Problem:

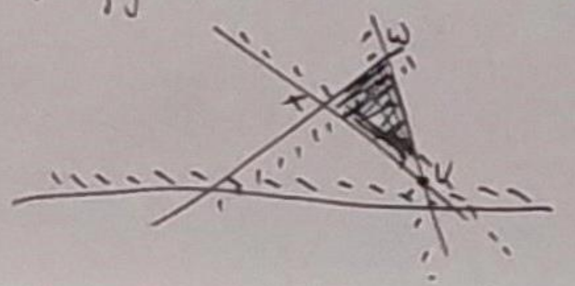
Given a set of n half-planes:

$$H = \{h_1, h_2, \dots, h_n\}$$

note: h_i is defined by line ℓ_i

Want: Find (possibly empty) convex region defined by $\cap H := \bigcap_{i=1}^{n} h_i$

note: Since all boundaries are linear, this is a polygon. So, we want list of verts in ccw



$\cap H = [v, w, x]$

Divide + Conquer Algo

HalfPlaneInt(H)

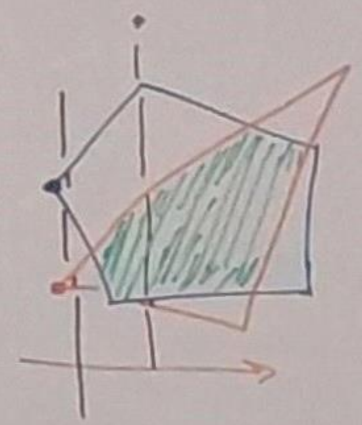
$\Theta(1)$ 1: if $|H| \leq 1$, return H
$\Theta(1)^*$ 2: Partition H into $H_1 \sqcup H_2$, equal sizes ± 1
$T(n/2)$ 3: $P_1 \leftarrow$ HalfPlaneInt(H_1)
$T(n/2)$ 4: $P_2 \leftarrow$ HalfPlaneInt(H_2)
$\Theta(n)$ 5: return $P_1 \cap P_2$ ⟵ a plane sweep!

$T(n)$ = worst case RT when $|H| = n$

$$= 2T(n/2) + \Theta(n)$$

$$= \Theta(n \log n)$$

Note: worst case, P_i is $\Theta(n)$ since all $n/2$ lines can be used

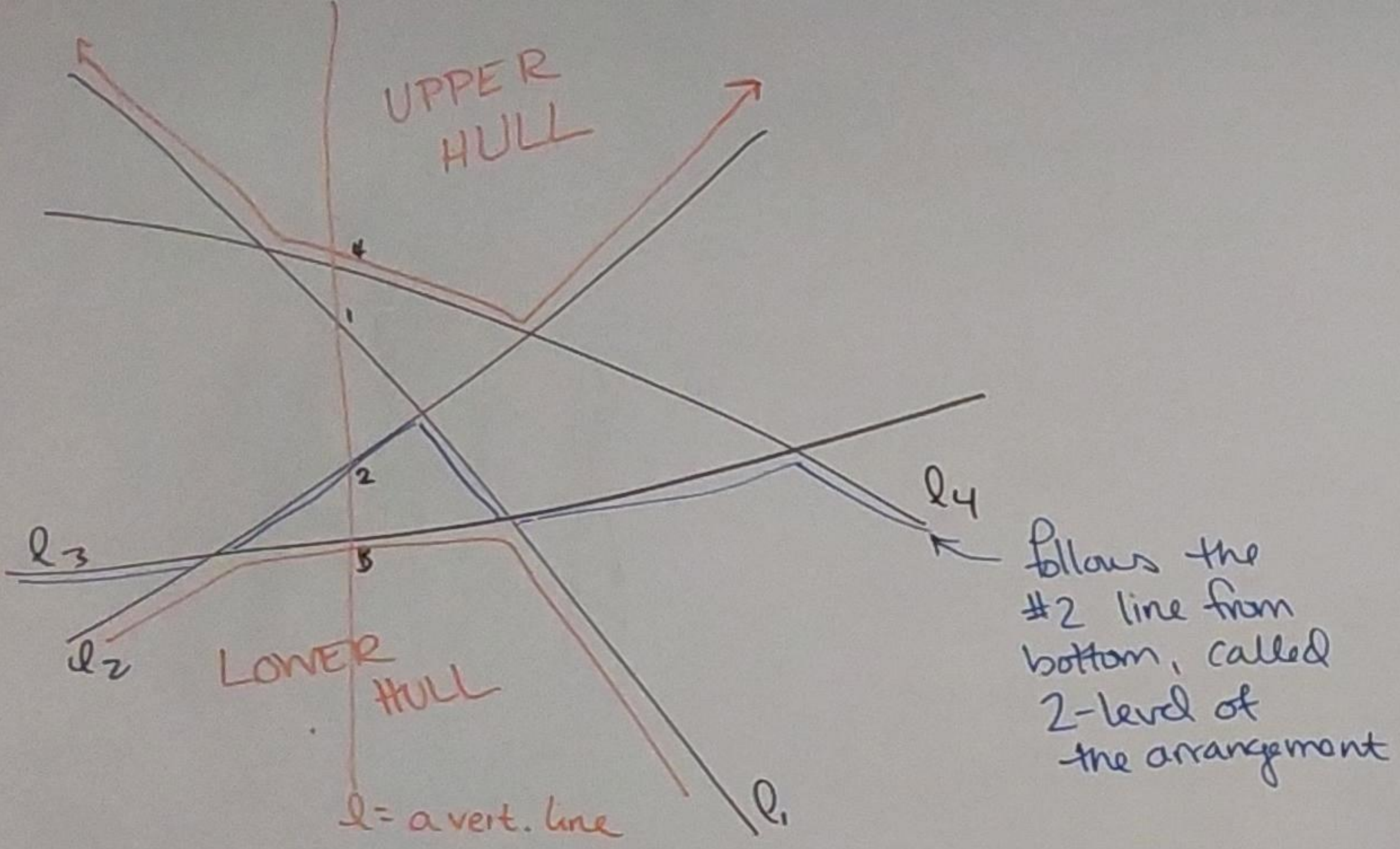


Def: A **Arrangement** of lines is a set of lines (in $\mathbb{R}^2$), along with the induced spatial decomposition

→ to compute an arrangement is to compute the DCEL (or other d.s. if we prefer something else)

→ note: can also talk about arrangements of other objects (hyperplanes, circles, algebraic curves, etc.) in higher-d.

→ pseudolines: set of curves that pairwise intersect at most once.



L = a set of non-vertical lines
(oriented R→L, so + side is above)

Linear Programming,

an ex of constrained optimization

↳ min/max optimization fcn subject to constraints

linear fcn

half spaces def. by lines/hyperplanes (linear constraints)

Constraints in $\mathbb{R}^d$

$$\left.\begin{array}{l} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,d}x_d \le b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,d}x_d \le b_2 \\ \vdots \qquad\qquad \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \dots + a_{n,d}x_d \le b_n \end{array}\right\}$$

a feasible sol'n is a $x \in \mathbb{R}^d$ that satisfies all of these constraints!

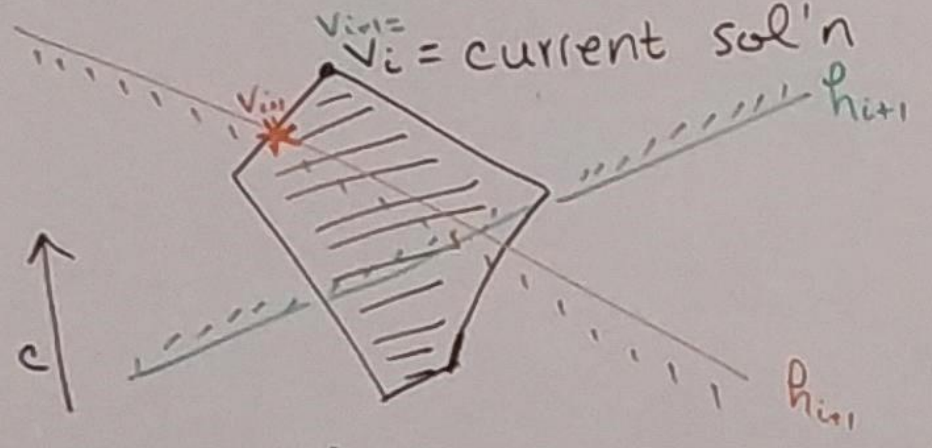
Geom: we have a polygon/polytope, possibly empty (infeasible constraints) & possibly unbounded

$\boxed{Ax \le b}$ matrix form

The idea of Incremental algos:

- Start w/ some sol'n
- add a constraint } *
- update
- repeat until all constraints are added.

Suppose we have the Solution to $H_i := \{h_1, h_2, \dots, h_i\}$ (optimizing in dir c.)



Case 1: $v_i \in h_{i+1}$

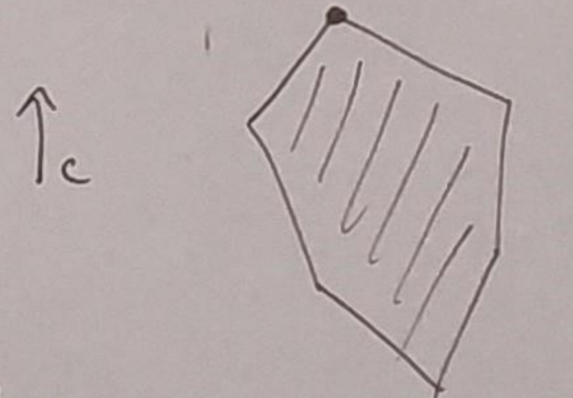
Still a valid sol'n. Try adding the next constraint.
(If keeping track of feasible space, need to update that.)

Case 2: $v_i \notin h_{i+1}$

Optimization fn

max $c_1 x_1 + c_2 x_2 + \dots + c_d$

max $c^T x$



find extreme pt in dir c.

Observe:

At any pt (assuming no h_i is vert & feasible space bounded), v_i (the current sol'n) is a vertex defining the feasible space $\Rightarrow$ a vertex of the arrangement $\{l_i\}$

Lemma: If we're in case 2 and $\cap H_{i+1} \neq \emptyset$, then the solution $v_{i+1} \in l_{i+1}$

(true in higher-d too!)

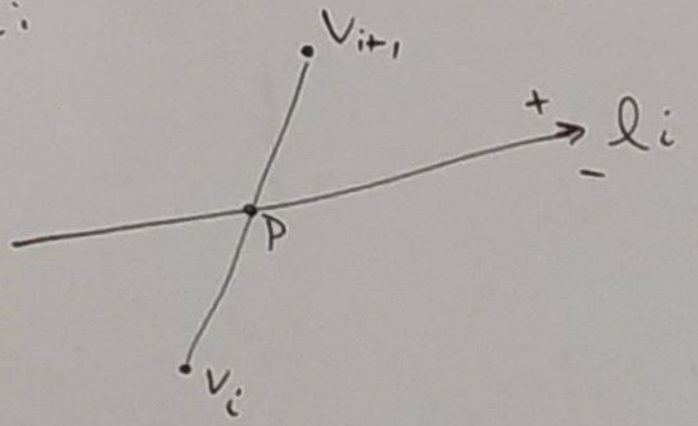
Proof: (By contradiction)

Assume $v_{i+1} \notin l_{i+1}$.

But, since feasible, $v_{i+1} \in h_{i+1}$.

By assumption of being in case 2, $v_i \notin h_{i+1}$.

Pic:



The line segment $\overline{v_i v_{i+1}}$ intersects l_i.

Let p be that intersection.

Since v_{i+1} was feasible for constraints H_i, but v_i was sol'n

$\rightarrow$ we know $c^T v_i > c^T v_{i+1}$

But since optimization was linear,

$$c^T v_i > c^T p > c^T v_{i+1},$$

which means p is feasible and bigger optimization fcn ※