CS880: Approximations Algorithms

Scribe: Matt Elder **Lecturer:** Shuchi Chawla
Topic: LP Rounding and Randomized Rounding **Date:** 2/22/2007

In out last lecture, we discussed the LP Rounding technique for producing approximation algorithms. The idea behind LP Rounding is to write the problem as an integer linear program, relax its integrality restraints to efficiently solve the general linear program, and then move the LP solution to a nearby integral point in the feasible solution space. The difficulty of this process lies in the rounding step, which demands that a bound on its suboptimality.

We discussed how to apply this method to vertex cover, set cover, and network flow. Here, we give somewhat more complicated rounding methods for facility location, and introduce the technique of randomized rounding in application to set cover and min-congestion rounding.

11.1 Facility Location

Again, the facility location problem gives a collection of facilities and a collection of customers, and asks which facilities we should open to minimize the total cost. We accept a facility cost of f_i if we decide to open facility i, and we accept a routing cost of $c(i,j)$ if we decide to route customer j to facility i. Furthermore, we know that the routing costs form a metric.

First, we design a linear program to answer a "relaxed" version of this problem. We let the variable x_i denote the extent to which facility i is open, and let y_{ij} denote the extent to which customer j is assigned to facility i. The following linear program then expresses the problem:

$$\begin{aligned} \text{minimize} \quad & \sum_i f_i x_i + \sum_{i,j} c(i,j) y_{ij}, \\ \text{where} \quad & 0 \le y_{ij} \le x_i \le 1 \qquad \forall i,j \end{aligned}$$

For convenience, let $C_f(x)$ denote the total factory cost induced by x, i.e., $\sum_i f_i x_i$. Similarly, let $C_r(y)$ denote the total routing cost induced by y, $\sum_{i,j} c(i,j) y_{ij}$.

If this LP were modified to require that these variables each equal 0 or 1, this system would be precisely the ILP we need to solve. But, again, solving general ILPs is NP-hard problem, so we solve this related real-valued LP instead.

Let x^*, y^* be the optimal solution to this linear program. Since every feasible solution to the original ILP lies in the feasible region of this LP, the cost $C(x^*, y^*)$ is less than the optimal solution to the ILP. Since x^* and y^* are almost certainly non-integral, we need a way to round this solution to a feasible, integral solution without increasing the cost function much.

To do so, we first employ the filtering technique of Lin and Vitter [1] to produce $\tilde{x}, \tilde{y}$. This filtering will later allow us to put upper bounds on the routing cost that we accept.

1. For each customer j, compute the average cost $\tilde{c}_j = \sum_i c(i,j) y^*_{ij}$.

2. For each customer j, let the S_j denote the set $\{i \mid c(i,j) \leq 2\tilde{c}_j\}$.

3. For all i and j: if $i \notin S_j$, then set $\tilde{y}_{ij} = 0$; else, set $\tilde{y}_{ij} = y^*_{ij} / \sum_{i \in S_j} y^*_{ij}$.

4. For each facility i, let $\tilde{x}_i = \min(2x^*_i, 1)$.

Lemma 11.1.1 *For all i and j, $\tilde{y}_{ij} \leq 2y^*_{ij}$.*

Proof: If we fix j and treat y^*_{ij} as a probability distribution, then we can show this by Markov's inequality. However, the proof of Markov's Inequality is simple enough to show precisely how it applies here:

$$\tilde{c}_j = \sum_i c(i,j) y^*_{ij} \geq \sum_{i \notin S_j} c(i,j) y^*_{ij} \geq \sum_{i \notin S_j} 2\tilde{c}_j y^*_{ij} \geq 2\tilde{c}_j \sum_{i \notin S_j} y^*_{ij}.$$

So, $1/2 \geq \sum_{i \notin S_j} y^*_{ij}$. For any fixed j, y^*_{ij} is a probability distribution, so $\sum_{i \in S_j} y^*_{ij} \geq 1/2$. Therefore, $\tilde{y}_{ij} = y^*_{ij} / \left(\sum_{i \in S_j} y^*_{ij} \right) \leq 2y^*_{ij}$. ■

Lemma 11.1.2 *$\tilde{x}, \tilde{y}$ is feasible, and $C(\tilde{x}, \tilde{y}) \leq 2C(x^*, y^*)$.*

Proof: For any fixed j, the elements $\tilde{y}_{ij}$ form a probability distribution. For every i and j, $\tilde{y}_{ij} \leq 2y^*_{ij}$ and thus $\tilde{x}_i \geq \sum_i \tilde{y}_{ij}$. It is clear that $0 \leq x_i, y_{ij} \leq 1$ for all i and j, so $\tilde{x}$ and $\tilde{y}$ are feasible solutions to the LP. ■

Now, given $\tilde{x}$ and $\tilde{y}$, we perform the following algorithm:

1. Pick the unassigned j that minimizes $\tilde{c}_j$.

2. Open factory i, where $i = \text{argmin}_{i \in S_j}(f_i)$.

3. Assign customer j to factory i.

4. For all j' such that $S_j \cap S_{j'} \neq \emptyset$, assign customer j' to factory i.

5. Repeat steps 1-4 until all customers have been assigned to a factory.

Let L be the set of facilities that we open in this way. We now show that the solution that this algorithm picks has reasonably limited cost.

Lemma 11.1.3 *$C_f(L) \leq C_f(x^*)$ and $C_r(L) \leq 6C_r(y^*)$.*

Proof: For any two customers j_1 and j_2 that were picked in Step 1, $S_{j_1} \cap S_{j_2} = \emptyset$.

Consider the facility cost incurred by one execution of Steps 1 through 4. Let j be the customer chosen in Step 1, and let i be the facility chosen in Step 2. Since $\tilde{x}$ is part of a feasible solution, $1 \leq \sum_{k \in S_j} \tilde{x}_k$. So, $f_i \leq f_i \sum_{k \in S_j} \tilde{x}_k$; and since f_i is chosen to be minimal, $f_i \leq \sum_{k \in S_j} f_k \tilde{x}_k$. Facility i is the only member of S_j that the algorithm can open.

Let J be the set of all customers selected in Step 1. Considering the above across the algorithm's whole execution yields

$$C_f(L) \leq \sum_{j \in J} \sum_{k \in S_j} f_k \tilde{x}_k = \sum_i f_i \tilde{x}_i = C_f(\tilde{x}) \leq C_f(x^*).$$

Consider now the routing cost C_r. If j was picked in Step 1, then its routing cost is $c(i,j)$ for some facility i; so $C_r(j) \leq 2\tilde{c}_j$.

Now, suppose instead that j' was not picked in Step 1. By the algorithm, there is some j that picked in Step 1 such that $S_j \cap S_{j'} \neq \emptyset$. Suppose that facility i' is in this intersection, and say that facility i is the facility to which customers j and j' are routed. Now, at long last, we use the fact that $c(i,j)$ forms a metric: we know that $C_r(j') \leq c(i',j') + c(i',j) + c(i,j)$. Because i is in both S_j and $S_{j'}$, we know by their definition that $c(i',j') \leq 2\tilde{c}_{j'}$ and that $c(i',j) + c(i,j) \leq 4\tilde{c}_j$. The customer j' was not picked in Step 1, and customer j was, so $\tilde{c}_j \leq \tilde{c}_{j'}$, and thus, $C_r(j') \leq 6\tilde{c}_{j'}$.

Now, $\tilde{c}_j$ was the routing cost of customer j in the y^* LP solution. So, $C_r(L) \leq 6C_r(y^*)$. ■

This lemma yields the following as a corollary:

Theorem 11.1.4 *This algorithm is a 6-approximation to Facility Location.*

Notice that, in the preceding construction, we picked S_j to be all i such that the cost $c(i,j) \leq 2\tilde{c}_j$. This 2 is actually fairly arbitrary. Suppose we replace it with α, some parameter of the construction. If you redo the above arithemetic, you find that $C_f(L) \leq (1/(1-\alpha))C_f(x^*)$ and that $C_r(L) \leq (3/\alpha)C_r(y^*)$. Thus, if we let $\alpha = 3/4$ instead of $1/2$, this method yields a 4-approximation. If we let α be a variable in the actual computed values of $C_f(x^*)$ and $C_r(y^*)$, we would get a somewhat better approximation.

11.2 Set Cover

Again, the set cover problem is: given a set of elements E, a collection of sets $\mathcal{S} \in \mathcal{P}(E)$, and a cost for each set $c\colon \mathcal{S} \to \mathbb{R}$, find a collection of sets $C \subseteq \mathcal{S}$ such that every element in E is contained in a set in C, and the total cost of C is minimized.

As an integer linear program, we can state this problem as follows:

$$\begin{aligned} \text{Minimize} \quad & \sum_{S \in C} c(S) x_S, \\ \text{where} \quad & \sum_{\substack{S \in C \\ S \ni e}} x_S \geq 1 \;\; \forall e, \\ \text{and} \quad & x_S \in \{0,1\} \;\text{ is 1 iff } S \in C \end{aligned}$$

To find an approximate solution to this ILP, we relax the condition $x_S \in \{0,1\}$ to $x_S \in [0,1]$, getting an LP. Then, we perform the following:

1. Solve the LP to get x^*.
2. For each set S, pick S with probability x^*_S.
3. Repeat Step 2 until all elements are covered.

Lemma 11.2.1 *The expected cost of Step 2 is the cost of x^*.*

Proof: Let Z_S be an indicator variable, which is 1 iff we pick set S in this run of Step 2. We compute:

$$\mathbf{E}[\text{cost of Step 2}] = \mathbf{E}\left[\sum_S Z_S c(S)\right] = \sum_S \mathbf{E}[Z_S] = c(x^*).$$

■

We now need to estimate the number of times that Step 2 is executed. To do so, we estimate the probability that any one element is covered in a particular execution of Step 2. Fix some element a. We know that $\sum_{S \ni a} x_S^* \geq 1$. This gives us the following reasoning:

$$\mathbf{Pr}[a \text{ is picked}] = 1 - \prod_{S \ni a}(1 - x_S^*) \geq 1 - \prod_{S \ni a} \exp(-x_S^*) = 1 - \exp\left(-\sum_{S \ni a} x_S^*\right) \geq 1 - e^{-1}.$$

So, the probability that e is unpicked after k steps is no more than e^{-k}, because each execution of Step 2 is independent. So, the probability that any particular element is unpicked after, say, $2\ln n$ steps is no more than $(1/n^2)$. By the union bound, the probability that there exists an unpicked element after $2\ln n$ steps is at most $n(1/n^2) = 1/n$.

Thus, with high probability, the number of executions of Step 2 is $O(\log n)$. So the expected total cost of the algorithm is $c(x^*)O(\log n)$, and this algorithms is a $O(\log n)$-approximation in expectation. Standard methods can convert this to an arbitrarily high-probability result.

References

[1] JH Lin and JS Vitter. Approximation Algorithms for Geometric Median Problems. In *Information Processing Letters*, 1992.