Algorithms

Linear programming: introduction

Recap

- Last HW: due Wed.
- Final: Monday of exam week at 8am

Review Friday before:

Have your finals schedule by <u>this</u> friday.

- Final topic: linear programming

Review worksheet
one question will be on final

NP-Completeness Recap

- Most useful:
  List of problems: YES!
  How to select a problem

Linear program

In a linear program, we are given a set of variables

The goal is to give these real values so that:

1. We satisfy some set of linear equations or inequalities

2. We maximize or minimize some linear objective function

↳ often, "profit"

An example: Maximize profit

A chocolate shop produces 2 products

x_1 - Type 1, worth \$1 each

x_2 - Type 2, worth \$6 each

Constraints:

eqn 1 - Can only produce 200 / of type 1 per day

2 - And at most 300 of type 2

3 - Total output per day of both is ≤ 400

LP: maximize $1 \cdot x_1 + 6 \cdot x_2$

eqn 1: $x_1 \leq 200$

2: $x_2 \leq 300$

3: $x_1 + x_2 \leq 400$

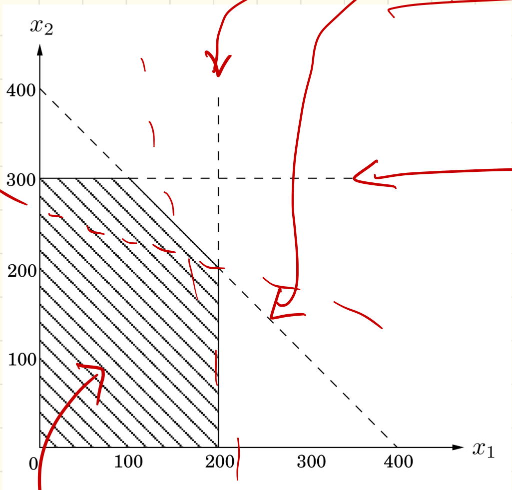
LP:

maximize $\underbrace{1\cdot x_1 + 6\cdot x_2}_{\text{line}}$

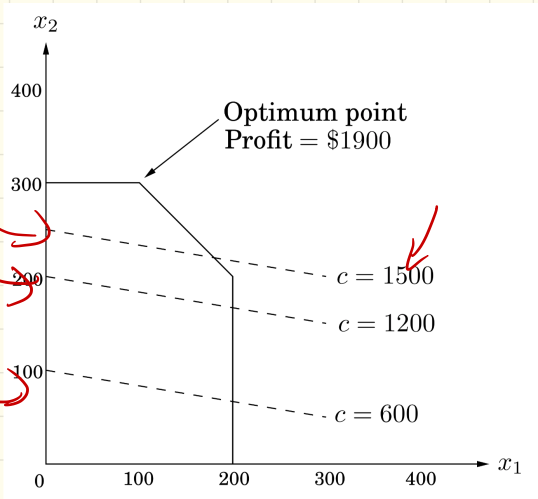
eqn 1: $x_1 \leq 200$

$x_2 \leq 300$

$\underline{x_1 + x_2 \leq 400}$



feasible soln



These go up in dimention with more x_i's:

new chocolate!

Maximize $x_1 + 6x_2 + 13x_3$

s.t.

$x_1 \le 200$ ①

$x_2 \le 300$ ②

$x_1 + x_2 + x_3 \le 400$

$x_2 + 3x_3 \le 600$

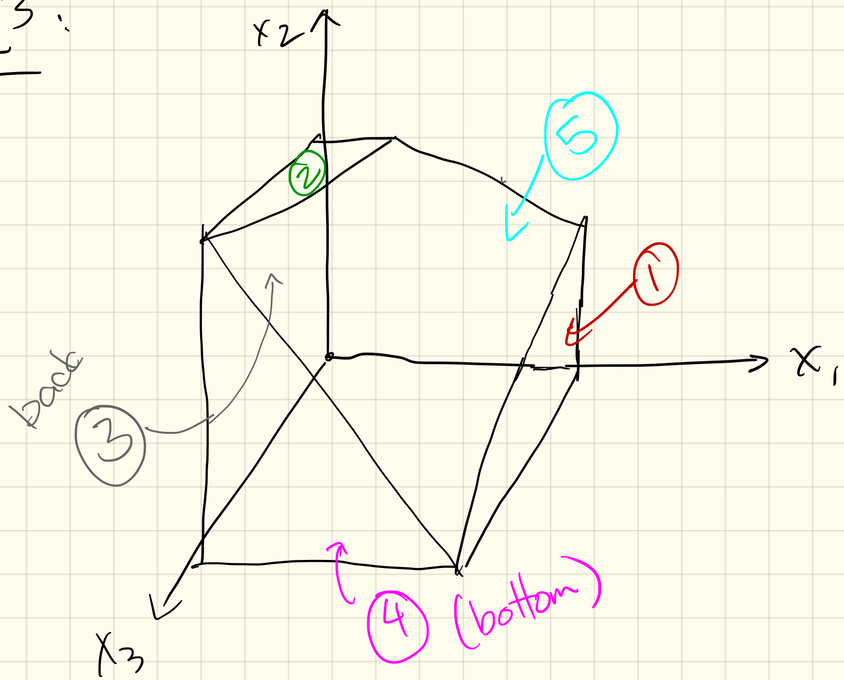
and

$x_1 \ge 0$ ③

$x_2 \ge 0$ ④

$x_3 \ge 0$ ⑤

$\mathbb{R}^3$:



Another (more general)

n foods, m nutrients

A_{ij} Let $a_{i,j}$ = amount of nutrient i in food j

vector $\vec{r}$: r_i = requirement of nutrient i

$\vec{x}$: x_j = amount of food j purchased

$\vec{C}$: c_j = cost of food j

Goal: Buy food so you satisfy nutrients while minimizing cost

The LP

$$\min \underbrace{\sum_j c_j \cdot x_j}_{\text{my cost}}$$

s.t. $a_{1,1} \cdot x_1 + a_{2,1} x_2 + \cdots + a_{n,j} \cdot x_n \geq r_1$

$a_{1,2} \cdot x_1 + a_{2,2} \cdot x_2 + \cdots + \_ x_n \geq r_2$

$\geq r_3$

$\vdots$

$\geq r_m$

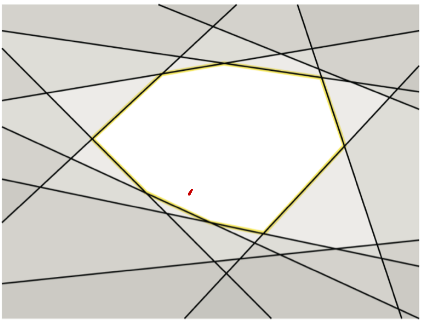
minimize $\bar{c} \cdot \vec{x}^T$

s.t. $A\vec{x} \leq \vec{r}$

In general, get systems like this:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{d} c_j x_j \\
\text{subject to} \quad & \sum_{j=1}^{d} a_{ij} x_j \le b_i \quad \text{for each } i = 1 \,..\, p \\
& \sum_{j=1}^{d} a_{ij} x_j = b_i \quad \text{for each } i = p+1 \,..\, p+q \\
& \sum_{j=1}^{d} a_{ij} x_j \ge b_i \quad \text{for each } i = p+q+1 \,..\, n
\end{aligned}
$$

Geometric picture:



A two-dimensional polyhedron (white) defined by 10 linear inequalities.

Canonical form:

Avoid having both $\leq$ and $\geq$.

So get something more like our first example:

$$\text{maximize} \sum_{j=1}^{d} c_j x_j$$

$$\text{subject to} \sum_{j=1}^{d} a_{ij} x_j \leq b_i \quad \text{for each } i = 1..n$$

$$x_j \geq 0 \quad \text{for each } j = 1..d$$

Or, given a vector $\vec{c}$ & matrix A:

$$\max \ \vec{c}\,\vec{x}^{T}$$

$$A\vec{x} \leq \vec{b}$$

Anything can be put into canonical form:

① Avoid $=$: $\sum_{i=1}^{d} a_{ij} x_i = b_j$ eqn j

change: 2 eqns:

$$\sum_{i=1}^{d} a_{ij} x_i \leq b_j$$

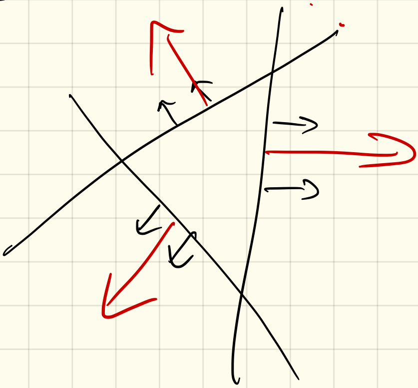
$$\sum_{i=1} a_{ij} x_i \geq b_j$$

② Avoid $\geq$:

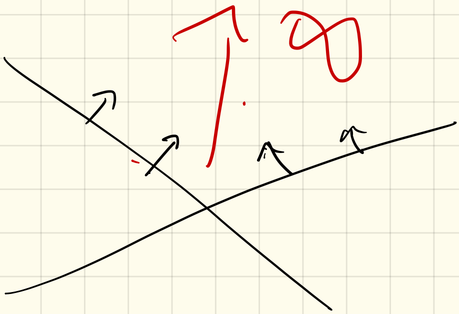
turn it into $-$

How could these not have a solution?

2 ways:

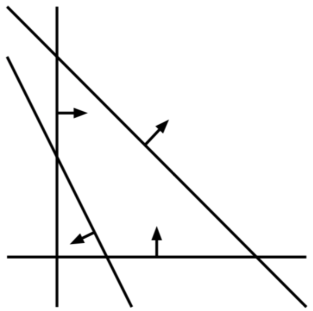


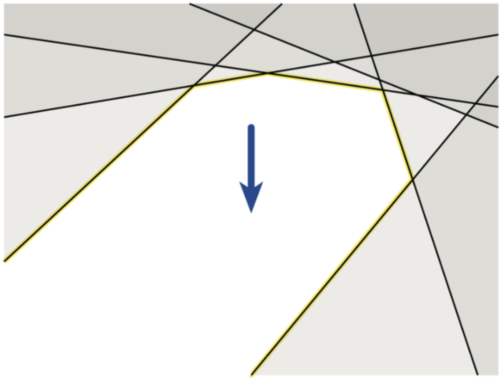
or



Better pictures (still 2-d):

$$\begin{aligned} \text{maximize } & x - y \\ \text{subject to } & 2x + y \le 1 \\ & x + y \ge 2 \\ & x, y \ge 0 \end{aligned}$$





Note:

① Multiplying by -1 turns any maximization problem into a minimization one:

② Can turn inequalities into equalities via slack variables:

$$\sum_{i=0}^{n} a_i x_i \leq b$$

$$\Downarrow$$

(3) Can change equalities into inequalities, also!

$$\sum_{i=1}^{n} a_i x_i = b$$

$$\Downarrow$$

Solving LP's:

The simplex algorithm