Design and Analysis of Algorithms (XXI)

Linear Programming based Approximation

Guoqiang Li
School of Software

Shanghai Jiao Tong University

Rounding

Set Cover of LP-Rounding

Set Cover of LP-Rounding

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

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

$$
\begin{aligned}
\min \quad & \sum_{S \in \mathcal{S}} c(S) x_S & \\
& \sum_{S : e \in S} x_S \geq 1, & e \in U \\
& x_S \geq 0, & S \in \mathcal{S}
\end{aligned}
$$

A Simple Rounding Algorithm

Algorithm

A Simple Rounding Algorithm

Algorithm

1. Find an optimal solution to the LP-relaxation.

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.

Analysis

Theorem

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

Proof.

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.

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.

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

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.

The rounding process increases x_S, for each set $S \in \mathcal{C}$, by a factor of at most f.

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.

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.

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

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

Randomized Rounding for Set Cover

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

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.

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.

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

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

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

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

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 $\mathcal{S}$.

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 $p_1 + p_2 + \cdots + p_k \geq 1$.

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

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

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

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

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 $\mathbf{E}[\mathcal{C}'] \leq \mathrm{OPT}_f \cdot c \log n$.

Analysis (IV)

Clearly $\mathbf{E}[\mathcal{C}'] \leq \mathrm{OPT}_f \cdot c \log n$.

Applying Markov's Inequality, we get

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

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

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

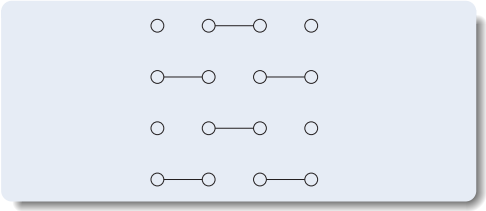
Primal-Dual Schema

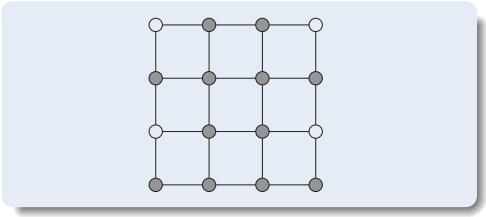
Vertex Cover, Revisited Again

Cardinality Vertex Cover

Algorithm

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





The algorithm

Consider the following algorithm:

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

Cardinality Vertex Cover, Revisit

$$
\begin{aligned}
\min \quad & \sum_{v \in V} x_v \\
& x_u + x_v \geq 1, && (u, v) \in E \\
& x_v \geq 0, && v \in V
\end{aligned}
$$

Cardinality Vertex Cover, Revisit

$$
\begin{aligned}
\min \quad & \sum_{v \in V} x_v \\
& x_u + x_v \geq 1, && (u,v) \in E \\
& x_v \geq 0, && v \in V
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & \sum_{e \in E} y_e \\
& \sum_{e=(u,v) \in E} y_e \leq 1, && u \in V \\
& y_e \geq 0, && e \in E
\end{aligned}
$$

Vertex Cover

VERTEX COVER PROBLEM is the special case of set cover problem when $f = 2$.

Vertex Cover

VERTEX COVER PROBLEM is the special case of set cover problem when $f = 2$.

The dual of vertex cover problem is MAXIMAL MATCHING PROBLEM.

Vertex Cover

VERTEX COVER PROBLEM is the special case of set cover problem when $f = 2$.

The dual of vertex cover problem is MAXIMAL MATCHING PROBLEM.

The duality theorem implies

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

The algorithm

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

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} c_j x_j \\
& \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad && i = 1, 2, \ldots, m \\
& x_j \geq 0, && j = 1, 2, \ldots, n
\end{aligned}
$$

Primal and Dual

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} c_j x_j \\
& \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, 2, \ldots, m \\
& x_j \geq 0, \qquad\qquad j = 1, 2, \ldots, n
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{m} b_i y_i \\
& \sum_{i=1}^{m} a_{ij} y_i \leq c_i, \quad j = 1, 2, \ldots, n \\
& y_i \geq 0, \qquad\qquad j = 1, 2, \ldots, m
\end{aligned}
$$

Complementary Slackness

Complementary Slackness

Let $\mathbf{x}$ and $\mathbf{y}$ be primal and dual feasible solutions, respectively. Then, $\mathbf{x}$ and $\mathbf{y}$ 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

Discussion

The theorem ensures

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

Discussion

The theorem ensures

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

In general, we cannot hope

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

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 $\mathbf{x}$ and $\mathbf{y}$ be primal and dual feasible solutions, respectively. Then, $\mathbf{x}$ and $\mathbf{y}$ 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 $\mathbf{x}$ *and* $\mathbf{y}$ *are primal and dual feasible solutions satisfying the conditions stated in the previous page, then*

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

Primal-Dual Schema

Theorem

If $\mathbf{x}$ *and* $\mathbf{y}$ *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.

Primal-Dual Schema

Theorem

If $\mathbf{x}$ *and* $\mathbf{y}$ *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.

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

Primal-Dual Schema

Theorem

If $\mathbf{x}$ *and* $\mathbf{y}$ *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

Algorithm

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

$$\begin{aligned} \min \quad & \sum_{S \in \mathcal{S}} c(S) x_S \\ & \sum_{S : e \in S} x_S \geq 1, && e \in U \\ & x_S \geq 0, && S \in \mathcal{S} \end{aligned}$$

$$\begin{aligned} \max \quad & \sum_{e \in U} y_e \\ & \sum_{e : e \in S} y_e \leq c(S), && S \in \mathcal{S} \\ & y_e \geq 0, && e \in U \end{aligned}$$

Primal-Dual to Set Cover

$$\begin{aligned} \min \quad & \sum_{S \in \mathcal{S}} c(S) x_S \\ & \sum_{S : e \in S} x_S \geq 1, \qquad e \in U \\ & x_S \geq 0, \qquad S \in \mathcal{S} \end{aligned}$$

$$\begin{aligned} \max \quad & \sum_{e \in U} y_e \\ & \sum_{e : e \in S} y_e \leq c(S), \qquad S \in \mathcal{S} \\ & y_e \geq 0, \qquad e \in U \end{aligned}$$

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

.

Primal Condition

Primal conditions

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

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 $\sum_{e:e\in S} y_e = c(S)$.

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:

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

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

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:

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

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.

The Guarantee Factor

Theorem

The algorithm achieves an approximation factor of f.

Proof.

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

Referred Materials

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