CS 170 — 3.4 LECTURE 13

Linear programming is about expressing & solving linear optimization problems, and is not related to DP.

An optimization problem has the following form:

- n variables $x_1, \dots, x_n \in \mathbb{R}$
- objective function $f(x_1, \dots, x_n) \in \mathbb{R}$
- m constraints $c_1, \dots, c_m$ $\quad c_i(x_1, \dots, x_n) \in \{\text{true}, \text{false}\}$

We want $\max(f(x_1, \dots, x_n))$ s.t. $\forall i \in [m]$ $\quad c_i(x_1, \dots, x_n) = \text{true}$

LINEAR PROGRAMMING

f is a linear function, each c_i is a linear constraint

c_i can be inequalities, but cannot be strict ($\leq \geq$ ok, $<>$ not ok)

item	profit	max demand
chestnut box	\$2	50
pretzel	\$4	80

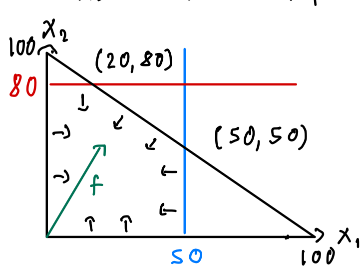
space in cart = 100

EXAMPLE

2 variables: x_1 = # chestnut boxes $\quad x_2$ = # pretzels

function: $f(x_1, x_2) = 2x_1 + 4x_2$

5 constraints: $x_1 \geq 0 \quad x_2 \geq 0 \quad x_1 + x_2 \leq 100 \quad x_1 \leq 50 \quad x_2 \leq 80$



orthogonal to f are "profit lines" where the value of f is the same

optimal here is $f(20, 80) = 360$

CONVEXITY

CLAIM Set of feasible solutions to a LP problem is convex: if $\vec{x}$ and $\vec{y}$ are feasible, then $\forall \lambda \in [0,1]$ so is $\vec{z} = \lambda \vec{x} + (1-\lambda)\vec{y}$

PROOF Follows from linearity of constraints.

$$\left.\begin{array}{l} \vec{a}^T \vec{x} \leq b \\ \vec{a}^T \vec{y} \leq b \\ \lambda \in [0,1] \end{array}\right\} a^T(\lambda \vec{x} + (1-\lambda)\vec{y}) = \lambda(\vec{a}^T\vec{x}) + (1-\lambda)(\vec{a}^T\vec{y}) \leq \lambda b - (1-\lambda) b = b$$

So, either no feasible solutions, no optimum in unbounded space, or optimum is a vertex of the polytope (intersection of half spaces).

In the last case, if we're in the interior, we can improve. If we're on an edge, we can move to a vertex and we'll improve or stay the same. Thus, considering just vertices will suffice.

Solving LPs

ALGO — Simplex method: start at a vertex, move to any adjacent vertex of higher value, and return when no such vertex exists

PROOF — Correctness follows from convexity, since no local maxima. Say we're at an assumed optimal v. Consider profit line through v. The feasible region is below this line, so v must be true optimum.

Most of the LP in this class will not involve solving high dimensions.

EFFICIENCY — Can we improve? # vertices $\le \binom{m}{n}$, so good news is finite time. Bad news is that this bound is exponential. There are polynomial time algorithms such as ellipsoid method and interior point methods.

EXAMPLE — A company wants to minimize cost of helmet production.

- monthly demand $d_1, \dots, d_{12}$
- employees cost 4000/month and produce 20 helmets/month $\quad w_0 = 50$
- overtime is 1.5x cost and maxes out at 25% time
- hiring is 300/worker, firing is 400/worker
- surplus storage is 10/helmet/month (0 at start and end)

w_i	workers in month i	$x_i = 20w_i + o_i$
x_i	production	$w_i = w_{i-1} + h_i - f_i$
o_i	overtime production	$s_i = s_{i-1} + x_i - d_i$
h_i, f_i	hires, fires at start of month i	$o_i \le 0.25(20w_i)$
s_i	storage at end of month i	

$$f = \min\left(4000\sum_i w_i + 300\sum_i h_i + 400\sum_i f_i + 10\sum_i s_i + \frac{3}{2}\cdot\frac{4000}{20}\sum_i o_i\right)$$

You may get a fraction that needs to be rounded, and may increase cost!