

Randomized Rounding

Lecture 1

Probability

Events, probabilities,
conditional probabilities,
union bound

Random variables,
expected value,
linearity of expectation
Markov's inequality

Algorithms

Approx. 3-SAT,

Big idea: we only want the average

$$\begin{array}{l} \text{Linearity of Expectation} \\ E[X+Y] = E[X] + E[Y] \quad (\text{stochastic variables}) \\ \text{"average sum = sum of averages"} \end{array}$$

let $Y_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{otherwise} \end{cases}$ ($i=1, \dots, m$)

$$E[Y_i] = 7/8 \quad (1 \text{ clause case})$$

$$E[\text{\# clauses satisfied}] = E\left[\sum_{i=1}^m Y_i\right] = \sum_{i=1}^m E[Y_i] = 7/8 m$$

let $Y_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{otherwise} \end{cases}$

$$Z = \sum_{i=1}^m Y_i = \text{\# clauses satisfied}$$

$$E[Z] = \frac{1}{2} E[Z | x_1 = \text{true}] + \frac{1}{2} E[Z | x_1 = \text{false}]$$

$\Rightarrow$ either $E[Z | x_1 = \text{true}]$ or $E[Z | x_1 = \text{false}]$ is $\geq E[Z]$!

QuickSort,

Random divide and conquer

① pick random 'pivot'

② Divide A into two parts



Expected running time of QuickSort

For each $i, j \in [n]$ w/ $i < j$,

$X_{ij} = \begin{cases} 1 & \text{if } i\text{th smallest element is compared to } j\text{th smallest} \\ 0 & \text{otherwise} \end{cases}$

$\sum_{i < j} X_{ij}$ = total # comparisons made by the algorithm
 $\approx$ total running time

$E[\sum_{i < j} X_{ij}]$ = average running time

$$E[\sum_{i < j} X_{ij}] = \sum_{i < j} E[X_{ij}]$$

now analyze $E[X_{ij}]$ in isolation

$$E[X_{ij}] = \Pr[X_{ij} = 1] = \Pr[i\text{th element is compared to } j\text{th}]$$

i th and j th elem selected before any element in between happens w/ probability $\frac{2}{j-i+1}$

$$E[X_{ij}] = \frac{2}{j-i+1}$$

by linearity of expectation

$$E[\sum_{i < j} X_{ij}] = \sum_{i < j} \frac{2}{j-i+1}$$

$$= 2 \sum_{i=1}^n \sum_{j=i+1}^n \frac{1}{j-i+1} = 2 \sum_{k=1}^{n-1} \sum_{i=1}^k \frac{1}{k+1-i+1}$$

$$= 2 \sum_{k=1}^{n-1} \sum_{i=1}^k \frac{1}{k+1-i+1} = 2 \sum_{k=1}^{n-1} \sum_{i=1}^k \frac{1}{k+1-i+1}$$

$$= 2 \sum_{k=1}^{n-1} \sum_{i=1}^k \frac{1}{k+1-i+1} = 2 \sum_{k=1}^{n-1} \sum_{i=1}^k \frac{1}{k+1-i+1}$$

$$= 2 \sum_{k=1}^{n-1} \sum_{i=1}^k \frac{1}{k+1-i+1} = 2 \sum_{k=1}^{n-1} \sum_{i=1}^k \frac{1}{k+1-i+1}$$

$$= 2 \sum_{k=1}^{n-1} \sum_{i=1}^k \frac{1}{k+1-i+1} = 2 \sum_{k=1}^{n-1} \sum_{i=1}^k \frac{1}{k+1-i+1}$$

Quickselect (HW)

Lecture 2

Probability

Hash Functions,

Universal Hash Functions

$h: [n] \rightarrow k$ s.t. for all pairs $i, j \in [n]$,
 $\Pr[h(i) = h(j)] \leq \frac{1}{k}$ (much weaker than ideal hash)

e.g. $h(x) = (ax + b) \bmod p \bmod k$

where • p prime, $> k$

• a in $\{1, \dots, p-1\}$ unif. random

• b in $\{0, \dots, p-1\}$ unif. random



Markov's Inequality

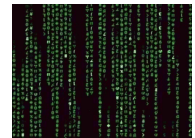
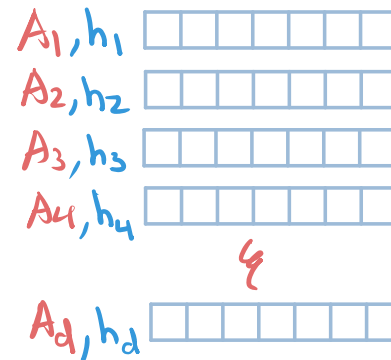
Markov's Inequality (special case)

Let $X \geq 0$ be nonnegative RV

$$\Pr[X \geq 2E[X]] \leq \frac{1}{2}$$

Algorithms

Frequency Estimation, Heavy Hitters (in streams)



estimate f_i w/
 $\min_{j=1, \dots, d} A_j[h_j(i)]$

Lecture 3 (last Tuesday)

Probability

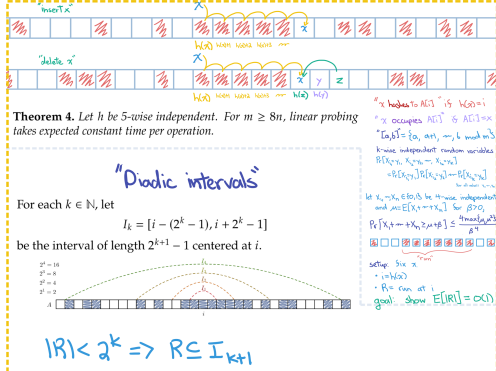
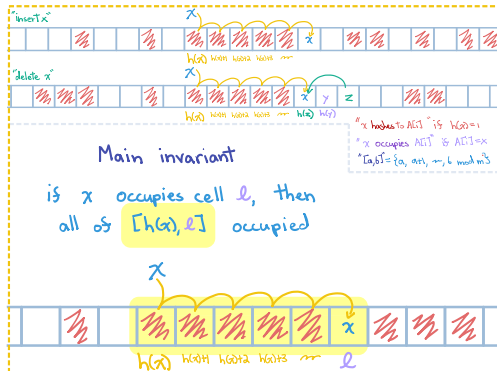
k-wise independence,
higher order moments
(generalizing universal hash
functions and Markov's inequality)

Algorithms

Hash tables w/
chaining +
linear probing

k-wise independent hash functions
 $h: [U] \rightarrow [m]$
s.t. for any k keys $x_1, x_2, \dots, x_k \in [U]$,
 k values $v_1, v_2, \dots, v_k \in [m]$,
 $P[h(x_1)=v_1, h(x_2)=v_2, \dots, h(x_k)=v_k] = \frac{1}{m^k}$
eg. $h(x) = a_0 + a_1x + a_2x^2 + \dots + a_{k-1}x^{k-1} \pmod p$
where $p \geq U$ prime,
 $a_0, a_1, \dots, a_{k-1} \in [p]$ indep, uniformly random

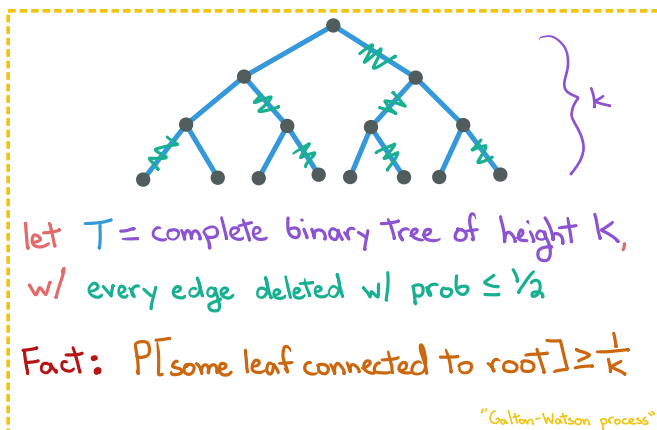
Important lemma
k-wise independent random variables
 $P[X_1=y_1, X_2=y_2, \dots, X_k=y_k]$
 $= P[X_1=y_1] P[X_2=y_2] \dots P[X_k=y_k]$
for all values $y_1, \dots, y_k$
let $X_1, \dots, X_n \in \{0,1\}$ be 4-wise independent
and $\mu = E[X_1 + \dots + X_n]$. then for $\beta > 0$,
 $P[X_1 + \dots + X_n \geq \mu + \beta] \leq \frac{4 \max\{\mu, \mu^3\}}{\beta^4}$



Lecture 4 (last Thursday)

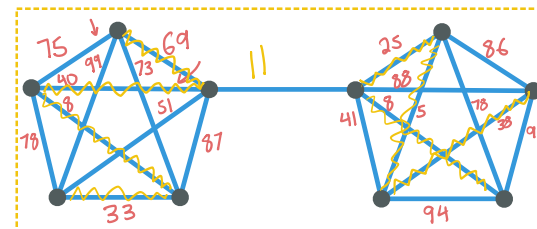
Probability

Galton-Watson processes



Algorithms

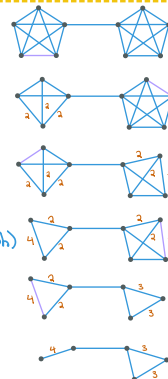
Random Contractions for min cut



Randomized contractions

Input: $G=(V,E)$, weights $c \in \mathbb{R}_{>0}^E$

- ① while $|E| > 1$
 - Ⓐ sample $e \in E$ in proportion to c
 - Ⓑ "contract" e
- ② return set of edges (in original graph) contracted to remaining edge

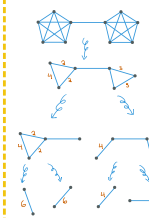


Amplification by branching

Instead of restarting at the beginning, restart partway through

High-level algorithm

- ① randomly contract until prob. of ruining the min-cut is $\sim 1/2$
- ② Branch:
make two copies of current graph
recurse on both copies



Lecture 5

Probability

Multiplicative Chernoff Bound

Multiplicative Chernoff bound

let $X_1, \dots, X_n \in [0, 1]$ be independent

$$\mu = E[X_1 + \dots + X_n]$$

$$\epsilon \in (0, 1)$$

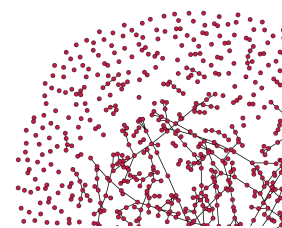
- $P[X_1 + \dots + X_n \geq (1 + \epsilon)\mu] \leq e^{-\epsilon^2 \mu / 3}$
- $P[X_1 + \dots + X_n \leq (1 - \epsilon)\mu] \leq e^{-\epsilon^2 \mu / 2}$

Algorithms (/ results)

Random graphs & the giant component

Erdős-Rényi graphs $G(n, p)$

- n vertices
- each edge sampled independently w/ probability p



Theorem 5.3. Consider a random graph $G \sim \mathcal{G}(n, p)$ for $p = c/n$, where c is a constant.

- If $c > 1$, then with high probability, there is exactly one connected component of G with $\Omega(n)$ vertices, and all other components have size $\leq O(\log n)$.
- For $c < 1$, then with high probability, all connected components of G has size $< O(\log n)$.

Lecture 6 (today)

Probability (& techniques)

Approximations via
linear programming
& randomized rounding

Algorithms

Max Sat

Set Cover

Covering
Integer Programs

Lecture 6 (today)

Probability (& techniques)

Approximations via
linear programming
& randomized rounding

Algorithms

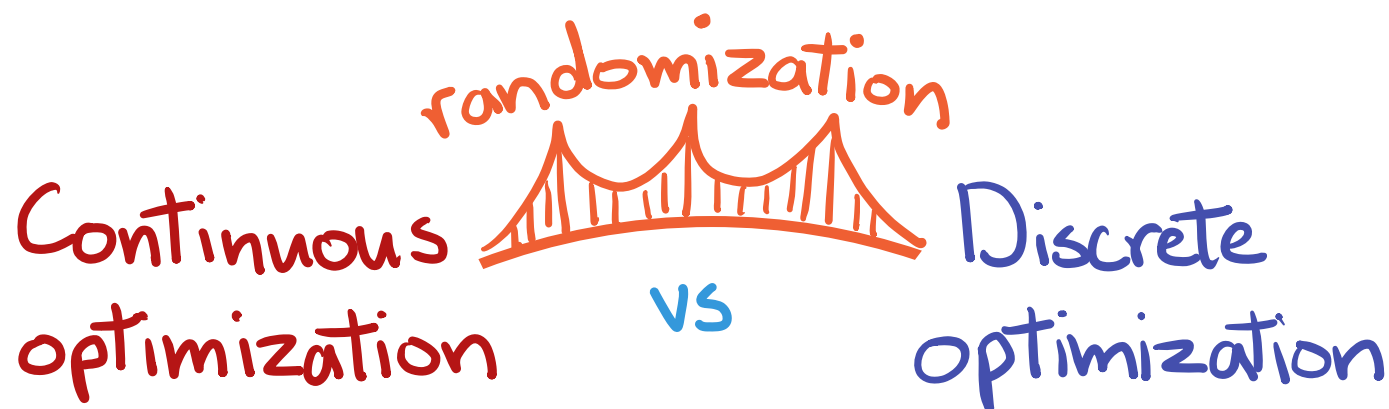
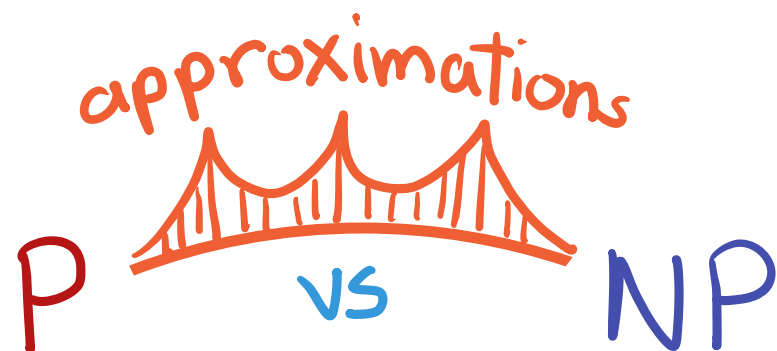
Max Sat

Set Cover

Covering
Integer Programs

Randomized Rounding

Story of



Integer Programs (IPs)

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

linear objective

over $x \in \mathbb{Z}^n$

integral variables

s.t. $A_1 x \leq b_1$
 $A_2 x = b_2$
 $A_3 x \geq b_3$

linear
constraints

for fixed vectors c, b_1, b_2, b_3 and matrices A_1, A_2, A_3

NP-Hard

Linear programs (LPs)

similar to IPs, except no integrality constraints

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

linear objective

$$\text{Over } x \in \mathbb{R}^n$$

~~linear~~
~~integral~~ variables

$$\text{s.t. } A_1 x \leq b_1$$

$$A_2 x = b_2$$

$$A_3 x \geq b_3$$

linear
constraints

for fixed vectors c, b_1, b_2, b_3 and matrices A_1, A_2, A_3

Linear Programs are in P (!!)

↑
(weakly polynomial)

1947 - Simplex algorithm
- Dantzig

(not polytime but practical)

1979 - Ellipsoid algorithm
- Khachiyan

polytime

1984 - Interior point algorithm
- Karmarkar

what's the difference?

Discrete
optimization

vs

Continuous
optimization

Integer
Programs

vs

Linear
Programs



NP-Hard



polynomial time!

LP-Based Approximations

① model as IP

$$\begin{array}{ll}\max & \sum_{j=1}^n c_j x_j \\ \text{Over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1\end{array}$$

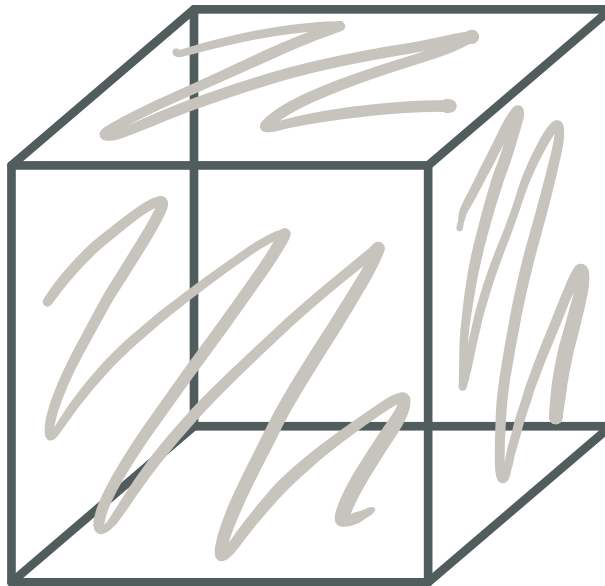


$$\begin{array}{ll}\max & \sum_{j=1}^n c_j x_j \\ \text{Over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1\end{array}$$

② relax to LP



LP solver



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$



Lecture 6 (today)

Probability (& techniques)

Approximations via
linear programming
& randomized rounding

Algorithms

Max Sat

Set Cover

Covering
Integer Programs

3-SAT

"conjunctive normal form"

"conjunctive normal form"
CNF formula

$$\varphi(x_1, \dots, x_n) = (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_4 \vee \dots) \wedge \dots$$

m clauses, n variables

3 (distinct) variables per clause

The silliest APX algorithm

1. for $i = 1, \dots, n$

a. randomly set $x_i = \begin{cases} \text{TRUE} & \text{w/ prob .5} \\ \text{FALSE} & \text{w/ prob .5} \end{cases}$

We will:

① prove silly algo is $\frac{7}{8}$ -APX

② derandomize silly algo $\Rightarrow$ deterministic $7/8$ -APX

Even more surprising:

"probabilistically checkable proofs"

PCP Theorem

$(\frac{7}{8} + \epsilon)$ -APX is NP-Hard $\forall \epsilon > 0$

Approximating 3SAT: "Max 3-SAT"

$$\varphi(x_1, \dots, x_n) = (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_4 \vee \dots) \wedge \dots$$

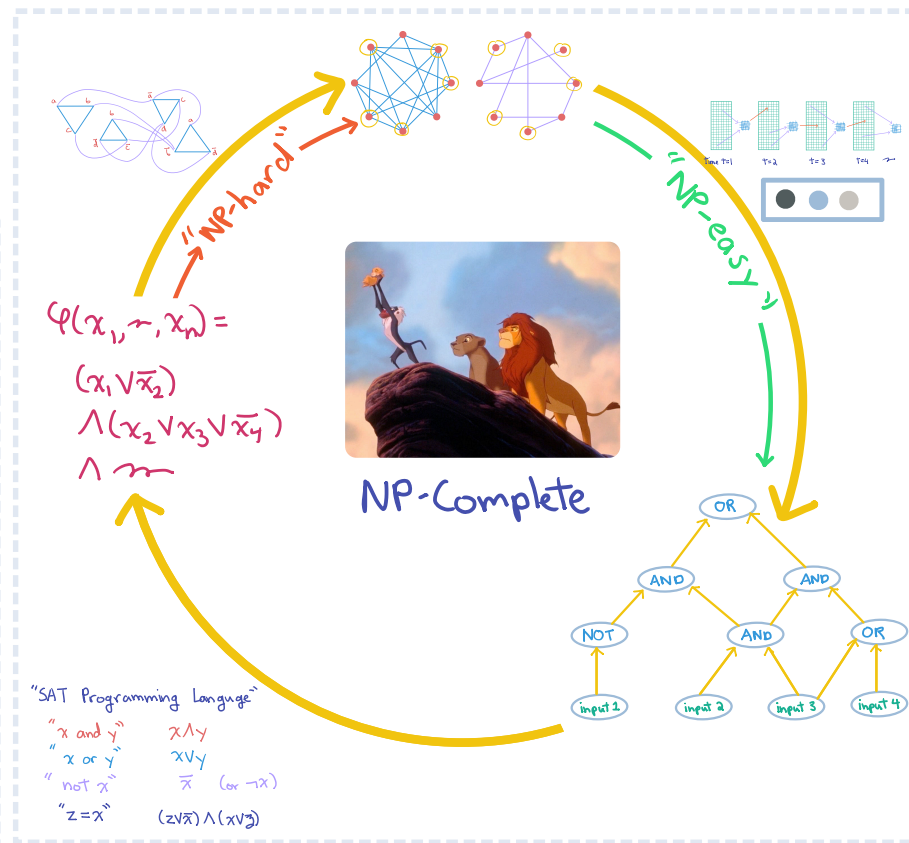
Goal: satisfy Max # clauses

Interesting even if ϕ is unsatisfiable

"d-Approximation algorithm" $d \in (0, 1)$

Algorithm satisfies $d \cdot \text{OPT}$ clauses

Optimum (max) value



2 disjoint clauses

① prove silly algo is $\frac{7}{8}$ -APX

1. for $i=1, \dots, n$ The silliest APX algorithm
 a. randomly set $x_i = \begin{cases} \text{TRUE} & \text{w/ prob. } .5 \\ \text{FALSE} & \text{w/ prob. } .5 \end{cases}$

$$\mathcal{S}(x) = (\underbrace{x_1 \vee x_2 \vee \bar{x}_3}_{C_1}) \wedge (\underbrace{x_4 \vee \bar{x}_5 \vee x_6}_{C_2})$$

E_1 = event that C_1 satisfied

E_2 = $\sim$ C_2 $\sim$

$$\Pr[E_1, E_2] = \frac{7}{8} \cdot \frac{7}{8}$$

$$= \Pr[E_1] \cdot \Pr[E_2]$$

2 clauses w/
1 common variable

① prove silly algo is $\frac{7}{8}$ -APX

1. for $i=1, \dots, n$ The silliest APX algorithm
 a. randomly set $x_i = \begin{cases} \text{TRUE} & \text{w/ prob. } .5 \\ \text{FALSE} & \text{w/ prob. } .5 \end{cases}$

$$\mathcal{S}(x) = (\underbrace{x_1 \vee \bar{x}_2 \vee x_3}_{C_1}) \wedge (\underbrace{x_3 \vee x_4 \vee \bar{x}_5}_{C_2})$$

E_1 = event that C_1 satisfied

E_2 = $\sim$ C_2 $\sim$

$$\Pr[E_1, E_2] =$$

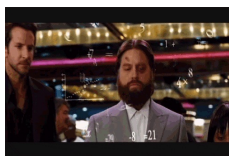
$$\underbrace{\Pr[E_1, E_2 | x_3 = T] \cdot \frac{1}{2}}_1 + \underbrace{\Pr[E_1, E_2 | x_3 = F] \cdot \frac{1}{2}}_{(\frac{3}{4})^2}$$

m clauses w/
common variables

① prove silly algo is $\frac{7}{8}$ -APX

1. for $i=1, \dots, n$ The silliest APX algorithm
 a. randomly set $x_i = \begin{cases} \text{TRUE} & \text{w/ prob. } .5 \\ \text{FALSE} & \text{w/ prob. } .5 \end{cases}$

$$\mathcal{S}(x) = (\underbrace{x_1 \vee \bar{x}_2 \vee x_3}_{C_1}) \wedge (\underbrace{x_3 \vee x_4 \vee \bar{x}_5}_{C_2}) \wedge (\underbrace{x_2 \vee \bar{x}_4 \vee x_6}_{C_3}) \wedge \dots$$



E_1 = event that C_1 satisfied

E_2 = $\sim$ C_2 $\sim$

E_3 = $\sim$ C_3 $\sim$

$\vdots$

general question

how to analyze a randomized mechanism
w/ a combinatorial explosion of possibilities?

Big idea: we only want the average

Linearity of Expectation

$$E[X+Y] = E[X] + E[Y] \quad (\text{ridiculously useful})$$

"average sum = sum of averages"

$$\text{let } Y_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{otherwise} \end{cases}$$

($i=1, \dots, m$)

$$E[Y_i] = \frac{7}{8} \quad (1 \text{ clause case})$$

$$E[\text{\# clauses satisfied}] = E\left[\sum_{i=1}^m Y_i\right] = \sum_{i=1}^m E[Y_i] = \frac{7}{8}m$$

(actually easy!)

Max-SAT

"conjunctive normal form"

CNF formula

$$\varphi(x_1, \dots, x_n) = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_4 \vee \dots) \wedge \dots$$

m clauses, n variables

any # of variables per clause

goal: satisfy max # clauses

Approximation ratios for k-SAT vs max-SAT

k-sat

max-SAT

k oblivious APX

$$1 \quad 1/2$$

$$2 \quad 3/4$$

$$3 \quad 7/8$$

$$4 \quad 15/16$$

$$5 \quad 31/32$$

$\vdots$

$$k \quad 1 - 1/2^k$$

$$1/2$$

better?

LP-Based Approximations

① model as IP

$$\begin{array}{ll}\max & \sum_{j=1}^n c_j x_j \\ \text{Over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1\end{array}$$

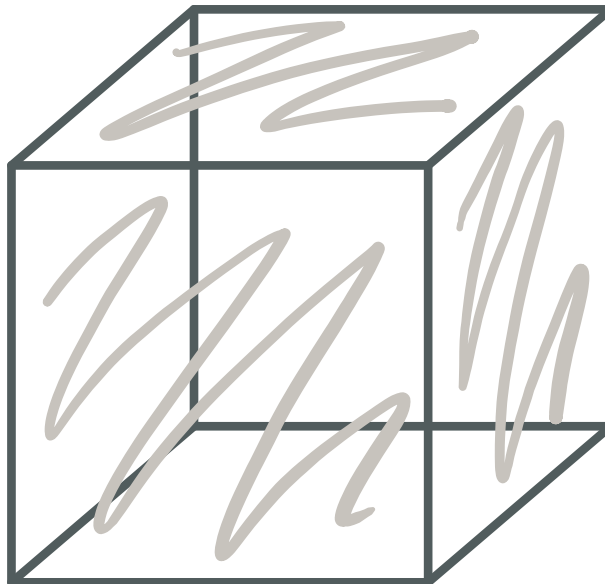


$$\begin{array}{ll}\max & \sum_{j=1}^n c_j x_j \\ \text{Over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1\end{array}$$

② relax to LP



LP solver



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$



IP for max-SAT

$$\max \sum_{i=1}^m z_i$$

$$\text{over } y \in \{0,1\}^n, z \in \{0,1\}^m$$

$$\text{s.t. } \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i$$

for each clause C_i

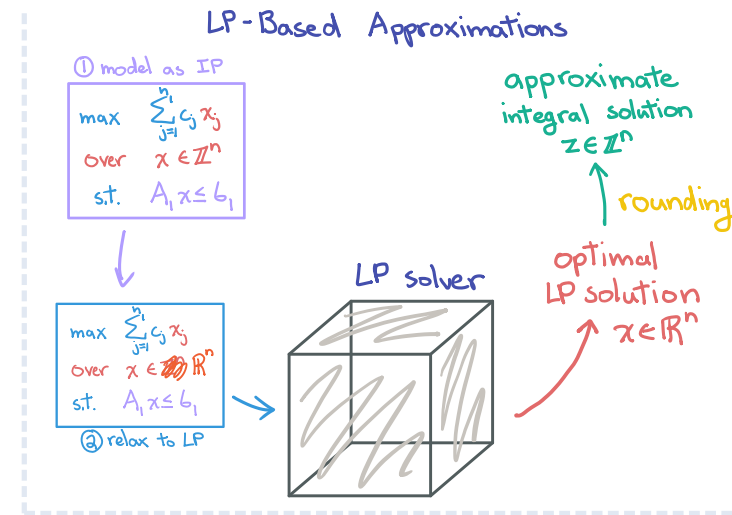
$$z_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{if not} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if } x_j = \text{TRUE} \\ 0 & \text{if } x_j = \text{FALSE} \end{cases}$$

e.g. $C_i = x_1 \vee \bar{x}_2 \vee x_3$

$\Rightarrow$ constraint

$$y_1 + (1 - y_2) + y_3 \geq z_1$$



LP for max-SAT

$$\max \sum_{i=1}^m z_i$$

$$\text{over } y \in [0,1]^n, z \in [0,1]^m$$

$$\text{s.t. } \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i \quad \text{for each clause } C_i$$

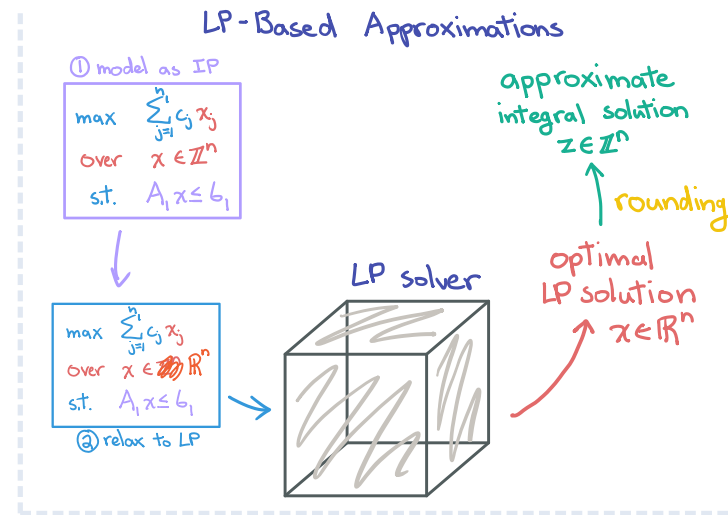
$$z_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{if not} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if } x_j = \text{TRUE} \\ 0 & \text{if } x_j = \text{FALSE} \end{cases}$$

e.g. $C_i = x_1 \vee \bar{x}_2 \vee x_3$

$\Rightarrow$ constraint

$$y_1 + (1 - y_2) + y_3 \geq z_1$$



IP for max-SAT

$$\max \sum_{i=1}^m z_i$$

over $y \in \{0,1\}^n, z \in \{0,1\}^m$

$$\text{s.t. } \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i$$

NP-Hard

LP for max-SAT

$$\max \sum_{i=1}^m z_i$$

over $y \in [0,1]^n, z \in [0,1]^m$

$$\text{s.t. } \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i$$

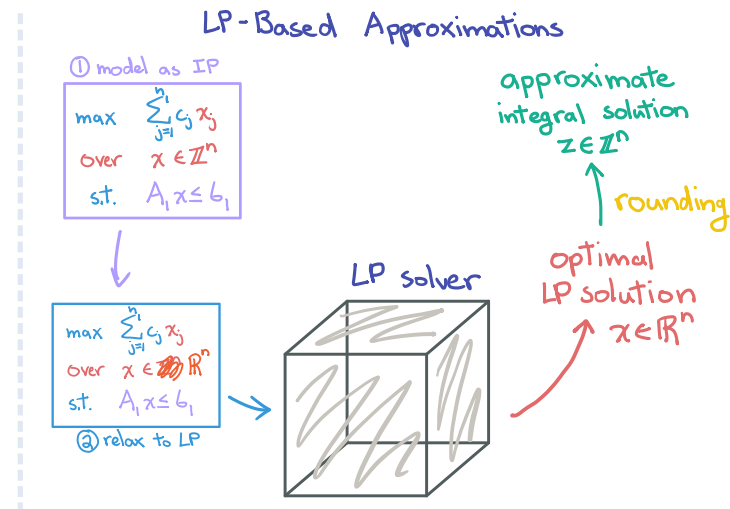
Solvable

$$\text{OPT}_{\text{IP}} \leq \text{OPT}_{\text{LP}}$$

Rounding LP solution

$$y \in [0,1]^n \rightsquigarrow x \in \{\text{TRUE}, \text{FALSE}\}$$

ideas?



$$\begin{array}{ll} \max & \sum_{i=1}^m z_i \\ \text{over} & y \in [0,1]^n, z \in [0,1]^m \\ \text{s.t.} & \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i \\ & \text{for each clause } C_i \end{array}$$

$z_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{if not} \end{cases}$

$y_j = \begin{cases} 1 & \text{if } x_j = \text{TRUE} \\ 0 & \text{if } x_j = \text{FALSE} \end{cases}$

e.g., how to interpret $y_i = \frac{1}{2}$?

Rounding LP solution

$$y \in [0,1]^n \rightsquigarrow x \in \{\text{TRUE}, \text{FALSE}\}$$

For each variable x_j , independently, randomly set $x_j = \text{t}$ with probability y_j , and $x_j = \text{f}$ otherwise.

LP-Based Approximations

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$



$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



LP solver

approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$

$$\max \sum_{i=1}^m z_i$$

$$\text{over } y \in [0,1]^n, z \in [0,1]^m$$

$$\text{s.t. } \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i$$

for each clause C_i

$$z_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{if not} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if } x_j = \text{TRUE} \\ 0 & \text{if } x_j = \text{FALSE} \end{cases}$$

For each variable x_j , independently, randomly set $x_j = t$ with probability y_j , and $x_j = f$ otherwise.

claim: each clause C_i is satisfied
w/ probability $(1 - 1/e) z_i$

LP-Based Approximations

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$

$$\max \sum_{i=1}^m z_i$$

$$\text{over } y \in [0,1]^n, z \in [0,1]^m$$

$$\text{s.t. } \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i$$

for each clause C_i

$$z_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{if not} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if } x_j = \text{TRUE} \\ 0 & \text{if } x_j = \text{FALSE} \end{cases}$$

For each variable x_j , independently, randomly set $x_j = t$ with probability y_j , and $x_j = f$ otherwise.

claim: each clause C_i is satisfied w/ probability $(1 - 1/e) z_i$

claim:

$$E[\# \text{ sat. clauses}] \geq (1 - 1/e) \text{OPT}$$

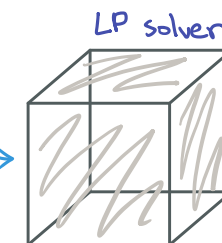
LP-Based Approximations

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$

$$\begin{array}{ll} \max & \sum_{i=1}^m z_i \\ \text{over} & y \in [0, 1]^n, z \in [0, 1]^m \\ \text{s.t.} & \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i \end{array}$$

for each clause C_i

$$z_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{if not} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if } x_j = \text{TRUE} \\ 0 & \text{if } x_j = \text{FALSE} \end{cases}$$

For each variable x_j , independently, randomly set $x_j = \text{t}$ with probability y_j , and $x_j = \text{f}$ otherwise.

$$E[\# \text{ sat. clauses}] \geq (1 - 1/e) \text{OPT}$$

Better?

LP-Based Approximations

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

③ relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$

$$\max \sum_{i=1}^m z_i$$

$$\text{over } y \in [0,1]^n, z \in [0,1]^m$$

$$\text{s.t. } \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i$$

for each clause C_i

$$z_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{if not} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if } x_j = \text{TRUE} \\ 0 & \text{if } x_j = \text{FALSE} \end{cases}$$

For each variable x_j , independently, randomly set $x_j = t$ with probability y_j , and $x_j = f$ otherwise.

claim: each clause C_i is satisfied
w/ probability $(1 - 1/e) z_i$

what if C_i has 1 variable?

~ 2 variables?

LP-Based Approximations

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$

$$\max \sum_{i=1}^m z_i$$

$$\text{over } y \in [0, 1]^n, z \in [0, 1]^m$$

$$\text{s.t. } \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i$$

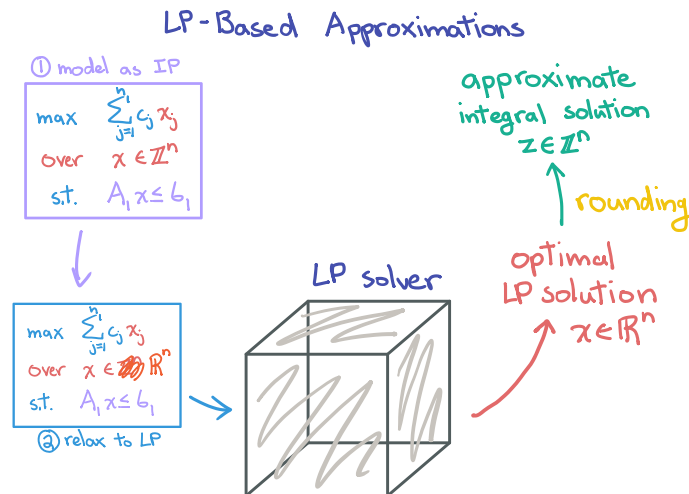
for each clause C_i

$$z_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{if not} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if } x_j = \text{TRUE} \\ 0 & \text{if } x_j = \text{FALSE} \end{cases}$$

For each variable x_j , independently, randomly set $x_j = \text{t}$ with probability y_j , and $x_j = \text{f}$ otherwise.

$$E[\# \text{ sat. clauses}] \geq (1 - 1/e) \text{OPT}$$



k	oblivious	LP	average
1	1/2	z_i	$\geq (3/4)z_i$
2	3/4	$(3/4)z_i$	$\geq (3/4)z_i$
3	7/8	$(1 - 1/e)z_i$	$\geq (3/4)z_i$
4	15/16	$(1 - 1/e)z_i$	$\geq (3/4)z_i$
5	31/32	$(1 - 1/e)z_i$	$\geq (3/4)z_i$
$\vdots$			
k	$1 - 1/2^k$	$(1 - 1/e)z_i$	$\geq (3/4)z_i$

$$\begin{array}{ll} \max & \sum_{i=1}^m z_i \\ \text{over} & y \in [0,1]^n, z \in [0,1]^m \\ \text{s.t.} & \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i \\ & \text{for each clause } C_i \end{array}$$

$z_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{if not} \end{cases}$

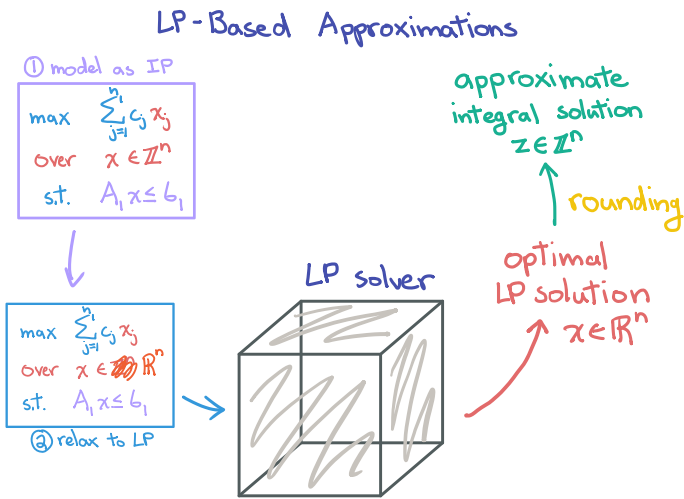
$y_j = \begin{cases} 1 & \text{if } x_j = \text{TRUE} \\ 0 & \text{if } x_j = \text{FALSE} \end{cases}$

Can we get $\frac{3}{4}$ -APX?

- ① Flip a coin
- ② If heads, do oblivious rounding
- ③ If tails, do LP rounding

For each variable x_j , independently, randomly set $x_j = t$ with probability y_j , and $x_j = f$ otherwise.

claim: each clause C_i is satisfied
w/ probability $\frac{3}{4} z_i$



$$\begin{array}{ll} \max & \sum_{i=1}^m z_i \\ \text{over} & y \in [0,1]^n, z \in [0,1]^m \\ \text{s.t.} & \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i \\ & \text{for each clause } C_i \end{array}$$

$z_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{if not} \end{cases}$

$y_j = \begin{cases} 1 & \text{if } x_j = \text{TRUE} \\ 0 & \text{if } x_j = \text{FALSE} \end{cases}$

k	oblivious	LP	average
1	1/2	z_i	$\geq (3/4)z_i$
2	3/4	$(3/4)z_i$	$\geq (3/4)z_i$
3	7/8	$(1 - 1/e)z_i$	$\geq (3/4)z_i$
4	15/16	$(1 - 1/e)z_i$	$\geq (3/4)z_i$
5	31/32	$(1 - 1/e)z_i$	$\geq (3/4)z_i$
⋮			
k	$1 - 1/2^k$	$(1 - 1/e)z_i$	$\geq (3/4)z_i$

Lecture 6 (today)

Probability (& techniques)

Approximations via
linear programming
& randomized rounding

Algorithms

Max Sat

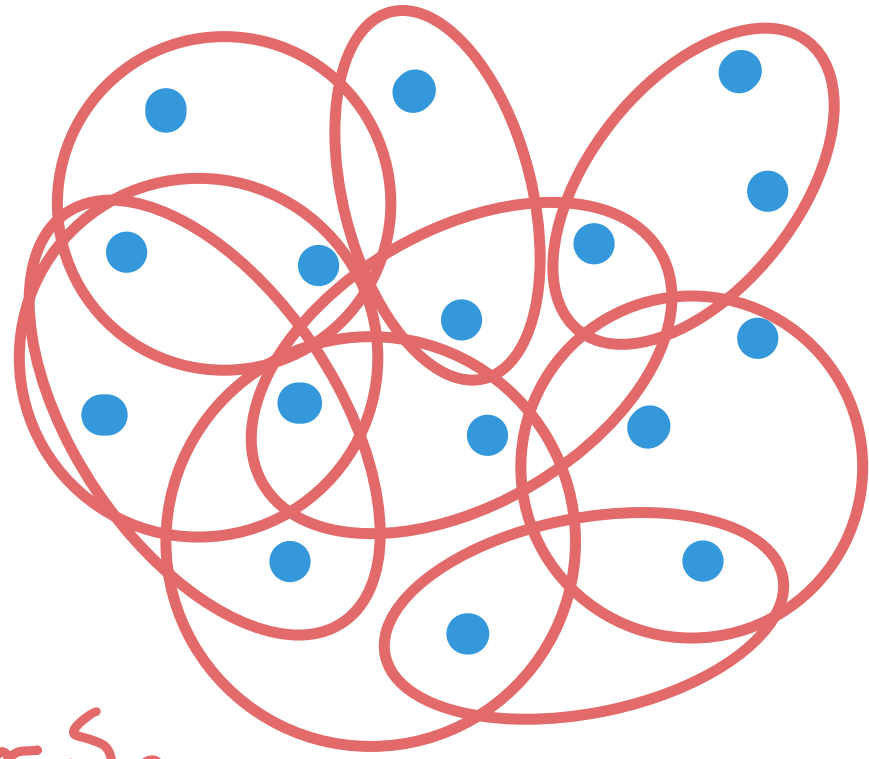
Set Cover

Covering
Integer Programs

Minimum Set Cover

m points $[m] \equiv \{1, \dots, m\}$

n sets $S_1, \dots, S_n \subseteq [m]$



Goal: select k sets $S_{e_1}, S_{e_2}, \dots, S_{e_k}$
minimizing k ,

$$\text{s.t. } S_{e_1} \cup S_{e_2} \cup \dots \cup S_{e_k} = [m]$$

NP-Hard

IP for set cover

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

$$\text{over } x \in \{0, 1\}^n$$

$$\text{s.t. } \sum_{j: i \in S_j} x_j \geq 1 \quad \forall i \in [m]$$

$$x_j = \begin{cases} 1 & \text{if } S_j \text{ in set cover,} \\ 0 & \text{otherwise} \end{cases}$$

LP-Based Approximations

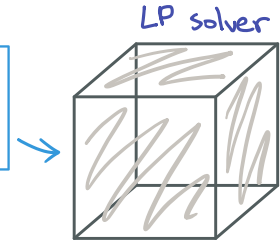
① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$



$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$

Minimum Set Cover

m points $[m] \equiv \{1, \dots, m\}$

n sets $S_1, \dots, S_n \subseteq [m]$

Goal: select k sets $S_{e_1}, \dots, S_{e_k}$

minimizing k ,

$$\text{s.t. } S_{e_1} \cup \dots \cup S_{e_k} = [m]$$

LP for set cover

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

$$\text{over } x \in [0,1]^n$$

$$\text{s.t. } \sum_{j: i \in S_j} x_j \geq 1 \quad \forall i \in [m]$$

$$x_j = \begin{cases} 1 & \text{if } S_j \text{ in set cover,} \\ 0 & \text{otherwise} \end{cases}$$

LP-Based Approximations

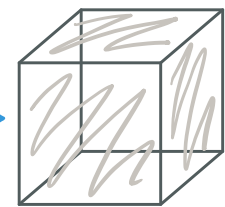
① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$



$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



optimal LP solution $x \in \mathbb{R}^n$

rounding

approximate integral solution $z \in \mathbb{Z}^n$

Minimum Set Cover
m points $[m] \equiv \{1, \dots, m\}$
n sets $S_1, \dots, S_n \subseteq [m]$
Goal: select k sets $S_{e_1}, \dots, S_{e_k}$
minimizing k,
s.t. $S_{e_1} \cup \dots \cup S_{e_k} = [m]$

IP for set cover

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

over $x \in \{0,1\}^n$

$$\text{s.t. } \sum_{j: i \in S_j} x_j \geq 1 \\ \forall i \in [m]$$

NP-Hard

LP for set cover

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

over $x \in [0,1]^n$

$$\text{s.t. } \sum_{j: i \in S_j} x_j \geq 1 \\ \forall i \in [m]$$

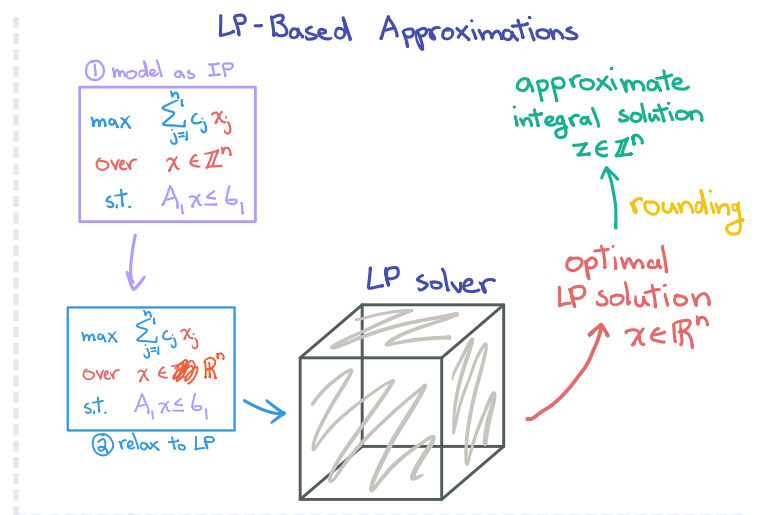
Polytime solvable

Rounding LP solution

$$x \in [0,1]^n \rightsquigarrow F \subseteq \{S_1, \dots, S_n\}$$

Ideas?

How to interpret $x_i = .5$?



Minimum Set Cover

m points $[m] \equiv \{1, \dots, m\}$

n sets $S_1, \dots, S_n \subseteq [m]$

Goal: select k sets $S_{e_1}, \dots, S_{e_k}$
minimizing k ,

$$\text{s.t. } S_{e_1} \cup \dots \cup S_{e_k} = [m]$$

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

$$\text{over } x \in [0,1]^n$$

$$\text{s.t. } \sum_{j: i \in S_j} x_j \geq 1 \quad \forall i \in [m]$$

$$x_j = \begin{cases} 1 & \text{if } S_j \text{ in set cover,} \\ 0 & \text{otherwise} \end{cases}$$

Rounding LP solution

$$x \in [0,1]^n \rightsquigarrow F \subseteq \{S_1, \dots, S_n\}$$

For each set S_j , independently, take S_j with probability x_j .

$$E[|F|] = \sum_j x_j = \text{OPT} \quad (\text{good!})$$

Fix a point i .

$$E[\text{sets in } F \text{ covering } i] = \sum_{j: i \in S_j} x_j \geq 1$$

$$P[F \text{ not cover } i] = \prod_{j: i \in S_j} (1 - x_j) \leq \prod_{j: i \in S_j} e^{-x_j}$$

$$= e^{-\sum_{j: i \in S_j} x_j} \leq e^{-1}$$

LP-Based Approximations

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_j x \leq b_j \end{array}$$



$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_j x \leq b_j \end{array}$$

② relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$

Minimum Set Cover

m points $[m] \equiv \{1, \dots, m\}$

n sets $S_1, \dots, S_n \subseteq [m]$

Goal: select k sets $S_{e_1}, \dots, S_{e_k}$
minimizing k ,

$$\text{s.t. } S_{e_1} \cup \dots \cup S_{e_k} = [m]$$

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

over $x \in [0,1]^n$

$$\text{s.t. } \sum_{j: i \in S_j} x_j \geq 1 \quad \forall i \in [m]$$

$$x_j = \begin{cases} 1 & \text{if } S_j \text{ in set cover,} \\ 0 & \text{otherwise} \end{cases}$$

Rounding LP solution

$$x \in [0,1]^n \rightsquigarrow F \subseteq \{S_1, \dots, S_n\}$$

For each set S_j , independently, take S_j with probability x_j .

$$E[|F|] = \sum_j x_j = \text{OPT} \quad (\text{good!})$$

$$P[F \text{ not cover } i] = e^{-1}$$

so F covers each point w/
constant probability

$\Rightarrow$ F is a set cover w/ constant prob.

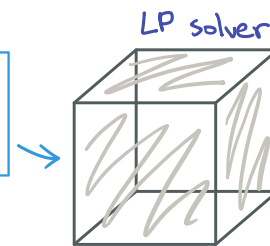
LP-Based Approximations

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$

Minimum Set Cover

m points $[m] \equiv \{1, \dots, m\}$

n sets $S_1, \dots, S_n \subseteq [m]$

Goal: select k sets $S_{e_1}, \dots, S_{e_k}$

minimizing k ,

$$\text{s.t. } S_{e_1} \cup \dots \cup S_{e_k} = [m]$$

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

$$\text{over } x \in [0,1]^n$$

$$\text{s.t. } \sum_{j: i \in S_j} x_j \geq 1 \quad \forall i \in [m]$$

$$x_j = \begin{cases} 1 & \text{if } S_j \text{ in set cover,} \\ 0 & \text{otherwise} \end{cases}$$

Rounding LP solution

$$x \in [0,1]^n \rightsquigarrow F \subseteq \{S_1, \dots, S_n\}$$

2. For each set S_j independently, take S_j with probability $\min\{1, \alpha x_j\}$.

for $d > 1$ TBD.

$$E[|F|] = d \sum_j x_j = d \text{OPT}$$

$$P[F \text{ not cover } i] = \prod_{S_j \ni i} (1 - d x_j) \leq e^{-d \sum_{S_j \ni i} x_j} \leq e^{-d}$$

$$P[F \text{ not set cover}] \leq \sum_i P[F \text{ not cover } i] \leq m \cdot e^{-d} \leq \frac{1}{m^{\alpha d}} \text{ for } d = O(\log m)$$

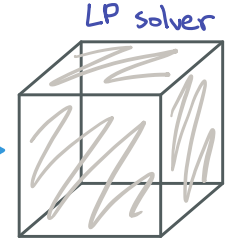
LP-Based Approximations

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_j x \leq b_j \end{array}$$

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_j x \leq b_j \end{array}$$

② relax to LP



approximate integral solution $z \in \mathbb{Z}^n$

rounding

optimal LP solution $x \in \mathbb{R}^n$

Minimum Set Cover

m points $[m] \equiv \{1, \dots, m\}$

n sets $S_1, \dots, S_n \subseteq [m]$

Goal: select k sets $S_{e_1}, \dots, S_{e_k}$ minimizing k ,

$$\text{s.t. } S_{e_1} \cup \dots \cup S_{e_k} = [m]$$

$$\begin{array}{ll} \min & \sum_{j=1}^n x_j \\ \text{over} & x \in [0,1]^n \end{array}$$

$$\text{s.t. } \sum_{j: i \in S_j} x_j \geq 1 \quad \forall i \in [m]$$

$$x_j = \begin{cases} 1 & \text{if } S_j \text{ in set cover,} \\ 0 & \text{otherwise} \end{cases}$$

Rounding LP solution

$$x \in [0,1]^n \rightsquigarrow F \subseteq \{S_1, \dots, S_n\}$$

2. For each set S_j independently, take S_j with probability $\min\{1, \alpha x_j\}$.

for $d > 1$ TBD.

$$E[|F|] = d \sum_j x_j = d \text{OPT}$$

$$P[F \text{ not set cover}] \leq \frac{1}{m^{\alpha d}}$$

$$\text{for } d = O(\log m)$$

$$\Rightarrow O(\log m)\text{-APX}$$

LP-Based Approximations

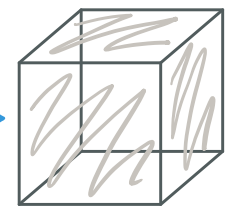
① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$



$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



LP solver

optimal LP solution $x \in \mathbb{R}^n$

rounding

approximate integral solution $z \in \mathbb{Z}^n$

Minimum Set Cover

m points $[m] \equiv \{1, \dots, m\}$

n sets $S_1, \dots, S_n \subseteq [m]$

Goal: select k sets $S_{e_1}, \dots, S_{e_k}$

minimizing k ,

$$\text{s.t. } S_{e_1} \cup \dots \cup S_{e_k} = [m]$$

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

$$\text{over } x \in [0,1]^n$$

$$\text{s.t. } \sum_{j: i \in S_j} x_j \geq 1 \quad \forall i \in [m]$$

$$x_j = \begin{cases} 1 & \text{if } S_j \text{ in set cover,} \\ 0 & \text{otherwise} \end{cases}$$

Lecture 6 (today)

Probability (& techniques)

Approximations via
linear programming
& randomized rounding

Algorithms

Max Sat

Set Cover

Covering
Integer Programs

Covering IPs

$$\min \sum_j c_j x_j$$

$$\text{over } x \in \mathbb{Z}_{\geq 0}^n$$

$$\text{s.t. } \sum_{j=1}^n A_{ij} x_j \geq b_i \text{ for } i \in [m]$$

$$\text{where } A \in \mathbb{R}_{\geq 0}^{m \times n}, \quad b \in \mathbb{R}_{> 0}^m, \quad c \in \mathbb{R}_{> 0}^n$$

"demands" "costs"

WLOG $A_{ij} \leq b_i \quad \forall i, j$ (why?)

Set cover is special case w/ $A_{ij} \in \{0, 1\}, b_i = 1, c_j = 1$

LP relaxation

$$\min \sum_j c_j x_j$$

$$\text{over } x \in \mathbb{Z}_{\geq 0}^n \quad \mathbb{R}_{\geq 0}^n$$

$$\text{s.t. } \sum_{j=1}^n A_{ij} x_j \geq b_i \quad \text{for } i \in [m]$$

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$



$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



LP solver

approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$

Covering IPs

$$\min \sum_j c_j x_j$$

$$\text{over } x \in \mathbb{Z}_{\geq 0}^n$$

$$\text{s.t. } \sum_{j=1}^n A_{ij} x_j \geq b_i \quad \text{for } i \in [m]$$

$$A_{ij} \leq b_i \quad \forall i, j$$

Rounding LP solution

$$x \in \mathbb{R}_{\geq 0}^n \rightsquigarrow z \in \mathbb{Z}_{\geq 0}^n$$

Ideas?

LP-Based Approximations

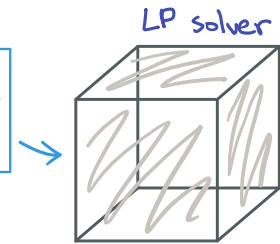
① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$



$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

↑ rounding

optimal
LP solution
 $x \in \mathbb{R}^n$

LP relaxation

$$\begin{array}{ll} \min & \sum_j c_j x_j \\ \text{over} & x \in \mathbb{R}_{\geq 0}^n \\ \text{s.t.} & \sum_{j=1}^n A_{ij} x_j \geq b_i \\ & \text{for } i \in [m] \end{array}$$

Covering IPs

$$\begin{array}{ll} \min & \sum_j c_j x_j \\ \text{over} & x \in \mathbb{Z}_{\geq 0}^n \\ \text{s.t.} & \sum_{j=1}^n A_{ij} x_j \geq b_i \\ & \text{for } i \in [m] \\ & A_{ij} \leq b_i \quad \forall i, j \end{array}$$

Rounding LP solution

$$x \in \mathbb{R}_{\geq 0}^n \rightsquigarrow z \in \mathbb{Z}_{\geq 0}^n$$

For each $j \in [n]$, independently, let

$$z_j = \begin{cases} \lceil \alpha x_j \rceil & \text{with probability } \alpha x_j - \lfloor \alpha x_j \rfloor \\ \lfloor \alpha x_j \rfloor & \text{otherwise.} \end{cases}$$

$$E[z_j] = \alpha x_j$$

$$E\left[\sum_j c_j z_j\right] = \alpha \sum_j c_j x_j = \alpha \text{OPT}$$

LP-Based Approximations

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

optimal
LP solution
 $x \in \mathbb{R}^n$

rounding

LP relaxation

$$\begin{array}{ll} \min & \sum_j c_j x_j \\ \text{over} & x \in \mathbb{R}_{\geq 0}^n \\ \text{s.t.} & \sum_{j=1}^n A_{ij} x_j \geq b_i \end{array}$$

for $i \in [m]$

Covering IPs

$$\begin{array}{ll} \min & \sum_j c_j x_j \\ \text{over} & x \in \mathbb{Z}_{\geq 0}^n \\ \text{s.t.} & \sum_{j=1}^n A_{ij} x_j \geq b_i \end{array}$$

for $i \in [m]$
 $A_{ij} \leq b_i \quad \forall i, j$

For each $j \in [n]$, independently, let

$$z_j = \begin{cases} \lceil \alpha x_j \rceil & \text{with probability } \alpha x_j - \lfloor \alpha x_j \rfloor \\ \lfloor \alpha x_j \rfloor & \text{otherwise.} \end{cases}$$

$$P[\sum_j A_{ij} z_j \leq b_i]$$

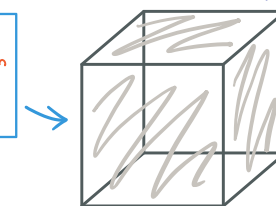
LP-Based Approximations

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding
optimal
LP solution
 $x \in \mathbb{R}^n$

LP relaxation

$$\begin{array}{ll} \min & \sum_j c_j x_j \\ \text{over} & x \in \mathbb{R}_{\geq 0}^n \\ \text{s.t.} & \sum_{j=1}^n A_{ij} x_j \geq b_i \\ & \text{for } i \in [m] \end{array}$$

Covering IPs

$$\begin{array}{ll} \min & \sum_j c_j x_j \\ \text{over} & x \in \mathbb{Z}_{\geq 0}^n \\ \text{s.t.} & \sum_{j=1}^n A_{ij} x_j \geq b_i \\ & \text{for } i \in [m] \\ & A_{ij} \leq b_i \quad \forall i, j \end{array}$$

For each $j \in [n]$, independently, let

$$z_j = \begin{cases} \lceil \alpha x_j \rceil & \text{with probability } \alpha x_j - \lfloor \alpha x_j \rfloor \\ \lfloor \alpha x_j \rfloor & \text{otherwise.} \end{cases}$$

$$P[\sum_j A_{ij} z_j \leq b_i]$$

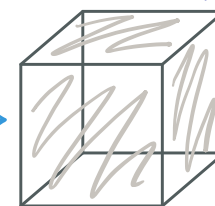
LP-Based Approximations

① model as IP

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j \\ \text{over } & x \in \mathbb{Z}^n \\ \text{s.t. } & A_i x \leq b_i \end{aligned}$$

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j \\ \text{over } & x \in \mathbb{R}^n \\ \text{s.t. } & A_i x \leq b_i \end{aligned}$$

② relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding

optimal
LP solution
 $x \in \mathbb{R}^n$

LP relaxation

$$\begin{aligned} \min \quad & \sum_j c_j x_j \\ \text{over } & x \in \mathbb{R}_{\geq 0}^n \\ \text{s.t. } & \sum_{j=1}^n A_{ij} x_j \geq b_i \\ & \text{for } i \in [m] \end{aligned}$$

Covering IPs

$$\begin{aligned} \min \quad & \sum_j c_j x_j \\ \text{over } & x \in \mathbb{Z}_{\geq 0}^n \\ \text{s.t. } & \sum_{j=1}^n A_{ij} x_j \geq b_i \\ & \text{for } i \in [m] \\ & A_{ij} \leq b_i \quad \forall i, j \end{aligned}$$

Multiplicative Chernoff bound

let $X_1, \dots, X_n \in [0, 1]$ be independent

$$\mu = E[X_1 + \dots + X_n] \in (0, 1)$$

$$P[X_1 + \dots + X_n \leq (1 - \epsilon)\mu] \leq e^{-\epsilon^2 \mu / 2}$$

For each $j \in [n]$, independently, let

$$z_j = \begin{cases} \lceil \alpha x_j \rceil & \text{with probability } \alpha x_j - \lfloor \alpha x_j \rfloor \\ \lfloor \alpha x_j \rfloor & \text{otherwise.} \end{cases}$$

LP-Based Approximations

① model as IP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{Z}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

↓

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{over} & x \in \mathbb{R}^n \\ \text{s.t.} & A_1 x \leq b_1 \end{array}$$

② relax to LP



approximate
integral solution
 $z \in \mathbb{Z}^n$

rounding
optimal
LP solution
 $x \in \mathbb{R}^n$

LP relaxation

$$\begin{array}{ll} \min & \sum_j c_j x_j \\ \text{over} & x \in \mathbb{R}_{\geq 0}^n \\ \text{s.t.} & \sum_{j=1}^n A_{ij} x_j \geq b_i \\ & \text{for } i \in [m] \end{array}$$

Covering IPs

$$\begin{array}{ll} \min & \sum_j c_j x_j \\ \text{over} & x \in \mathbb{Z}_{\geq 0}^n \\ \text{s.t.} & \sum_{j=1}^n A_{ij} x_j \geq b_i \\ & \text{for } i \in [m] \\ & A_{ij} \leq b_i \quad \forall i, j \end{array}$$

Multiplicative Chernoff bound

let $X_1, \dots, X_n \in [0, 1]$ be independent

$$\mu = E[X_1 + \dots + X_n] \in (0, 1)$$

$$P[X_1 + \dots + X_n \leq (1 - \epsilon)\mu] \leq e^{-\epsilon^2 \mu / 2}$$

Multiplicative Chernoff bound

let $X_1, \dots, X_n \in [0, 1]$ be independent

$$\mu = E[X_1 + \dots + X_n]$$

$$\epsilon \in (0, 1)$$

$$\bullet \quad P[X_1 + \dots + X_n \geq (1 + \epsilon)\mu] \leq e^{-\epsilon^2 \mu / 3}$$

$$\bullet \quad P[X_1 + \dots + X_n \leq (1 - \epsilon)\mu] \leq e^{-\epsilon^2 \mu / 2}$$

(proof at end if time permits)

Chapter 6

Randomized Rounding

Many of the problems we are interested in are inherently discrete, and unfortunately many discrete problems are NP-Hard. One general class of problems in this category is *integer programs* (IPs), which are mathematical optimization problems where the variables are required to be integers.

One example is a *covering integer program*, which is an optimization problem of the form

$$\begin{aligned} & \text{minimize} && \sum_{j=1}^n c_j x_j \\ & \text{over} && x \in \mathbb{Z}_{\geq 0} \\ & \text{s.t.} && \sum_{j=1}^n A_{ij} x_j \geq b_i \text{ for } i \in [m], \end{aligned} \tag{6.1}$$

where $A \in \mathbb{R}_{\geq 0}^{m \times n}$, $b \in \mathbb{R}_{> 0}^m$, and $c \in \mathbb{R}_{> 0}^n$. Here, each variable $x_j \in \mathbb{Z}_{\geq 0}$ models a binary decision. Each c_j can be interpreted as the *cost* of taking $x_j = 1$. For $i \in [m]$, the constraint

$$\sum_{j=1}^n A_{ij} x_j \geq b_i \text{ for } i \in [m]$$

is a *covering constraint*, saying we must set $x_j = 1$ for enough variables x_j so that the sum of A_{ij} over these variables is at least b_i .

The special case where $c = \mathbf{1}$, $b = \mathbf{1}$, and $A \in \{0, 1\}^{m \times n}$ describes the *set cover* problem. There the coordinates $j \in [n]$ correspond to sets, and the coordinates $i \in [m]$ correspond to points; set j covers point i iff $A_{ij} = 1$.

Besides covering integer programs, there are also packing integer programs, which

are maximization problems subject to packing (i.e., $\leq$) constraints, of the form

$$\begin{aligned} & \text{maximize } \sum_{j=1}^n b_j y_j \\ & \text{over } y \in \mathbb{Z}_{\geq 0}^n \\ & \text{s.t. } \sum_{j=1}^m A_{ij} y_j \leq c_j \text{ for } i \in [m] \end{aligned}$$

where $A \in \mathbb{R}_{\geq 0}^{m \times n}$, $b \in \mathbb{R}_{> 0}^n$, and $c \in \mathbb{R}_{> 0}^m$. Here each b_j can be understood as the “profit” of taking $y_j = 1$, and we want to maximize the total profit. We are constrained by the n packing constraints. The knapsack problem is a special case of the covering integer programs where $m = 1$.

In general, integer programs maximize or minimize a set of integer variables over a linear objective, subject to linear equality and inequality constraints.¹ The basic appeal of integer programs is that they are very flexible for modeling discrete optimization problems. Unfortunately, their broad applicability also makes them NP-Hard.

Integer programs are NP-Hard because the output is required to be discrete. If we allowed the variables to vary *continuously* over the reals, we instead have a *linear program* (LP). For example, the following describes a linear program for covering problems similar to the covering integer program above. This time, however, each variable x_j is allowed to take any nonnegative value.

$$\begin{aligned} & \text{minimize } \sum_{j=1}^n c_j x_j \\ & \text{over } x \in \mathbb{R}_{\geq 0}^n \\ & \text{s.t. } \sum_{j=1}^n A_{ij} x_j \geq b_i \text{ for } i \in [m]. \end{aligned}$$

Note that any solution feasible to the CIP (6.1) is also feasible for the LP above. As such, the LP is said to be a *relaxation* of the CIP. It implies that the optimum value of the LP is less than or equal to the optimum value for the integer program.

Unlike integer programs, linear programs are polynomial time solvable (!). This allows for the following general approach to discrete optimization: given an IP

¹Integer programs may have both packing (i.e., $Ax \leq b$) and covering (i.e., $Ax \geq b$) constraints. “Packing integer programs” and “covering integer programs” refer to the special case that have only packing constraints or only covering constraints, respectively.

formulation of the problem, instead solve the corresponding LP. The LP provides a *fractional solution* satisfying the same constraints, which we can treat as a clue towards a good integer solution. The goal becomes to convert the fractional solution to an integer solution, while maintaining feasibility and the objective value. There are several strategies to *round* a fractional solution to an integer solution and they are covered in courses on approximation algorithms (e.g., [Vaz01; WS11]). We will study a technique called *randomized rounding*.

In this chapter, we will use randomized rounding to obtain approximation algorithms for max SAT (section 6.1), set cover (section 6.2), and CIPs (??).

6.1 SAT

Recall the max-SAT problem from chapter 1: given a boolean formula $f(x_1, \dots, x_n)$ in CNF, the goal is to find an assignment of $x_1, \dots, x_n$ to $\{\mathbf{t}, \mathbf{f}\}$ that satisfies the maximum number of clauses. There we showed that a random assignment gives a $7/8$ -approximation for 3-SAT. Moreover the algorithm can be derandomized, and the approximation factor is best possible unless $P = NP$.

More generally a random assignment gives a $(1 - 2^{-k})$ approximation for k -SAT, for any $k \in \mathbb{N}$. The following table lists the approximation factors for the first few values of k .

k	oblivious APX
1	$1/2$
2	$3/4$
3	$7/8$
4	$15/16$
5	$31/32$
$\vdots$	
k	$1 - 1/2^k$

The approximation factor gets better and better as k increases. Meanwhile, the first row — $k = 1$ — is rather embarrassing, since 1-SAT is trivial. 2-SAT is not as trivial but there is a polynomial time algorithm for this problem as well. So oblivious rounding is not so great for very small values of k .

This is relevant as we now consider the more general form of max-SAT, where we are given a formula in CNF where each clause can have any number of clauses. We can still apply our oblivious randomized algorithm that flips a coin for every variable. If every clause had *at least* k variables, then we obtain a $1 - 1/2^k$ approximation ratio. But the presence of single-variable clauses — which one might expect to cause the

least trouble — means we only have a $1/2$ -approximation ratio in general. We will use linear programming to improve the approximation ratio.

An integer program for SAT. We first translate max-SAT to an integer program. Fix a formula $f(x_1, \dots, x_n)$ of n variables, consisting of m clauses $C_1, \dots, C_m$. Let OPT denote the maximum number of satisfiable clauses. Let “ $x_j \in C_i$ ” indicate that the symbol x_j appears in C_i (without negation) and “ $\bar{x}_j \in C_i$ ” indicate that $\bar{x}_j$ appears in C_i .

Consider now the following integer program.

$$\begin{aligned} & \text{maximize } \sum_{i=1}^m z_i \text{ over } y \in \{0, 1\}^n, z \in \{0, 1\}^m \\ & \text{s.t. } \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i \text{ for all clauses } C_i. \end{aligned}$$

The integer program has $\{0, 1\}$ -variables for each variable x_j and each clause C_i , with the following interpretations.

1. For $j = 1, \dots, n$, let $y_j \in \{0, 1\}$ indicates whether we set $x_j = \mathbf{t}$ ($y_j = 1$) or $x_j = \mathbf{f}$ ($y_j = 0$).
2. For $i = 1, \dots, m$, let $z_i \in \{0, 1\}$ indicate whether we satisfy the i th clause ($z_i = 1$) or not ($z_i = 0$).

The integer program seeks to maximize the number of satisfied clauses, represented by $\sum_{i=1}^m z_i$. For each clause C_i , the corresponding constraint implies we can only set $z_i = 1$ if $y_j = 1$ for some $x_j \in C_i$ or $y_j = 0$ for some $\bar{x}_j \in C_i$.

A linear program for SAT. As discussed, while we cannot solve integer programs in polynomial time, we can solve linear programs. We relax the integer program above to a linear program by now allowing each variable y_j or z_i to lie anywhere in the interval $[0, 1]$.

$$\begin{aligned} & \text{maximize } \sum_{i=1}^m z_i \text{ over } y_1, \dots, y_n, z_1, \dots, z_m \in \mathbb{R} \\ & \text{s.t. } \sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j) \geq z_i \text{ for all clauses } C_i \\ & \quad 0 \leq z_i \leq 1 \text{ for all } i = 1, \dots, m \\ & \quad 0 \leq y_j \leq 1 \text{ for all } j = 1, \dots, n \end{aligned}$$

Let OPT_{LP} denote the optimum value of the LP. Since the LP is a relaxation of the IP above it, we have $\text{OPT}_{\text{LP}} \geq \text{OPT}$.

Rounding the LP solution. Let $y_1, \dots, y_n$ and $z_1, \dots, z_m$ be an optimum solution to the LP. We know that the objective value $\text{OPT}_{\text{LP}} = \sum_i z_i$ is very good. Our goal now is to convert the y_j 's into *discrete* decisions while keeping the objective value as close to OPT_{LP} as possible.

The basic question is: how do we interpret a *fractional* value such as $y_1 = .5$? The LP seems to suggest that we should set x_1 to be one-half true and one-half false. Of course there are no half values in boolean algebra and “one-half true and one-half false” is total nonsense. A different interpretation is that of a randomized experiment where we set $x_1 = \mathbf{t}$ *half the time*, and $x_1 = \mathbf{f}$ the other half. Consider the following *randomized rounding* algorithm:

Randomized rounding for SAT

1. Let $y_1, \dots, y_n$ be an optimum solution to the LP for max-SAT.
2. For each variable x_j , independently, randomly set $x_j = \mathbf{t}$ with probability y_j , and $x_j = \mathbf{f}$ otherwise.

Next we analyze the expected number of clauses satisfied by randomized rounding. By linearity of expectation this boils down to analyzing the probability of satisfying each individual clause.

Lemma 6.1. *Each clause C_i is satisfied with probability at least $(1 - 1/e)z_i$.*

Proof. A clause C_i is not satisfied iff we randomly set $x_j = \mathbf{f}$ for all $x_j \in C_i$, and set $x_j = \mathbf{t}$ for all $\bar{x}_j \in C_i$. Thus

$$\mathbf{P}[C_i \text{ not satisfied}] = \prod_{j: x_j \in C_i} (1 - y_j) \prod_{j: \bar{x}_j \in C_i} y_j.$$

Now, by the inequality $1 + x \leq e^x$ (for all x), we can simplify the RHS as

$$e^{-\sum_{j: x_j \in C_i} y_j + \sum_{j: \bar{x}_j \in C_i} (1 - y_j)} \leq e^{-z_i}.$$

Here the inequality follows from the LP constraint for C_i . Finally, by convexity of e^{-x} , we have

$$e^{-z_i} \leq (1 - z_i)e^0 + z_ie^{-1} = (1 - 1/e)z_i,$$

as desired. □

By linearity of expectation, the expected number of clauses that are satisfied equals the sum of probabilities of each clause being satisfied. Therefore, by lemma 6.1, we will satisfy at least $(1 - 1/e) \text{OPT}_{\text{LP}} \geq (1 - 1/e) \text{OPT}$ clauses in expectation.

Theorem 6.2. *There is a $(1 - 1/e)$ -approximation algorithm for max-SAT.*

Note that our bound is only interesting when there are clauses with one or two variables; otherwise oblivious rounding is still better.

The best of both worlds. Part of the problem is that lemma 6.1 is not tight for small k . Take for example $k = 1$. Obviously, if $|C_i| = 1$, then C_i is satisfied with probability z_i , not $(1 - 1/e)z_i$. For $k = 2$, we have the following better analysis.

Lemma 6.3. *If $|C_i| = 2$, then C_i is satisfied with probability $\geq 3z_i/4$.*

Proof. Suppose for simplicity that $C_i = x_1 \vee x_2$. (It will be obvious how to generalize the analysis to other pairs of variables.) We have

$$\begin{aligned} \mathbf{P}[C_i \text{ not satisfied}] &= (1 - y_1)(1 - y_2) \stackrel{(a)}{\leq} \left(\frac{(1 - y_1) + (1 - y_2)}{2} \right)^2 \\ &\leq \left(1 - \frac{z_j}{2} \right)^2 \stackrel{(b)}{\leq} (1 - z_j) \left(1 - \frac{0}{2} \right)^2 + z_j \left(1 - \frac{1}{2} \right)^2 \\ &= 1 - z_j + \frac{z_j}{4} = 1 - \frac{3}{4}z_j. \end{aligned}$$

(a) is by AM-GM. (b) is by convexity of $f(x) = \left(1 - \frac{x}{2}\right)^2$. □

Below we list the probability of clause being satisfied by the oblivious and LP rounding strategies, as a function of the number of variables in the clause, k . Observe that the average of the probabilities is at least $(3/4)z_i$ for all k , as indicated in the column on the right.

k	oblivious	LP	average
1	$1/2$	z_i	$\geq (3/4)z_i$
2	$3/4$	$(3/4)z_i$	$\geq (3/4)z_i$
3	$7/8$	$(1 - 1/e)z_i$	$\geq (3/4)z_i$
4	$15/16$	$(1 - 1/e)z_i$	$\geq (3/4)z_i$
5	$31/32$	$(1 - 1/e)z_i$	$\geq (3/4)z_i$
$\vdots$			
k	$1 - 1/2^k$	$(1 - 1/e)z_i$	$\geq (3/4)z_i$

The table suggests that we might be able to merge the oblivious sampling and LP rounding algorithms to obtain a $3/4$ -approximation ratio. But the two approaches seem like polar opposites: one approach is completely oblivious, and the other strongly depends on the formula (implicitly via the LP solver). But here's a trick: just pick one of the two strategies uniformly at random. The "averaging" will work itself out in the analysis.

Hybrid algorithm for max-SAT.

1. With probability $1/2$, return a uniformly random assignment.
2. Otherwise solve and randomly round the LP.

Theorem 6.4. *The hybrid algorithm gives a $(3/4)$ -approximation algorithm for max-SAT.*

We leave the analysis to the reader as exercise 6.1.

6.2 Set cover

In *set cover*, we are given m points $[m] = \{1, \dots, m\}$, and n sets $S_1, \dots, S_n \subseteq [m]$. The goal is to

find the minimum number of sets $S_{i_1}, \dots, S_{i_k}$ such that $S_{i_1} \cup \dots \cup S_{i_k} = [m]$.

Some natural extensions including adding costs for sets, pointwise demands that require points to be covered by multiple sets, and coefficients $A_{ij} \in [0, 1]$ that indicate the amount of coverage a set S_i gives to point j . These will be considered when we discuss CIPs later. Set Cover is NP-Hard. Instead we will design an approximation algorithm for set cover, via randomized rounding.

We first write an LP relaxation for set cover. For each set S_j we introduce a variable x_j that models our decision to take S_j in our set cover. Consider the following LP.

$$\text{minimize } \sum_{j=1}^n x_j \text{ over } x \in \mathbb{R}_{\geq 0}^n \text{ s.t. } \sum_{S_j \ni i} x_j \geq 1 \text{ for all } i \in [m].$$

The objective, $\sum_{j=1}^n x_j$, is the (fractional) number of sets in our (fractional) set cover. For each point $i \in [m]$, we require at least one fractional set among the family of sets that cover that point.

Randomized rounding. Let us consider the following randomized rounding algorithm

1. Let $x_1, \dots, x_n$ be an optimum solution to the set cover LP.
2. For each set S_j , independently, take S_j with probability x_j .

Let $F \subseteq \{S_1, \dots, S_m\}$ be the random family of sets produced by randomized rounding. We have two questions to address:

1. How big is F , relative to OPT?
2. Is F a set cover?

For the first question, we have

$$\mathbf{E}[|F|] = \sum_j \mathbf{P}[S_j \in F] = \sum_j x_j = \text{OPT}_{\text{LP}},$$

which is very good indeed. Now, is F a set cover? Fix a point $i \in [m]$. The expected number of sets covering i is

$$\mathbf{E}[\# \text{ sets in } F \text{ covering } i] = \sum_{S_j \ni i} \mathbf{P}[S_j \in F] = \sum_{S_j \ni i} x_j \geq 1.$$

So i expects to have at least 1 set in F containing it. But this does not imply that F is a set cover — that F covers all of $[m]$ simultaneously — with any nonnegligible probability. A better question is about the probability that F covers i , or rather, the probability that F doesn't cover i . F does not cover i iff it fails to sample any of the sets covering i , hence

$$\mathbf{P}[F \text{ doesn't cover } i] = \prod_{S_j \ni i} (1 - x_j) \leq e^{-\sum_{S_j \ni i} x_j} \leq e^{-1}.$$

So each point is covered with constant probability.

The question is: how do we increase the probability that *all points are covered*, simultaneously?

Scaling and rounding. The problem above is that having each point covered with constant probability does *not* imply that all points are covered with constant, or any nonnegligible, probability. However, having each point covered with *high probability* does imply, *by the union bound*, that all points are covered with *high probability*. Now, how can we ensure that each point is covered with high probability? By scaling up x before rounding.

Randomized rounding for set cover.

1. Let $x_1, \dots, x_n$ be an optimum solution to the set cover LP. Let $\alpha = 2 \log(m)$.
2. For each set S_j independently, take S_j with probability $\min\{1, \alpha x_j\}$.

As before, let F denote the random collection of sets returned by the algorithm.

Lemma 6.5. *Each point i is covered with probability $1 - 1/m^2$.*

Proof. If $x_j \geq 1/\alpha$ for any set S_j covering i , then i is covered deterministically. Otherwise, by similar calculations as before, we have

$$\mathbf{P}[F \text{ doesn't cover } i] = \prod_{S_j \ni i} (1 - \alpha x_j) \leq e^{-\sum_{S_j \ni i} \alpha x_j} \leq e^{-\alpha} = 1/m^2. \quad (6.2)$$

□

Lemma 6.6. *F is a set cover with probability at least $1 - 1/m$.*

Proof. By the union bound, we have

$$\mathbf{P}[F \text{ is not a set cover}] \leq \sum_i \mathbf{P}[F \text{ doesn't cover } i] \leq \frac{1}{m},$$

as desired.

□

Theorem 6.7. *Randomized rounding is a $O(\log m)$ -approximation algorithm for set cover.*

Proof. We have $\mathbf{E}[|F|] = 2 \log(m) \text{OPT}$, and by Markov's inequality, $|F| \leq 4 \log(m) \text{OPT}$ with probability $1/2$. F is also a set cover with probability $1 - 1/m$. By the union bound, with probability of error at most $1/2 + 1/m$, F is a set cover of size at most $4 \log(m) \text{OPT}$. □

6.3 Covering integer programs

, we consider covering integer programs. They are discrete problems generalizing set cover, of the form

$$\text{minimize } \langle c, x \rangle \text{ over } x \in \mathbb{Z}_{\geq 0}^n \text{ s.t. } Ax \geq b.$$

where $c \in \mathbb{R}_{>0}^n$, $A \in \mathbb{R}_{\geq 0}^{m \times n}$, and $b \in \mathbb{R}_{>0}^m$. The matrix notation expands out to the following.

$$\begin{aligned} & \text{minimize } \sum_{j=1}^n c_j x_j \\ & \text{over } x \in \mathbb{Z}_{\geq 0}^n \\ & \text{s.t. } \sum_{j=1}^n A_{ij} x_j \geq b_i \text{ for } i \in [m]. \end{aligned}$$

We assume without loss of generality that $A_{ij} \leq b_i$ for all i . (Why?) We also assume without loss of generality that m is at least some constant; say, 4. (Otherwise add a few empty constraints.)

The algorithm we analyze is essentially the same as for set cover. We scale up x by a $O(\log m)$ factor, and then randomly round each coordinate of x , independently, to integer values.

1. Let $\alpha = 8 \log(m)$.
2. For each $j \in [n]$, independently, let

$$z_j = \begin{cases} \lceil \alpha x_j \rceil & \text{with probability } \alpha x_j - \lfloor \alpha x_j \rfloor, \\ \lfloor \alpha x_j \rfloor & \text{otherwise.} \end{cases}$$

The algorithm is very similar to set cover. However, the coefficients A_{ij} make the analysis more difficult. For set cover, we were able to give an exact formula for the probability that a point i is uncovered. (Equation (6.2).) Here it is not so simple, and we instead appeal to the *multiplicative* (or *relative*) Chernoff bound introduced last chapter.

Theorem 6.8 (Multiplicative Chernoff bounds). *Let $X_1, \dots, X_n \in [0, 1]$ be independent random variables, and $\epsilon \in [0, 1]$. Let $\mu = \mathbf{E}[X_1 + \dots + X_n]$. Then*

$$\begin{aligned} \mathbf{P}[X_1 + \dots + X_n \geq (1 + \epsilon)\mu] &\leq e^{-\epsilon^2 \mu / 3}. \\ \mathbf{P}[X_1 + \dots + X_n \leq (1 - \epsilon)\mu] &\leq e^{-\epsilon^2 \mu / 2}. \end{aligned}$$

The main part of the analysis is to show that a single constraint is satisfied with high probability, as follows.

Lemma 6.9. *The i th constraint is satisfied with probability at least $1 - 1/m^2$.*

Proof. By scaling, we may assume that $b_i \leq 1$, hence $A_{ij} \leq 1$ for all j . We may assume that $\sum_j A_{ij} \lfloor \alpha x_j \rfloor < 1$, since otherwise the constraint is satisfied deterministically.

For each coordinate j , let $X_j = A_{ij}(z_j - \lfloor \alpha x_j \rfloor)$ for each j . Let $\mu = \mathbf{E}[\sum_{j=1}^n X_j]$. We have

$$\mu = \sum_{j=1}^n A_{ij}(\alpha x_j - \lfloor \alpha x_j \rfloor) \geq \alpha - \sum_{j=1}^n A_{ij} \lfloor \alpha x_j \rfloor.$$

This implies that both

$$\mu \geq \alpha \left(1 - \sum_{j=1}^n A_{ij} \lfloor \alpha x_j \rfloor\right), \text{ and } \mu \geq \alpha - 1 \geq 8 \log(m).$$

We have

$$\begin{aligned} \mathbf{P}[z \text{ fails constraint } i] &= \mathbf{P}\left[\sum_{j=1}^n A_{ij} z_j \leq 1\right] \\ &= \mathbf{P}\left[\sum_{j=1}^n X_j \leq 1 - \sum_{j=1}^n A_{ij} \lfloor \alpha x_j \rfloor\right] \\ &\leq \mathbf{P}\left[\sum_{j=1}^n X_j \leq \frac{\mu}{\alpha}\right]. \end{aligned}$$

By the Chernoff inequality, we have

$$\mathbf{P}\left[\sum_{j=1}^n X_j \leq \frac{\mu}{\alpha}\right] \leq e^{-(1-1/\alpha)^2 \mu/2}.$$

Now,

$$\frac{1}{2} \left(1 - \frac{1}{\alpha}\right)^2 \mu \geq \frac{1}{2} \left(1 - \frac{1}{\alpha}\right)^3 \alpha \geq \frac{\alpha}{4} \geq 2 \log(m).$$

Thus

$$\mathbf{P}[z \text{ fails constraint } i] \leq e^{-(1-1/\alpha)^2 \mu/2} \leq e^{-2 \ln(m)} = \frac{1}{m^2},$$

as desired. □

Theorem 6.10. *With constant probability, the randomized rounding algorithm returns a feasible solution of cost at most $O(\ln(m)) \text{OPT}_{LP}$.*

Proof. By Markov's inequality, we have

$$\sum_j c_j z_j \leq 2 \mathbf{E} \left[\sum_j c_j z_j \right] = 2\alpha \text{OPT}_{\text{LP}}$$

with probability of error at most $1/2$. Each constraint i is satisfied with probability of error at most $1/m^2$.

By the union bound, the probability of z not being a feasible solution of cost at most 2OPT_{LP} is at most

$$\frac{1}{2} + \mathbf{P}[z \text{ fails any constraint}] \leq \frac{1}{2} + \sum_i \mathbf{P}[z \text{ fails constraint } i] \leq \frac{1}{2} + \frac{1}{m},$$

as desired. □

6.4 Exercises

Exercise 6.1. . Complete the proof of theorem 6.4.

Exercise 6.2. Extend the approximation algorithm for set cover to positive costs. For each set, there is a positive cost $c_j > 0$. The goal is to compute the minimum cost collection of sets that covers all the points.

-

Exercise 6.3. 1. Design and analyze a deterministic $3/4$ -approximation algorithm for max-SAT.²

2. Design and analyze a deterministic $(1 - 1/e)$ -approximation algorithm for set cover.

3. (Harder) Design and analyze a deterministic $(1 - 1/e)$ -approximation algorithm for covering integer programs.³

²You may first want to design and analyze a deterministic $(1 - 1/e)$ -approximation algorithm for max-SAT.

³For CIP's, unlike set cover, it is not clear how to compute exactly the probability of a constraint being satisfied by a randomly rounded solution. But you might find a useful upper bound on this probability by studying the proof of the multiplicative Chernoff bound.

Exercise 6.4. Consider an instance of (weighted) set cover defined by sets $S_1, \dots, S_n \subseteq [m]$ and costs $c_i > 0$ for each set S_i . The goal is to compute the minimum cost collection of sets covering $[m]$. We say that solving the LP and then randomly rounding gives a $O(\log m)$ approximation. Here we consider a special case where all the sets are small and obtain a better approximation factor by a standard extension of randomized rounding called *alterations*.

Let $\Delta \in \mathbb{N}$ be such that $|S_j| \leq \Delta$ for all j . Consider the algorithm **round-and-fix** for which some pseudocode is given below. **round-and-fix** is similar to randomized rounding and has two stages. The first stage solves the LP and then *rounds* the solution scaled up by some factor $\alpha \geq 1$. It is possible that some of the elements $i \in [m]$ may not be covered. In the second stage, we *fix* each uncovered element by (deterministically) taking the cheapest set that covers it.

round-and-fix(sets $S_1, \dots, S_n \subseteq [m]$, costs $c \in \mathbb{R}_{>0}^n$, $\alpha \geq 1$)

1. let $x \in [0, 1]^n$ solve the set cover LP
2. let $F \subseteq \{S_1, \dots, S_n\}$ sample each set S_i independently with probability $\min\{1, \alpha x_i\}$
3. for each $i \in [m]$
 - A. if i is not covered by F
 1. add the cheapest set covering i to F
4. return F

Show that for an appropriate choice of α , this algorithm returns a $O(\log \Delta)$ approximation to the set cover instance (in expectation). (It is possible to get $\log \Delta + \log \log \Delta + O(1)$ with care.)

Exercise 6.5. The defining characteristic of LPs is that the objective and all linear constraints are given by linear functions. It is natural to generalize this notion and consider mathematical programs where the objective and linear constraints are all given by low-degree polynomials; say, bounded by a degree d . Let us call these “degree d polynomial programs”. Linear programs are degree 1 polynomial programs.

Prove that degree d polynomial programs are NP-Hard to solve for $d \geq 3$. To this end, pick a suitable NP-Hard problem, and design a degree 3 polynomial program that can be rounded to a discrete solution without any loss.

Exercise 6.6. or CIPs, we could assume without loss of generality that $A_{ij} \leq b_j$ for all i, j . (In fact this assumption was critical for applying the Chernoff bound.) Suppose now that we had $\lambda A_{ij} \leq b_j$ for all i, j for a parameter $1 \leq \lambda \leq \log(n)$. Design and analyzing a $O(\log(m)/\lambda)$ -approximation algorithm for this setting.