CMPUT 675: Topics on Approximation Algorithms and Approximability **Fall 2007**

Lecture 3: 13 Sept. 2007

Lecturer: Mohammad R. Salavatipour *Scribe: Pirooz Chubak*

3.1 Linear Programming (Continued)

As discussed in Section 2.2, A Linear Programming (LP) problem with n variables $x_1, \ldots, x_n$ is defined as:

$$\begin{aligned} \text{minimize}: \quad & \sum_{i=1}^{n} c_i x_i \\ \text{subject to}: \quad & \sum_{i=1}^{n} a_{ji} x_i \geq b_j, \quad \forall 1 \leq j \leq m \\ & x_i \geq 0, \quad \forall 1 \leq i \leq n \end{aligned}$$

where c_i, a_{ji}, b_j are real values. Every solution to an LP is a vector in $\mathbb{R}^n$. Every constraint corresponds to a half-space. The set of feasible solutions forms a convex body, which is called a *polyhedron*. If this body is bounded it is called a *polytope*.

A vertex (extreme point) is a point that is the intersection of n half-spaces (i.e. a feasible solution that is tight for n constraints). Every solution corresponding to a vertex is called a basic solution. We can solve LP even if there are an exponential number of constants, as long as the following conditions hold:

- The number of variables is polynomial
- Given any assignment to the variables, there is an oracle that finds a violated constraint (if there is any), in polynomial time.

3.1.1 Example (Set Cover)

Let's formulate the Set Cover problem as an integer program and find the LP relaxation. Summarizing the definition of Set Cover in Section 2.1, we have:

- Input
  - $U = \{e_1, \ldots, e_n\}$ is the *universe* of elements.
  - $S = \{S_1, \ldots, S_m\}$ is a collection of subsets of U with costs $c(S_1), \ldots, c(S_m)$.
- Goal: to find the minimum cost set $S' \subseteq S$ such that the union of all elements of S' give U.

$$\begin{aligned} \text{minimize}: \quad & \sum_{i=1}^{n} c(S_i) \cdot x_i \\ \text{subject to}: \quad & \sum_{e \in S_j} x_j \geq 1, \quad \forall\ e_i \in U \\ & x_i \in \{0,1\} \end{aligned}$$

If we relax the that constraint to $0 \leq x_i \leq 1$ then we obtain the LP relaxation to the IP. Note that in any IP/LP, the solution to the LP is a lower bound for the IP. Note that, we can remove the condition $x_i \leq 1$ as in any optimal solution there cannot be a value larger than 1. The reason is that in any feasible solution we can round all the $x_i > 1$ to 1 and still get a feasible solution with a lower cost.

A common technique in design of approximation algorithms is to first formulate the problem as an IP/LP, then solve the LP relaxation, and then round this fractional solution to an integer one while keeping a bound on the total value of the objective function. One method of rounding the LP solution is randomized rounding. Below we show how to use this technique to design another $O(\log n)$-approximation for Set Cover.

3.1.2 Set cover using randomized rounding

We use randomized rounding algorithm to solve Set Cover. First consider the following simple algorithm:

1. Compute an optimal solution x^* to the LP relaxation. Let z^* be the value of this optimal solution.
2. For each set S_i, pick S_i with probability x_i^*; in other words let $\hat{x}_i = 1$ be 1 with probability x_i^* and zero with probability $1 - x_i^*$. Therefore, $\hat{x}_i$ is the rounded value of x_i^*.
3. Return the collection $C = \{S_j | \hat{x}_j = 1\}$

The expected value of the cost of the returned collection of selected sets is computed as follows:

$$\begin{aligned} E\left[cost(C)\right] &= \sum_{i=1}^{m} \Pr\left[S_i \text{ being selected}\right] \cdot c(S_i) \\ &= \sum_{i=1}^{m} x_i^* \cdot c(S_i) \\ &= z^* \end{aligned}$$

So the expected cost of the solution is the same as the optimum. However, this algorithm randomly picks sets of S_j with a probability that is given by solving the problem using LP, so the solution might not be feasible. To address this, it's enough to repeat the Step 2, $4 \ln n$ times, and return the union of collection of all sets selected in all these repetitions. That is, if C_i is the collection of sets that are selected at i^{th} iteration of Step 2, then in Step 3, we would $C = \cup_{i=1}^{4 \ln n} c_i$. We show that this new algorithm has a good chance of returning a feasible solution with approximation ratio $O(\log n)$.

Consider an arbitrary element e_i and WLOG assume that it belongs to sets $S_1, \ldots, S_k$. Since we start from a feasible solution, we must have $x_1^* + x_2^* + \ldots + x_k^* \geq 1$; in the worst case we should have $x_1^* + x_2^* + \ldots + x_k^* = 1$. What is the probability that none of the corresponding k sets be selected given the probabilities of x_i^*'s? The following lemma (whose proof is starightforward) suggests that this probability is the highest when all the x_i^*'s are equal.

Lemma 3.1 *Given $x_1 + \ldots + x_k = 1$, if we select each S_i with probability x_i^*, the probability that no S_i will be selected is highest (worst case) when all $x_i^* = \frac{1}{k}$.*

Using Lemma 3.1, we would like to compute the probability that randomized rounding would not be able to find a feasible solution. We take the worst case conditions into account in order to obtain an upper bound probability.

$$\Pr\left[e_i \notin c_j\right] \leq \left(1 - \frac{1}{k}\right)^k \leq e^{-1}$$

Therefore, for $4 \ln n$ iterations, we would have:

$$\Pr\left[e_i \in \cup_{j=1}^{4 \ln n} c_j\right] \leq \left(e^{-1}\right)^{4 \ln n} \leq \frac{1}{4n}$$

Therefore, using union bound and the fact that there are n elements:

$$\Pr\left[\text{some } e_i \text{ is not covered}\right] \leq n \cdot \frac{1}{4n} \leq \frac{1}{4}.$$

Using the analysis done earlier, the expected cost of this solution is bounded by:

$$\mathrm{E}\left[cost(C)\right] \leq 4 \ln n \cdot z^*$$

Using Markov's inequality, $\Pr[X \geq t] \leq \frac{E[X]}{t}$:

$$\Pr\left[cost(C) > 4 \ln n \cdot z^*\right] \leq \frac{1}{4}$$

Therefore, with a probability greater than or equal to $\frac{1}{2}$, we have a feasible solution with cost less than or equal to $4 \ln n \cdot z^*$ which is in $O(\log n) \cdot OPT$, where OPT is the cost of the optimal solution. To get better probability it's enough to repeat the algorithm a constant number of times and get the best answer.

3.1.3 Integrality Gap

For an instance I of a minimization problem, let $OPT_f(I)$ be the cost of the optimal LP solution. Then, integrality gap of this instance is given by $OPT(I)/OPT_f(I)$. The integrality gap of the problem Π is $\max_I OPT(I)/OPT_f(I)$, i.e. the largest ratio between the fractional and integral solution over all possible instances of Π. Note that in any LP rounding algorithm, we use the LP solution as a lower bound for the optimal solution. Therefore, any such algorithm cannot have a performance ratio better than the integrality gap. In general, it is difficult to design algorithms with ratio better than the integrality gap. Therefore, large integrality gaps typically are indications of difficulty of a problem. Consider for example the Vertex Cover problem, and let I be the complete graph K_n. Clearly by assigning a value of $\frac{1}{2}$ to each vertex we get a fractional solution of value $\frac{n}{2}$ whereas any integer solution must contain at least $n - 1$ nodes (or else one edge is not covered). Thus the integrality gap of V.C. is at least $2 - \frac{2}{n}$.

Exercise. Prove that the integrality gap for Set Cover problem is $\Omega(\log n)$.

3.1.4 Weighted Vertex Cover.

In this section we show how we can use LP rounding to obtain a 2-approximation for the weighted version of V.C. Consider the following LP relaxation of the vertex cover problem:

$$\begin{array}{rrcll} \text{minimize}: & \displaystyle\sum_{i=1}^{n} c_v \cdot x_v & & & \\ \text{subject to}: & x_u + x_v & \geq & 1, & \forall\ uv \in E \\ & x_v & \geq & 0 & \end{array}$$

Definition 3.2 (Half-integer) *A variable is half-integer, if it is an integer factor of* $\{0, \frac{1}{2}, 1\}$.

Our main goal is to prove the following lemma:

Lemma 3.3 *Any basic solution to the above LP for the Vertex Cover is half-integer*

It's easy to see that by rounding each $x_i \geq \frac{1}{2}$ to 1 we get a 2-approximation algorithm for V.C.

Proof: In order to prove this lemma, we take advantage of the fact that all non-basic feasible solutions can be written as a convex combination of two other feasible solutions. However, basic solutions cannot be written in such a way. Let x be any basic solution to the LP. Assume x is not half-integer. Define:

$$V^{>} = \left\{ v \;\middle|\; \frac{1}{2} < x_v < 1 \right\}$$
$$V^{<} = \left\{ v \;\middle|\; 0 < x_v < \frac{1}{2} \right\}$$

Assuming that x is not half-integer, we must have $V^{>} \cup V^{<} \neq \emptyset$. We would like to show that there exist feasible solutions y_v and z_v such that x can be written as a convex combination of them, which is a contradiction. Define y_v and z_v as follows:

$$y_v = \begin{cases} x_v + \epsilon & x \in V^{>} \\ x_v - \epsilon & x \in V^{<} \\ x_v & \text{Otherwise} \end{cases} \quad \text{and} \quad z_v = \begin{cases} x_v - \epsilon & x \in V^{>} \\ x_v + \epsilon & x \in V^{<} \\ x_v & \text{Otherwise} \end{cases}$$

Observation 1. If $V^{>} \cup V^{<} \neq \emptyset$, then $y \neq x$ and $z \neq x$.

Observation 2. $x = \frac{1}{2}(y + z)$; i.e. x is written as a combination of y and z.

We can easily make sure that $0 \leq y_v, z_v \leq 1$ by choosing ϵ sufficiently small. In order to prove y and z are both feasible solutions, we consider two cases:

1. For every edge uv with $x_u + x_v > 1$, we can easily choose ϵ so small that we have $(x_u - \epsilon) + (x_v - \epsilon) \geq 1$. In this case, both $y_u + y_v \geq 1$ and $z_v + z_u \geq 1$ since y_u and z_u are both $\geq x_u - \epsilon$ (similarly for y_v and z_v)
2. Now, suppose $x_u + x_v = 1$. In this case one of x_u and x_v belongs to $V^{>}$ and the other belongs to $V^{<}$. Assume $x_u \in V^{>}$ and $x_v \in V^{<}$. Therefore $(\underbrace{x_u + \epsilon}_{y_u} - \epsilon) + (\underbrace{x_v - \epsilon}_{y_v} + \epsilon) = 1 \Rightarrow y_u + y_v = 1$ and $(\underbrace{x_u - \epsilon}_{z_u} + \epsilon) + (\underbrace{x_v + \epsilon}_{z_v} - \epsilon) = 1 \Rightarrow z_u + z_v = 1$

Therefore, in z, y are both feasible and x (which is a basic solution) is a convex combination of two feasible solution, a contradiction.

■