CS 787: Advanced Algorithms	Homework 2
Out: 10/17/07	**Due: 11/07/07**

Guidelines

- This homework consists of a few exercises followed by some problems. The exercises are meant for *your practice only*, and do not have to be turned in. You are required to turn in the problems. We will provide solutions to all the questions.
- Collaboration is not allowed. You may consult class notes as well as reading material posted on the class website, but please refrain from using extra reading material from the web or related textbooks. Some of the problems are difficult, so please get started early. Late submissions do not get any credit.

Exercises

1. In class we studied the following two LP relaxations of the min-congestion routing problem, and claimed that they are equivalent. Prove that given a solution to the second LP, you can come up with a solution of equal value to the first one in time polynomial in the size of the graph n. (Note that in order to do this in polynomial time your solution to the first LP must assign non-zero values to at most a polynomial number of variables.)

$$\begin{array}{lrcll} \min t & \text{s. t.} & & & \\ & \sum_{P\in\mathcal{P}_i} x_P & = & 1 & \forall i \\ & \sum_i \sum_{P\in\mathcal{P}_i, P\ni e} x_P & \leq & t & \forall e \\ & x_P & \geq & 0 & \forall i, P\in\mathcal{P}_i \\ & t & \geq & 0 & \end{array}$$

$$\begin{array}{lrcll} \min t & \text{s. t.} & & & \\ & \sum_{e\in\delta^-(s_i)} x_{i,e} & = & 1 & \forall i \\ & \sum_{e\in\delta^+(v)} x_{i,e} & = & \sum_{e\in\delta^-(v)} x_{i,e} & \forall i, v\neq s_i, t_i \\ & \sum_i x_{i,e} & \leq & t & \forall e \\ & x_{i,e} & \geq & 0 & \forall i, e \\ & t & \geq & 0 & \end{array}$$

2. Let $\{a_i\}_{i\in[n]}$ be a probability distribution over the set $[n]$, i. e. $\sum_{i\in[n]} a_i = 1$.
   (a) Give a procedure for picking a number in $[n]$ from this distribution. You may assume that you have as a black box a routine that generates a real number in $[0,1]$ uniformly at random, and that you can perform computations on real numbers.
   (b) Perform the same task using an unbiased coin.
3. Recall that we showed in class that the natural LP relaxation for the vertex cover problem has an integrality gap of at most 2. Give an example where the integrality gap of the LP is $2 - o(1)$.

Problems

1. Recall that in the set cover problem our goal is to cover a given set of elements using a collection of subsets of the least total cost. In the **maximum coverage problem** our goal is complementary—we want to cover the most number of elements using a collection of subsets with total cost no larger than a given bound B.
   (a) Give a constant factor approximation for the maximum coverage problem. Try to obtain as good an approximation as possible.

(b) Prove that if maximum coverage can be approximated within a factor of $1-1/e+\epsilon$ in polynomial time for some constant $\epsilon > 0$, then there exists a constant $\epsilon' > 0$ such that set cover can be approximated within a factor of $(1-\epsilon')\log n$ in polynomial time. You may assume that you know the cost of the optimal solution for the set cover problem (but not the actual solution).

2. A **legal k-coloring** of a graph is an assignment of colors $1, 2, \cdots, k$ to the vertices of the graph such that no two adjacent vertices receive the same color. A graph is **k-colorable** if there exists a legal k-coloring of its vertices. The problem of finding a legal k-coloring of a k-colorable graph is NP-complete for $k \geq 3$.

   (a) Prove that graphs with maximum degree Δ are $(\Delta+1)$-colorable. Also give a polynomial time algorithm for finding a $(\Delta+1)$-coloring.

   (b) Give a polynomial time algorithm for 2-coloring a bipartite graph.

   (c) Using parts (a) and (b) above, give a polynomial time algorithm for finding an $O(\sqrt{n})$-coloring of a 3-colorable graph.

   (Hint: Verify and use the fact that the neighborhood of any vertex in a 3-colorable graph is 2-colorable.)

   (d) Extend the algorithm from part (c) to obtain an $O(n^{2/3})$-coloring for a 4-colorable graph in polynomial time.

   (Aside: The best known algorithm for coloring 3-colorable graphs uses $O(n^{0.2111})$ colors.)

3. Recall that in class we developed a local search algorithm for the facility location problem in which, at every step, we consider adding a facility or removing a facility or swapping one facility for another from our solution, in order to improve the cost of the solution. The running time of this algorithm can be really large (even exponential) if we make only tiny improvements to the objective function at every step.

   Consider the following modification to the algorithm in order to improve its running time: we perform a local improvement step only if the cost decreases by a *factor* of at least $(1-\epsilon/m^2)$, where $\epsilon > 0$ is some constant, and m is the number of facilities (locations).

   (a) Prove that this new algorithm gives us a $5/(1-\epsilon)$ approximation.

   (b) Assume that all the facility opening and routing costs are at least 1, and let $M = \max\{\max_{i,j} c_{i,j}, \max_j f_j\}$ denote the maximum over all facility opening and routing costs. Prove that the new algorithm terminates in at most poly($\log M, m, n, 1/\epsilon$) steps, where n is the number of customers, and m is the number of facilities.

   (c) A natural question is whether we can improve the bound of 5 on the gap between the cost of a locally optimal solution and that of a globally optimal solution. Give an example where this gap is at least a factor of 2.

4. (a) Give a linear programming relaxation for the facility location problem with variables x_{ij} for routing customer i to facility j and y_j for opening facility j.

   (b) Find the dual of this LP, and give an interpretation to the dual problem (that is, explain what the dual feasible solutions represent).

   (c) Give an integrality gap example for the LP in part (a). Aim for an integrality gap lying between 1 and 2, and try to obtain as large a factor as you can.

5. We will now develop an approximation for the *non-metric* facility location problem based on your LP relaxation from the previous problem. Let LPC(x, y) denote the LP's cost for a feasible solution (x, y). Consider the optimal LP solution $(x^*, y^*) = \text{argmin}_{(x,y)}$ LPC(x, y), and let $A_i = \sum_j c_{ij}x^*_{ij}$ denote the "average" routing cost of customer i in this solution.

   (a) Prove that there exists a feasible LP solution $(\tilde{x}, \tilde{y})$ such that LPC$(\tilde{x}, \tilde{y}) \leq 2\,LPC(x^*, y^*)$, and for all pairs (i, j) with $\tilde{x}_{i,j} > 0$, the corresponding routing cost is small, specifically $c_{ij} \leq 2A_i$.

   Hint: Modify (x^*, y^*) to obtain $(\tilde{x}, \tilde{y})$ by setting to zero all x^*_{ij} for which $c_{ij} > 2A_i$, and modifying other values appropriately to obtain a feasible solution.

(b) We will now round $(\tilde{x}, \tilde{y})$ to an integral solution. For every customer i, let $F(i)$ denote the set of facilities j for which $\tilde{x}_{i,j} > 0$. Give an algorithm for picking a subset of facilities that contains at least one facility from each set $F(i)$, and approximately minimizes the total facility opening cost of this subset. Prove that the total cost of your solution is at most $O(\log n)$ times LPC$(\tilde{x}, \tilde{y})$, where n denotes the number of customers. Conclude that you obtain an $O(\log n)$ approximation.

Hint: Interpret this problem of picking a subset of facilities as a set cover problem, and apply the greedy algorithm.

Note that you should not have to use the triangle inequality anywhere in your argument.