22C:253 Lecture 6

Scribe: George Thomas

September 25, 2002

The integer program for SET COVER is the following:
Let x_i be an indicator variable for set S_i.
Minimize

$$\sum_{i=1}^{k} x_i \cdot c(S_i)$$

subject to

$$\begin{aligned} \sum_{i: j \in S_i} x_i &\geq 1 \text{ for } j = 1, 2, \cdots, n \\ x_i &\in \{0, 1\} \text{ for } i = 1, 2, \cdots, k \end{aligned}$$

The corresponding LP-relaxation replaces $x_i \in \{0, 1\}$ by $x_i \geq 0$ for each $i = 1, 2, \cdots, k$. Recall that $x_i \leq 1$ is unnecessary.

Here is a deterministic rounding approximation algorithm for SET COVER that uses the above LP relaxation. Let f_j be the frequency of element $j = 1, 2, \cdots, n$ (that is, the number of sets S_i that j appears in). Let $f = \max_j f_j$. The algorithm provides a factor-f approximation for SET COVER.

Algorithm

1. Solve the LP-relaxation (using your favorite polynomial-time LP solver).

2. For any variable $x_i \geq \frac{1}{f}$ in the solution of the LP-relaxation computed in step 1, round x_i to 1. Round all other x_is down to 0.

Lemma 1 *The above algorithm produces a feasible solution for SET COVER.*

Proof: Note that for each element $j = 1, 2, \ldots, n$, the constraint

$$\sum_{i: j \in S_i} x_i \geq 1$$

contains f_j terms (one term for every set j belongs to). Therefore, the maximum number of terms in any such constraint is f. This implies that for each such constraint, there is a variable x_i, $j \in S_i$, such that $x_i \geq 1/f_j \geq 1/f$. This implies that x_i is rounded to 1 by the above algorithm and hence the inequality continues to be satisfied even after the rounding step, implying feasibility. □

Lemma 2 *The cost of the solution produced by the above algorithm is at most $f \cdot OPT$.*

Proof: First note that if C^* is the optimal cost of the solution to the LP-relaxation, then

$$C^* \leq OPT$$

This follows from the fact that the feasible region of the LP-relaxation contains everything that is feasible for original SET COVER IP.

Let C be the cost of the solution produced by our algorithm. Let $x = (x_1, x_2, \ldots, x_n)$ denote an optimal solution of the LP-relaxation and let $x' = (x'_1, x'_2, \ldots, x'_n)$ denote the solution after rounding. Now

$$C = \sum_{i=1}^{k} c(S_i) \cdot x'_i.$$

Also note that

$$x'_i \leq f \cdot x_i.$$

This implies that

$$C \leq f \sum_{i=1}^{k} c(S_i) \cdot x_i = f \cdot C^* \leq f \cdot OPT.$$

□

How good is this algorithm?

1. It yields a factor-2 approximation algorithm for Vertex Cover.
2. This is incomparable to the O(log n)-factor greedy approximation algorithm for Set Cover discussed earlier. (Performance varies depending on the value of f.)

LP-Based Techniques

LP-based techniques can be partitioned into two groups:

1. Algorithms that work by *rounding*:
   - Simpler, more intuitive.
   - More costly because solving an LP is relatively costly.
2. *Primal-dual schema* algorithms:
   - They are based on LP-relaxation but eventually have *combinatorial* versions.
   - Faster, because they are combinatorial.
   - More amenable to fine-tuning.

Elementary LP Theory

An LP has a linear objective function subject to linear constraints. There are various forms of writing LPs, such as standard, canonical, slack, etc.

Standard Form of LP

Minimize

$$\sum_{j=1}^{n} c_j x_j$$

subject to

$$\begin{aligned} \sum_{j=1}^{n} a_{ij} x_j &\leq b_i \text{ for } i = 1, 2, \cdots, n \\ x_j &\geq 0 \text{ for } j = 1, 2, \cdots, n \end{aligned}$$

All other forms of LP(maximization of objective,non-negativity and equality constraints,etc) can be easily transformed into standard form.
More compactly, given $c \in \Re^n$, $b \in \Re^m$ and $A \in \Re^{m \times n}$, LP in standard form is

$$\min c^T x$$

subject to

$$Ax \leq b, x \geq 0.$$

Note that the solution vector x belongs to $\Re^n$.

Geometric aspects of LP

The $(m+n)$ constraints define a feasible region of the LP. Each constraint corresponds to an *n-dimensional half-space*. Therefore, the feasible region is the intersection of $(m+n)$ n-dimensional half-spaces. It is well known that this is a *convex polytope* (in $\Re^n$).
If the LP has an optimal solution, then it has one at a vertex of the feasibility polytope. The LP may not have an optimal solution because either

1. Feasible region is empty
2. Feasible region is unbounded

But this is not an issue for us as we will usually be working with non-empty, bounded feasible regions.
There are three well known algorithmic techniques for solving an LP:

1. *Simplex method* (Dantzig, 1949): This is fast, but exponential in worst case.
2. *Ellipsoid method* (Khachiyan, 1979): Polynomial time, but expensive; this was an important theoretical result showing that LP was in P.
3. *Interior point methods* (Karmarkar, 1980s): Polynomial time, it competes with Simplex. Its worst case is large polynomial time.

Integrality Gap

Let Π be an optimization problem, P be an IP for it, and L be an LP-relaxation of P. Let $OPT(I)$ denote the cost of an optimal solution of Π for instance I. Let $OPT_f(I)$ denote the cost of the optimal solution of L. For a minimization problem, $OPT_f(I) \leq OPT(I)$ for all I. The ratio

$$\sup_{I} \frac{OPT(I)}{OPT_f(I)}$$

is the integrality gap of the (P, L) pair.

Examples

CVC: For K_3, $OPT = 2$ and $OPT_f = 1.5$

$$\Rightarrow Integrality\ Gap\ for\ CVC \geq 2/1.5$$

MMS: Consider the case of 1 job ($n = 1$) of time P and m machines , $OPT = P$ and $OPT_f = P/m$

$$\Rightarrow Integrality\ Gap\ for\ MMS \geq m,\ ie.\ unbounded$$

Situations in which good integrality gap is guaranteed: Best possible integrality gap is 1. In some cases, this is achieved. eg. Vertex cover for bipartite graphs.

Total Unimodularity. A square matrix B is *unimodular* if $det(B) \in \{+1, -1\}$. A matrix A is *totally unimodular*(TUM) if for every non-singular, square submatrix B of A, $det(B) \in \{+1, -1\}$.

Theorem 3 *Given an LP, min $c^T x$ subject to $Ax \leq b$ and $x \geq 0$, if A is TUM then every vertex of the feasibility polytope is integral, provided b is integral.*