in \mathcal{D}'$, we open exactly one facility, choosing facility $i \in F_j$ with probability $y^*_i = x^*_{ij}$. Also, we open every unclustered facility $i \notin \bigcup_{j \in \mathcal{D}'} F_j$ independently with probability y^*_i. We assign every client to the nearest open facility.

(b) Consider a client $j \in \mathcal{D}'$. Let X_j be the random variable denoting the distance between j and the facility opened from F_j. Prove that $\mathrm{E}\big[X_j\big] = C^*_j$, which thus uper bounds the expected assignment cost of j. Consider any set $S \subseteq F_j$. Prove that $\min_{i \in S} c_{ij} + \mathrm{E}\big[X_j \big| \text{ no facility from } S \text{ is open}\big] \le C^*_j + \alpha^*_j$.

(c) Now consider a client $k \in \mathcal{D} \setminus \mathcal{D}'$. Prove that the expected assignment cost of k is at most $C^*_k + 2\alpha^*_k \cdot \prod_{i \in F_k}(1 - y^*_i)$. You may use the following inequality.

Let $0 \le d_1 \le d_2 \le \ldots \le d_r$, and $y_1, y_2, \ldots, y_r \in [0,1]$. Then

$$d_1 y_1 + d_2 y_2 (1-y_1) + d_3 y_3 (1-y_1)(1-y_2) + \ldots + d_r y_r \prod_{i=1}^{r-1}(1-y_i) \le \frac{\sum_{i=1}^r d_i y_i}{\sum_{i=1}^r y_i} \cdot \Big(1 - \prod_{i=1}^{r}(1-y_i)\Big). \quad (1)$$

(**Hint:** Let X_k be the random variable denoting the assignment cost of k, and N be the event that no facility in F_k is opened. Bound $\mathrm{E}\big[X_k | \overline{N}\big]$ by an expression that looks similar to the left-hand-side of (1); use part (b) to bound $\mathrm{E}\big[X_k | N\big]$.)

(d) Deduce that the above algorithm returns a solution of expected cost at most $\big(1 + \frac{2}{e}\big)\big(F^* + \sum_j C^*_j\big)$, and hence is a $\big(1 + \frac{2}{e}\big)$-approximation algorithm.

(e) **(Bonus part)** Prove inequality (1).

Q5: Lagrangian-multiplier-preserving (LMP) algorithms

(a) Recall that in the algorithm for k-facility location (k-FL), we used an LMP ρ-approximation algorithm for (metric) UFL to find two solutions $\big(F^1, \{i^1(j)\}\big)$ with $|F^1| = k_1 < k$, and $\big(F^2, \{i^2(j)\}\big)$ with $|F^2| = k_2 > k$ such that the fractional solution obtained by taking a convex combination of the two with weights $a = \frac{k_2 - k}{k_2 - k_1}$ and $b = \frac{k - k_1}{k_2 - k_1}$ respectively has cost at most $(\rho + \epsilon) OPT_{k\text{-FL}}$ (where ϵ can be made arbitrarily small). We showed that this fractional solution can be rounded to an integer solution that opens k facilities, while blowing up the solution cost by a factor of at most 2.

Show that if the two solutions obtained have the additional property that $F^1 \subseteq F^2$, then we can round the fractional solution without losing any additional factor, thus obtaining a solution of cost at most $(\rho + \epsilon) OPT_{k\text{-FL}}$.

(b) In the *prize-collecting set-cover* (PCSC) problem, the input is a set-cover instance, but we are allowed to not cover an element e at the expense of incurring a nonnegative penalty π_e. The goal therefore is to choose a collection $\mathcal{S}'$ of sets so as to minimize $\sum_{S \in \mathcal{S}'} w_S + \sum_{e \text{ not covered by } \mathcal{S}'} \pi_e$. An LMP ρ-approximation algorithm for PCSC is an algorithm that returns a solution $\mathcal{S}'$ satisfying $\sum_{S \in \mathcal{S}'} w_S + \rho \cdot \sum_{e \text{ not covered by } \mathcal{S}'} \pi_e \leq \rho \cdot OPT_{\text{PCSC}}$. Design an LMP H_Δ-approximation for PCSC, where Δ is the size of the largest set. Recall that $H_k = 1 + \frac{1}{2} + \ldots + \frac{1}{k}$ is the k-th harmonic number.

(c) **(Bonus part)** Use the LMP H_Δ-approximation algorithm for PCSC to obtain an $O(H_\Delta)$-approximation algorithm for the partial set-cover problem, where we have an input parameter k and we seek to find the minimum-cost collection of sets that cover at least k elements.