Advanced Approximation Algorithms **CMU 15-854B, Spring 2008**

Homework 2
Due: Tuesday, February 12

---

1. Approximation Algorithms for weighted Min-Vertex-Cover. For each of the following approximation algorithms for Min-Vertex-Cover with positive vertex weights, prove the best approximation ratio guarantee that you can. (If you can say something better in the special case when all weights are 1, please do.)

a) *Super Naive:* Consider all the edges in some order. If the edge $\{u, v\}$ being considered is not covered yet, pick whichever of u or v has less weight.

b) *Naive:* Consider all the edges in some order. If the edge $\{u, v\}$ being considered is not covered yet, pick *both* the vertices u and v.

c) *Randomized:* Consider all the edges in some order. If the edge $\{u, v\}$ being considered is not covered yet, with probability $\frac{w_v}{w_u+w_v}$ pick the vertex u, and with the remaining probability, pick v.

d) *LP rounding:* The standard Vertex-Cover LP is the following: minimize $\sum w_v x_v$ subject to $x_u + x_v \geq 1$ for all edges $\{u, v\} \in E$, and $x \geq 0$. Given a fractional solution for this LP, define $V_\alpha = \{v \in V \mid x_v \geq \alpha\}$. What value of α ensures that V_α is a vertex cover? What approximation guaratee can you get?

e) *Primal-dual:* Write the dual of the above LP: the variables in this LP correspond to edges of the graph. By raising these variables uniformly, design a primal-dual algorithm with an approximation guarantee of 2.

f) *Local search:* Define two solutions $S \subseteq V$ and $S' \subseteq V$ to be neighbors if S can be obtained from S' by adding, deleting, or swapping a vertex. The local search moves are simple: Start with any solution $S \subseteq V$; if you are at some solution S, move to any neighboring solution S' that has less weight. If you are at a *local optimum* — where all the neighbors have at least as much weight — output this local optimum. (Don't worry about the running time for this algorithm.)

g) *Greedy:* Repeatedly pick a vertex v that maximizes $\frac{\text{number of edges newly covered}}{w_v}$, until all the edges are covered.

2. Facility Location (the non-metric case). The reduction from this problem to Min-Set-Cover gives an $O(\log n)$ factor approximation. Take the standard LP relaxation for the problem and use randomized rounding to find a solution whose cost is at most $O(\log n) \cdot F^* + O(1) \cdot C^*$. (Recall: fixing some optimal solution for the non-metric facility location instance, F^* is the facility cost and C^* is the connection cost of this optimal solution.)

Also, show that one shouldn't hope to get an algorithm whose cost is $o(\log n) \cdot F^* + f(n) \cdot C^*$, even with, say, $f(n) = n$.

3. Another Algorithm for Max-Cut. Consider the local search algorithm for Max-Cut, where two solutions (i.e., 2-colorings) (R, B) and (R', B') are neighbors if there exists a vertex $v \in V$ such that $R' \triangle R = \{v\} = B' \triangle B$. ($\triangle$ = "symmetric difference".) Again, the local search dynamics are the obvious ones: given a solution (R, B), move to any neighbor (R', B') with less cost.

Show that any local optimum for this local search dynamics yields an absolute 1/2 approximation.

4. Max-E3Sat-3. Show that this problem (and in general, the problem Max-EkSat-k) is solvable in polynomial time. (Hint: think perfect matchings.)

5. The Fire Station problem. The Fire Station problem is as follows: The input is a positive integer k and a complete undirected graph G with distances on the edges. The distances form a metric: $d(v, v) = 0$, $d(u, v) = d(v, u) \geq 0$, and $d(u, w) \leq d(u, v) + d(v, w)$. A valid output is a subset S of at most k vertices. The cost (to be minimized) of a solution is the maximum distance of any vertex from S; i.e., $\max_v \min_{s \in S} d(v, s)$. One can imagine being able to open k fire stations, in an attempt to minimize the distance of everyone to the closest fire station.

a) Show that approximating this problem to within ratio $2 - \epsilon$ is NP-hard for all $\epsilon > 0$. (Hint: show that Dominating-Set is NP-hard.)

b) Choose any one of the following two (very similar) algorithms, and show that it achieves an approximation ratio of 2.

> **Algorithm 1**. Assume the distinct distances in the graph are $0 = c_0 \leq c_1 \leq c_2 \cdots \leq c_m$. Let G_i denote the (unweighted) graph with the same vertices as G but only those edges (u, v) with $d(u, v) \leq c_i$, for $i = 0 \dots m$. Let G_i^2 denote the graph with the same vertices as G_i and an edge (u, v) if and only if $u \neq v$ and there is a path of length at most 2 between u and v in G_i. Compute a *maximal* independent set S_i in G_i^2. Output S_j, where j is the least value such that $|S_i| \leq k$. Show that this is a factor-2 approximation algorithm.
>
> **Algorithm 2**. Pick any vertex v_1 in the graph to begin. Pick a vertex v_2 furthest from v_1. Pick a vertex v_3 furthest from the set $S_2 = \{v_1, v_2\}$ — that is, v_3 is the vertex maximizing $\min_{s \in S_2} d(v, s)$, and so on. In general, pick v_j furthest from $S_{j-1} = \{v_1, v_2, \dots, v_{j-1}\}$. Stop when you have k vertices. Show that this is a 2-approximation.

6. Fun with set systems. A *set system* $\mathcal{F}$ over n ground elements is a collection of subsets of $[n]$. We often identify subsets of $[n]$ with strings in $\{0, 1\}^n$ in the natural way; thus we also think of $\mathcal{F}$ as a collection of strings. We can also think of $\mathcal{F}$ as a boolean function $\mathcal{F} : \{0, 1\}^n \to \{0, 1\}$, with $\mathcal{F}(x) = 1$ iff the string x is in $\mathcal{F}$.

a) Let $s \geq 2$ and $t \geq 1$ be integers. We say that $\mathcal{F}$ is "s-wise t-intersecting" if for every s sets $A_1, \dots, A_s \in \mathcal{F}$ it holds that $|A_1 \cap \cdots \cap A_s| \geq t$. Show that for each integer $\ell \geq 0$, the set system

$$\mathcal{F}_{s,t,\ell} = \{A \subseteq [n] : |A \cap \{1, \dots, t + \ell s\}| \geq t + \ell(s - 1)\}$$

is s-wise t-intersecting.

b) For integers $1 \leq i < j \leq n$, the left-shift operator S_{ij} applied to a set system $\mathcal{F} \subseteq \{0,1\}^n$ produces a new set system $S_{ij}(\mathcal{F})$, as follows: for each string $x \in \mathcal{F}$ with $x_i = 0$, $x_j = 1$, we shift the 1 from the jth position to the ith position — as long as the resulting string is not already in $\mathcal{F}$. (Remark: if $\mathcal{F}$ contains exactly m strings with Hamming weight exactly r, then the same is true of $S_{ij}(\mathcal{F})$.) We say that a set system $\mathcal{F}$ is "left-shifted" if $S_{ij}(\mathcal{F}) = \mathcal{F}$ for all $1 \leq i < j \leq n$. Show that each $\mathcal{F}_{s,t,\ell}$ is left-shifted.

c) Show that if $\mathcal{F}$ is s-wise t-intersecting then so is $S_{ij}(\mathcal{F})$.

d) [BONUS] Let $\mathcal{F}$ be left-shifted and s-wise t-intersecting. Show that $\mathcal{F} \subseteq \bigcup_{\ell \geq 0} \mathcal{F}_{s,t,\ell}$.

Exercises

This is for your edification: you do not have to submit solutions to this problem.

A. Chernoff Bounds. If you haven't seen these "large-deviation" bounds before, here's a useful form to remember.

> Let $X_1, X_2, \ldots$ be independent real-valued random variables taking on values in the range $[0,1]$. Let $E[X_i] = \mu_i$, $X = \sum_i X_i$ and $E[X] = \sum_i \mu_i = \mu$. Then
>
> $$\Pr[X \geq \mu + \lambda] \leq \exp\{-\frac{\lambda^2}{2\mu + \lambda}\}. \quad (1)$$
>
> and
>
> $$\Pr[X \leq \mu - \lambda] \leq \exp\{-\frac{\lambda^2}{3\mu}\}. \quad (2)$$
>
> In other words, the random variable X is "tightly concentrated" around its mean μ.

Here is a simple example where this can be used. Consider throwing n balls into n bins independently, where the probability that ball j falls into bin i is exactly p_{ij} (and hence $\sum_i p_{ij} = 1$ for every j). Note that the expected load of bin i is $L_i = \sum_j p_{ij}$. (Hence, if $p_{ij} = 1/n$ for all i, j, then $L_i = 1$.) Use bound (1) above to argue that any fixed bin i has at most $L_i + O(\log n)$ balls in it with probability n^{-c} for some constant c. Now use the trivial union bound (which says that for any events $\mathcal{B}_j$, $\Pr[\cup_j \mathcal{B}_j] \leq \sum_j \Pr[\mathcal{B}_j]$) to argue that with probability at least 1/2, every bin has at most $L_i + O(\log n)$ balls in it.

Aside: this result is almost tight—a better upper bound of $L_i + O(\frac{\log n}{\log\log n})$ can be proved (in fact, using a better Chernoff-type bound). Moreover, this improved bound is in fact tight for the case $p_{ij} = 1/n$: if n balls are thrown into n bins uniformly and independently, then with high probability, *at least one* bin contains $\Omega(\frac{\log n}{\log\log n})$ balls.