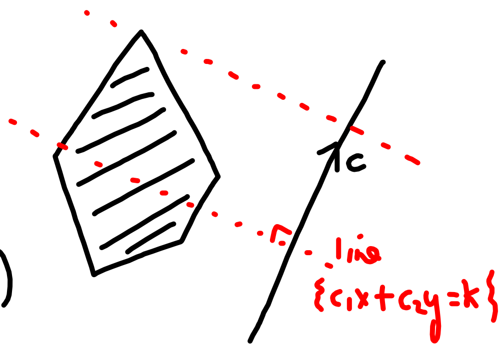
Lecture 7 – Linear programming.

- Consider $f: \mathbb{R}^2 \longrightarrow \mathbb{R} : (x,y) \longmapsto c_1x + c_2y$ where $(c_1, c_2) \neq (0,0)$, & set $H = \{h_1, \dots, h_n\}$ of half-planes.
- Goal: Find a point $(x,y) \in \bigcap_{i \in I} h_i = \bigcap H$ at which f attains maximal value.
- We will write
  $h_i : a_{i1}x + a_{i2}y \leqslant b_i$ for $i \in \{1, \dots, n\}$.

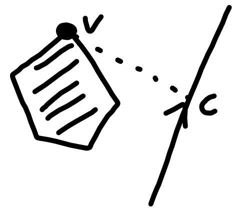
Geometrical interpretation

- f det. by a vector $\vec{c}=(c_1,c_2)$ non-zero.
- As we move in direction of $\vec{c}$, f increases:
ie. for $t>0$, $f((x,y)+t(c_1,c_2))$
$$= f(x,y)+tf(c_1,c_2)$$
$$= f(x,y)+t(c_1^2+c_2^2) > f(x,y).$$
- At lines $\{(x,y) : c_1x+c_2y=k\}$
  f has constant value. (ie. lines perp to $\vec{c}$)
- Hence f obtains maximal value at any point v in intersection which is extreme in direction of $\vec{c}$.

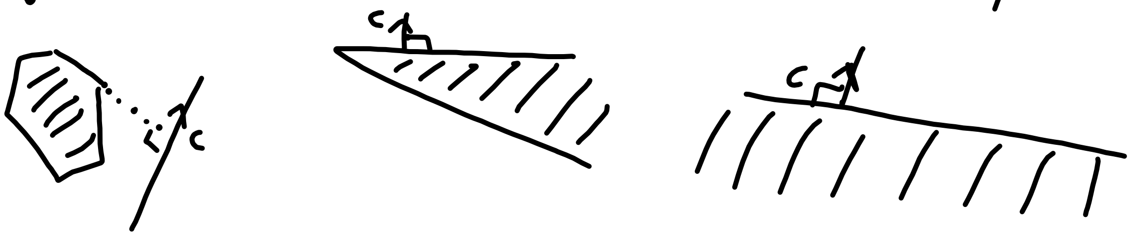


Different possibilities

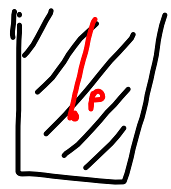
1) $\cap H$ is empty. No solution - problem is infeasible.
2) 1 point v at which f obtains maximal value

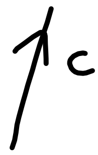


3) Infinitely many solutions - those form a segment, half-line or line:



4) F is unbounded on intersection:
there exists a half-line p in $\cap H$ along which f is increasing.





Input to algorithm: vector $\vec{c} \neq 0$ & $H = \{h_1, \ldots, h_n\}$ a set of half-planes.

Output: - If problem is infeasible, provide 3 half-planes with empty intersection.

- If f achieves a maximum, provide such a point in intersection.

- If f is unbounded above in intersection, provide a half-line in intersection along which f is increasing.

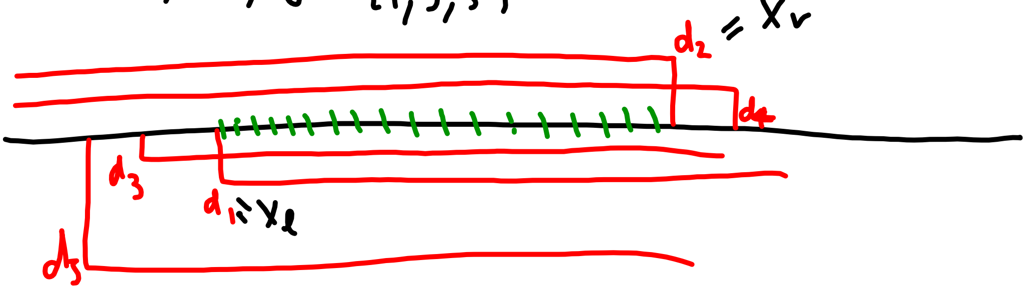
Firstly, solve 1-d case:

- $f\cancel{x} = cx$ for $c \neq 0$.
- half-planes $a_i x \leq b_i$ for $a_i \neq 0$ where $i = 1, \ldots, n$.

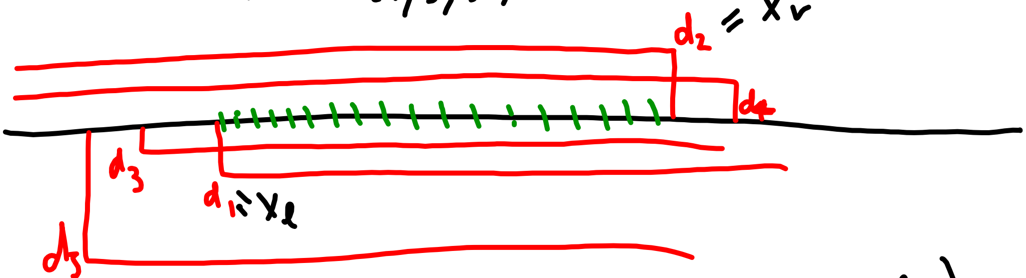
Goal: Find pt in int. at which f obtains $m\cancel{a}x$.

- let $I = \{ i : a_i > 0 \}$, $J = \{ j : a_j < 0 \}$.
- Inequalities become $x \leq b_i / a_i = d_i$ for $i \in I$.
  & $x \geq b_j / a_j = d_j$ for $j \in J$.
- let $x_l = \max\{-\infty, d_j : j \in J\}$
- $x_r = \min\{d_i, \infty : i \in I\}$.

$I = \{2, 4\}$, $J = \{1, 3, 5\}$



$= \{2,4\}$, $J = \{1,3,5\}$



Cases (1) $x_r < x_\ell$ (Empty intersection - problem infeasible)

(2) $x_\ell \leq x_r < \infty$ & $c > 0$ (f has max at x_r.)

(3) $-\infty < x_\ell \leq x_r$ & $c < 0$ (f has max at x_ℓ.)

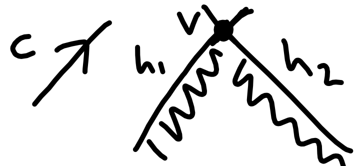
(4) $x_r = \infty$ (I is empty) & $c > 0$: $[x_\ell, \infty]$ is half-line along which f is increasing.

(5) $x_\ell = -\infty$ & $c < 0$ - Then $[-\infty, x_r]$ is half-line along which f is increasing.

Complexity $O(n)$ - Find max, min from sets of n points.

2-d bounded case

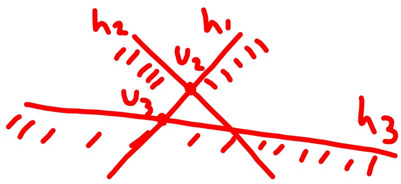
- In bounded case, we are given 2 half-planes h_1, h_2 such that f is bounded from above on $h_1 \cap h_2$.



boundary of h_1, h_2

- Here $h_1 \cap h_2$ has maximum at $l_1 \cap l_2 = v$.
- In this setting, if more than 1 sol. arises – choose least lexicographically.

- Alg. is incremental:
  given optimal point $v_{i-1} \in C_{i-1} = h_1 \cap \dots \cap h_{i-1}$
  we search for an optimal point $v_i \in h_i \cap C_{i-1} = C_i$.
- optimal point v_i: f achieves max at v_i in C_i & v_i is least w.r.t. lex. ordering.



- If $v_{i-1} \in C_i$ then $v_i = v_{i-1}$.
- Otherwise, C_i is empty or v_i lies on boundary ℓ_i of half-plane h_i.
- How to find v_i in this case?
- let $v_i = (x, y)$ - then $a_{i_1}x + a_{i_2}y = b_i$.
- Assuming $a_{i_2} \neq 0$ (otherwise $a_{i_1} \neq 0$), we have $y = \frac{b_i - a_{i_1}x}{a_{i_2}}$.

- We search for max. value of f on line $y = \frac{b_i - a_{i_1} x}{a_{i2}}$.
- On line, consider f as function of 1 variable

$$g(x) = c_1 x + c_2 \left(\frac{b_i - a_{i_1} x}{a_{i_2}} \right)$$
$$= \left(\frac{c_1 - c_2 a_{i_1}}{a_{i_2}} \right) x + c_2 \left(b_i / a_{i_2} \right)$$

Max val. of g not depend on constant, so must find max. value of

$$g^*(x) = \left(\frac{c_1 - c_2 a_{i_1}}{a_{i_2}} \right) x .$$

- Searching for max of f on $l_i \cap C_{i-1}$ ~ finding max of g^* at
  $$a_{j1}x + a_{j2}\left(\frac{b_i - a_{i1}x}{a_{i2}}\right) \leq b_j \text{ for } j = 1, \ldots, i-1.$$
- Rewrite as
  $$(*) \quad \left(a_{j1} - \frac{a_{j2}a_{i1}}{a_{i2}}\right)x \leq b_j - \frac{a_{j2}b_i}{a_{i2}}$$
- Now find v_i (or that C_i is empty) by solving 1-d linear program for g^* at cases $*$.

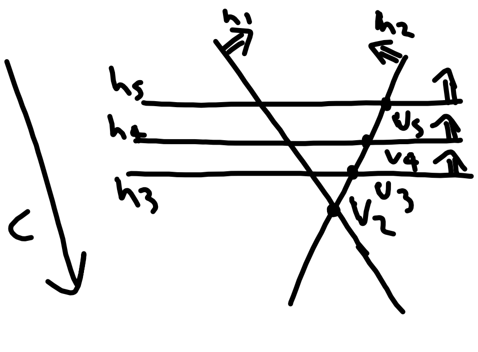
(See pseudocode in E-Learning - Lines 7-17, except line 9.)

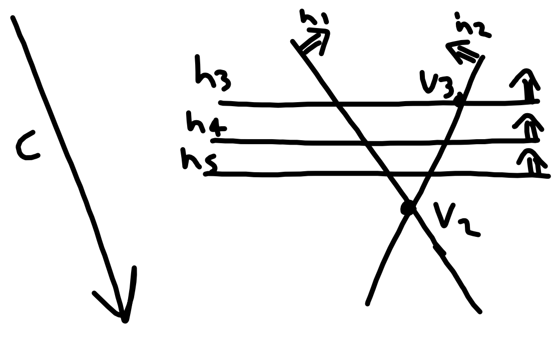
Running

Running time

- If $v_{i-1} \in h_i$, constant time to set $v_i = v_{i-1}$.
- Otherwise, time to calculate v_i is lin. in i - so $O(i)$.
- Complexity: $O(3) + O(4) + \ldots + O(n)$
$$= O(3 + 4 + \ldots + n)$$
$$= O(n^2)$$

Quite high running time & depends heavily on order of half-planes.





$V_3 =$
$U_4 =$
U_5.

- Introduce randomization into algorithm - consider a random ordering of the half-planes.
  (See L9 of code in E-learning.)
- Randomized expected time of algorithm is much lower: average time taking into account all possible orders.

Calculation of randomized expected time

- X_i random variable def. by $X_i = \begin{cases} 1 & \text{if } v_{i-1} \notin h_i \\ 0 & \text{if } \quad\quad \in h_i \end{cases}$
- Rand. expect time
$$E(X) = \sum_{i=3}^{n} O(i) E(X_i)$$
where $E(X_i) = \text{prob}(X_i = 1) = \text{prob}(v_{i-1} \notin h_i)$.
- Will show $p(v_{i-1} \notin h_i) = 2/i$.
- Therefore $E(X) = \sum_3^n O(i) \cdot 2/i = \sum_3^n O(1)$
$$= O(n).$$

Expected time is linear.

lis 11-19:44

- Show $p(v_{i-1} \notin h_i) = 2/i$.
- Now $v_i = l_j \cap l_k$ for $j,k \leq i$ & j,k are min. with these props.
- Then $p(v_{i-1} \notin h_i) = p(v_i \neq v_{i-1})$
  $= p(i=j \text{ or } i=k)$
- There are $i(i-1)$ choices of pairs $j,k \leq i$.
  - ...... $i-1$ choices in which $j=i$.
  - ...... $i-1$ .... $k=i$.
- So $2(i-1)$ choices in which j or k equals i.
  So $p(i=j \text{ or } i=k) = \frac{2(i-1)}{i(i-1)} = 2/i$.

$\Rightarrow$ Exp. times is $O(n)$.

- Unbouned cases covered in E-learning too.

lis 11-19:56