Randomized Algorithms: 2-dim Linear Programming

Deterministic algorithms: same behaviour on a fixed input

Randomized algorithms: random decisions

→ running time → random variable

→ output can be random

- With high probability, running time is small
- w.h.p., Output is correct.

Randomized Quicksort (randomly choose pivot)

Sampling, Avoid worst cases

- with some prob, hardware can fail.
- Algorithms: repeat multiple times
  $\Rightarrow$ error probability very small

Many cases: randomized algorithms are the fastest known.
Also, simplest.

2-dim Linear Programming

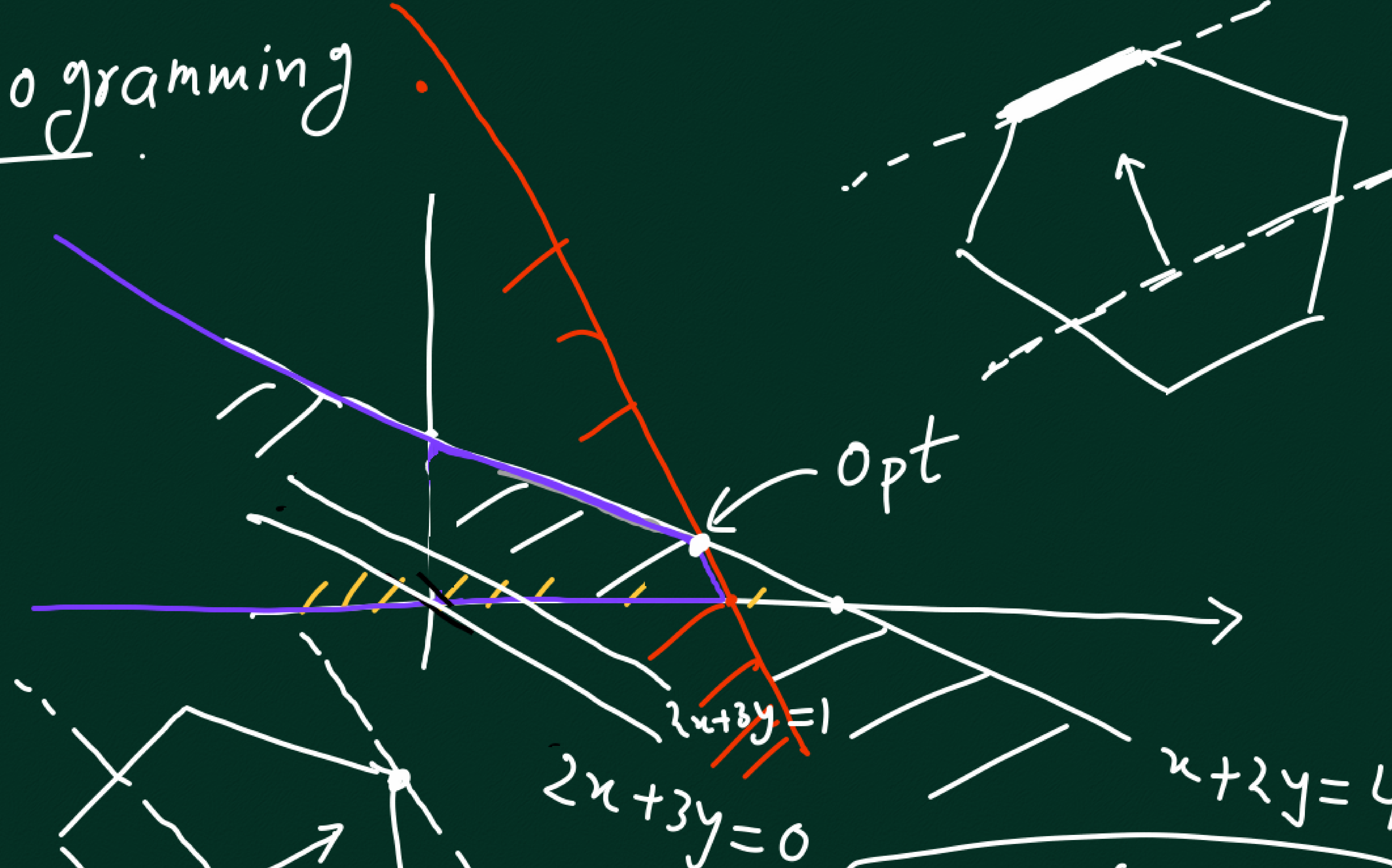
x, y

$$x + 2y \le 4$$

$$2x + y \le 6$$

$$y \ge 0$$

$$\max\ 2x + 3y$$



$$\left(\tfrac{8}{3}, \tfrac{2}{3}\right) \qquad \text{Opt } \frac{22}{3}$$

One of corners is an optimal point.

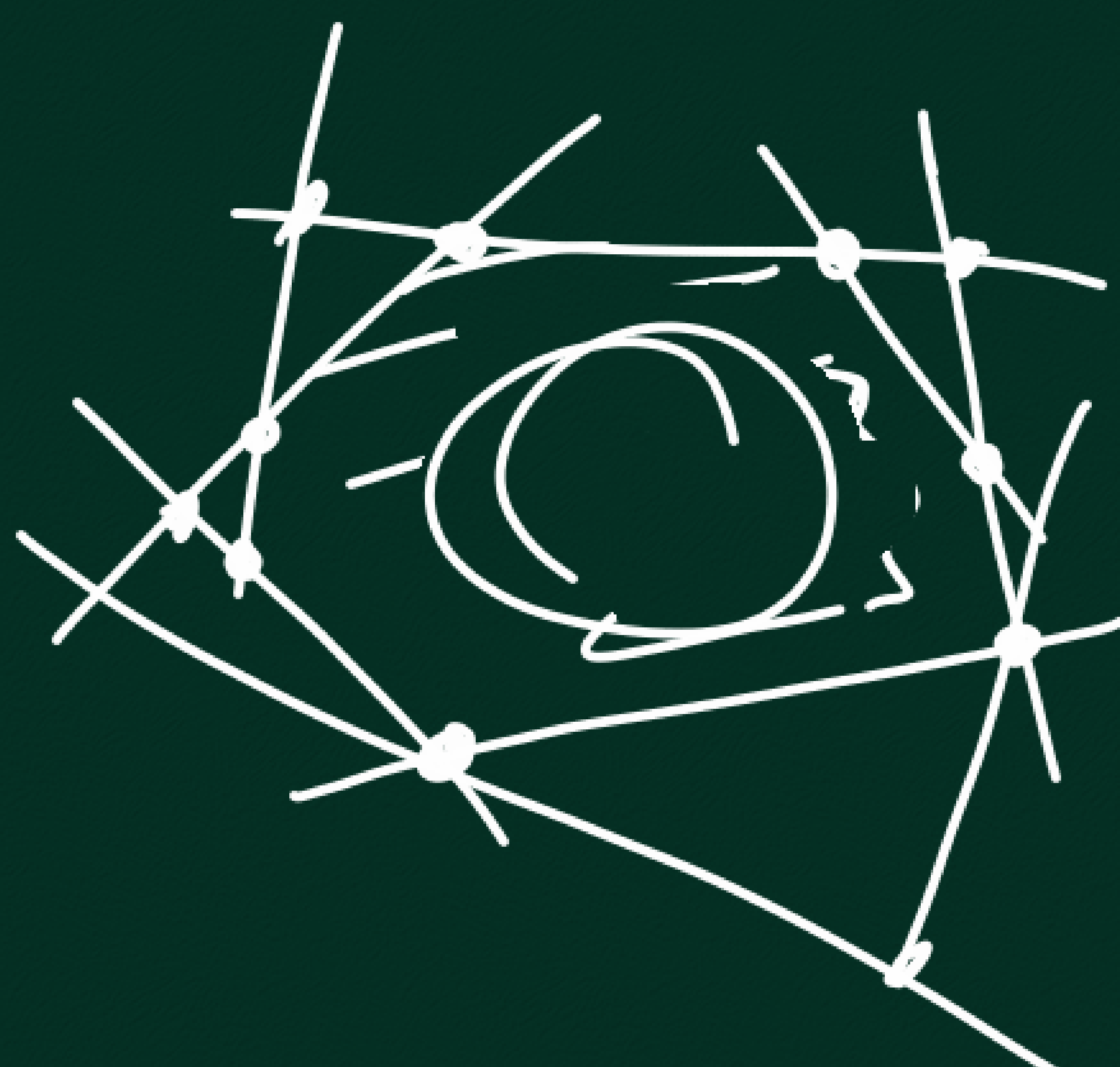
HW

n linear constraints.

$O(n^3)$

check all the corners

at most n



no. of intersection points

$O(n^2)$

HW $O(n \log n)$ compute all corners.

Randomization simple algorithm

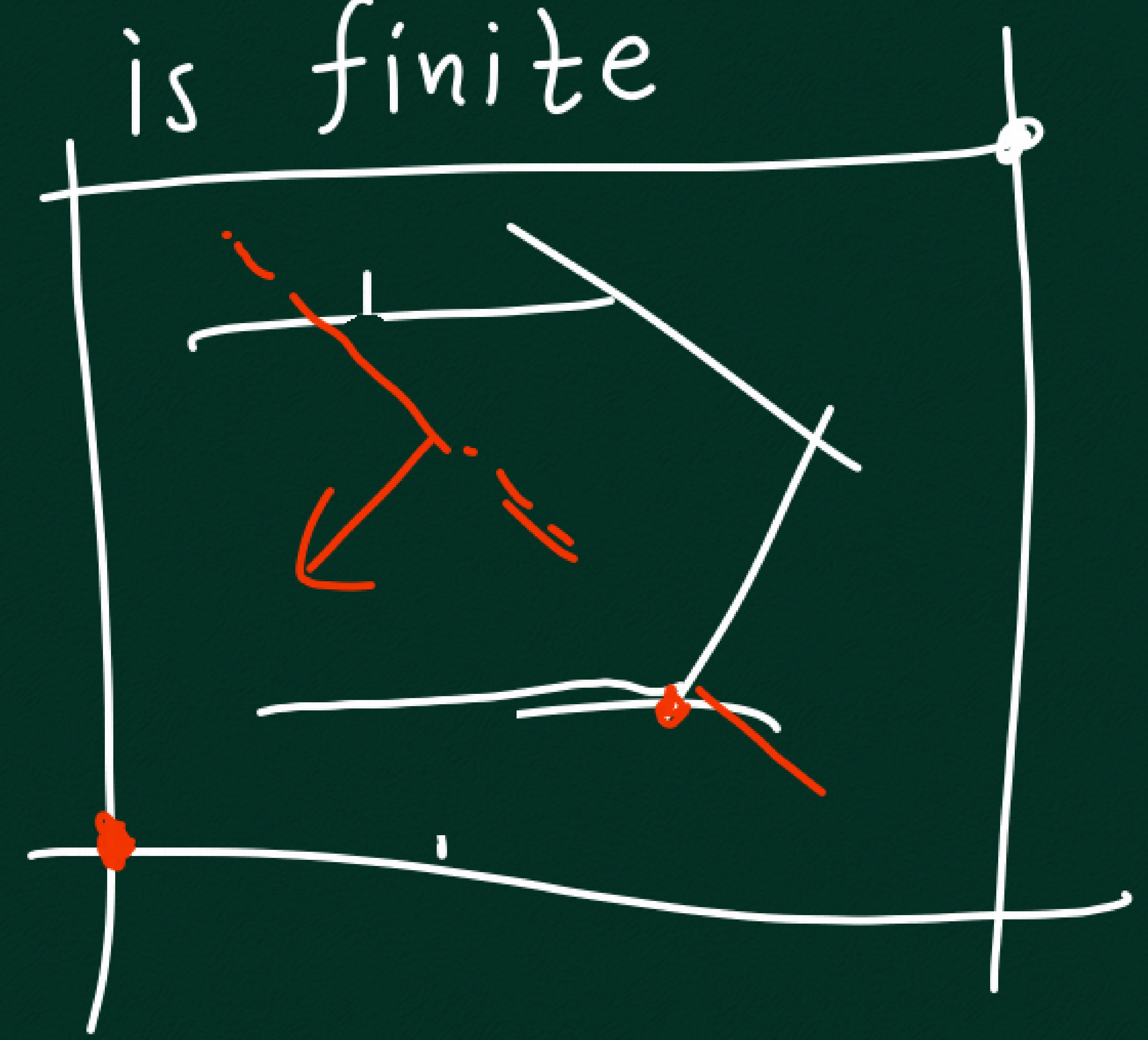
Find the optimal point $O(n)$ time (w.h.p.)

Assumptions: Optimal value is finite

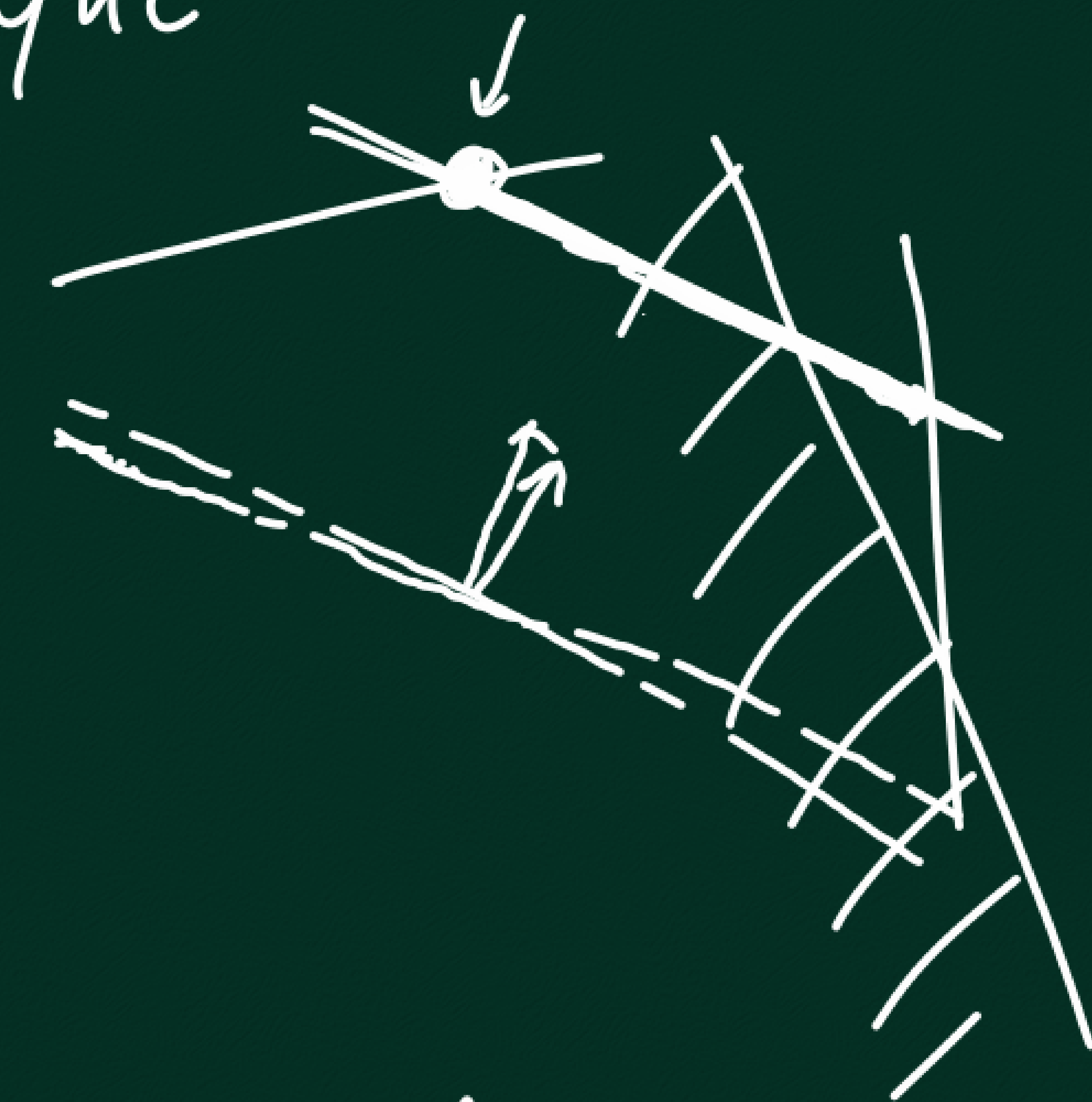
for every intersection point

$-L < x\text{-coord} < L$

$-L < y\text{-coord} < L$



Assumption: optimal solution is unique.



$C_0 \leftarrow$ Set of constraints $\leftarrow$ $-L \le x \le L$

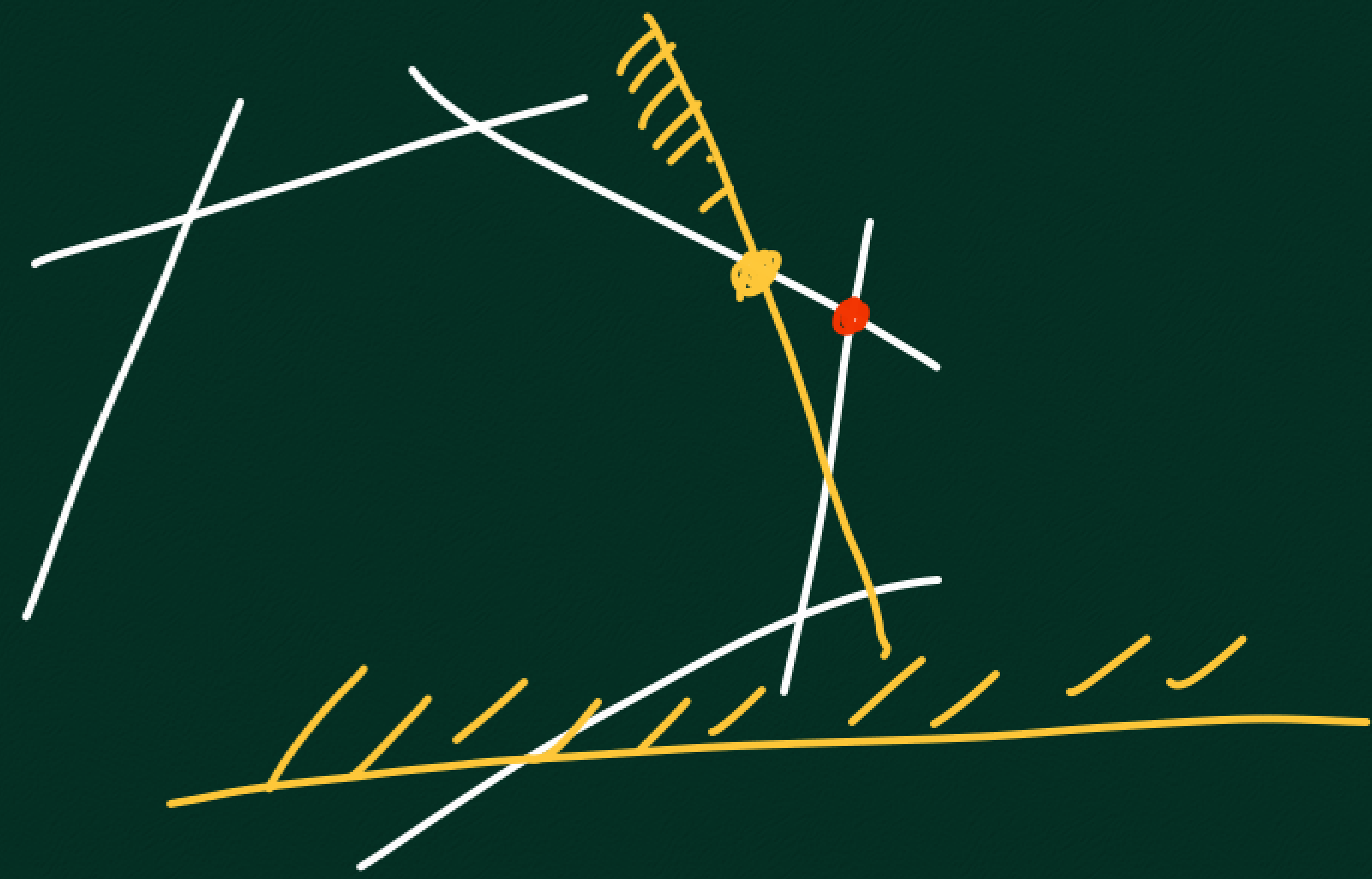
$-L \le y \le L$

$v_0 \leftarrow$ Optimal point $\leftarrow (L, L)$

Add constraints one by one and update the optimal point

$$(C_{i-1}, v_{i-1}) \longrightarrow (C_i, v_i)$$

$$C_i = C_{i-1} \cup \{h_i\}$$



- if v_{i-1} satisfies h_i
  then $v_i = v_{i-1}$