Chapter 18

Approximation Algorithms using Linear Programming

CS 573: Algorithms, Fall 2014
October 28, 2014

Part I

Weighted vertex cover

18.1 Weighted vertex cover

18.1.0.1 Weighted vertex cover

Weighted Vertex Cover problem $G = (V, E)$.

Each vertex $v \in V$: cost c_v.

Compute a vertex cover of minimum cost.

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

18.1.1 Weighted vertex cover

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

$$\begin{aligned} \min \quad & \sum_{v \in V} c_v x_v, & \\ \text{such that} \quad & x_v \in \{0,1\} & \forall v \in V \\ & x_v + x_u \geq 1 & \forall vu \in E. \end{aligned} \tag{18.1}$$

(A) ... **NP-Hard**.
(B) relax the integer program.
(C) allow x_v get values $\in [0, 1]$.
(D) $x_v \in \{0,1\}$ replaced by $0 \leq x_v \leq 1$. The resulting LP is

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

18.1.1.2 Weighted vertex cover – rounding the LP

(A) Optimal solution to this LP: $\widehat{x_v}$ value of var X_v, $\forall v \in V$.
(B) optimal value of LP solution is $\widehat{\alpha} = \sum_{v \in V} c_v \widehat{x_v}$.
(C) optimal integer solution: x_v^I, $\forall v \in V$ and α^I.
(D) **Any valid solution to IP is valid solution for LP!**
(E) $\widehat{\alpha} \leq \alpha^I$.
Integral solution not better than LP.
(F) Got fractional solution (i.e., values of $\widehat{x_v}$).
(G) Fractional solution is better than the optimal cost.
(H) Q: How to turn fractional solution into a (valid!) integer solution?
(I) Using ***rounding***.

18.1.1.3 How to round?

(A) consider vertex v and fractional value $\widehat{x_v}$.
(B) If $\widehat{x_v} = 1$ then include in solution!
(C) If $\widehat{x_v} = 0$ then do **not**not include in solution.

(D) if $\widehat{x_\mathsf{v}} = 0.9 \implies$ LP considers v as being 0.9 useful.
(E) The LP puts its money where its belief is...
(F) ...$\widehat{\alpha}$ value is a function of this "belief" generated by the LP.
(G) **Big idea**: Trust LP values as guidance to usefulness of vertices.
(H) Pick all vertices $\geq$ threshold of usefulness according to LP.
(I) $S = \left\{ \mathsf{v} \,\middle|\, \widehat{x_\mathsf{v}} \geq 1/2 \right\}$.
(J) **Claim**: S a valid vertex cover, and cost is low.
(K) Indeed, edge cover as: $\forall \mathsf{vu} \in \mathsf{E}$ have $\widehat{x_\mathsf{v}} + \widehat{x_\mathsf{u}} \geq 1$.
(L) $\widehat{x_\mathsf{v}}, \widehat{x_\mathsf{u}} \in (0,1)$
$\implies \widehat{x_\mathsf{v}} \geq 1/2$ or $\widehat{x_\mathsf{u}} \geq 1/2$.
$\implies \mathsf{v} \in S$ or $\mathsf{u} \in S$ (or both).
$\implies S$ covers all the edges of G.

18.1.1.4 Cost of solution

Cost of S:

$$\mathsf{c}_S = \sum_{\mathsf{v} \in S} \mathsf{c}_\mathsf{v} = \sum_{\mathsf{v} \in S} 1 \cdot \mathsf{c}_\mathsf{v} \leq \sum_{\mathsf{v} \in S} 2\widehat{x_\mathsf{v}} \cdot \mathsf{c}_\mathsf{v} \leq 2 \sum_{\mathsf{v} \in \mathsf{V}} \widehat{x_\mathsf{v}} \mathsf{c}_\mathsf{v} = 2\widehat{\alpha} \leq 2\alpha^I,$$

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

Theorem 18.1.1. *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.*

18.1.2 The lessons we can take away

18.1.2.1 Or not - boring, boring, boring.

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

18.2 Revisiting Set Cover

18.2.0.2 Revisiting Set Cover

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

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.

18.2.0.3 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.

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

18.2.0.4 Set Cover – IP & LP

(A) LP solution: $\forall U \in \mathcal{F}$, $\widehat{x_U}$, and $\widehat{\alpha}$.
(B) Opt IP solution: $\forall U \in \mathcal{F}$, x_U^I, and α^I.
(C) Use LP solution to guide in rounding process.
(D) If $\widehat{x_U}$ is close to 1 then pick U to cover.
(E) If $\widehat{x_U}$ close to 0 do not.
(F) **Idea:** Pick $U \in \mathcal{F}$: randomly choose U with ***probability*** $\widehat{x_U}$.
(G) Resulting family of sets $\mathcal{G}$.
(H) Z_S: indicator variable. 1 if $S \in \mathcal{G}$.
(I) Cost of $\mathcal{G}$ is $\sum_{S \in \mathcal{F}} Z_S$, and the expected cost is $\mathbf{E}\left[\text{cost of } \mathcal{G}\right] = \mathbf{E}\left[\sum_{S \in \mathcal{F}} Z_S\right] = \sum_{S \in \mathcal{F}} \mathbf{E}\left[Z_S\right] = \sum_{S \in \mathcal{F}} \mathbf{Pr}\left[S \in \mathcal{G}\right] = \sum_{S \in \mathcal{F}} \widehat{x_S} = \widehat{\alpha} \leq \alpha^I$.
(J) In expectation, $\mathcal{G}$ is not too expensive.
(K) Bigus problumos: $\mathcal{G}$ might fail to cover some element $s \in S$.

18.2.0.5 Set Cover – Rounding continued

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

18.2.0.6 The set $\mathcal{H}$ covers S

(A) For an element $s \in S$, we have that

$$\sum_{U \in \mathcal{F}, s \in U} \widehat{x_U} \geq 1, \tag{18.2}$$

(B) probability s not covered by $\mathcal{G}_i$ (ith iteration set).
$\mathbf{Pr}\left[s \text{ not covered by } \mathcal{G}_i\right]$

$= \mathbf{Pr}\left[\text{ no } U \in \mathcal{F}, \text{ s.t. } s \in U \text{ picked into } \mathcal{G}_i \right]$
$= \prod_{U \in \mathcal{F}, s \in U} \mathbf{Pr}\left[U \text{ was not picked into } \mathcal{G}_i \right]$
$= \prod_{U \in \mathcal{F}, s \in U} (1 - \widehat{x_U}) \leq \prod_{U \in \mathcal{F}, s \in U} \exp(-\widehat{x_U})$
$= \exp\left(- \sum_{U \in \mathcal{F}, s \in U} \widehat{x_U} \right) \leq \exp(-1) \leq \frac{1}{2}, \leq \frac{1}{2}$

(C) probability s is not covered in all m iterations $\leq \left(\frac{1}{2}\right)^m < \frac{1}{n^{10}}$,
(D) ...since $m = O(\log n)$.
(E) probability one of n elements of S is not covered by $\mathcal{H}$ is $\leq n(1/n^{10}) = 1/n^9$.

18.2.0.7 Cost of solution

(A) Have: $\mathbf{E}\left[\text{cost of } \mathcal{G}_i\right] \leq \alpha^I$.
(B) $\implies$ Each iteration expected cost of cover $\leq$ cost of optimal solution (i.e., α^I).
(C) Expected cost of the solution is

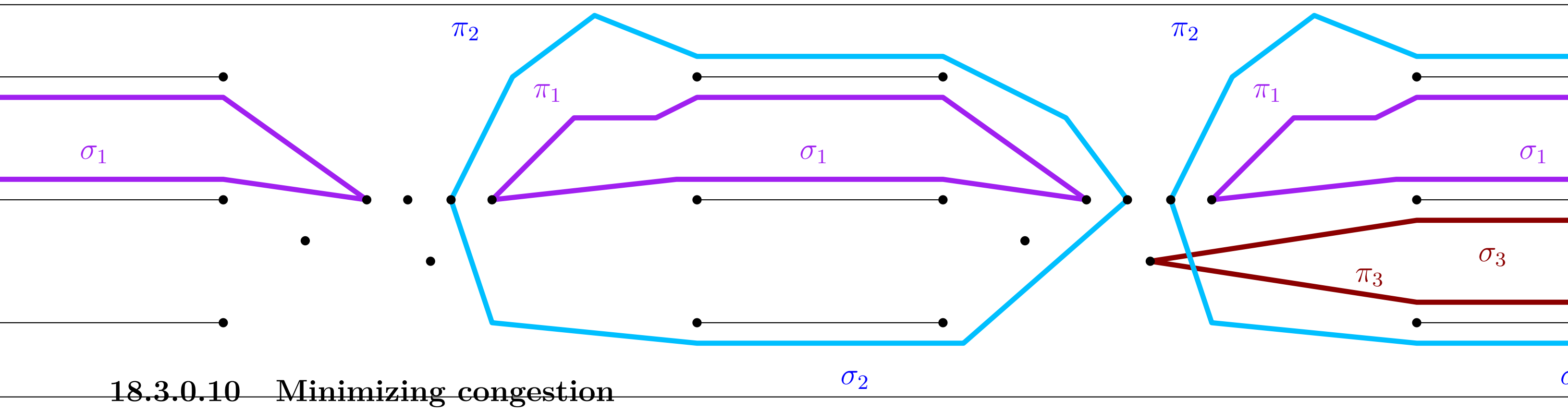
$$c_{\mathcal{H}} \leq \sum_i c_{B_i} \leq m\alpha^I = O\left(\alpha^I \log n\right).$$

18.2.0.8 The result

Theorem 18.2.1. *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.*

18.3 Minimizing congestion

18.3.0.9 Minimizing congestion by example



18.3.0.10 Minimizing congestion

(A) G: graph. n vertices.
(B) π_i, σ_i paths with the same endpoints $v_i, u_i \in V(G)$, for $i = 1, \ldots, t$.
(C) Rule I: Send one unit of flow from v_i to u_i.
(D) Rule II: Choose whether to use π_i or σ_i.
(E) Target: No edge in G is being used too much.

Definition 18.3.1. 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.

18.3.0.11 Minimizing congestion

(A) IP $\implies$ LP:

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

(B) $\widehat{x_i}$: value of x_i in the optimal LP solution.
(C) $\widehat{w}$: value of w in LP solution.
(D) Optimal congestion must be bigger than $\widehat{w}$.
(E) X_i: random variable one with probability $\widehat{x_i}$, and zero otherwise.
(F) If $X_i = 1$ then use π to route from v_i to u_i.
(G) Otherwise use σ_i.

18.3.0.12 Minimizing congestion

(A) Congestion of e is $Y_{\mathsf{e}} = \sum_{\mathsf{e} \in \pi_i} X_i + \sum_{\mathsf{e} \in \sigma_i} (1 - X_i)$.
(B) And in expectation

$$\begin{aligned} \alpha_{\mathsf{e}} = \mathbf{E}\Big[Y_{\mathsf{e}}\Big] &= \mathbf{E}\Bigg[\sum_{\mathsf{e} \in \pi_i} X_i + \sum_{\mathsf{e} \in \sigma_i} (1 - X_i)\Bigg] \\ &= \sum_{\mathsf{e} \in \pi_i} \mathbf{E}\Big[X_i\Big] + \sum_{\mathsf{e} \in \sigma_i} \mathbf{E}\Big[(1 - X_i)\Big] \\ &= \sum_{\mathsf{e} \in \pi_i} \widehat{x_i} + \sum_{\mathsf{e} \in \sigma_i} (1 - \widehat{x_i}) \leq \widehat{w}. \end{aligned}$$

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

18.3.0.13 Minimizing congestion - continued

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

18.3.0.14 Minimizing congestion - continued

(A) By Chernoff inequality:

$$\mathbf{Pr}\Big[Y_{\mathsf{e}} \geq (1 + \delta)\alpha_{\mathsf{e}}\Big] \leq \exp\left(-\frac{\alpha_{\mathsf{e}} \delta^2}{4}\right) \leq \exp\left(-\frac{\widehat{w} \delta^2}{4}\right).$$

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

$$\mathbf{Pr}\Big[Y_{\mathsf{e}} \geq (1 + \delta)\alpha_{\mathsf{e}}\Big] \leq \exp\left(-\frac{\delta^2 \widehat{w}}{4}\right) \leq \frac{1}{t^{100}},$$

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

18.3.0.15 Minimizing congestion - continued

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

$$\mathbf{Pr}\Big[Y_{\mathsf{e}} \geq (1+\delta)\alpha_{\mathsf{e}}\Big] \leq \exp\left(-\frac{\delta^2 \widehat{w}}{4}\right) \leq \frac{1}{t^{100}},$$

(B) 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.

18.3.0.16 Minimizing congestion: result

Theorem 18.3.2. *(A)* G*: Graph n vertices.*
(B) $(s_1, t_1), \ldots, (s_t, t_t)$*: pairs o vertices*
(C) π_i, σ_i*: two different paths connecting s_i to t_i*
(D) $\widehat{w}$*: Fractional congestion at least $n^{1/2}$.*
(E) opt*: Congestion of optimal solution.*
(F) $\implies$ *In polynomial time (LP solving time) choose paths*
  (A) congestion $\forall$ edges: $\leq (1+\delta)\mathrm{opt}$
  (B) $\delta = \sqrt{\frac{20}{\widehat{w}} \ln t}.$

18.3.0.17 When the congestion is low

(A) Assume $\widehat{w}$ is a constant.
(B) Can get a better bound by using the Chernoff inequality in its more general form.
(C) set $\delta = c \ln t / \ln \ln t$, where c is a constant. For $\mu = \alpha_{\mathsf{e}}$, we have that

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

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

18.3.0.18 When the congestion is low

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

18.4 Reminder about Chernoff inequality

18.4.0.19 The Chernoff Bound — General Case
18.4.0.20 Chernoff inequality

Problem 18.4.1. Let $X_1, \ldots X_n$ be n independent Bernoulli trials, where

$$\mathbf{Pr}\Big[X_i = 1\Big] = p_i, \qquad \mathbf{Pr}\Big[X_i = 0\Big] = 1 - p_i,$$
$$Y = \sum_i X_i, \qquad \text{and} \qquad \mu = \mathbf{E}\Big[Y\Big].$$

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

18.4.0.21 Chernoff inequality

Theorem 18.4.2 (Chernoff inequality). *For any* $\delta > 0$*,*

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

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

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

and

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

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

18.4.0.22 More Chernoff...

Theorem 18.4.3. *Under the same assumptions as the theorem above, we have*

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