Description of LPs
~~Motivation~~ by a simple example: the diet problem.

- Go to doctor & he says you need to eat healthier:

  energy 2000 kcal/day
  protein 55 g/day
  calcium 800 mg/day

- You are picky ~~but also poor~~. So you write out all the foods you like eating.

	Energy/s	Protein/s	Calcium/s	Cents/s	Max servings
- oatmeal	110	4	2	3¢	4
- chicken	205	32	12	24¢	3
- eggs	160	13	54	13¢	2
- milk	160	8	285	9¢	8
- pie	420	4	22	20¢	2
- pork w/ beans	260	14	80	19¢	2

- You are also poor, so you note the cost per serving

- Question: what is the cheapest way to get all your nutrients, in terms of servings?

let x_o be # servings of oatmeal, x_c, x_e, x_m, x_p, x_b.

min. or. max.
(i) linear fcn

$$\boxed{\begin{array}{l} \text{minimize } 3\cdot x_o + 24\cdot x_c + 13\cdot x_e + 9\cdot x_m + 20\cdot x_p + 19 x_b \\ 110\cdot x_o + 205\cdot x_c + 160\cdot x_e + 160\cdot x_m + 420\cdot x_p + 260\cdot x_b \geq 2000 \\ 4\cdot x_o + 32\cdot x_c + 13 x_e + 8\cdot x_m + 4\cdot x_p + 14\cdot x_b \geq 55 \\ 2\cdot x_o + 12\cdot x_c + 54 x_e + 285 x_m + 22\cdot x_p + 80 x_b \geq 800 \\ x_* \geq 0 \\ x_o \leq 4;\ x_c \leq 3;\ x_e \leq 2;\ x_m \leq 8;\ x_p \leq 2;\ x_b \leq 2 \end{array}}$$

(ii) constraints are linear equalities or inequalities

show blank then go straight to solution file

LAB 0

this is an LP for diet prob.
what makes it an LP? (i) & (ii)

Linear Programs generally

(1)

n variables $x_1, x_2, \dots, x_n \in \mathbb{R}^n$

LP: $\max. \sum_{j=1}^{n} c_j x_j$ ← obj fcn
s.t.

$$\left.\begin{array}{l} \sum_{j=1}^{n} a_{1j} x_j \leq b_1 \\ \sum_{j=1}^{n} a_{2j} x_j \leq b_2 \\ \vdots \\ \sum_{j=1}^{n} a_{mj} x_j \leq b_m \end{array}\right\} \text{constr} \qquad \begin{bmatrix} \leftarrow a_{1j} \rightarrow \\ \leftarrow a_{2j} \rightarrow \\ \vdots \\ \leftarrow a_{mj} \rightarrow \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \leq \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

$x_j \geq 0 \quad \forall j \in \{1, \dots, n\}$) non-negativity constrs

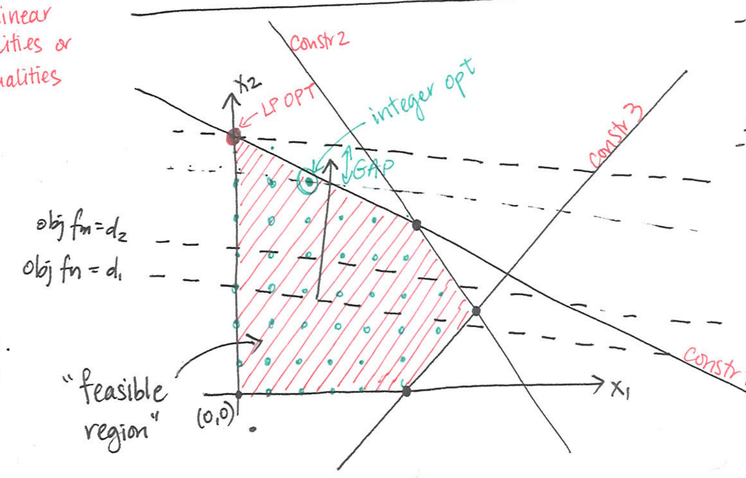
$x_j \in \mathbb{Z}$ for some of j's in $\{1, \dots, n\}$ ← makes it an IP!

Compact form: $\vec{x}$ variables $\vec{x} = (x_1, \dots, x_n)$
$\vec{c} = (c_1, \dots, c_n)$
$\vec{b} = (b_1, \dots, b_m)$
A

problem size: (n, m)

LP: $\max.\ \vec{c}\cdot\vec{x}$ ← obj fcn
s.t.
$A\vec{x} \leq \vec{b}$ ← m constrs
$\vec{x} \geq \vec{0}$ ← non-neg. constrs

"standard form"
- ineqs in either direction, equalities

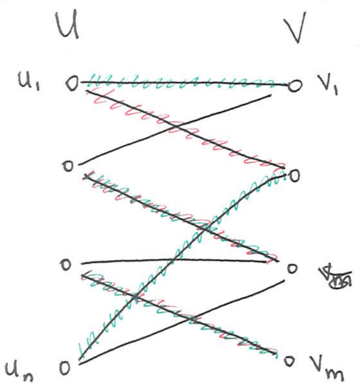


- for n vars, feas. region is n-dim. polytope
- feas reg can be
  - empty
  - unbounded

Sometimes we want to enforce that vars x_j take on integral values.

- show how formulation changes
- visually, superimpose integer 2-d grid
- Note! our feasible region is not ~~continuous~~ convex or compact anymore!
- Note! integral opt ~~doe~~ may not be at a vertex pt of the linear "relaxed" feasible region!
- Note! diff. b/t LP OPT & IP OPT is the "integrality gap" for this IP.
  - ↪ we'll more often be interested in a family of IPs associated with a problem. "worst" biggest integrality gap over

Assigning children to beds (patients to organ donors, advertisers to adslots, classes to classrooms, jobs to machines, ...)



bipartite graph $G = (U, V, E)$

$E \subseteq U \times V$. ← no edges b/t nodes on same side
⇒ no odd cycles ⟺ G is 2-colorable (would have seen in CMSC451)

a matching is an "valid assignment" b/t U & V
↪ subset M of edges s.t. no node is incident to more than 1 edge in M

goal: find matching w/ maximum number of edges in it. (i.e. max. $|S|$).

②

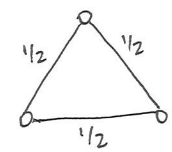
IP: $x_e \in \{0,1\}$ $\forall e \in E$ indicates whether $e \in M$.

$$\boxed{\begin{aligned} &\max. \sum_{e \in E} x_e \text{ s.t.} \\ &\sum_{u \sim e} x_e \le 1 \quad \forall u \in U \\ &\sum_{v \sim e} x_e \le 1 \quad \forall v \in V \\ &\cancel{x_e \in \{0,1\}} \quad x_e \in [0,1] \\ &\text{LP relaxation} \end{aligned}}$$

Solve IP.
Solve LP. } LAB 1 PREP?

(I) This problem has ~~as f~~ the property that:

- for G bipartite, the LP OPT is always integral! (show in visual)
- for general G, this is not true:



← this non-integral assignment is feasible in the LP relaxation, w/ value 1.5, but ~~the~~ a max. matching has value 1. !

⇒ for general G, need additional constraints to "stop this from happening".
  ⇒ it is a common technique to impose additional consts that don't change the IP OPT, but ~~the~~ brings LP OPT closer to IP OPT ("reducing gap")

(II) Solving IPs is NP-hard, but ~~okay~~ good for small, "nice" instances!

LPs ∈ P.

Next: algs & heuristics for LPs & IPs.

in practice, common alg is not poly time, but does well...

(II) this is a toy prob, as we have poly-time algs to solve max. matching. HOWEVER! MWPM's best poly-time alg is LP-based! Beyond scope...

Simplex method for Solving LPs: Suppose $\vec{b} > \vec{0}$.

max. $\vec{c}\cdot\vec{x}$ s.t.

$\vec{A}_1 \cdot \vec{x} \le b_1$

$\vec{A}_2 \cdot \vec{x} \le b_2$

$\vdots$

$\vec{A}_i \cdot \vec{x} \le b_i$

$\vdots$

$\vec{A}_m \cdot \vec{x} \le b_m$

$\vec{x} \ge 0$

orig LP

$z = \vec{c}\cdot\vec{x}$

$s_1 := b_1 - A_1 \cdot x$

$s_2 := b_2 - A_2 \cdot x$

$\vdots$

$s_m := b_m - A_m \cdot x$

$\underbrace{s_i}_{\text{positive LHS}} := \underbrace{b_i - A_i \cdot x}_{\text{zero RHS}}$

max. z s.t.

$s_1 \ge 0$

$\vdots$

$s_m \ge 0$

{all the equalities}

$x_j \ge 0,\ j = 1 \ldots n$

these LPs are equiv: 1-1 corresp. b/t feas x and feas x

(i) introduce new vars $s_1, \ldots, s_m$.

(ii) we will ~~maintain~~ express LP in an equivalent form & maintain invariant that for the equalities, LHS are the variables that are positive in our feas sln & they are expressed in terms of RHS vars, which are 0 in our feas sln.

Simplex alg

- start w/ feas sln $\vec{x} = \vec{0}$. This sets $\vec{s} > \vec{0}$.
- do repeatedly:
  - over RHS vars whose increase would increase z, (look at coeffs) choose one & increase it until some inequality becomes tight, i.e. $s_i = 0$. This is the next feas. sln.
  - set up for the next iteration: s.t. positive vars LHS & zero vars RHS:
    - may mv. x_j to LHS
    - will move at least s_i to RHS

$$\begin{bmatrix} s_i \overset{(>)}{\ge} 0 \ \downarrow \text{to zero} \\ x_j \overset{>}{\underset{(=)}{=}} 0 \ \uparrow \\ s_i = \ldots + a_{ij}x_j + \ldots \end{bmatrix} \rightarrow \begin{bmatrix} s_i \overset{(=)}{\ge} 0 \\ x_j \overset{(>)}{\ge} 0 \\ x_j = \frac{1}{a_{ij}}(s_i - \ldots) \end{bmatrix}$$

  - represent $x_j = \ldots$ by tight constraint
  - represent rest of constraints by subbing x_j for the new RHS in first constraint
  - also update z to be in terms of RHS constraints (sub x_j for its RHS)
- until $\nexists$ RHS vars whose incr. would incr. z.

- this is optimal to orig LP: all feas slns satisfy $z = \text{const} - \text{RHS}_1 - \text{RHS}_2 - \ldots -$ our ending feas sln satisfies $z = \text{const}$, since RHS vars are zero

---

Solving LPs:

- Simplex method: traverses vertex pts of feas. region. not poly-time. good in practice.
- 1970's ellipsoid method. increasingly shrinking (by volume) ellipsoids to contain OPT: $O(n^6 L)$ poly-time. not good in practice (showed LP ∈ P)
  - use ellipsoid's center to construct new ellipsoid
- interior point methods: e.g. Karmarkers. $O(n^{3.5} L)$ poly time, reasonable in practice. iteratively traverses "interior" of feas. region:
  - determine direction twd optimal
  - determine scale to next pt to stay in feas. region.

Solving IPs:

- IPs are very powerful. As expressive as SAT. ⇒ NP-hard.
- branch-and-bound: for min. probs
  - relax IP to LP.
  - solve it.
    1. if LP OPT is integral, do not branch, update curr. best known
    2. or its not feas, or its
    3. LP OPT is not better than curr best → do not branch
  - pick fractional variable, $x_j = c \notin \mathbb{Z}$. "branching variable"
  - branch into 2 IPs: $\boxed{\text{OLD IP}}$ $x_j \le \lfloor c \rfloor$, $\boxed{\text{OLD IP}}$ $x_j \ge \lceil c \rceil$ } recurse, then pick better sln

→ Gives a search tree

Note: curr best known $\ge$ IP OPT (an integral sln)

also get LBs from at least LP OPTs.
- ~~when~~ gap is ever 0, we can finish early.

LAB 2: K-median

- Some big company, want to strategically place k new operations centers

- n cities $\{u_1, \dots, u_n\} = V$
- distances b/t cities $d(u,v)$

how to open to min. cumulative cost to "connect" each city to its nearest center? $d(\text{city}, \text{center})$

IP:

$y_u \in \{0,1\}$ $= 1$ iff we open u as a center.

$x_{v,u} \in \{0,1\}$ $= 1$ iff we connect city v to center u.

$$\min. \sum_{u,v} d(u,v) \cdot x_{v,u} \quad \text{s.t.}$$

$\sum_u y_u \leq k$ // do not open more than k centers

$x_{v,u} \leq y_u \quad \forall u,v \in V \times V$ // cities can only connect to open centers

$\sum_u x_{v,u} = 1 \quad \forall v \in V$ // every city gets connected to some center

$y_u \in \{0,1\}$

$x_{v,u} \in \{0,1\}$

Note: these constraints are very similar to k-center problem!

LAB 2.5: K-center

how to open to minimize the max dist of any city to its nearest center?

$$\min. \ r \quad \text{s.t.}$$

$d(u,v) \cdot x_{v,u} \leq r \quad \forall v,u \in V \times V$ // city v can only connect to center u if they are dist r. This is still linear!

$d(u,v) \cdot x_{v,u} - r \leq 0$

(same) // open $\leq k$ centers

(same) // connect only to open centers

(same) // connect every city to some center

$y_u \in \{0,1\}$

$x_{v,u} \in \{0,1\}$