CSC 2421 - Approximation Algorithms

Lecture #3: 21 January 2003

Lecturer: A. Borodin

Scribe Notes by: S. Horn and T. Kral

1 Agenda

- IP, LP and rounding for set cover
- Integrality gap
- Randomized rounding
- Primal-dual applications

2 Integer programming, linear programming and rounding for set cover

Let $c(S)$ be the cost of the set S, and let x_s represent whether S is included in the set cover. x_s is equal to 1 if and only if set S is picked in the set cover.

Thus the integer programming problem for set cover is:

min $\sum_{S\in\mathcal{S}} c(S)x_s$

subject to: $\sum_{S:e\in S} x_s \geq 1\ \forall e \in U$

$x_s \in \{0,1\}\ \forall S \in \mathcal{S}$

The LP-relaxation of the problem is obtained by letting $0 \leq x_s \leq 1$.

The $x_s \leq 1$ is redundant, however, since if $x_s > 1$, we can always find a smaller x_s for which the constraint holds, namely $x_s = 1$. Thus we can consider $x_s \geq 0\ \ \forall S \in \mathcal{S}$.

Algorithm for set cover using LP-rounding

- Compute an optimal fractional solution, x^* to the LP-relaxation, where x^* is a vector of values x^*_s $\forall S \in \mathcal{S}$.
- Use rounding to produce $\hat{x}$, where $\hat{x}$ is a vector of values $\hat{x}_s$, as follows:

  If $x^*_s \geq 1/f$, where f is frequency of the most frequent element, then $\hat{x}_s = 1$, otherwise $\hat{x}_s = 0$.

Note: The above algorithm can be modified so that it picks all sets that are non-zero in the fractional solution. In other words, the rounding step can map anything non-zero to 1, and the algorithm still achieves a factor of f. This modification is discussed in question 14.1 of the text.

Claim 1: The LP/Rounded solution is an integral feasible solution.

Proof: Let $\mathcal{C}$ be the collection of sets selected. For any arbitrary element, $e \in U$, e is in at most f sets. Thus one of these sets, S, must be selected with $x_s^* \geq 1/f$ in the optimal fractional cover, x^*. Thus e is covered by $\mathcal{C}$, so $\mathcal{C}$ is a valid set cover.

Claim 2: The above algorithm is an f-approximation algorithm for the set cover problem.

Proof: After the rounding $x_s^* \rightarrow \hat{x}_s, \hat{x}_s$ is at most f times x_s^*. Also $f \cdot OPT_{frac} \leq f \cdot OPT$ since allowing a fractional solution can only improve the optimal solution. Thus,

$$LP/Rounded \leq f \cdot OPT_{frac} \leq f \cdot OPT$$

3 Integrality gap

Let $OPT_{frac}(I)$ denote the cost of an optimal fractional solution to the instance I.

For a minimization problem, the integrality gap of the LP-relaxation is:

$$max_I \frac{OPT(I)}{OPT_{frac}(I)}$$

3.1 Integrality gap for vertex cover

Now consider IP formulation for vertex cover:

$$\min \sum_{u \in V} c(u)\, x_u$$
$$\text{subject to } x_u + x_v \geq 1 \ \forall (u,v) \in E$$

The integrality gap for this problem can be found as follows:

Let I be K_n with unweighted edges, where K_n is the complete graph with n vertices.

In this case, x* = (1/2, 1/2,... ,1/2) is the fractional, optimal solution.

Thus OPT_{frac} = n/2.

In the optimal solution, we must select all but one vertex in order to cover all the edges of the complete, K_n graph. Thus $OPT = n - 1$.

Therefore, the integrality gap for vertex cover $\geq 2 - 2/n$

Using the LP-rounding algorithm we gave above, $x_u = 1 \quad \forall u \in V$, resulting in a feasible solution of n to the original integer program for K_n. Note that this is only 1 away from the optimal solution, a very pleasing result.

Now, consider a second instance of vertex cover, C_n with unweighted edges and n odd, where C_n is the n-cycle.

In this case, x* = (1/2, 1/2, ... , 1/2), and $OPT_{frac} = n/2$. With a naive rounding procedure that is not dependent on the problem instance, we would obtain a rounded, integral solution n, as with the previous example. However, the optimal integral solution for C_n is $OPT = \lceil \frac{n}{2} \rceil$. This is a less satisfying approximation.

Intuitively, any rounding procedure that does not consider the particular instance of the problem cannot produce a solution that has an approximation factor better than the integrality gap of the relaxation.

4 Randomized rounding

A second rounding procedure is randomized rounding. In this procedure, fractions in the optimal fractional solution are viewed as probabilities.

For the LP relaxation of set cover, we have

$x^* = (x_1^*, ..., x_i^*)$ where $i = |\mathcal{S}|$; for $0 \leq x_s^* \leq 1$

Represent each fraction x_s^* as the probability p_s.

Randomized algorithm for set cover

- Compute an optimal fractional solution x^* to the LP-relaxation
- Represent the fractional solution as probabilities $\{p_s : S \in \mathcal{S}\}$
- Use rounding to produce $\hat{p}_s$ as follows:

  round p_s to 1 with probability p_s, otherwise round to 0

Consider an arbitrary element e, which is contained in $k \leq n$ sets, where $n = |U|$. Let the corresponding probabilities for these sets be $p_1, p_2, ..., p_k$. Since e is fractionally covered in the optimal solution, we know $\sum_{i=1}^{k} p_i \geq 1$.

Let $\mathcal{C} = \{S | \hat{p}_s = 1\}$. Notice that $\mathcal{C}$ is not necessarily a set cover.

Note: The probability that e is covered by $\mathcal{C}$ is minimized when p$_i$ = 1/k $\forall$ i.

Prob[e is not covered by $\mathcal{C}$] $\leq (1 - 1/k)^k \leq 1/e$

The expected cost of $\mathcal{C}$ is:

E[cost($\mathcal{C}$)]

= $\sum E[\hat{p}_s] c(S)$

= $\sum p_s c(S)$

= OPT_{frac}

4.1 Finding a set cover using randomized rounding

Pick λ big enough so that $(1/e)^{\lambda \lg n} \leq \frac{1}{4n}$. Now run the experiment described above $\lambda \lg n$ times, where $n = |U|$.

Let $\mathcal{C}' = \bigcup_{1 \leq i \leq \lambda \lg n} \mathcal{C}_i$

Prob[e not covered by $\mathcal{C}'$] $\leq (1/e)^{\lambda \lg n} \leq \frac{1}{4n}$

Summing over all elements $e \in U$:

Prob[$\mathcal{C}'$ is not a cover] $\leq \frac{1}{4}$

Also, we can find the expected cost:

E[cost($\mathcal{C}'$)] $\leq \lambda \lg n \cdot OPT_{frac} = O(\lg n) \cdot OPT_{frac}$

Now we use Markov's Inequality: $P[X \geq t] \leq \frac{E[X]}{t}$ with $t = \lambda \lg n \cdot OPT_{frac}$, to find:

Prob[cost($\mathcal{C}'$) $> 4\lambda \lg n \cdot OPT_{frac}$] $\leq$ 1/4

The probability of the union of the two events

1) $\mathcal{C}'$ is not a cover and
2) cost($\mathcal{C}'$) $> 4\lambda \lg n \cdot OPT_{frac}$

is $\leq$ 1/2. Thus we have the following result:

Prob[cost($\mathcal{C}'$) $\leq 4\lambda \lg n \cdot OPT_{frac}$ AND $\mathcal{C}'$ is a cover] $\geq 1/2$.

We can verify in polynomial time whether $\mathcal{C}'$ satisfies both the conditions above. If not, we can repeat the entire algorithm. Notice that the expected number of repetitions needed is at most 2.

5 Primal-Dual Applications

5.1 Background and a motivating example

Because it's always nice to know why you're doing something. For variety's sake, we'll use an example that's not directly from the text. (Or, at least, not directly from the text we're using.)

Consider the minimization problem:

$$\begin{array}{ll} \text{minimize} & 10x_1 + 6x_2 + 4x_3 \\ \text{subject to} & 2x_1 + x_2 - x_3 \geq 2 \\ & x_1 + x_2 + x_3 \geq 3 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

Suppose we wished to prove that the optimal value of the function we wish to minimize (the objective function) is $\leq$ 50. Easily done - simply produce a feasible solution that is less than 50. In this case $\bar{x} = (1, 1, 1)$, evaluating to 20, would be sufficient. But what can we do if we wish to prove that the optimal is ≥ 10?

First, observe that the coefficients in the conditions are less than the corresponding coefficients in the objective function. So $2x_1 + x_2 - x_3 \geq 2$ immediately tells us that the objective function's minimum must also be $\geq$ 2. Similarly the coefficients of $3x_1 + 2x_2 \geq 5$, the sum of the two conditions, are still less than those in the objective function, and so we can say that the minimum must be ≥ 5.

In general, we consider a linear combination of the two conditions, specifically non-negative multiples, say

y_1 and y_2.

eg $\quad y_1(2x_1 + x_2 - x_3) \geq 2y_1$
$\quad\quad y_2(x_1 + x_2 + x_3) \geq 3y_2$

The goal is to find y_1 and y_2 such that the coefficients of x_1, x_2, and x_3 are no greater than those in the objective function, but the linear combination of the right-hand sides of the conditions is maximized.

More precisely, for our example we get the LP

$$\begin{array}{ll} \text{maximize} & 2y_1 + 3y_2 \\ \text{subject to} & 2y_1 + y_2 \leq 10 \\ & y_1 + y_2 \leq 6 \\ & -y_1 + y_2 \leq 4 \\ & y_1, y_2 \geq 0 \end{array}$$

This is called the dual of our original LP. Note that it is a maximization linear program in standard form. (As an aside, if we work out the two problems, we get an optimal solution of 17 for both, at $x^* = (0, 5/2, 1/2)$ for the primal minimization problem and at $y^* = (1, 5)$ for the dual.)

Every LP has a dual. For a minimization LP, the dual is a maximization LP, and vice versa, forming primal/dual pairs.

$$\begin{array}{ll} \text{minimize} & c_1x_1 + ... + c_nx_n \\ \text{subject to} & \sum_{j=1}^{n} a_{ij}x_j \geq b_i, \;\; 1 \leq i \leq m \\ & x_j \geq 0, \;\; 1 \leq j \leq n \end{array} \quad \text{and} \quad \begin{array}{ll} \text{maximize} & b_1y_1 + ... + b_mx_m \\ \text{subject to} & \sum_{i=1}^{m} a_{ij}y_i \leq c_j, \;\; 1 \leq j \leq n \\ & y_i \geq 0, \;\; 1 \leq i \leq m \end{array}$$

For a minimization LP, all feasible solutions of the dual will be less than or equal to all feasible solutions of the primal.

5.2 Dual-fitting based analysis of the "obvious" greedy set cover algorithm

We already know the problem of set cover can be represented by the LP

$$\begin{array}{ll} \text{minimize} & \sum_{S \in \mathcal{S}} c(S)x_s \\ \text{subject to} & \sum_{S: e \in S} x_s \geq 1 \;\forall e \in \mathcal{U} \\ & x_s \geq 0 \;\forall S \in \mathcal{S} \end{array}$$

The dual to this problem would be

$$\begin{array}{ll} \text{maximize} & \sum_{e \in \mathcal{U}} y_e \\ \text{subject to} & \sum_{e \in S} y_e \leq c(S) \;\forall S \in \mathcal{S} \\ & y_e \geq 0 \;\forall e \in \mathcal{U} \end{array}$$

The intuition behind the dual is that we can think of it as what you can pack into a set without exceeding its capacity (the cost of the set).

Now, recall the "obvious" greedy algorithm for set cover, where $cost_{GREEDY}(I) = \sum_{k=1}^{n} price(e_k)$. If $y_e = price(e)$ were dual feasible, then we would have an optimal algorithm. Unfortunately it's not. But...

Claim: $\left\{ y_e = \frac{price(e)}{H_\kappa} \mid e \in \mathcal{U} \right\}$, where κ is the maximum size of any set, is dual feasible.

Proof: Let S be an arbitrary set in $\mathcal{S}$. We wish to show that $\sum_{e \in S} y_e \leq c(S)$.

Say the elements of S are covered in the order $e_1, e_2, ..., e_l$ for $l \leq \kappa$.

Consider e_i. Suppose e_i is first covered by some set $\hat{S}$.

Then we have $price(e_i) \leq \frac{c(S)}{l-i+1}$ (since we took $\hat{S}$, which is at least as good as S)

This gives us
$$\begin{aligned} \sum_{e:e\in S} y_e &\leq \frac{1}{H_\kappa}\sum_{i=1}^{l}\frac{c(S)}{l-i+1} \\ &= \frac{1}{H_\kappa}H_l c(S) \qquad l \leq \kappa \\ &\leq \text{c(S)} \end{aligned}$$

Given the claim, we know that the "obvious" greedy algorithm is an H_κ-approximation, as $OPT_{frac} \geq \frac{\sum_{e\in\mathcal{U}} price(e)}{H_\kappa} = \frac{greedy(I)}{H_\kappa}$

5.3 Using primal-dual schema in the design of algorithms

We've now seen how to use primal-dual methods for analysis, but it is also a useful technique in designing approximation algorithms

Weak Duality Theorem 5.1. *If $\vec{x} = (x_1, ..., x_n)$ is a feasible solution to the primal program and $\vec{y} = (y_1, ..., y_m)$ is a feasible solution to the dual, then $\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{n} b_i y_i$.*

Proof:
$$\begin{aligned} \sum_{j=1}^{n} c_j x_j &\geq \sum_{j=1}^{n}\left(\sum_{i=1}^{m} a_{ij} y_i\right) x_j \\ &= \sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij} x_j\right) y_i \\ &\geq \sum_{i=1}^{m} b_i y_i \end{aligned}$$

Strong Duality Theorem 5.1. *If $\vec{x} = (x_1, ..., x_n)$ is an optimal solution to the primal program then the dual has an optimal solution $\vec{y} = (y_1, ..., y_m)$ and $\sum_{i=1}^{m} b_i y_i = \sum_{j=1}^{n} c_j x_j$*

Complementary Slackness Conditions 5.1. *Let $\vec{x}$ and $\vec{y}$ be feasible solutions for the primal and dual, respectively. $\vec{x}$ and $\vec{y}$ are both optimal if and only if the following conditions are satisfied:*

Primal slackness condition: $x_j = 0$ *or* $\sum_{i=1}^{m} a_{ij} y_i = c_j \quad 1 \leq j \leq n$
Dual slackness condition: $y_j = 0$ *or* $\sum_{j=1}^{n} a_{ij} x_j = b_i \quad 1 \leq i \leq m$

Relaxed Complementary Slackness Conditions 5.1. *Let $\vec{x}$ and $\vec{y}$ be feasible solutions for the primal and dual, respectively. If $\vec{x}$ and $\vec{y}$ satisfy:*

- *Primal (relaxed) complementary slackness condition: Let $\alpha \geq 1$. For each $1 \leq j \leq n$: Either $x_j = 0$ or $c_j/\alpha \leq \sum_{i=1}^{m} a_{ij} y_i \leq c_j$*
- *Dual (relaxed) complementary slackness condition: Let $\beta \geq 1$. For each $1 \leq i \leq m$: Either $y_i = 0$ or $b_i \leq \sum_{j=1}^{n} a_{ij} x_j \leq b_i\beta$*

then $\sum_{j=1}^{n} c_j x_j \leq \alpha \cdot \beta \cdot \sum_{i=1}^{m} b_i y_i$.

Note that if $\alpha = 1$ and $\beta = 1$, then the relaxed complementary slackness conditions are equivalent to the original complementary slackness conditions.

Many approximation algorithms that use a primal-dual schema obtain a solution to one problem by relaxing the conditions of its dual. For instance, in a one-pass primal-dual approach, one starts with a feasible solution to the dual and an integral (not necessarily feasible) solution to the primal. As the algorithm proceeds, we use the relaxed complementary slackness conditions to help find a feasible solution to the primal, maintaining integrality of the primal solution at each step.