Design and Analysis of Algorithms (XI)

LP based Approximation

Guoqiang Li

School of Software, Shanghai Jiao Tong University

Rounding

Set Cover of LP-Rounding

We will design two approximation algorithms for set cover problem using the method of LP-rounding.

The first one is a simple rounding algorithm achieving a guarantee of f.

The second one is based on randomized rounding and achieves a guarantee of $O(\log n)$.

The Set Cover LP-Relaxation

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

$$\sum_{S : e \in S} x_S \geq 1, \qquad e \in U$$

$$x_S \geq 0, \qquad S \in \mathcal{S}$$

A Simple Rounding Algorithm

Algorithm

1. Find an optimal solution to the LP-relaxation.
2. Pick all sets S for which $x_S \geq 1/f$ in this solution.

Analysis

Theorem

This algorithm achieves an approximation factor of f for the set cover problem.

Proof.

Let $\mathcal{C}$ be the collection of picked sets. We first show that $\mathcal{C}$ is indeed a set cover.

- Consider an element e. Since e is in at most f sets, one of these sets must be picked to the extent of at least $1/f$ in the fractional cover, due to the pigeonhole principle.
- Thus, e is covered by $\mathcal{C}$, and hence $\mathcal{C}$ is a valid set cover.

Analysis

- The rounding process increases x_S, for each set $S \in \mathcal{C}$, by a factor of at most f.
- Therefore, the cost of $\mathcal{C}$ is at most f times the cost of the fractional cover, thereby proving the desired approximation guarantee.

Randomized Rounding

Why Randomization

The performance guarantee of a random approximation algorithm holds with high probability.

In most cases a randomized approximation algorithms can be derandomized, and

the randomization gains simplicity in the algorithm design and analysis.

Randomized Rounding for Set Cover

Intuitively, a set with larger value is more likely to be chosen in the optimal solution.

This motivates us to view the fractions as probability and round accordingly.

Analysis (I)

- Let $\mathbf{x} = \mathbf{p}$ be an optimal solution to the linear program.
- For each set $S \in \mathcal{S}$, our algorithm pick S with probability p_S.
- The expected cost of $\mathcal{C}$ is

$$\mathbf{E}[\text{cost}(\mathcal{C})] = \sum_{S \in \mathcal{S}} \mathbf{Pr}[S \text{ is picked}] \cdot c_S = \sum_{S \in \mathcal{S}} p_S \cdot c_S = \text{OPT}_f.$$

- Next, let us compute the probability that an element $a \in U$ is covered by $\mathcal{C}$.

Analysis (II)

- Suppose that a occurs in k sets of $\mathcal{S}$.
- Let the probabilities associated with these sets be $p_1, \ldots, p_k$. Then $p_1 + p_2 + \cdots + p_k \geq 1$.
- Then we have

$$\mathbf{Pr}[a \text{ is covered by } \mathcal{C}] \geq 1 - (1 - \frac{1}{k})^k \geq 1 - \frac{1}{e}$$

- To get a complete set cover, independently pick $c \log n$ such subcollections, and let $\mathcal{C}'$ be their union, where c is a constant such that

$$(\frac{1}{e})^{c \log n} \leq \frac{1}{4n}$$

Analysis (III)

- Now we have

$$\mathbf{Pr}[a \text{ is not covered by } \mathcal{C}'] \leq (\frac{1}{e})^{c \log n} \leq \frac{1}{4n}$$

- Summing over all elements $a \in U$, we get

$$\mathbf{Pr}[\mathcal{C}' \text{ is not a valid set cover }] \leq n \cdot \frac{1}{4n} \leq \frac{1}{4}.$$

Analysis (IV)

- Clearly $\mathbf{E}[\mathcal{C}'] \leq \mathrm{OPT}_f \cdot c \log n$.
- Applying Markov's Inequality, we get

$$\mathbf{Pr}[\mathrm{cost}(\mathcal{C}') \geq \mathrm{OPT}_f \cdot 4c \log n] \leq \frac{1}{4}$$

- Thus

$$\mathbf{Pr}[\mathcal{C}' \text{ is a valid set cover and has cost} \leq \mathrm{OPT}_f \cdot 4c \log n] \geq \frac{1}{2}$$

MAX-SAT

MAX-SAT

Given n boolean variables $x_1, \ldots, x_n$, a CNF

$$\varphi(x_1, \ldots, x_n) = \bigwedge_{j=1}^{m} C_j$$

and a nonnegative weight w_j for each C_j.
Find an assignment to the x_is that maximizes the weight of the satisfied clauses.

Flipping a Coin

A very straightforward randomized approximation algorithm is to set each x_i to `true` independently with probability $1/2$.

Setting each x_i to `true` with probability $1/2$ independently gives a randomized $\frac{1}{2}$-approximation algorithm for weighted MAX-SAT.

Proof

Proof.

Let W be a random variable that is equal to the total weight of the satisfied clauses. Define an indicator random variable Y_j for each clause C_j such that $Y_j = 1$ if and only if C_j is satisfied. Then

$$W = \sum_{j=1}^{m} w_j Y_j$$

We use OPT to denote value of optimum solution, then

$$E[W] = \sum_{j=1}^{m} w_j E[Y_j] = \sum_{j=1}^{m} w_j \cdot \Pr[\text{clause } C_j \text{ satisfied}]$$

Proof (cont'd)

Since each variable is set to `true` independently, we have

$$\Pr[\text{clause } C_j \text{ satisfied}] = \left(1 - \left(\frac{1}{2}\right)^{l_j}\right) \geq \frac{1}{2}$$

where l_j is the number of literals in clause C_j. Hence,

$$E[W] \geq \frac{1}{2}\sum_{j=1}^{m} w_j \geq \frac{1}{2} OPT.$$

A Finer Analysis

Observe that if $l_j \geq k$ for each clause j, then the analysis above shows that the algorithm is a $(1 - (\frac{1}{2})^k)$-approximation algorithm for such instances. For instance, the performance guarantee of MAX E3SAT is $7/8$.

From the analysis, we can see that the performance of the algorithm is better on instances consisting of long clauses.

Theorem
If there is an $(\frac{7}{8} + \epsilon)$-approximation algorithm for MAX E3SAT for any constant $\epsilon > 0$, then **P = NP**.

Derandomization

Derandomization

The previous randomized algorithm can be derandomized. Note that

$$\begin{aligned} E[W] &= E[W \mid x_1 \leftarrow \texttt{true}] \cdot \Pr[x_1 \leftarrow \texttt{true}] \\ &\quad + E[W \mid x_1 \leftarrow \texttt{false}] \cdot \Pr[x_1 \leftarrow \texttt{false}] \\ &= \frac{1}{2}(E[W \mid x_1 \leftarrow \texttt{true}] + E[W \mid x_1 \leftarrow \texttt{false}]) \end{aligned}$$

We set b_1 `true` if $E[W \mid x_1 \leftarrow \texttt{true}] \geq E[W \mid x_1 \leftarrow \texttt{false}]$ and set b_1 `false` otherwise. Let the value of x_1 be b_1.

Continue this process until all b_i are found, i.e., all n variables have been set.

An Example

$$x_3 \vee \overline{x_5} \vee \overline{x_7}$$

- $\texttt{Pr}[\text{clause satisfied} \mid x_1 \leftarrow \texttt{true}, x_2 \leftarrow \texttt{false}, x_3 \leftarrow \texttt{true}] = 1$
- $\texttt{Pr}[\text{clause satisfied} \mid x_1 \leftarrow \texttt{true}, x_2 \leftarrow \texttt{false}, x_3 \leftarrow \texttt{false}] = 1 - (\frac{1}{2})^2 = \frac{3}{4}$

Derandomization

This is a deterministic $\frac{1}{2}$-approximation algorithm because of the following two facts:

1. $E[W \mid x_1 \leftarrow b_1, \ldots, x_i \leftarrow b_i]$ can be computed in polynomial time for fixed $b_1, \ldots, b_i$.
2. $E[W \mid x_1 \leftarrow b_1, \ldots, x_i \leftarrow b_i, x_{i+1} \leftarrow b_{i+1}] \geq E[W \mid x_1 \leftarrow b_1, \ldots, x_i \leftarrow b_i]$ for all i, and by induction, $E[W \mid x_1 \leftarrow b_1, \ldots, x_i \leftarrow b_i, x_{i+1} \leftarrow b_{i+1}] \geq E[W]$.

Flipping Biased Coins

Flipping Biased Coins

- Previously, we set each x_i `true` or `false` with probability $\frac{1}{2}$ independently. $\frac{1}{2}$ is nothing special here.
- In the following, we set each x_i `true` with probability $p \geq \frac{1}{2}$.
- We first consider the case that no clause is of the form $C_j = \bar{x}_i$.

Lemma

If each x_i is set to `true` with probability $p \geq 1/2$ independently, then the probability that any given clause is satisfied is at least $\min(p, 1 - p^2)$ for instances with no negated unit clauses.

Proof

Proof.

- If the clause is a unit clause, then the probability the clause is satisfied is p.
- If the clause has length at least two, then the probability that the clause is satisfied is $1 - p^a(1-p)^b$, where a is the number of negated variables and b is the number of unnegated variables. Since $p > \frac{1}{2} > 1 - p$, this probability is at least $1 - p^2$.

Flipping Biased Coins

Armed with previous lemma, we then maximize $\min(p, 1-p^2)$, which is achieved when $p = 1-p^2$, namely $p = \frac{1}{2}(\sqrt{5}-1) \approx 0.618$.

We need more effort to deal with negated unit clauses, i.e., $C_j = \bar{x}_i$ for some j.

We distinguish between two cases:

1. Assume $C_j = \bar{x}_i$ and there is no clause such that $C = x_i$. In this case, we can introduce a new variable y and replace the appearance of $\bar{x}_i$ in φ by y and the appearance of x_i by $\bar{y}$.

Flipping Biased Coins

2. $C_j = \bar{x}_i$ and some clause $C_k = x_i$. W.L.O.G we assume $w(C_j) \leq w(C_k)$. Note that for any assignment, C_j and C_k cannot be satisfied simultaneously. Let v_i be the weight of the unit clause $\bar{x}_i$ if it exists in the instance, and let v_i be zero otherwise, we have

$$\text{OPT} \leq \sum_{j=1}^{m} w_j - \sum_{i=1}^{n} v_i$$

We set each x_i `true` with probability $p = \frac{1}{2}(\sqrt{5} - 1)$, then

$$\begin{aligned} E[W] &= \sum_{j=1}^{m} w_j E[Y_j] \\ &\geq p \cdot \left(\sum_{j=1}^{m} w_j - \sum_{i=1}^{n} v_i \right) \\ &\geq p \cdot \text{OPT} \end{aligned}$$

Rounding by Linear Programming

The Use of Linear Program

Integer Program Characterization: Linear Program Relaxation:

$$\max \quad \sum_{j=1}^{m} w_j z_j$$

$$\sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j, \quad \forall C_j = \bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i,$$

$$y_i \in \{0,1\} 0 \leq y_i \leq 1, \qquad i = 1, \ldots, n,$$
$$z_j \in \{0,1\} 0 \leq z_j \leq 1, \qquad j = 1, \ldots, m.$$

where y_i indicate the assignment of variable x_i and z_j indicates whether clause C_j is satisfied.

Flipping Different Coins

- Let (y^*, z^*) be an optimal solution of the linear program.
- We set x_i to `true` with probability y_i^*.
- This can be viewed as flipping different coins for every variable.

Randomized rounding gives a randomized $(1 - \frac{1}{e})$-approximation algorithm for MAX-SAT.

Analysis

$$
\begin{aligned}
& \Pr[\text{clause } C_j \text{ not satisfied}] \\
= \quad & \prod_{i \in P_j} (1 - y_i^*) \prod_{i \in N_j} y_i^* \\
\leq \quad & \left[\frac{1}{l_j} \left(\sum_{i \in P_j} (1 - y_i^*) + \sum_{i \in N_j} y_i^* \right) \right]^{l_j} \\
= \quad & \left[1 - \frac{1}{l_j} \left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \right) \right]^{l_j} \leq \left(1 - \frac{z_j^*}{l_j} \right)^{l_j}
\end{aligned}
$$

Arithmetic-Geometric Mean Inequality

Analysis

$$
\begin{aligned}
& \Pr[\text{clause } C_j \text{ satisfied}] \\
\geq\ & 1 - \left(1 - \frac{z_j^*}{l_j}\right)^{l_j} \\
\geq\ & \left[1 - \left(1 - \frac{1}{l_j}\right)^{l_j}\right] z_j^*
\end{aligned}
$$

Jensen's Inequality

Therefore, we have

$$
\begin{aligned}
E[W] &= \sum_{j=1}^{m} w_j \Pr[\text{clause } C_j \text{ satisfied}] \\
&\geq \sum_{j=1}^{m} w_j z_j^* \left[1 - \left(1 - \frac{1}{l_j}\right)^{l_j}\right] \\
&\geq \left(1 - \frac{1}{e}\right) \cdot \text{OPT}
\end{aligned}
$$

The Combined Algorithm

Choosing the Better of Two

- The randomized rounding algorithm performs better when l_j-s are small. ($\left(1 - \frac{1}{k}\right)^k$ is nondecreasing)
- The unbiased randomized algorithm performs better when l_j-s are large.
- We will combine them together.

Choosing the better of the two solutions given by the two algorithms yields a randomized $\frac{3}{4}$-approximation algorithm for MAX SAT.

Analysis

Let W_1 and W_2 be the r.v. of value of solution of randomized rounding algorithm and unbiased randomized algorithm respectively. Then

$$
\begin{aligned}
E[\max(W_1, W_2)] &\geq E[\frac{1}{2}W_1 + \frac{1}{2}W_2] \\
&\geq \frac{1}{2}\sum_{j=1}^{m} w_j z_j^* \left[1 - \left(1 - \frac{1}{l_j}\right)^{l_j}\right] + \frac{1}{2}\sum_{j=1}^{m} w_j \left(1 - 2^{-l_j}\right) \\
&\geq \sum_{j=1}^{m} w_j z_j^* \left[\frac{1}{2}\left(1 - \left(1 - \frac{1}{l_j}\right)^{l_j}\right) + \frac{1}{2}\left(1 - 2^{-l_j}\right)\right] \\
&\geq \frac{3}{4} \cdot \mathrm{OPT}
\end{aligned}
$$

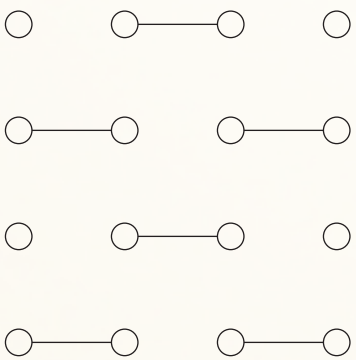
Primal-Dual Schema

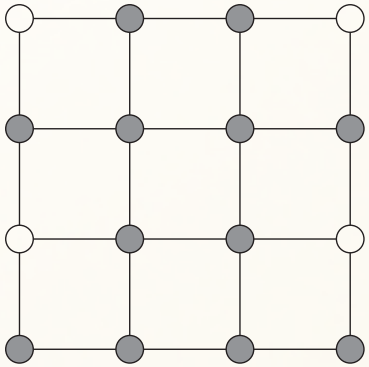
Vertex Cover, Revisited Again

Cardinality Vertex Cover

Algorithm

Find a maximal matching in G and output the set of matched vertices.





The algorithm

Consider the following algorithm:

1. $M \leftarrow \varnothing$
2. $S \leftarrow \varnothing$
3. **while** G is not empty **do**
   - 3.1 Choose an edge $e = \{u, v\} \in E(G)$ and let $M \leftarrow M \cup \{e\}$
   - 3.2 $S \leftarrow S \cup \{u, v\}$
   - 3.2 $G \leftarrow G[V \setminus \{u, v\}]$
4. **return** S

Cardinality Vertex Cover, Revisit

$$\min \quad \sum_{v \in V} x_v$$

$$x_u + x_v \geq 1, \quad (u,v) \in E$$

$$x_v \geq 0, \quad v \in V$$

$$\max \quad \sum_{e \in E} y_e$$

$$\sum_{e=(u,v) \in E} y_e \leq 1, \quad u \in V$$

$$y_e \geq 0, \quad e \in E$$

Vertex Cover

- Vertex Cover Problem is the special case of set cover problem when $f = 2$.
- The dual of vertex cover problem is Maximum Matching Problem.
- The duality theorem implies

$$\text{maximum matching} \leq \text{minimum vertex cover}$$

The algorithm

Consider the following algorithm:

1. $M \leftarrow \varnothing$
2. $S \leftarrow \varnothing$
3. **while** G is not empty **do**
   - 3.1 Choose an edge $e = \{u, v\} \in E(G)$ and let $M \leftarrow M \cup \{e\}$
   - 3.2 $S \leftarrow S \cup \{u, v\}$
   - 3.2 $G \leftarrow G[V \setminus \{u, v\}]$
4. **return** S

This is the combinatorial interpretation of a primal-dual algorithm.

Overview of the Schema

Primal and Dual

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

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, 2, \ldots, m$$

$$x_j \geq 0, \qquad j = 1, 2, \ldots, n$$

$$\max \quad \sum_{i=1}^{m} b_i y_i$$

$$\sum_{i=1}^{m} a_{ij} y_i \leq c_i, \quad j = 1, 2, \ldots, n$$

$$y_i \geq 0, \qquad j = 1, 2, \ldots, m$$

Complementary Slackness

Let **x** and **y** be primal and dual feasible solutions, respectively. Then, **x** and **y** are both optimal iff all of the following conditions are satisfied:

- Primal complementary slackness conditions:
  For each $1 \leq j \leq n$: either $x_j = 0$ or $\sum_{i=1}^{m} a_{ij} y_i = c_j$; and
- Dual complementary slackness conditions:
  For each $1 \leq i \leq m$: either $y_i = 0$ or $\sum_{j=1}^{n} a_{ij} x_j = b_i$.

Discussion

The theorem ensures

$$x_j > 0 \implies \sum_{i=1}^{m} a_{ij} y_i = c_j$$

In general, we cannot hope

$$y_i > 0 \implies \sum_{j=1}^{n} a_{ij} x_j = b_i$$

We want to show it is not too slack, i.e.

$$y_i > 0 \implies \sum_{j=1}^{n} a_{ij} x_j \leq \beta \cdot b_i$$

Complementary Slackness Relaxation

Let **x** and **y** be primal and dual feasible solutions, respectively. Then, **x** and **y** are both optimal iff all of the following conditions are satisfied:

- Primal complementary slackness conditions:
  let $\alpha \geq 1$
  For each $1 \leq j \leq n$: either $x_j = 0$ or $c_j/\alpha \leq \sum_{i=1}^{m} a_{ij} y_i \leq c_j$; and
- Dual complementary slackness conditions:
  Let $\beta \geq 1$
  For each $1 \leq i \leq m$: either $y_i = 0$ or $b_i \leq \sum_{j=1}^{n} a_{ij} x_j \leq \beta \cdot b_i$.

Primal-Dual Schema

Theorem If **x** and **y** are primal and dual feasible solutions satisfying the conditions stated in the previous page, then

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

Proof.

$$\sum_{j=1}^{n} c_j x_j \leq \alpha \sum_{j=1}^{n} (\sum_{i=1}^{m} a_{ij} y_i) x_j = \alpha \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j) y_i \leq \alpha\beta \sum_{i=1}^{m} b_i y_i$$

- If primal conditions are ensured, we set $\alpha = 1$
- If dual conditions are ensured, we set $\beta = 1$.

Primal-Dual Approximation Algorithms

1. Formulate a given problem as an IP. Relax the variable constraints to obtain the primal LP $\mathcal{P}$, then find the dual $\mathcal{D}$.
2. Starts with a primal infeasible solution $\mathbf{x}$ and a dual feasible solution $\mathbf{y}$, usually $\mathbf{x} = 0$ and $\mathbf{y} = 0$.
3. Until x is feasible, do
   - 3.1. Increase the value y_i in some fashion until some dual constraints go tight.
   - 3.2. Select some subset of tight dual constraints and increase the values of primal variables corresponding to these constraints by an integral amounts. This ensures that the final solution $\mathbf{x}$ is integral.
4. The cost of the dual solution is used as a lower bound on OPT. Note that the approximation guarantee of the algorithm is $\alpha\beta$.

Applied to Set Cover

Primal-Dual to Set Cover

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

$$\sum_{S: e \in S} x_S \geq 1, \quad e \in U$$

$$x_S \geq 0, \qquad S \in \mathcal{S}$$

$$\max \quad \sum_{e \in U} y_e$$

$$\sum_{e: e \in S} y_e \leq c(S), \quad S \in \mathcal{S}$$

$$y_e \geq 0, \qquad e \in U$$

$\alpha = 1$ and $\beta = f$

.

Primal Condition

Primal conditions

$$\forall S \in \mathcal{S} : x_S \neq 0 \Rightarrow \sum_{e:e \in S} y_e = c(S)$$

- Set S will be said to be tight if $\sum_{e:e \in S} y_e = c(S)$.
- Since we will increment the primal variables integrally, we can state the conditions as:
- Pick only tight sets in the cover.

Dual Condition

Dual conditions

$$\forall e \in U : y_e \neq 0 \Rightarrow \sum_{S:e \in S} x_S \leq f$$

- Since we will find a $0/1$ solution for x, these conditions are equivalent to:
- Each element having a nonzero dual value can be covered at most f times.

The Algorithm

1. Initialization: $x \leftarrow 0$, $y \leftarrow 0$
2. Until all elements are covered, do:
   - 2.1 Pick an uncovered element, say e, and raise y_e until some set goes tight.
   - 2.2 Pick all tight sets in the cover and update x.
   - 2.3 Declare all the elements occurring in these sets as "covered".
3. Output the set cover x.

The Guarantee Factor

Theorem

The algorithm achieves an approximation factor of f.

Proof.

Clearly there will be no uncovered elements and no overpacked sets at the end of the algorithm. Thus, the primal and dual solutions will both be feasible. Since they satisfy the relaxed complementary slackness conditions with $\alpha = 1$ and $\beta = f$.

Referred Materials

Content of this lecture comes from Chapter 14 and 15 Chapter 16 in [Vaz04], and Section 5.1-5.5 7.1-7.2 in [WS11].