CS6363

Linear Programming (LP) → Chap. 29 → 29.1 (2-D)

(1)

- Many problems can be formulated as max/min an objective function subject to a set of constraints

- Linear Programming Problem:
  - objective ↦ linear function
  - constraints ↦ linear equalities / inequalities
  - if d variables are involved → d-D problem.

- linear function : $f(x_1, x_2, \dots, x_d) = a_1 x_1 + a_2 x_2 + \dots + a_d x_d$
- linear equality: $f(x_1, x_2, \dots, x_d) = b$

  inequality: $f(x_1, x_2, \dots, x_d) \leq b$, $\geq b$

  Note: strict inequalities not allowed!!

General LP form: maximize/minimize linear function of n variables subject to m constraints (linear inequalities)

Constant dimension LP: d variables subject to n constraints

- if $X = (x_1, x_2, \dots x_d)$ satisfies all constraints ⟹ feasible solution
- the set of all feasible solutions : $\emptyset$ / a convex region ↦ feasible domain / region

Example:

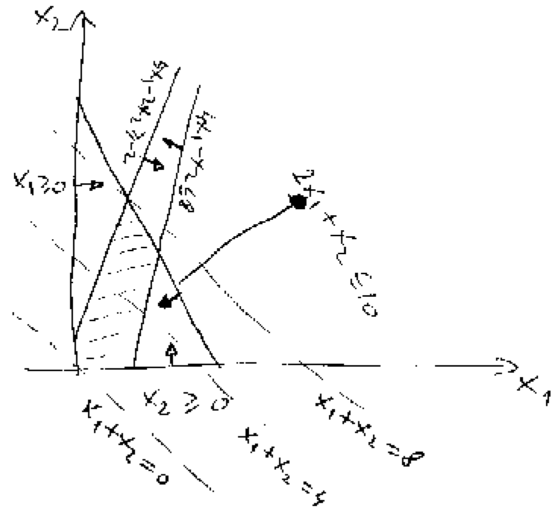
$\max \quad x_1 + x_2$

$\text{s.t.} \quad 4x_1 - x_2 \le 8$

$2x_1 + x_2 \le 10$

$5x_1 - 2x_2 \ge -2$

$x_1, x_2 \ge 0$



* Note: optimal solution at a vertex!

LP input:

- set of constraints: linear inequalities
- linear objective function
    - min / max s.t. constraints

- d-D: $\text{Max / Min}: \; c_1 x_1 + c_2 x_2 + \cdots + c_d x_d$

$$\begin{aligned} \text{s.t.}: \; & a_{11} x_1 + a_{12} x_2 + \cdots + a_{1d} x_d \le b_1 \\ & a_{21} x_1 + a_{22} x_2 + \cdots + a_{2d} x_d \le b_2 \\ & \vdots \\ & a_{n1} x_1 + a_{n2} x_2 + \cdots + a_{nd} x_d \le b_n \end{aligned}$$

} $\cap$ of halfspaces → convex polyhedron

- Can think of objective function as a vector

  $c = (c_1, c_2, \ldots, c_d)$

  - find vertex of feasible region furthest ~~lowest~~ in direction c $(-c)$.

  * $c \cdot x = b$ → equivalue plane (the set of all x)

- if FR is empty ⇒ LP is infeasible (no solution)
- if FR unbounded on $\vec{c} \to \infty$ ⇒ LP is unbounded (∞ solution)
- possible ∞ number of optimal solutions

Methods

- for high dimensional LP → simplex & interior point methods

- Simplex: find a vertex on FR then walk edge by edge on boundary until reaching a local max/min ↦ by convexity in global max/min.
  - exponential in worst case but fast in practice

- Interior point: polynomial time algorithm
  - move through interior of FR
  - polynomial in n, d, #bits in numbers

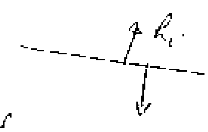
    (Open: strong. poly. algo: do not depend on #bits)

- in 2-D:
  - $O(n \log n)$: find FR then identify vertex
  - hope to do better by avoiding to compute FR.
  - $O(n)$ possible for any d-D LP, d constant ↳ $[O(d!\,n)]$

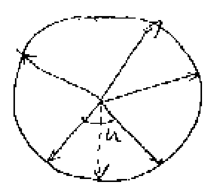
Unbounded LP

- can find if unbounded in $O(n)$ time or find a pair of bounding halfplanes in $O(n)$
  - extends to higher dimensions.

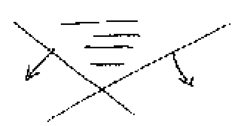
- For 2-D: consider <u>outward</u> normal vectors to halfplanes
  - partition unit circle of directions in n angular sectors



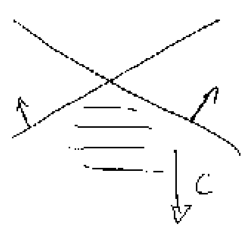
  - consider sector containing c



- if $u < 180°$ $\Rightarrow$ bounded (if not empty)



(the 2 halfplanes converge at a point)

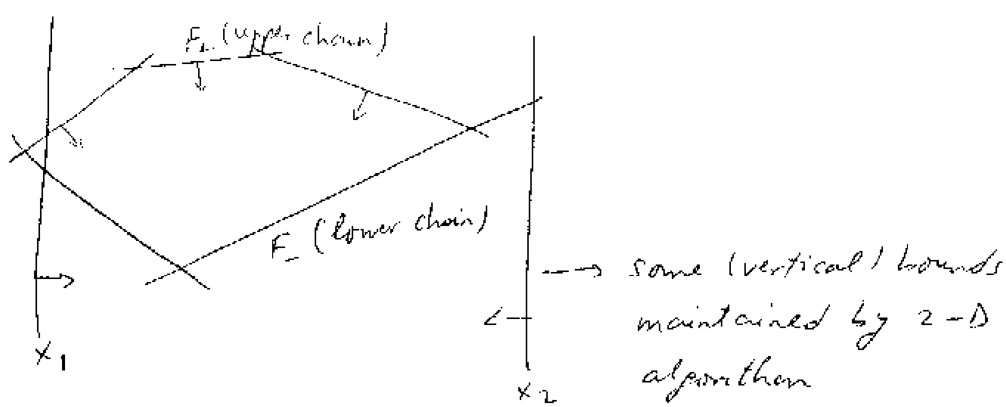


Linear Time LP $\rightarrow$ 2-D

- min: $ax + by$
  s.t.: $a_i x + b_i y \le c_i, \quad i = 1, 2, \dots, n$

- Change of variables:
  - set $\begin{cases} Y = ax + by \\ X = x \end{cases}$ $\Rightarrow$ min Y
    s.t. $d_i X + e_i Y \le f_i, \quad i = 1, 2, \dots, n$
    where: $\begin{cases} d_i = \left(a_i - \frac{a}{b} b_i\right) \\ e_i = \frac{b_i}{b} \end{cases}$

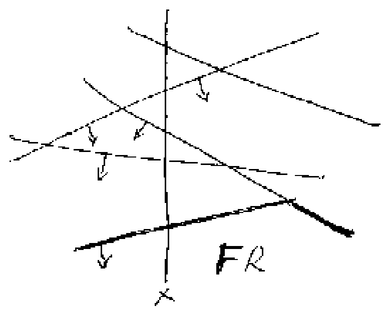
$\Rightarrow$ find feasible point with min y-coord.



$\Rightarrow$ want min feasible point on F_-

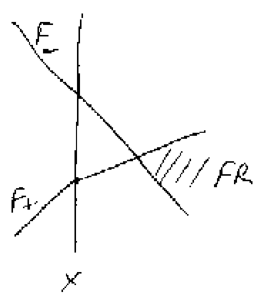
Observations:

- solution at some vertex of FR ⇒ discrete problem
  - consider only x values where 2 constraints intersect.



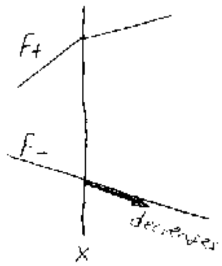
- For F_+ → $O(n)$ time to find min intersection point at x
- Same for F_- (max intersection)

- In $O(n)$ time can decide:
  1. x not feasible and all feasible points to right / left of x
     - Need slopes of F_+, F_- at x
       - at most 4 → identify constraints of F_+, F_- to left / right of x
       → slope: $-\frac{a_i}{b_i}$



(and the symmetric cases)

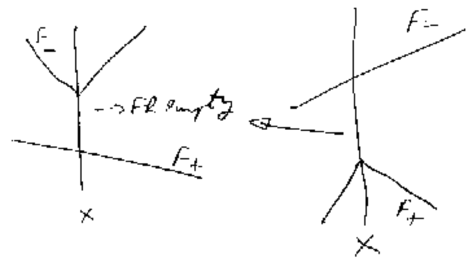
2. x is feasible and optimal x^* is to the right/left of x.



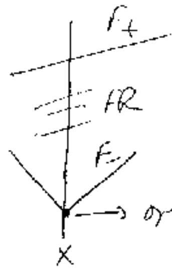
(and the symmetric cases)

3. No solution to LP

- F_+ bellow F_-



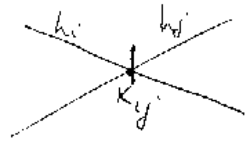
4. x is the min of F_- ⟹ Done!

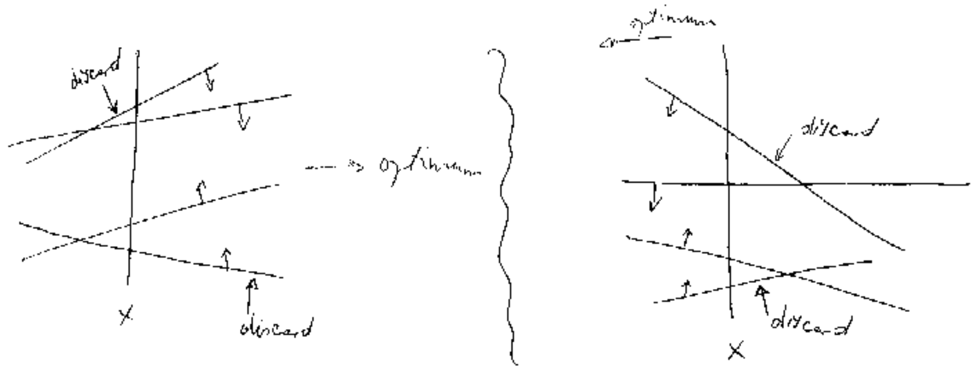


- At each step, make smart choice of x and decide which side optimal x^* is.
  - decision allows to discard (prune) a "large" number of constraints.

- $F_+ \longrightarrow S_1$
- $F_- \longrightarrow S_2$
- pair constraints in each set, S_1 and S_2 and try to eliminate one constraint in each pair by comparing x_{ij} with x

($x_{ij} \rightarrow$ abscisa of intersection point of pair h_i, h_j)





Analysis:
- linear time to get S_1, S_2
- linear time to pair constraints
- linear time to compute intersection at x and decide which side to look for x^*
- how to choose x ?!
  - take median of intersection points of pairs in S_1, S_2.
  - $\leq \lfloor \frac{N}{2} \rfloor$ pairs $\Rightarrow \lfloor \frac{N}{2} \rfloor$ points
  - find median $\rightarrow O(N)$ time
    x-coord.
  - locate side
  - for each intersection point on "wrong" side $\Rightarrow$ eliminate a constraint $\Rightarrow O(\frac{N}{4})$ eliminated.

$$T(n) = T\left(\frac{3n}{4}\right) + O(n)$$
$$\Downarrow$$
$$T(n) = O(n)$$