OLD CS 473: Fundamental Algorithms, Spring 2015

Approximation Algorithms using Linear Programming

Lecture 26

April 30, 2015

Weighted vertex cover

Weighted Vertex Cover problem

$G = (V, E)$.
Each vertex $\mathbf{v} \in V$: cost $\mathbf{c_v}$.
Compute a vertex cover of minimum cost.

1. vertex cover: subset of vertices V so each edge is covered.
2. **NP-Hard**
3. ...unweighted **Vertex Cover** problem.
4. ... write as an integer program (IP):
5. $\forall \mathbf{v} \in V$: $x_v = 1 \iff \mathbf{v}$ in the vertex cover.
6. $\forall \mathbf{vu} \in E$: covered. $\implies x_v \vee x_u$ **true**. $\implies x_v + x_u \geq 1$.
7. minimize total cost: $\mathbf{min} \sum_{v \in V} x_v \mathbf{c_v}$.

Weighted vertex cover

State as IP $\implies$ Relax $\implies$ LP

$$\begin{aligned} \mathbf{min} \quad & \sum_{v \in V} \mathbf{c_v} x_v, & & \\ \text{such that} \quad & x_v \in \{0, 1\} & \forall \mathbf{v} \in V & \qquad (1) \\ & x_v + x_u \geq 1 & \forall \mathbf{vu} \in E. & \end{aligned}$$

1. ... **NP-Hard**.
2. relax the integer program.
3. allow x_v get values $\in [0, 1]$.
4. $x_v \in \{0, 1\}$ replaced by $0 \leq x_v \leq 1$. The resulting LP is

$$\begin{aligned} \mathbf{min} \quad & \sum_{v \in V} \mathbf{c_v} x_v, & \\ \text{s.t.} \quad & 0 \leq x_v & \forall \mathbf{v} \in V, \\ & x_v \leq 1 & \forall \mathbf{v} \in V, \\ & x_v + x_u \geq 1 & \forall \mathbf{vu} \in E. \end{aligned}$$

Weighted vertex cover – rounding the LP

1. Optimal solution to this LP: $\widehat{x_v}$ value of var X_v, $\forall \mathbf{v} \in V$.
2. optimal value of LP solution is $\widehat{\alpha} = \sum_{v \in V} \mathbf{c_v} \widehat{x_v}$.
3. optimal integer solution: x'_v, $\forall \mathbf{v} \in V$ and α'.
4. **Any valid solution to IP is valid solution for LP!**
5. $\widehat{\alpha} \leq \alpha'$.
   Integral solution not better than LP.
6. Got fractional solution (i.e., values of $\widehat{x_v}$).
7. Fractional solution is better than the optimal cost.
8. Q: How to turn fractional solution into a (valid!) integer solution?
9. Using **rounding**.

How to round?

1. consider vertex $\mathbf{v}$ and fractional value $\widehat{x}_\mathbf{v}$.
2. If $\widehat{x}_\mathbf{v} = 1$ then include in solution!
3. If $\widehat{x}_\mathbf{v} = 0$ then do not include in solution.
4. if $\widehat{x}_\mathbf{v} = 0.9 \implies$ LP considers $\mathbf{v}$ as being 0.9 useful.
5. The LP puts its money where its belief is...
6. ...$\widehat{\alpha}$ value is a function of this "belief" generated by the LP.
7. Big idea: Trust LP values as guidance to usefulness of vertices.

II: How to round?

$$\begin{aligned} \min \ & \sum_{\mathbf{v}\in \mathsf{V}} c_\mathbf{v} x_\mathbf{v}, & \\ \text{s.t.}\ & 0 \leq x_\mathbf{v} & \forall \mathbf{v} \in \mathsf{V} \\ & x_\mathbf{v} \leq 1 & \forall \mathbf{v} \in \mathsf{V} \\ & x_\mathbf{v} + x_\mathbf{u} \geq 1 & \forall \mathbf{vu} \in \mathsf{E} \end{aligned}$$

1. Pick all vertices $\geq$ threshold of usefulness according to LP.
2. $S = \left\{ \mathbf{v} \;\middle|\; \widehat{x}_\mathbf{v} \geq 1/2 \right\}$.
3. **Claim**: S a valid vertex cover, and cost is low.

1. Indeed, edge cover as: $\forall \mathbf{vu} \in \mathsf{E}$ have $\widehat{x}_\mathbf{v} + \widehat{x}_\mathbf{u} \geq 1$.
2. $\widehat{x}_\mathbf{v}, \widehat{x}_\mathbf{u} \in (0,1)$
   $\implies \widehat{x}_\mathbf{v} \geq 1/2$ or $\widehat{x}_\mathbf{u} \geq 1/2$.
   $\implies \mathbf{v} \in S$ or $\mathbf{u} \in S$ (or both).
   $\implies S$ covers all the edges of G.

Cost of solution

Cost of S:

$$c_S = \sum_{\mathbf{v}\in S} c_\mathbf{v} = \sum_{\mathbf{v}\in S} 1 \cdot c_\mathbf{v} \leq \sum_{\mathbf{v}\in S} 2\widehat{x}_\mathbf{v} \cdot c_\mathbf{v} \leq 2 \sum_{\mathbf{v}\in \mathsf{V}} \widehat{x}_\mathbf{v} c_\mathbf{v} = 2\widehat{\alpha} \leq 2\alpha^I,$$

since $\widehat{x}_\mathbf{v} \geq 1/2$ as $\mathbf{v} \in S$.
α^I is cost of the optimal solution $\implies$

Theorem

The **Weighted Vertex Cover** *problem can be* **2***-approximated by solving a single LP. Assuming computing the LP takes polynomial time, the resulting approximation algorithm takes polynomial time.*

The lessons we can take away

Or not - boring, boring, boring.

1. Weighted vertex cover is simple, but resulting approximation algorithm is non-trivial.
2. Not aware of any other **2**-approximation algorithm does not use LP. (For the weighted case!)
3. Solving a **relaxation** of an optimization problem into a LP provides us with insight.
4. But... have to be creative in the rounding.

Revisiting Set Cover

1. Purpose: See new technique for an approximation algorithm.
2. Not better than greedy algorithm already seen $O(\log n)$ approximation.

Problem: Set Cover

Instance: $(S, \mathcal{F})$
S - a set of n elements
$\mathcal{F}$ - a family of subsets of S, s.t. $\bigcup_{X \in \mathcal{F}} X = S$.
Question: The set $\mathcal{X} \subseteq F$ such that $\mathcal{X}$ contains as few sets as possible, and $\mathcal{X}$ covers S.

Set Cover – IP & LP

$$\begin{aligned} \min \quad & \alpha = \sum_{U \in \mathcal{F}} x_U, & \\ \text{s.t.} \quad & x_U \in \{0, 1\} & \forall U \in \mathcal{F}, \\ & \sum_{U \in \mathcal{F}, s \in U} x_U \geq 1 & \forall s \in S. \end{aligned}$$

Next, we relax this IP into the following LP.

$$\begin{aligned} \min \quad & \alpha = \sum_{U \in \mathcal{F}} x_U, & \\ & 0 \leq x_U \leq 1 & \forall U \in \mathcal{F}, \\ & \sum_{U \in \mathcal{F}, s \in U} x_U \geq 1 & \forall s \in S. \end{aligned}$$

Set Cover – IP & LP

1. LP solution: $\forall U \in \mathcal{F}$, $\widehat{x_U}$, and $\widehat{\alpha}$.
2. Opt IP solution: $\forall U \in \mathcal{F}$, x^I_U, and α^I.
3. Use LP solution to guide in rounding process.
4. If $\widehat{x_U}$ is close to 1 then pick U to cover.
5. If $\widehat{x_U}$ close to 0 do not.
6. Idea: Pick $U \in \mathcal{F}$: randomly choose U with **probability** $\widehat{x_U}$.
7. Resulting family of sets $\mathcal{G}$.
8. Z_S: indicator variable. 1 if $S \in \mathcal{G}$.
9. Cost of $\mathcal{G}$ is $\sum_{S \in \mathcal{F}} Z_S$, and the expected cost is $\mathbf{E}\left[\text{cost of } \mathcal{G}\right] = \mathbf{E}\left[\sum_{S \in \mathcal{F}} Z_S\right] = \sum_{S \in \mathcal{F}} \mathbf{E}\left[Z_S\right] = \sum_{S \in \mathcal{F}} \mathbf{Pr}\left[S \in \mathcal{G}\right] = \sum_{S \in \mathcal{F}} \widehat{x_S} = \widehat{\alpha} \leq \alpha^I$.
10. In expectation, $\mathcal{G}$ is not too expensive.
11. Bigus problumos: $\mathcal{G}$ might fail to cover some element $s \in S$.

Set Cover – Rounding continued

1. **Solution**: Repeat rounding stage $m = 10\lceil \lg n \rceil = O(\log n)$ times.
2. $n = |S|$.
3. $\mathcal{G}_i$: random cover computed in ith iteration.
4. $\mathcal{H} = \cup_i \mathcal{G}_i$. Return $\mathcal{H}$ as the required cover.

The set $\mathcal{H}$ covers S

1. For an element $s \in S$, we have that

$$\sum_{U \in \mathcal{F}, s \in U} \widehat{x_U} \geq 1, \qquad (2)$$

2. probability s not covered by $\mathcal{G}_i$ (ith iteration set).

$\mathbf{Pr}\left[s \text{ not covered by } \mathcal{G}_i \right]$

$= \mathbf{Pr}[\text{ no } U \in \mathcal{F}, \text{ s.t. } s \in U \text{ picked into } \mathcal{G}_i]$

$= \prod_{U \in \mathcal{F}, s \in U} \mathbf{Pr}[U \text{ was not picked into } \mathcal{G}_i]$

$= \prod_{U \in \mathcal{F}, s \in U} (1 - \widehat{x_U}) \leq \prod_{U \in \mathcal{F}, s \in U} \exp(-\widehat{x_U})$

$= \exp\left(- \sum_{U \in \mathcal{F}, s \in U} \widehat{x_U}\right) \leq \exp(-1) \leq \frac{1}{2}, \leq \frac{1}{2}$

The set $\mathcal{H}$ covers S

Probability of a single item to be covered

1. $\mathbf{Pr}\left[s \text{ not covered by } \mathcal{G}_i \right] \leq 1/2.$
2. Number of iterations of rounding $m = O(\log n)$.
3. Covering with sets in $\mathcal{G}_1, \ldots, \mathcal{G}_m$.
4. probability s is not covered in all m iterations

$$\begin{aligned} P_s &= \mathbf{Pr}\left[s \text{ not covered by } \mathcal{G}_1, \ldots, \mathcal{F}_m \right] \\ &\leq \mathbf{Pr}\left[(s \notin \mathcal{F}_1) \cap (s \notin \mathcal{F}_2) \cap \ldots \cap (s \notin \mathcal{F}_m) \right] \\ &\leq \mathbf{Pr}\left[s \notin \mathcal{F}_1 \right] \mathbf{Pr}\left[s \notin \mathcal{F}_2 \right] \cdots \mathbf{Pr}\left[s \notin \mathcal{F}_m \right] \\ &= \frac{1}{2} \times \frac{1}{2} \times \cdots \times \frac{1}{2} = \left(\frac{1}{2}\right)^m < \frac{1}{n^{10}}, \end{aligned}$$

The set $\mathcal{H}$ covers S

Probability of all items to be covered

1. $n = |S|$,
2. Probability of $s \in S$, not to be in $\mathcal{G}_1 \cup \ldots \cup \mathcal{F}_m$ is

$$P_s < \frac{1}{n^{10}}.$$

3. probability one of n elements of S is not covered by $\mathcal{H}$ is

$$\sum_{s \in S} \mathbf{Pr}[s \notin \mathcal{G}_1 \cup \ldots \cup \mathcal{F}_m] = \sum_{s \in S} P_s < n(1/n^{10}) = 1/n^9.$$

XXX

Reminder: LP for Set Cover

$$\min \quad \alpha = \sum_{U \in \mathcal{F}} x_U,$$
$$0 \le x_U \le 1 \qquad \forall U \in \mathcal{F},$$
$$\sum_{U \in \mathcal{F}, s \in U} x_U \ge 1 \qquad \forall s \in S.$$

1. Solve the LP.
2. $\widehat{x_U}$: Value of x_u in the optimal LP solution.
3. Fractional solution: $\widehat{\alpha} = \sum_{U \in \mathcal{F}} \widehat{x_U}$.
4. Integral solution (what we want): $\alpha' \ge \widehat{\alpha}$.

Cost of solution

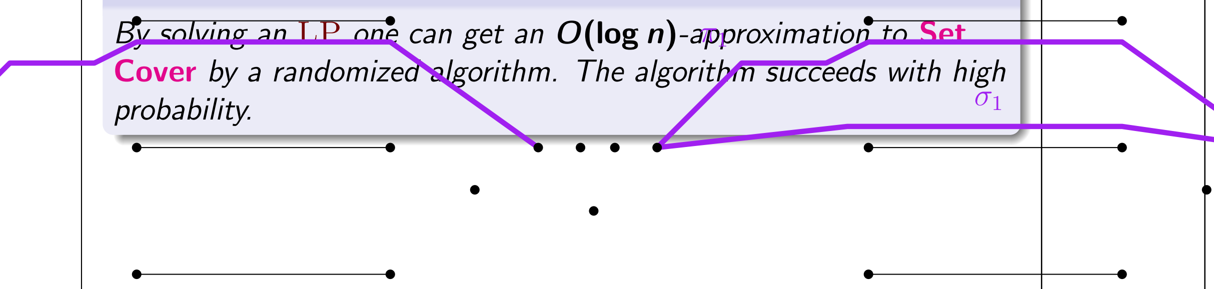
1. $(S, \mathcal{F})$: Given instance of **Set Cover**.
2. For $U \in \mathcal{F}$, $\widehat{x_U}$: LP value for set U in optimal solution.
3. For $\mathcal{G}_i$: Indicator variable $Z_u = 1 \iff U \in \mathcal{G}_i$.
4. Expected number of sets in the ith sample: $\mathbf{E}[|\mathcal{G}_i|] = \mathbf{E}\left[\sum_{U \in \mathcal{F}} Z_U\right] = \sum_{U \in \mathcal{F}} \mathbf{E}[Z_U] = \sum_{U \in \mathcal{F}} \widehat{x_U} = \widehat{\alpha} \le \alpha'$.
5. $\implies$ Each iteration expected cost of cover $\le$ cost of optimal solution (i.e., α'). XXX
6. Expected size of the solution is

$$\mathbf{E}[|\mathcal{H}|] = \mathbf{E}[|\cup_i \mathcal{G}_i|] \le \mathbf{E}\left[\sum_i |\mathcal{G}_i|\right] \le m\alpha' = O(\alpha' \log n).$$

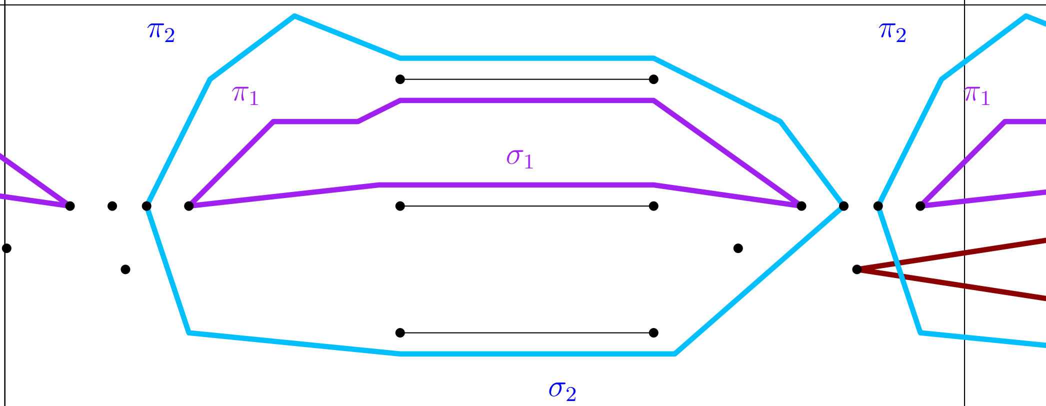
The result

Theorem

By solving an LP one can get an $O(\log n)$-approximation to **Set Cover** *by a randomized algorithm. The algorithm succeeds with high probability.*



Minimizing congestion by example



Minimizing congestion

1. G: graph. n vertices.
2. π_i, σ_i paths with the same endpoints $\mathbf{v}_i, \mathbf{u}_i \in V(G)$, for $i = 1, \ldots, t$.
3. Rule I: Send one unit of flow from $\mathbf{v}_i$ to $\mathbf{u}_i$.
4. Rule II: Choose whether to use π_i or σ_i.
5. Target: No edge in G is being used too much.

Definition

Given a set X of paths in a graph G, the **congestion** of X is the maximum number of paths in X that use the same edge.

Minimizing congestion

1. IP $\implies$ LP:

$$\begin{aligned} \min \quad & w & \\ \text{s.t.} \quad & x_i \geq 0 & i = 1, \ldots, t, \\ & x_i \leq 1 & i = 1, \ldots, t, \\ & \sum_{e \in \pi_i} x_i + \sum_{e \in \sigma_i} (1 - x_i) \leq w & \forall e \in E. \end{aligned}$$

2. $\widehat{x}_i$: value of x_i in the optimal LP solution.
3. $\widehat{w}$: value of w in LP solution.
4. Optimal congestion must be bigger than $\widehat{w}$.
5. X_i: random variable one with probability $\widehat{x}_i$, and zero otherwise.
6. If $X_i = 1$ then use π to route from $\mathbf{v}_i$ to $\mathbf{u}_i$.
7. Otherwise use σ_i.

Minimizing congestion

1. Congestion of $\mathbf{e}$ is $Y_e = \sum_{e \in \pi_i} X_i + \sum_{e \in \sigma_i} (1 - X_i)$.
2. And in expectation

$$\begin{aligned} \alpha_e = \mathbf{E}\Big[Y_e\Big] &= \mathbf{E}\Bigg[\sum_{e \in \pi_i} X_i + \sum_{e \in \sigma_i} (1 - X_i)\Bigg] \\ &= \sum_{e \in \pi_i} \mathbf{E}\Big[X_i\Big] + \sum_{e \in \sigma_i} \mathbf{E}\Big[(1 - X_i)\Big] \\ &= \sum_{e \in \pi_i} \widehat{x}_i + \sum_{e \in \sigma_i} (1 - \widehat{x}_i) \leq \widehat{w}. \end{aligned}$$

3. $\widehat{w}$: Fractional congestion (from LP solution).

Minimizing congestion - continued

1. $Y_e = \sum_{e \in \pi_i} X_i + \sum_{e \in \sigma_i} (1 - X_i)$.
2. Y_e is just a sum of independent $0/1$ random variables!
3. Chernoff inequality tells us sum can not be too far from expectation!

Minimizing congestion - continued

1. By Chernoff inequality:

$$\Pr\left[Y_e \geq (1+\delta)\alpha_e\right] \leq \exp\left(-\frac{\alpha_e \delta^2}{4}\right) \leq \exp\left(-\frac{\widehat{w}\delta^2}{4}\right).$$

2. Let $\delta = \sqrt{\frac{400}{\widehat{w}} \ln t}$. We have that

$$\Pr\left[Y_e \geq (1+\delta)\alpha_e\right] \leq \exp\left(-\frac{\delta^2 \widehat{w}}{4}\right) \leq \frac{1}{t^{100}},$$

3. If $t \geq n^{1/50} \implies \forall$ edges in graph congestion $\leq (1+\delta)\widehat{w}$.
4. t: Number of pairs, n: Number of vertices in G.

Minimizing congestion - continued

1. Got: For $\delta = \sqrt{\frac{400}{\widehat{w}} \ln t}$. We have

$$\Pr\left[Y_e \geq (1+\delta)\alpha_e\right] \leq \exp\left(-\frac{\delta^2 \widehat{w}}{4}\right) \leq \frac{1}{t^{100}},$$

2. Play with the numbers. If $t = n$, and $\widehat{w} \geq \sqrt{n}$. Then, the solution has congestion larger than the optimal solution by a factor of

$$1 + \delta = 1 + \sqrt{\frac{20}{\widehat{w}} \ln t} \leq 1 + \frac{\sqrt{20 \ln n}}{n^{1/4}},$$

which is of course extremely close to 1, if n is sufficiently large.

Minimizing congestion: result

Theorem

1. *G: Graph n vertices.*
2. $(s_1, t_1), \ldots, (s_t, t_t)$*: pairs o vertices*
3. π_i, σ_i*: two different paths connecting s_i to t_i*
4. $\widehat{w}$*: Fractional congestion at least $n^{1/2}$.*
5. opt*: Congestion of optimal solution.*
6. $\implies$ *In polynomial time (LP solving time) choose paths*
   1. *congestion $\forall$ edges: $\leq (1+\delta)\mathrm{opt}$*
   2. $\delta = \sqrt{\frac{20}{\widehat{w}} \ln t}$.

When the congestion is low

1. Assume $\widehat{w}$ is a constant.
2. Can get a better bound by using the Chernoff inequality in its more general form.
3. set $\delta = c \ln t / \ln \ln t$, where c is a constant. For $\mu = \alpha_e$, we have that

$$\begin{aligned}
\Pr\left[Y_e \geq (1+\delta)\mu\right] &\leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu \\
&= \exp\Big(\mu\big(\delta - (1+\delta)\ln(1+\delta)\big)\Big) \\
&= \exp\Big(-\mu c' \ln t\Big) \leq \frac{1}{t^{O(1)}},
\end{aligned}$$

where c' is a constant that depends on c and grows if c grows.

When the congestion is low

1. Just proved that...
2. if the optimal congestion is $O(1)$, then...
3. algorithm outputs a solution with congestion $O(\log t / \log\log t)$, and this holds with high probability.

Chernoff inequality

Problem

Let $X_1, \ldots X_n$ be n independent Bernoulli trials, where

$$\Pr\Big[X_i = 1\Big] = p_i, \qquad \Pr\Big[X_i = 0\Big] = 1 - p_i,$$

$$Y = \sum_i X_i, \qquad \text{and} \qquad \mu = \mathbf{E}\Big[Y\Big].$$

We are interested in bounding the probability that $Y \geq (1+\delta)\mu$.

Chernoff inequality

Theorem (Chernoff inequality)

For any $\delta > 0$,

$$\Pr\Big[Y > (1+\delta)\mu\Big] < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu.$$

Or in a more simplified form, for any $\delta \leq 2e - 1$,

$$\Pr\Big[Y > (1+\delta)\mu\Big] < \exp\left(-\mu\delta^2/4\right),$$

and

$$\Pr\Big[Y > (1+\delta)\mu\Big] < 2^{-\mu(1+\delta)},$$

for $\delta \geq 2e - 1$.

More Chernoff...

Theorem

Under the same assumptions as the theorem above, we have

$$\Pr\Big[Y < (1-\delta)\mu\Big] \leq \exp\left(-\mu\frac{\delta^2}{2}\right).$$