Linear Programming (LP)

Some Optimization Problems we've seen:

shortest paths, min cost spanning tree, min cut

LPs can be used to solve optimization problems with linear objective function & linear constraints.

Example:

Company makes 2 products A & B.
How much of each should it make to maximize profit?

For each unit of A, makes a profit of \$1
& for each unit of B makes \$6.

Demand is ≤ 300 units of A per day,
& ≤ 200 units of B per day.

Total supply is ≤ 700 hours per day,
where A takes 1 hour per unit & B takes 3 hours per unit.

Let x_1 = # of units of A produced per day
x_2 = # of " B " per day

associated LP:

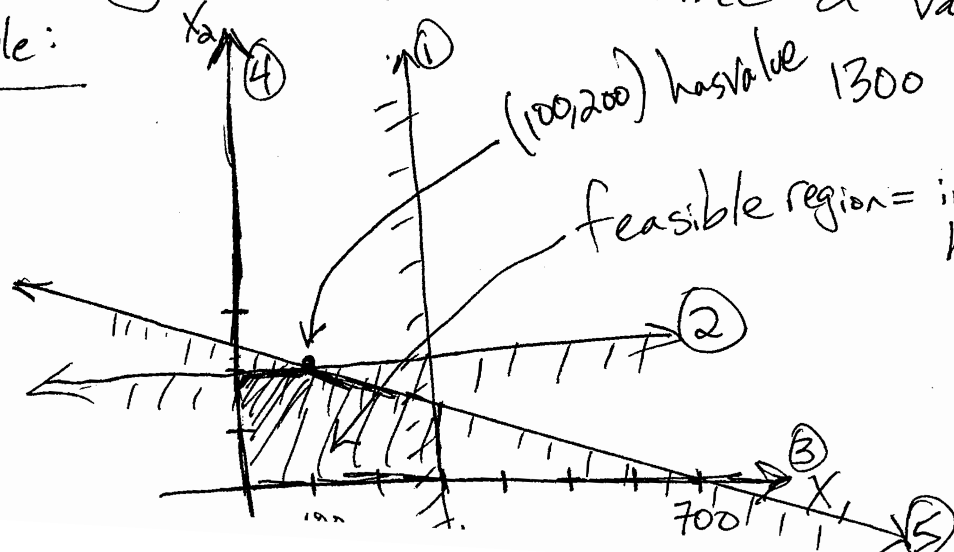
objective function: max $x_1 + 6x_2$

constraints:

$$x_1 \leq 300 \quad (1)$$
$$x_2 \leq 200 \quad (2)$$
$$x_1 \geq 0 \quad (3)$$
$$x_2 \geq 0 \quad (4)$$
$$x_1 + 3x_2 \leq 700 \quad (5)$$

Each constraint is a linear inequality = half-space lying in 2 dimensions since 2 variables x_1, x_2.

Above example:



Goal: find max c

where $x_1 + 6x_2 = c$ intersects the feasible region

such a point ~~is~~ (x_1, x_2) in the feasible region with max c is the optimum.

In the above example, the optimum is $c = 1300$ for $(x_1, x_2) = (100, 200)$

has equality for 2 & 5

& satisfies 1, 3, 4

Another LP example:

3 products A, B, C

Profit: \$1 per unit of A, \$6 per unit of B, \$10 per unit of C.

Demand: ≤ 300 units/day for A & ≤ 200 for B

Supply: ≤ 1000 hours total/day,
A takes 1 hour/unit, B takes 3 hours
& C takes 2 hours.

Packaging: ≤ 500 units total. ~~A~~
B takes 1 unit of packaging & C takes 3 units

LP:

Variables x_1, x_2, x_3.

$$\max \quad x_1 + 6x_2 + 10x_3$$

$$\text{s.t.} \quad x_1 \leq 300 \quad (1)$$

$$x_2 \leq 200 \quad (2)$$

$$x_1 + 3x_2 + 2x_3 \leq 1000 \quad (3)$$

$$x_2 + 3x_3 \leq 500 \quad (4)$$

$$x_1, x_2, x_3 \geq 0 \quad 5a, 5b, 5c$$

Line fitting:

Given points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

find the line $y = ax + b$ which minimizes the max distance.

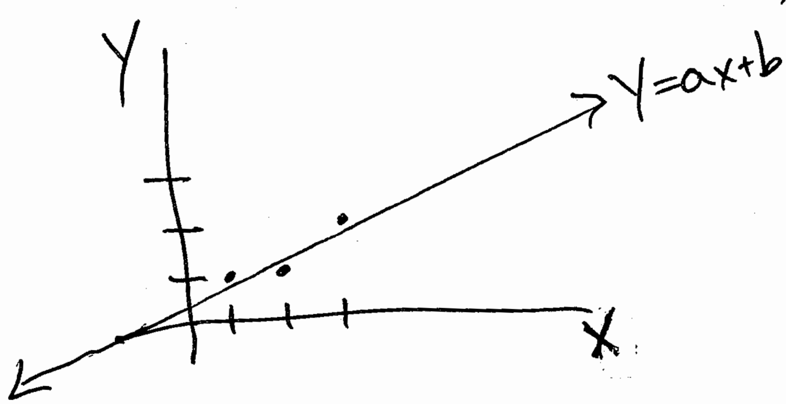
In other words, find a & b so that the line $y = ax + b$ achieves

minimum e where:

$$|\underbrace{y_i}_{\text{y-coordinate of } i^{th} \text{ point}} \underbrace{- ax_i - b}_{\text{y-coordinate of line at } x_i}| \le e \quad \text{for all } i$$

Example:

$(x_1, y_1) = (1, 1), \ (x_2, y_2) = (2, 1), \ (x_3, y_3) = (3, 2)$



LP: Variables a, b, e

$\min\ e$

s.t. $|1-a-b| \leq e \Rightarrow 1-a-b \leq e$ ①
$1-a-b \geq -e$ ②

$|1-2a-b| \leq e \Rightarrow 1-2a-b \leq e$ ③
$1-2a-b \geq -e$ ④

$|2-3a-b| \leq e \Rightarrow 2-3a-b \leq e$ ⑤
$2-3a-b \geq -e$ ⑥

Solution: $a=\frac{1}{2}, b=\frac{1}{4}, e=\frac{1}{4}$

Can we verify that this is optimal?

Look at ① + $-2\times$④ + ⑤

$$\begin{aligned}
① &= 1 - a - b \leq e \\
-2\times④ &= -2 + 4a + 2b \leq 2e \\
⑤ &= 2 - 3a - b \leq e \\
\hline
& 1 \leq 4e
\end{aligned}$$

$$\tfrac{1}{4} \leq e$$

So $e = 1/4$ is best possible.

Standard form for LPs:

Variable vector $x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$

objective function $c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}$

constraint matrix: $m \times n$ matrix A

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ & \vdots & \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$ m constraints

Generic form for the LP:

$$\max \; c^T x$$

$$Ax \leq b \qquad m \text{ constraints}$$

$$x \geq 0 \qquad n \text{ constraints}$$

For 2 variables, in 2 dimensions

constraint = linear inequality = half-plane

feasible region = points satisfying all constraints

= convex polygon for 2-dimension

In general, in n dimensions

constraint = half-space

feasible region = intersection of $(m+n)$ half-spaces

= convex polyhedron in n dimensions

Vertices = corners of the ~~f~~ feasible region

= intersections of sides of feasible region.

= point satisfying

n inequalities with equality

& satisfying remaining m inequalities

Thus, $\leq \binom{m+n}{n}$ vertices

Neighboring vertices share $n-1$ of their equality constraints.

How to solve LPs?

Simplex algorithm [Dantzig '47]
Widely used but exponential time in worst case
Ellipsoid algorithm [Khachiyan '79] = 1st poly-time algorithm
but huge polynomial.

Interior-point approach [Karmarkar '84] =
poly-time & also widely used.

Simplex algorithm:

starts at $x = \overline{0}$
Looks for neighboring vertex that has better objective value & then moves there & repeats

Once we're at a vertex that's better than all of its neighbors then we stop
$\Rightarrow$ this is the optimum by convexity.

Each step of simplex:

Naive approach: Gaussian elimination $\Rightarrow O(n^3)$ time to find a neighbor

$\times\, O(nm)$ neighbors

$\Rightarrow O(n^4 m)$ time per step

Better approach: $O(mn)$ time per step.

<u>But</u> worst case $O(m^n)$ steps.

<u>Key</u>:

Optimum of LP is achieved at a vertex of the feasible region except if:

a) infeasible

example: $x_1 \leq 100,\ x_1 \geq 200$

b) unbounded

example: $\max\ x_1 + x_2$

$x_1, x_2 \geq 100.$

Running simplex on earlier example with 3 products:

$$\max\ x_1 + 6x_2 + 10x_3$$

s.t.

$$x_1 \le 300 \quad (1)$$
$$x_2 \le 200 \quad (2)$$
$$x_1 + 3x_2 + 2x_3 \le 1000 \quad (3)$$
$$x_2 + 3x_3 \le 500 \quad (4)$$
$$x_1, x_2, x_3 \ge 0 \quad (5a, 5b, 5c)$$

Start at $(0,0,0)$ defined by (5a, 5b, 5c)
Profit $= 0$

$\longrightarrow (300, 0, 0)$ (1), 5b, 5c
Profit $= 300$

$\longrightarrow (300, 200, 0)$ (1), (2), (5c)
Profit $= 1500$

$\longrightarrow (300, 200, 50)$ (1), (2), (3)
Profit $= 2000$

$\longrightarrow (200, 200, 100)$ (2), (3), (4)
Profit $= 2400$

How do we know it's optimal?

Let $y = (y_1, y_2, y_3, y_4) = (0, \tfrac{1}{3}, 1, \tfrac{8}{3})$

Look at $y_1 \times (1) + y_2 \times (2) + y_3 \times (3) + y_4 \times (4)$

$$\begin{aligned}
y_1 \times (1) &= 0\,(x_1 \le 300)\\
y_2 \times (2) &= \tfrac{1}{3}\,(x_2 \le 200)\\
y_3 \times (3) &= 1\,(x_1 + 3x_2 + 2x_3 \le 1000)\\
y_4 \times (4) &= \tfrac{8}{3}\,(x_2 + 3x_3 \le 500)\\
\hline
& \underbrace{x_1 + 6x_2 + 10x_3}_{\text{Profit}} \le 2400
\end{aligned}$$

Profit ≤ 2400 ✓

How to convert an LP into generic form?

- for constraint: $\sum_{i=1}^{n} a_i x_i \geq b \Rightarrow \sum_{i=1}^{n} (-a_i) x_i \leq -b$

- for constraint: $\sum_{i=1}^{n} a_i x_i = b$

  replace with: $\sum_{i=1}^{n} a_i x_i \leq b$

  & $\sum_{i=1}^{n} a_i x_i \geq b \Rightarrow \sum_{i=1}^{n} (-a_i) x_i \leq -b$

- for objective function: $\min \ c^T x$

  replace with: $\max -c^T x$

- for x_i that are unrestricted (so no $x_i \geq 0$ constraint)

  add variables x_i^+ & x_i^-

  & replace x_i by $(x_i^+ - x_i^-)$

  & add constraints $x_i^+ \geq 0, x_i^- \geq 0$.