Crash Course on Linear Programming

$$\min \ \vec{c}\cdot\vec{x}$$

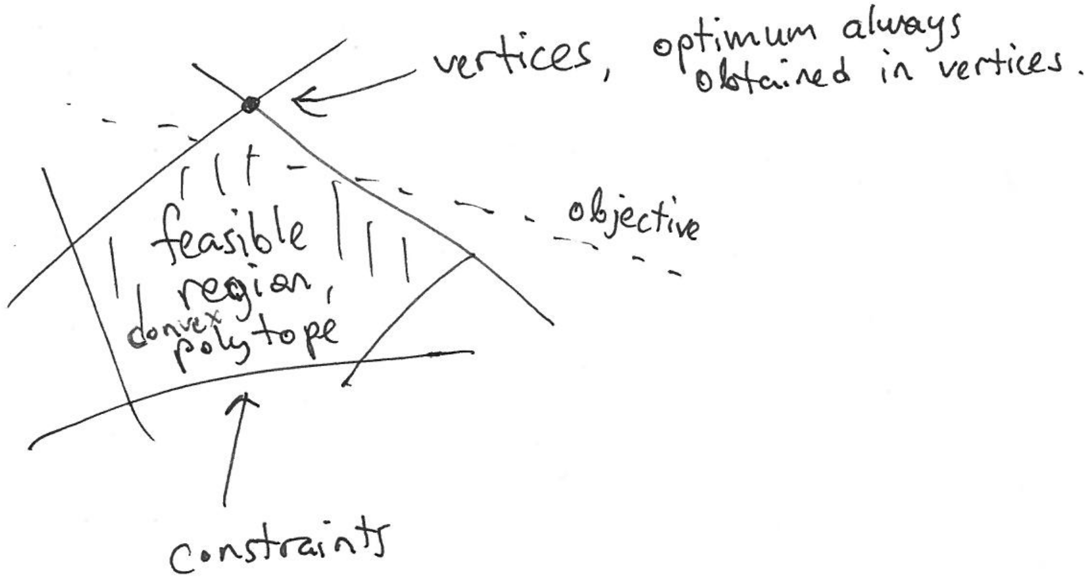
Subject to

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

$$\vec{x} \geq 0$$

- may change min→max
- may add $\leq$, $=$ constraints
- may remove $x \geq 0$
- can always transfer to normal form

When x is 2-dimensional:



Integer linear programming add constraints like $\vec{x} \in \{0,1\}^n$, $\vec{x} \in \mathbb{Z}^n$ and is NP-hard.

For real linear programming we have efficient algorithms.

Efficient algorithms for LP

<u>Simplex</u> walks from vertex to vertex in direction $\vec{c}$.

- Widely used in practice.
- Takes exponential time in worst case.

} "smoothed analysis" aims to explain that.

- There are variants that run in linear time.

<u>Ellipsoid</u> guarantee that opt is in ellipsoid and keep shrinking ellipsoid, overall polynomial time.

<u>Key</u> polynomial time <u>separation oracle</u>: given $\vec{x}$ either verify that $\vec{x}$ is feasible or give unsat constraint.

<u>Note</u> may be possible to solve exponential sized programs if they have a poly time separation oracle.

<u>Interior Point Method</u> random walk in interior of polytope guided by $\vec{c}$.

- polynomial time, active area of research.

Duality If someone gives you an optimal solution, you can check it efficiently!

More generally, you can efficiently show that

$$\min cx \text{ s.t } \begin{matrix} Ax \geq b \\ x \geq 0 \end{matrix} \quad \geq \ldots$$

Example

$$\min x+y$$
$$\text{s.t.} \quad 3x - y \geq 5$$
$$2x + 3y \geq 6$$
$$x, y \geq 0$$

$\min \geq 2$ since $\quad x+y \geq \frac{1}{3}(2x+3y) \geq 2$

Can we do better? want a, b, s.t.

$x+y \geq b(2x+3y) + a(3x-y)$ & $5a+6b$ is as large as possible.

In general, finding best certificate is a linear program:

primal $\quad \min cx$ s.t. $Ax \geq b$ $\qquad$ dual $\quad \max by$ s.t. $A^T y \leq c$

weak duality $\quad$ value of feasible to primal $\geq$ value of feasible to dual

strong duality $\quad$ equality when opt attained

Plan

- Crash course on linear programming.
- Approximation algorithms based on LP.

↳

1. Integer linear program
2. **Relaxation:** Real linear program (fractional)

Integrality gap: $\sup_{\text{inputs}} \dfrac{OPT_{\text{integer program}}}{OPT_{\text{real program}}}$ for minimization

3. **Rounding:** convert real solution to integral solution

Metric Uncapacitated Facility Location

where to open facilities so close to every vertex?

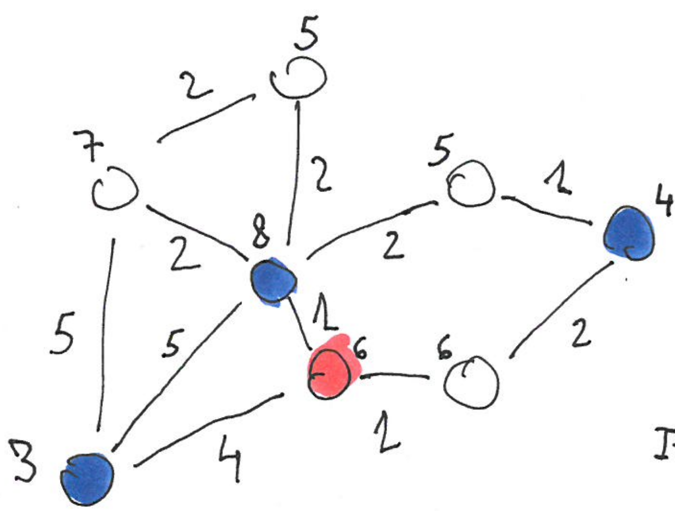
<u>Input</u> f_i $\quad i = 1 .. n$

c_{ij} $\quad 1 \le i, j \le n$ symmetric, triangle inequality

Find $F \subseteq V$ that minimizes

$$\text{cost} = \sum_{i \in F} f_i + \sum_{j \in V} c(j, F)$$

<u>Example</u>



cost of opening facilities:
$8 + 4 + 3 = 15$

cost of connecting:
$2 + 2 + 1 + 2 + 1 = 8$

23

If switch $8 \to 6$:

$6 + 4 + 3 = 13$

&

$3 + 3 + 1 + 1 + 1 = 9$

22

Integer linear program

$y_i = i \in F?$

$x_{ij} =$ is j assigned to facility i?

$$\min \sum_{i \in V} f_i y_i + \sum_{i,j \in V} c_{ij} x_{ij}$$

s.t.

(1) $\sum x_{ij} = 1 \quad \forall j$ (each vertex assigned to 1 facility)

(2) $x_{ij} \leq y_i \quad \forall i,j$ (no assiging to non-facility)

(3) $x_{ij} \in \{0,1\} \quad \forall i,j$ (indicator variables)

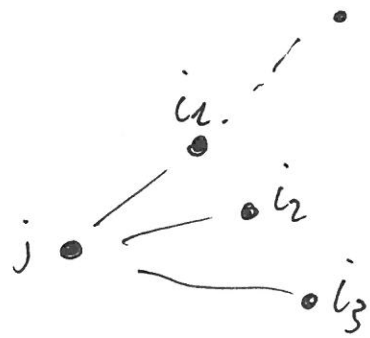
(4) $y_i \in \{0,1\}$ integral solution

Relaxation: replace (3), (4) w/ $x_{ij} \geq 0$, $y_i \geq 0$.

Rounding Fractional solution $\rightarrow$ integral solution without increasing cost much.

Step 1

Let fractional connection cost



$$\Delta_j = \sum_i c_{ij} x_{ij}$$

Define ball $B_j = \{ i \mid c_{ij} \le 2\Delta_j \}$

set $x'_{ij} = 0$ if $i \notin B_j$

$$x'_{ij} = \frac{x_{ij}}{\sum_{i \in B_j} x_{ij}} \quad \text{if } i \in B_j \qquad \text{to satisfy (1)}$$

Note that by Markov,

$$\left.\sum_{i \notin B_j} x_{ij} = P(c_{ij} > 2\Delta_j) < \frac{1}{2} \right\} \rightarrow \text{scaling } x'_{ij} \text{ by } \le 2.$$

Set $y'_i = \min\{1, 2y_i\}$ to satisfy (2).

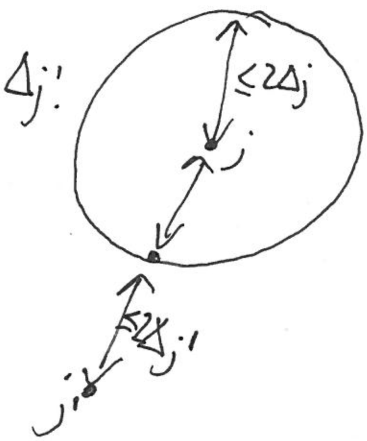
<u>Overall</u> cost increases by factor ≤ 2 & every vertex is assigned to ~~a~~ facilities close to it

Step 2

- Pick vertex j w/ smallest connection cost.
- Pick facility in B_j w/ smallest opening cost.
- Assign all vertices in B_j to that facility, don't open other facilies in B_j.

By that:

* opening cost didn't increase.
* connection cost increased by $\lesssim 6\Delta_{j'}$.



Overall, cost increased by $\leq$ factor 6.

---

can improve approximation factor to 4:

Instead of balls of radius $2\Delta_j$, take radius $(1+\alpha)\Delta_j$.

Scale up by $\leq \frac{1+\alpha}{\alpha}$ $\left(= 1 + \frac{1}{\alpha}\right)$

approx factor $\leq \max\left\{1+\frac{1}{\alpha},\ 3(1+\alpha)\right\}$

There's a 1.463 hardness

$\alpha = \frac{1}{3} \rightarrow$ factor ≤ 4