20

[written]

cs473 Algorithms: Lecture 18 (2022-03-29)

logistics: - pset 7 due F17

last lecture: randomized algo
- recap
- global mincut in undirected G
  - vs directed (s,t)-cut
  - random contraction [w/ repeat] [simple]

today: linear programming

recall: reductions [convert to existing problem - forward - backward] [don't redo algo from scratch]

bipartite matching ≤ max flow [flow] ≤

(single source) shortest paths [dyna program] ≤ ?

... ≤

[are also for all problems?]

SSSP vs maxflow vs mincut maxflow

Q: common generalization? [obvious approach, not really true] [specialized algs could still do better] P completeness. CS:

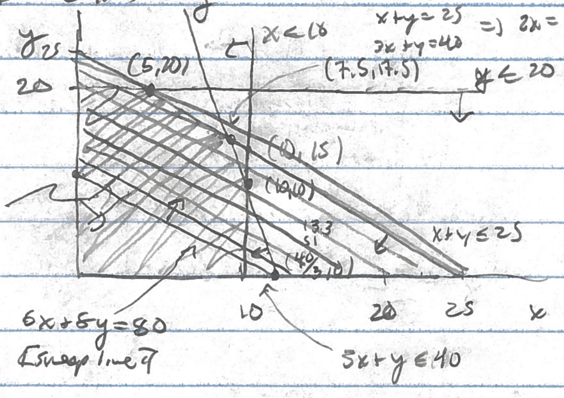
ex: [pandemic art, time to quit for dream job] opening new bubble tea shop [how to maximize profit]

two offerings
- black tea, w/ tapioca pearls ← x = # gallons/day [prepare daily]
- jasmine green tea, w/ tapioca pearls ← y = # gallons/day

constraints
- $x \le 10$ [# black tea bags available]
- $y \le 20$ [# green tea bags]
- $x + y \le 25$ [30] [amount of pearls available] [each drink has same # pearls]
- $3x + y \le 40$ [amount of sugar] [black better]

Maximize [profit per gallon] $6x + 8y$ [black] [green, premium]

linear program



x+y=25, 3x+y=40 ⇒ 2x = 15 ⇒ x = 7.5 ⇒ y = 17.5

feasible region

[extreme points] $6 \cdot 5 + 8 \cdot 20 = 190$

$6 \cdot 7.5 + 8 \cdot 17.5 = \underbrace{3 \cdot 15}_{45} + \underbrace{4 \cdot 35}_{140} = 185 < 190$

$6 \cdot 10 + 8 \cdot 10 = 140 < 190$

⇒ max profit per day is 190

rmk:
- continuous region to optimize over ⇒ no $\boxed{\text{obvious}}$ finite time algo
- possibly $\boxed{\text{high}}$ dimensional [dimensional] [high-dimensional space difficult to work with] [det] [will see] [finite #]
- optimum at extreme points ⇒ finite time algs

def = A *linear program* (LP) is given by

$c \in \mathbb{R}^n$

$A_\le, A_=, A_\ge \in \mathbb{R}^{m\times n}$ [matrices]

$b_\le, b_=, b_\ge \in \mathbb{R}^m$

and asks for

- x vars
- linear constraints

$$\text{max} \sum_{i=1}^n c_i x_i$$

st [constraints]

$$\forall\, 1\le i\le m \quad \sum_{j=1}^n (A_\le)_{ij} x_j \le (b_\le)_j \quad [\le \text{ constraint}]$$
$$\forall\, 1\le i\le m \quad \sum_{j=1}^n (A_=)_{ij} x_j = (b_=)_i \quad [= \text{ constraint}]$$
$$\forall\, 1\le i\le m \quad \sum_{j=1}^n (A_\ge)_{ij} x_j \ge (b_\ge)_j \quad [\ge \text{ constraint}]$$

Π

2019 typo

- draw

- $|\Pi|$

equiv [matrix notation] $|\Pi| = \max \ \langle c, x\rangle$ [inner product]

st [matrix-vector]

$$A_\le \cdot x \le b_\le \quad \text{coord-wise}$$
$$A_= \cdot x = b_=$$
$$A_\ge \cdot x \ge b_\ge$$

conversion vs size

the *input size* is n = # variables [x]

m = # constraints [3m in above]

total bit complexity of $c, A_\le, A_=, A_\ge, b_\le, b_=, b_\ge$

Π is *feasible* if exists $x\in\mathbb{R}^n$ satisfying constraints [no max yet]

feasible point

else *infeasible*

- vs polytope

Π is bounded if $|\Pi| < \infty$, else *unbounded*.

Q: given Π, compute $|\Pi|$? [compute maximizing point]

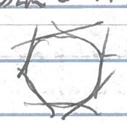
obj vs linear

rmk: - linear programming ← like "dynamic programming"

- linear constraints
- linear optimization

- maximize ≡ minimize as $\min \langle c,x\rangle = -\max\langle -c, x\rangle$

- m $\boxed{\text{not}}$ bounded function of n [vs max $\le n^2$ edges in graphs]

eg in $\mathbb{R}^2$  circle needs ∞-many constraints

- [counted] continuous optimization ⇒ no $\boxed{\text{obvious}}$ finite time alg. [no brute force]

[long def]

def = A *canonical form* linear program is of the form

$$\max \ \langle c, x\rangle$$
$$\text{st} \quad Ax \le b$$
$$x \ge 0$$

circulated, xx/yy.

lem - Π linear program $\max \langle c, x\rangle$ $A_{\le} x \le b_{\le}$ $A_{=} x = b_{=}$ $A_{\ge} x \ge b_{\ge}$ then exists canonical Π' $\max \langle c, x\rangle$ st $A'x' \le b'$, $x \ge 0$

[reduction, like next two] then exist efficient maps $x \mapsto x'$ [forward] $x \leftarrow x'$ [backwards]

(open ref.)

st x feasible iff x' feasible, $\langle c, x\rangle = \langle c', x'\rangle$] $\Rightarrow |\Pi| = |\Pi'|$ ["same" LP]

pf. $A_{\ge} x \ge b_{\ge} \equiv (-A_{\ge}) x \le (-b_{\ge})$ iso LP.

$A_{=} x = b_{=} \equiv \begin{bmatrix} A_{=} \\ -A_{=} \end{bmatrix} x \le \begin{bmatrix} b_{=} \\ -b_{=} \end{bmatrix}$ block matrix notation. $-A_{=}$.

$\Rightarrow$ Π wlog $\max \langle c, x\rangle$ st $Ax \le b$ [correctness] [efficiency] [need "$x' \ge 0$"]

create x' as $x^+, x^- \in \mathbb{R}^n_{\ge 0}$ [2n vars]

by x_i^+ [pos part] $= \begin{cases} x_i & x_i \ge 0 \\ 0 & \text{else} \end{cases}$ x_i^- [neg part] $= \begin{cases} -x_i & x_i \le 0 \\ 0 & \text{else} \end{cases}$

$\Rightarrow$ $x = x^+ - x^-$, $x^+, x^- \ge 0$ [forward and backwards] $x = x^+ - x^-$ but not at constructed form

define A' by $Ax = Ax^+ - Ax^- \le b$ $= \underbrace{\begin{bmatrix} A & -A \end{bmatrix}}_{A'} \underbrace{\begin{bmatrix} x^+ \\ x^- \end{bmatrix}}_{x'} \le \underbrace{b}_{b'}$

define c' by $\langle c, x\rangle = \langle c, x^+ - x^-\rangle$

$= \langle c, x^+\rangle - \langle c, x^-\rangle = \langle c', x'\rangle$ details.

complexity: clear

correctness: x feasible $= Ax \le b \mapsto x = x^+ - x^-$, $x^+, x^- \ge 0$; $A(x^+ - x^-) \le b$, $x^+, x^- \ge 0$ [x' feasible]

x' feasible $= A(x^+ - x^-) \le b$, $x^+, x^- \ge 0 \mapsto x = x^+ - x^-$ $Ax \le b$, $\langle c, x\rangle = \langle c', x'\rangle$ [clear] $\langle c', x'\rangle = \langle c, x\rangle$ [same cost] $\blacksquare$

[gives nicer form]

Q: max-flow vs LP?

prop: $G = (V, E)$ capacitated [directed] graph, $s \ne t \in V$ capacities $(c_e)_{e \in E} \ge 0$

max [value] flow in G $= \max f(s)$ [net flow] $= \sum_{e: s \to \cdot} f_e$ [out] $- \sum_{e: \cdot \to s} f_e$ [in] [linear]

st $\forall e$ $f_e \ge 0$ [non-neg flow] [linear]

$f_e \le c_e$ [capacities] [linear]

$\forall v \ne s, t$ $f(v) = 0$ [conservation] $= \sum_{e: v \to \cdot} f_e$ [out] $- \sum_{e: \cdot \to v} f_e$ [in] [linear]

[clear] interesting.

[mincut]

prop: $G = (V, E)$ capacitated graph, $s \ne t \in V$, capacities $(c_e)_{e \in E} \ge 0$

[min s-t cut] $\min_{V = S \sqcup T,\ s \in S,\ t \in T \text{ [s-t cut]}} |C(S, T)|$ [cut capacity] $= \sum_{e: u \to v,\ u \in S,\ v \in T} c_e$ $=$ [LP] $\min \sum_e c_e \cdot x_e$ st $\forall v \in V$ $d_s = 0$, $d_t = 1$, $\forall e: u \to v$ $d_v \le d_u + x_e$ [distance vs potential] $d, x \ge 0$?

maxflow mincut

pf: $\geq$: given $V = S \sqcup T$ ($s \in S$, $t \in T$) define $d_v = \begin{cases} 0 & v \in S \\ 1 & v \in T \end{cases}$

define $x_e = \begin{cases} 1 & e: u \to v \quad u \in S, v \in T \\ 0 & \text{else} \end{cases}$ $\rightarrow \begin{cases} d_s = 0 \\ d_t = 1 \end{cases}$

clm: $\forall e: u \to v \quad d_v \leq d_u + x_e$ [def of x_e]

pf: [4 possibility]

d_v	d_u	x_e	
0	0	0	✓
0	1	0	✓
1	0	1	✓
1	1	0	✓ □

$\Rightarrow (d_v)_v, (x_e)_e$ feasible

[defn]

$$C(S,T) = \sum_{\substack{e: u \to v \\ u \in S,\ v \in T}} c_e \cdot 1 = \sum_e c_e \cdot x_e = \langle c, x \rangle \geq \min_{\text{st} \ldots} \langle c, x \rangle$$

[$x_e = 1$ iff e crosses cut] [feasible] [min]

·approx·

$\leq$: given d optimal w/ $d_s = 0$, $d_t = 1$, $d_v \leq d_u + x_{e: u \to v}$ — possibly $\boxed{\text{non-integral!}}$

[cut indicators are integral] [general feasible soln non integral] [how to convert?]

idea - randomized rounding

• $(0,1]$ vs $[0,1]$

algo. pick $\theta \in (0,1]$ uniformly

define $S = \{v : d(v) < \theta\}$

output $V = S \sqcup T$ [cut]

clm: $s \in S$, $t \in T$

need $x_e \geq 0$? vs $c_e \geq 0$

pf: $d_s = 0 < \theta$ [any θ], $d_t = 1 \geq \theta$ [any θ]

clm: $\mathbb{E}[C(S,T)] \leq |\Pi|$

cor: $\boxed{\text{exists}}$ $V = S \sqcup T$ w/ $C(S,T) \leq |\Pi|$ [randomization cannot always be about mean] [prob method]

pf. $\mathbb{E}[C(S,T)] = \mathbb{E}[\sum_{e: u \to v} c_e \cdot \mathbb{1}[u \in S, v \in T]]$

[linearity] $= \sum_e c_e \cdot \underbrace{\Pr[u \in S, v \in T]}$

• $d_v < d_u$ possible, $x_e > 1$, $d_v, d_u \notin [0,1]$

$= \Pr_\theta[d_u < \theta \leq d_v]$ ← LP feasible

$d_v \leq d_u + x_e$, $x_e \geq 0$

$\leq x_e$

$= \langle c, x \rangle = |\Pi|$ □

rmk: use of randomness was $\boxed{\text{passenger}}$, can be made efficient

today: linear programming - example [bubble tea]
- def
- reduction to canonical form [forward, backward]
- LPs vs max flow [rescale], min cut [discrete vs continuous] [randomized rounding]

next lecture - linear programming

logistics - pset 7 due F 17