

Announcements

Assignments:

- P2: Optimization
 - Due Thurs 10/5, 10pm
- HW4 (online)
 - Covers LP, IP
 - Due Tues 9/26, 10 pm

EXAM 1: 9/28!!

Plan

Last Time

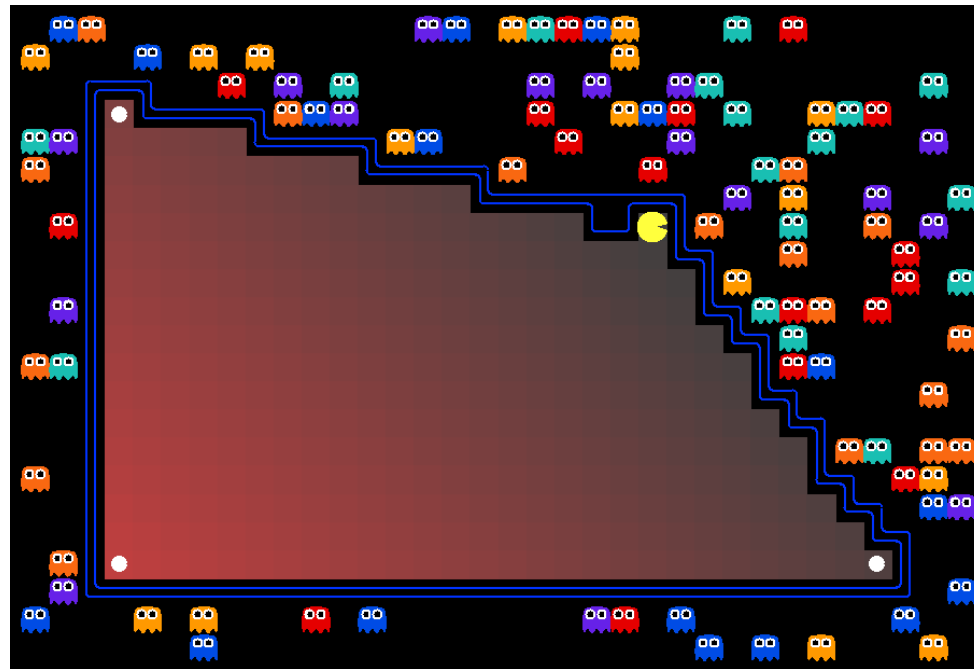
- Linear programming formulation
 - Problem description
 - Graphical representation
 - Optimization representation

Today

- Solving linear programs
- Higher dimensions than just 2
- Integer programs

AI: Representation and Problem Solving

Integer Programming



Instructors: Vincent Conitzer and Aditi Raghunathan

Slide credits: CMU AI with drawings from <http://ai.berkeley.edu>

Solving a Linear Program

Inequality form, with no constraints

$$\min_{\mathbf{x}} \quad \mathbf{c}^T \mathbf{x}$$

Solving a Linear Program

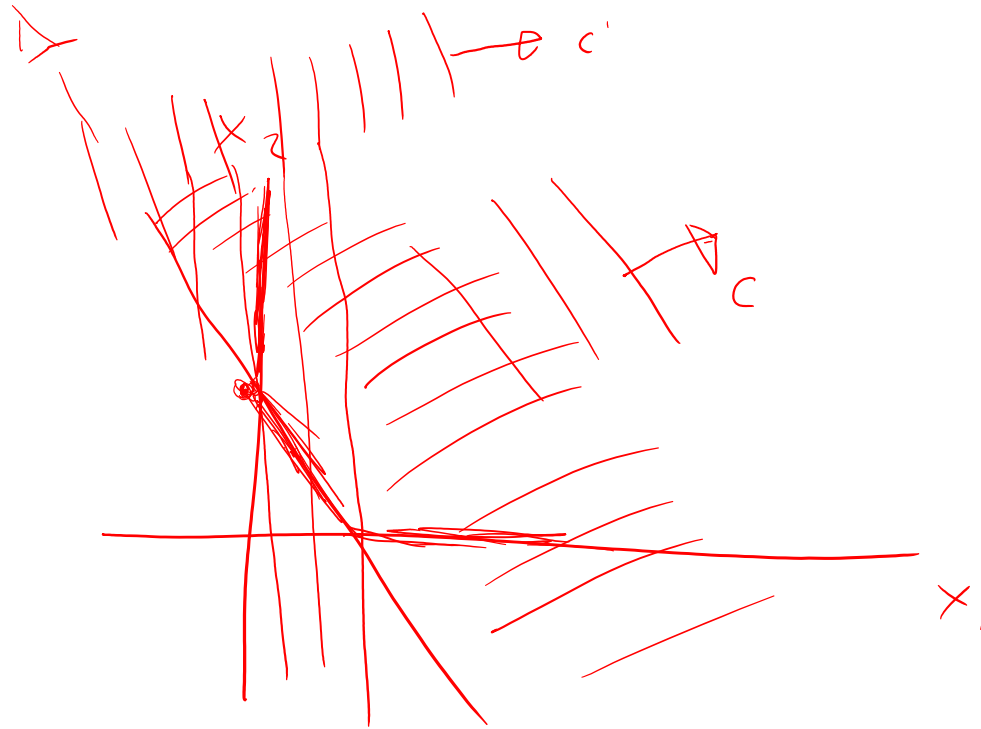
Inequality form, with one constraint

$$\begin{array}{ll} \min. & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & a_1 x_1 + a_2 x_2 \leq b \end{array}$$

Poll 1

True or False: A minimizing LP with exactly one constraint, will always have a minimum objective value of $-\infty$.

$$\begin{array}{ll} \min. & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & a_1 x_1 + a_2 x_2 \leq b \end{array}$$



Question

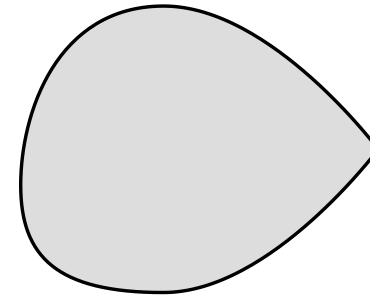
True or False: A minimizing LP with exactly two constraints, will always have a minimum objective value $> -\infty$.

$$\begin{array}{ll} \min. & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & a_{11}x_1 + a_{12}x_2 \leq b_1 \\ & a_{21}x_1 + a_{22}x_2 \leq b_2 \end{array}$$

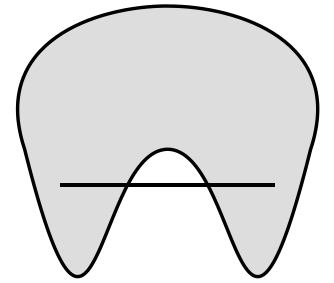
$$\begin{array}{l} x_1 \leq 0 \\ x_1 \geq 1 \\ -x_1 \leq -1 \end{array}$$

Convexity

Convex sets are those in which you can draw a line between two points and all the points between them are also in the set

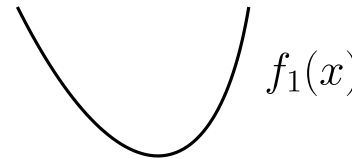


Convex set

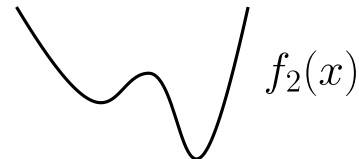


Nonconvex set

Convex optimization problems are ones in which the local minimum is also the global minimum

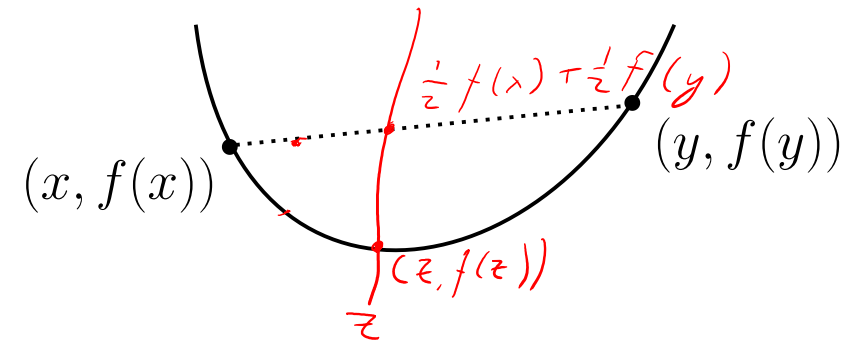


Convex function



Nonconvex function

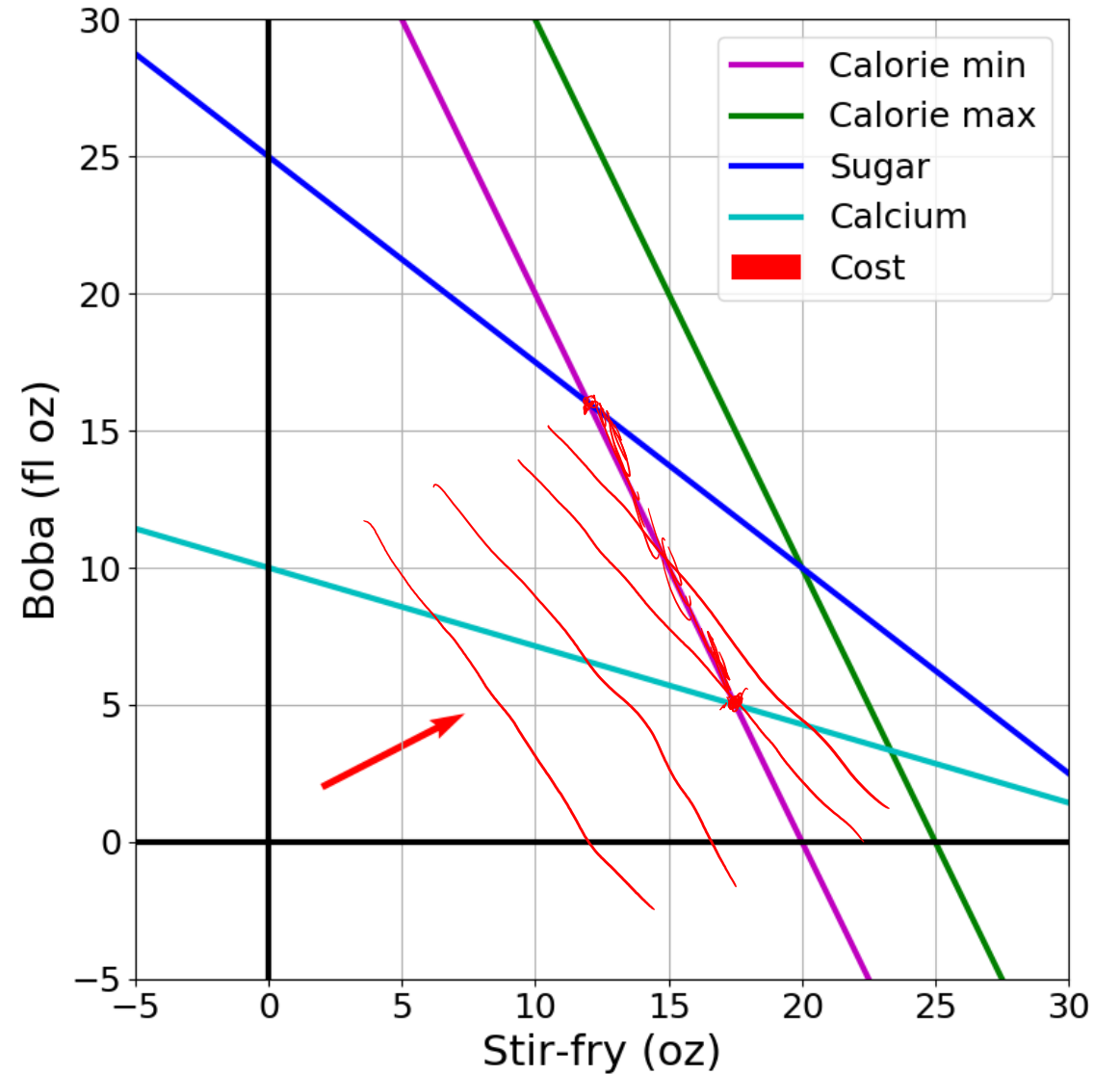
Convex functions have the property that for any point between two points x and y in a convex set:
 $f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$



Linear functions (like our costs) are convex!

LP Solutions

Solutions are at feasible intersections of constraint boundaries!!



Solving an LP

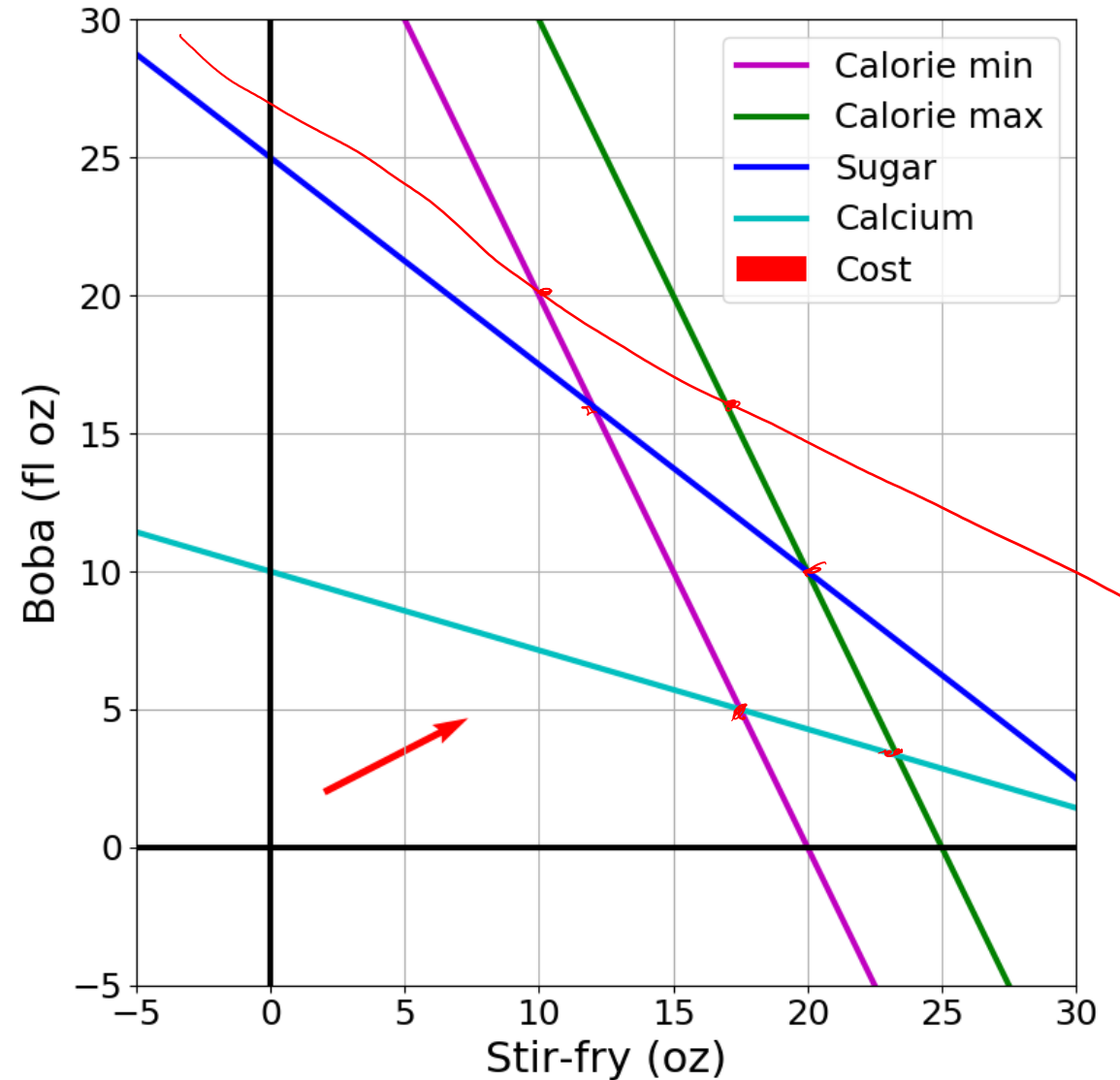
Solutions are at feasible intersections of constraint boundaries!!

Algorithm

- Check objective at all feasible intersections

In more detail:

1. Enumerate all intersections
2. Keep only those that are feasible (satisfy *all* inequalities)
3. Return feasible intersection with the lowest objective value



Solving an LP

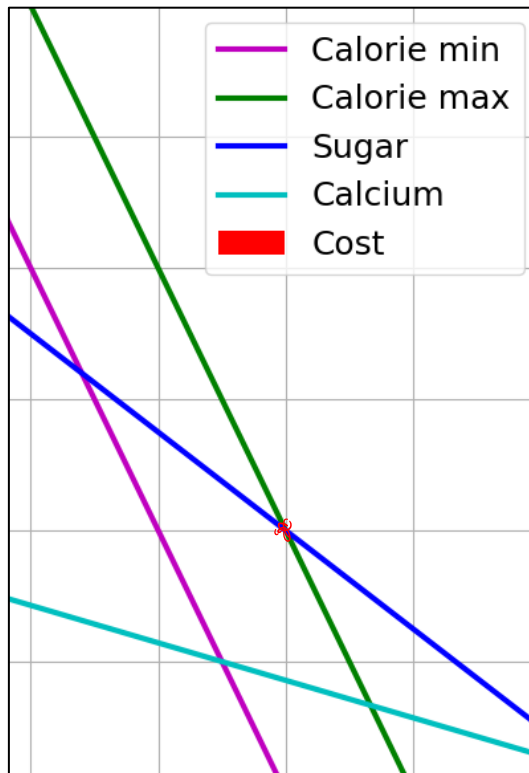
But, how do we find the intersection between boundaries?

$$\begin{array}{ll} \min & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & \mathbf{Ax} \leq \mathbf{b} \end{array}$$

$$\mathbf{A} = \begin{bmatrix} -100 & -50 \\ 100 & 50 \\ 3 & 4 \\ -20 & -70 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} -2000 \\ 2500 \\ 100 \\ -700 \end{bmatrix}$$

Calorie min
Calorie max
Sugar
Calcium



$$\begin{aligned} 100x_1 + 50x_2 &= 2500 \\ 3x_1 + 4x_2 &= 100 \end{aligned}$$

~~$\mathbf{Ax} \leq \mathbf{b}$~~

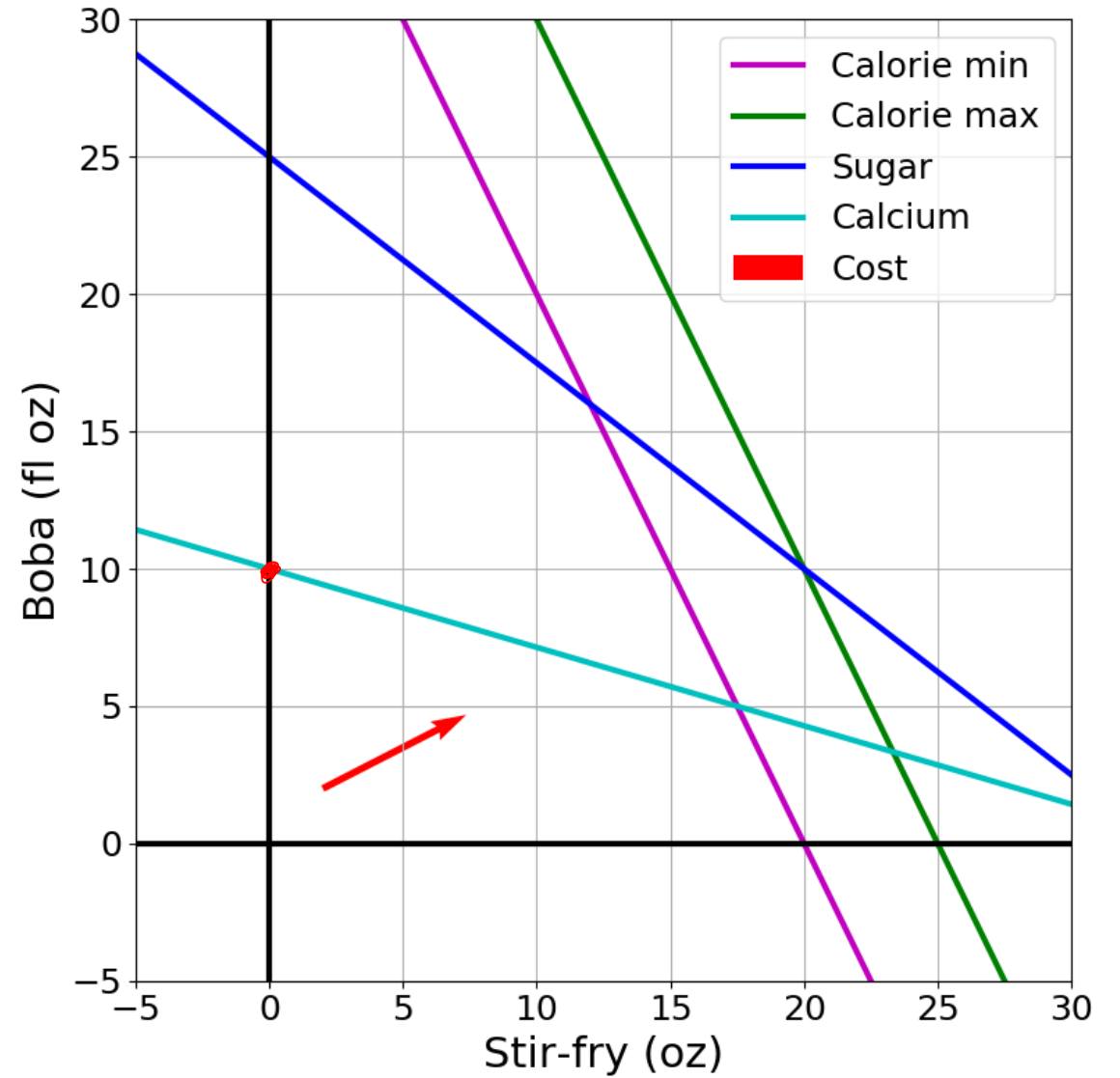
$$\begin{aligned} \mathbf{A}_{2,3} \mathbf{x} &= \mathbf{b}_{2,3} \\ \mathbf{x} &= \mathbf{A}_{2,3}^{-1} \mathbf{b}_{2,3} \end{aligned}$$

Cautions

Suppose we **drop** calorie min constraint

What is optimal?

$$x_i \geq 0$$

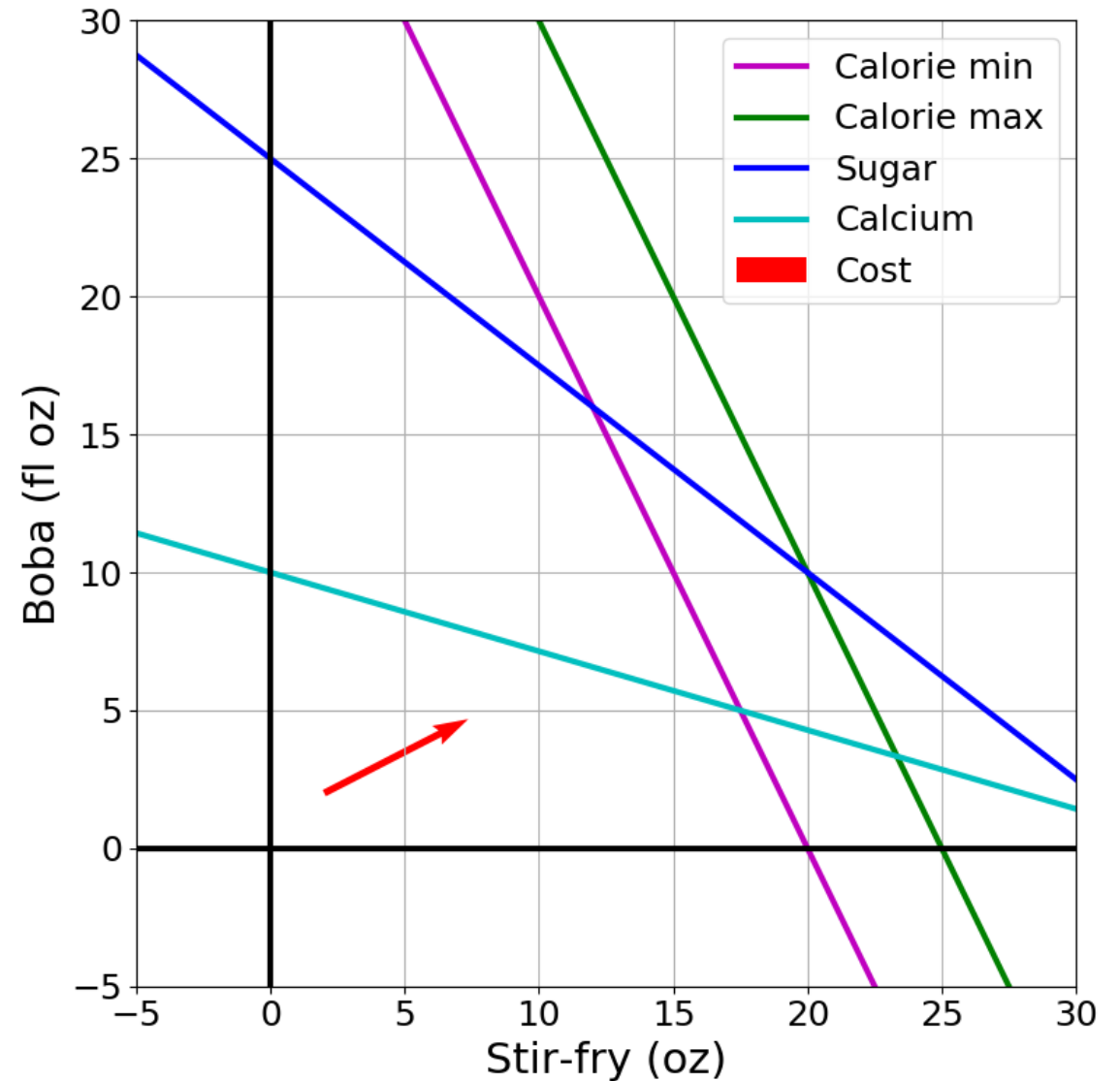


Solving an LP

Solutions are at feasible intersections of constraint boundaries!!

Algorithms

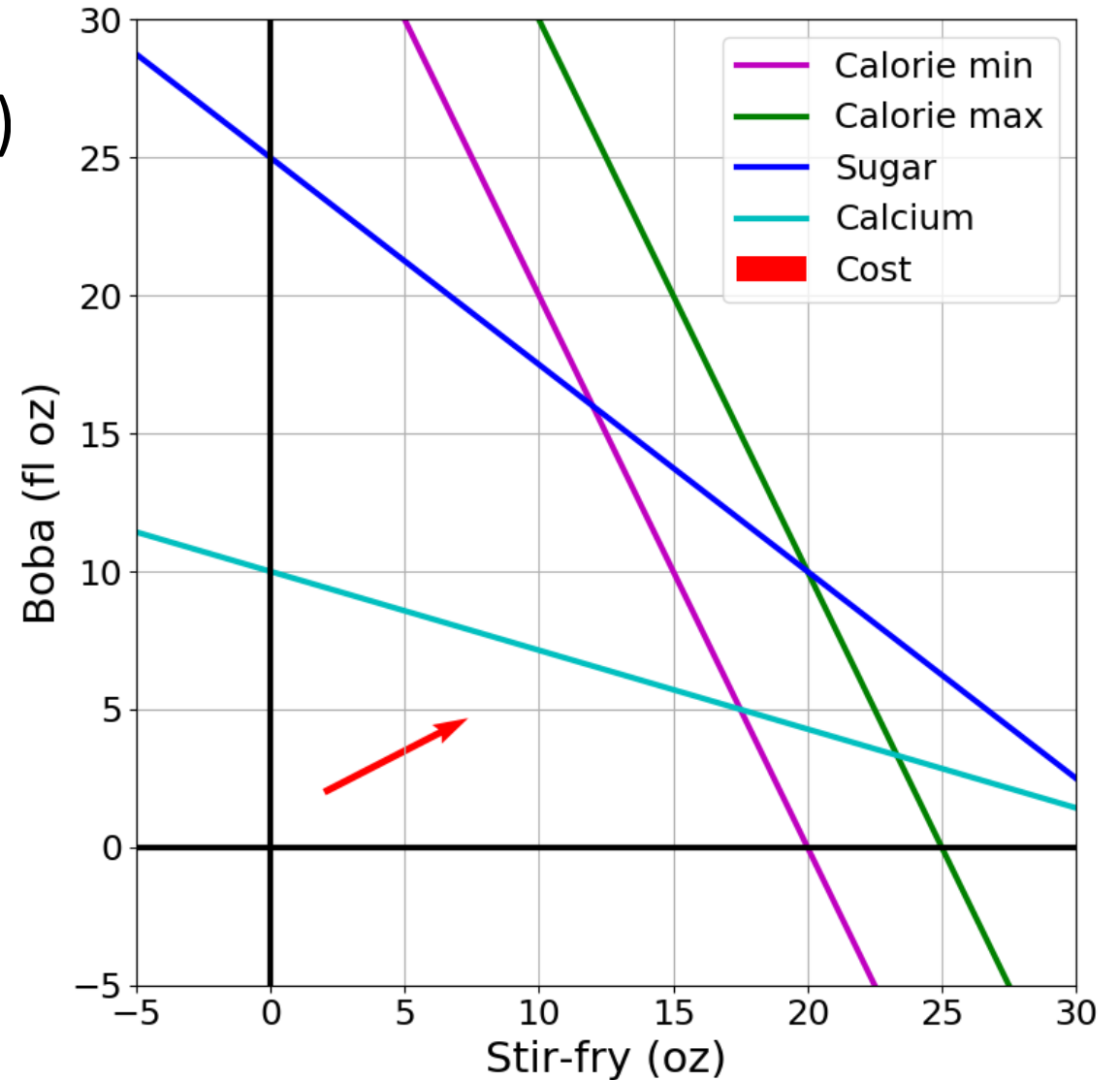
- Check objective at all feasible intersections
 - How many might there be?
- Simplex



Solving an LP

Simplex algorithm intuition (details missing)

- Start at a feasible intersection (if not trivial, can solve another LP to find one)
- Define successors as “neighbors” of current intersection
 - i.e., remove one row from our square subset of A, and add another row not in the subset; then check feasibility
- Move to any successor with lower objective than current intersection
 - If no such successors, we are done



Greedy local hill-climbing search! ... but always finds *optimal* solution (if defined right)

Solving an LP

Solutions are at feasible intersections
of constraint boundaries!!

Algorithms

- Check objective at all feasible intersections
- Simplex
- Interior Point

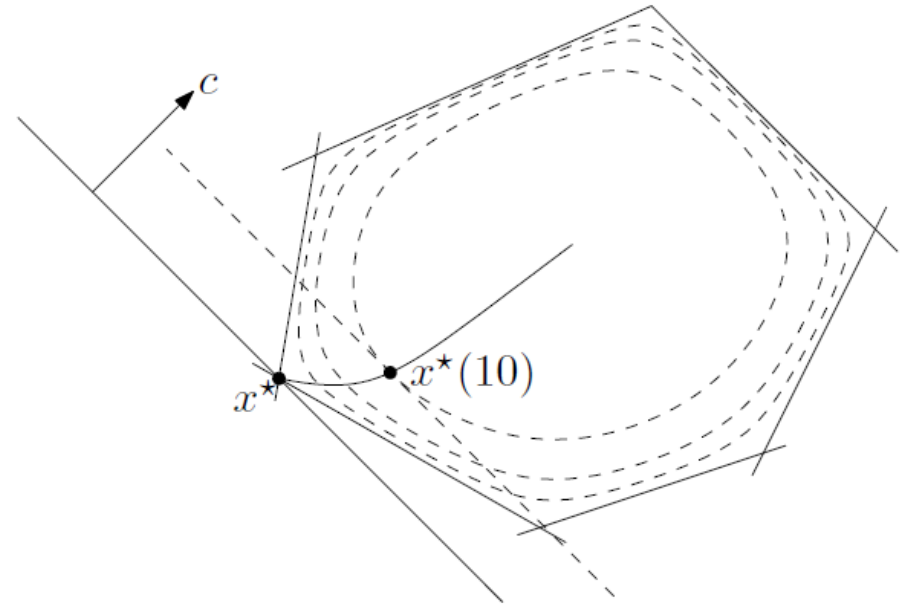


Figure 11.2 from Boyd and Vandenberghe, *Convex Optimization*

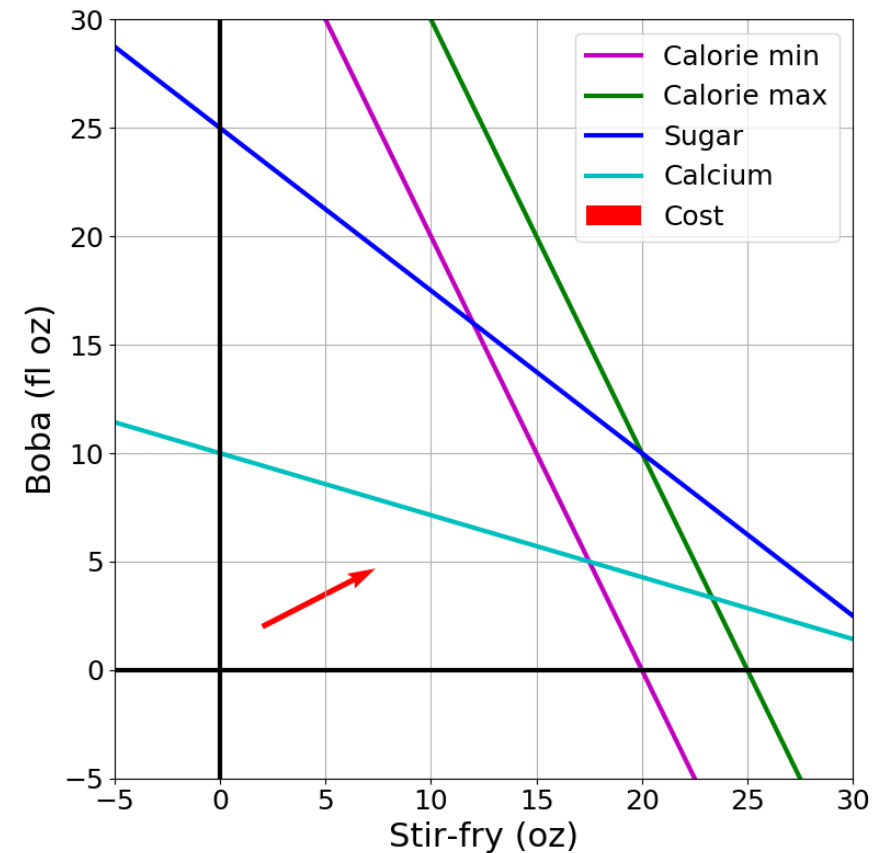
What about higher dimensions?

Problem Description

Optimization Representation

$$\begin{array}{ll} \min_{\mathbf{x}} & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \end{array}$$

Graphical Representation



Shapes in higher dimensions

How do these linear shapes extend to 3-D, N-D?

$$a_1 x_1 + a_2 x_2 = b_1$$

$$a_1 x_1 + a_2 x_2 \leq b_1$$

$$a_{1,1} x_1 + a_{1,2} x_2 \leq b_1$$

$$a_{2,1} x_1 + a_{2,2} x_2 \leq b_2$$

$$a_{3,1} x_1 + a_{3,2} x_2 \leq b_3$$

$$a_{4,1} x_1 + a_{4,2} x_2 \leq b_4$$

What are intersections in higher dimensions?

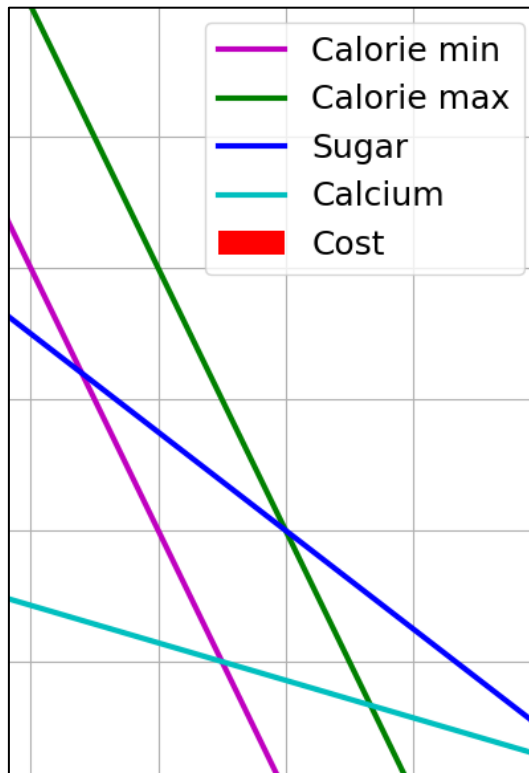
How do these linear shapes extend to 3-D, N-D?

$$\begin{array}{ll} \min & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & \mathbf{Ax} \leq \mathbf{b} \end{array}$$

$$\mathbf{A} = \begin{bmatrix} -100 & -50 & 0 \\ 100 & 50 & 0 \\ 3 & 4 & 0 \\ -20 & -70 & 0 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} -2000 \\ 2500 \\ 100 \\ -700 \end{bmatrix}$$

Calorie min
Calorie max
Sugar
Calcium



How do we find intersections in higher dimensions?

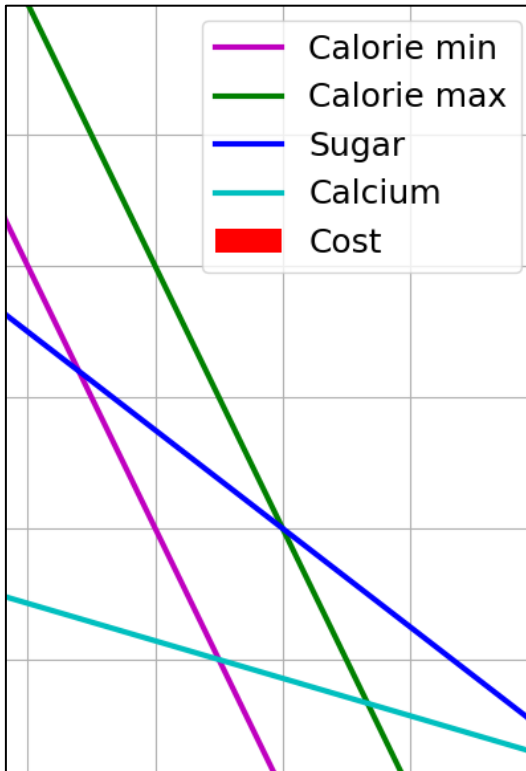
Still looking at subsets of A matrix

$$\begin{array}{ll} \min & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & \mathbf{Ax} \leq \mathbf{b} \end{array}$$

$$\mathbf{A} = \begin{bmatrix} -100 & -50 \\ 100 & 50 \\ 3 & 4 \\ -20 & -70 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} -2000 \\ 2500 \\ 100 \\ -700 \end{bmatrix}$$

Calorie min
Calorie max
Sugar
Calcium



Linear Programming

We are trying to stay healthy by finding the optimal food to purchase.

We can choose the amount of **stir-fry** (ounce) and **boba** (fluid ounces).

Healthy Squad Goals

- $2000 \leq \text{Calories} \leq 2500$
- $\text{Sugar} \leq 100 \text{ g}$
- $\text{Calcium} \geq 700 \text{ mg}$

Food	Cost	Calories	Sugar	Calcium
Stir-fry (per oz)	1	100	3	20
Boba (per fl oz)	0.5	50	4	70

What is the cheapest way to stay “healthy” with this menu?

How much **stir-fry** (ounce) and **boba** (fluid ounces) should we buy?

Linear Programming → Integer Programming

We are trying healthy by finding the optimal amount of food to purchase.

We can choose the amount of **stir-fry** (**bowls**) and **boba** (**glasses**).

Healthy Squad Goals

- $2000 \leq \text{Calories} \leq 2500$
- $\text{Sugar} \leq 100 \text{ g}$
- $\text{Calcium} \geq 700 \text{ mg}$

Food	Cost	Calories	Sugar	Calcium
Stir-fry (per bowl)	1	100	3	20
Boba (per glass)	0.5	50	4	70

What is the cheapest way to stay “healthy” with this menu?

How much **stir-fry** (ounce) and **boba** (fluid ounces) should we buy?

Linear Programming vs Integer Programming

Linear objective with linear constraints, but now with additional constraint that all values in \mathbf{x} must be integers

$$\begin{array}{ll} \min. & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \end{array}$$

$$\begin{array}{ll} \min. & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^N \end{array}$$

We could also do:

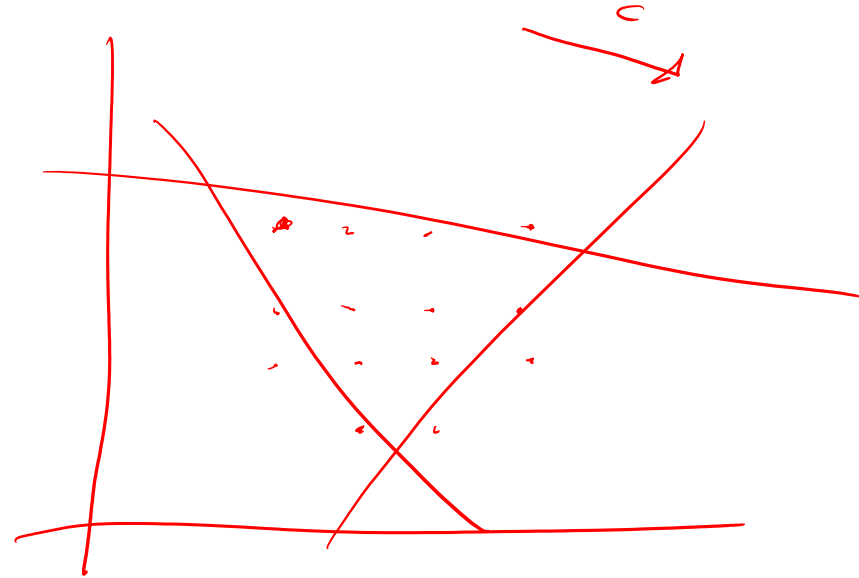
- Even more constrained: Binary Integer Programming $0 \leq x_i \leq 1, \text{ int}$
- A hybrid: Mixed Integer Linear Programming

Notation Alert!

Integer Programming: Graphical Representation

Just add a grid of integer points onto our LP representation

$$\begin{array}{ll} \min. & \mathbf{c}^T \mathbf{x} \\ \mathbf{x} & \\ \text{s.t.} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^N \end{array}$$



Integer Programming: Coloring

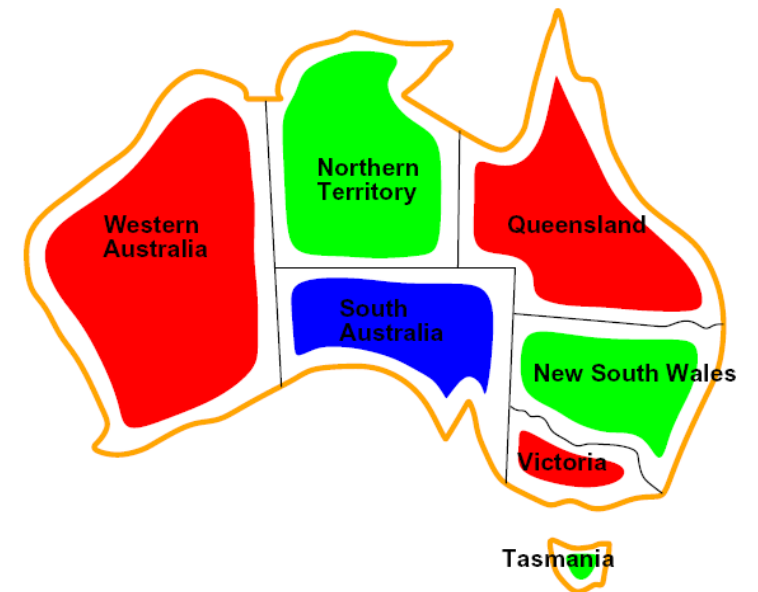
How would we formulate coloring as an **integer** program?

$$x \in \{0, 1\}$$

$$x_{NT,G}$$

$$x_{NT,G} + x_{NT,R} + x_{NT,B} = 1$$

$$x_{NT,G} + x_{WA,G} \leq 1$$



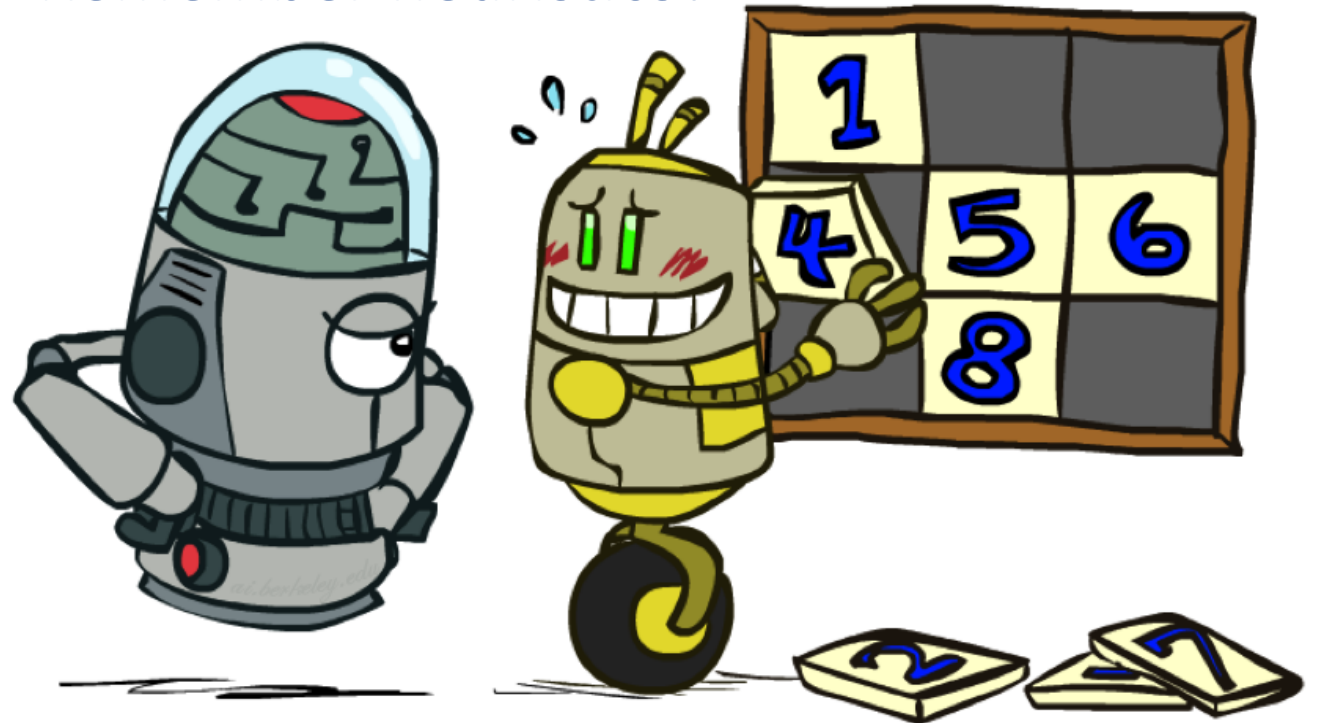
Convexity and IPs

Integer programs are not convex, but perhaps we can use LP solvers to help find solutions to integer programs?

Relax IP to LP by dropping integer constraints

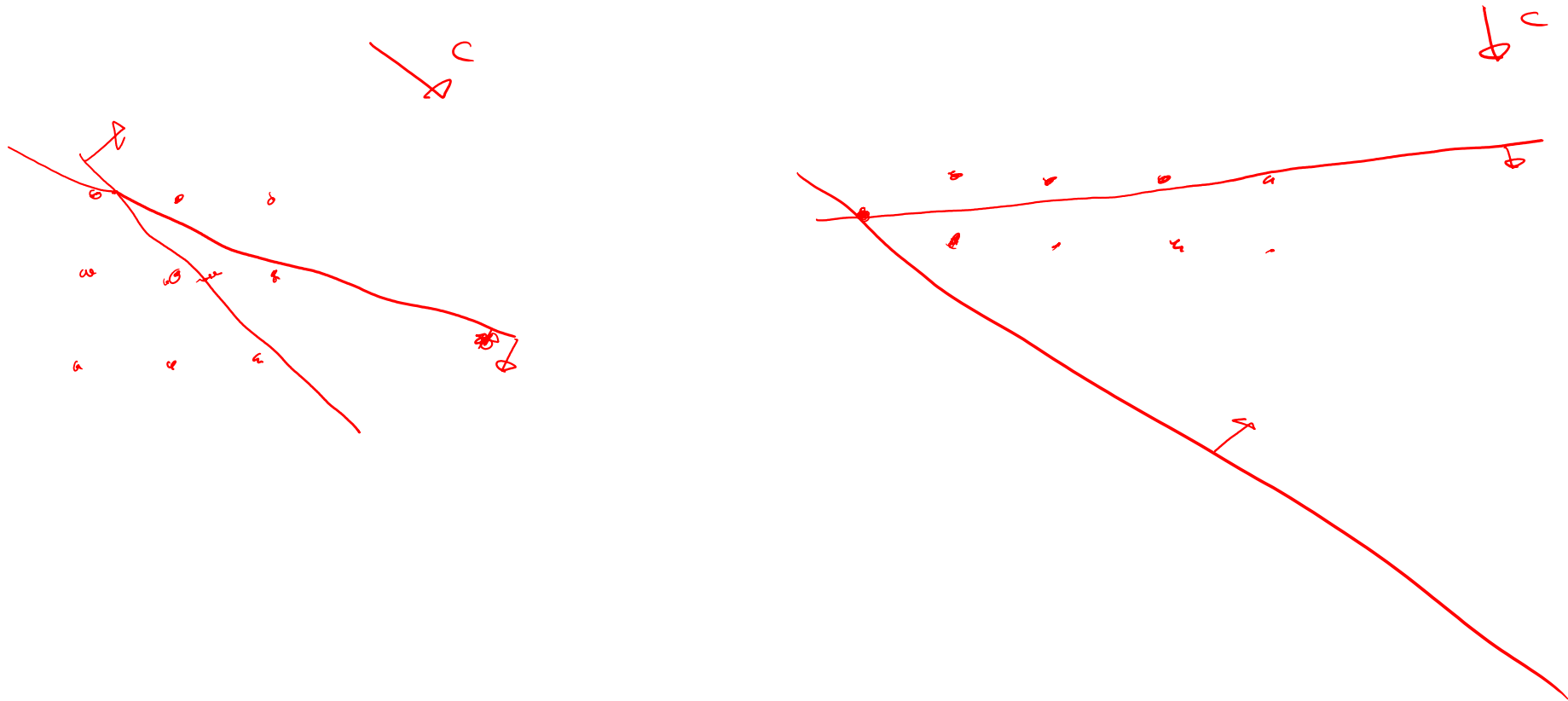
$$\begin{array}{ll} \min. & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^N \end{array}$$

Remember heuristics?



Poll 2:

True/False: It is sufficient to consider the integer points around the corresponding LP solution?



Poll 3:

Let y_{IP}^* be the optimal objective of an integer program P .

Let \mathbf{x}_{IP}^* be an optimal point of an integer program P .

Let y_{LP}^* be the optimal objective of the LP-relaxed version of P .

Let \mathbf{x}_{LP}^* be an optimal point of the LP-relaxed version of P .

Assume that P is a minimization problem.

Which of the following are true? Select all that apply.

A) $\mathbf{x}_{IP}^* = \mathbf{x}_{LP}^*$

B) $y_{IP}^* \leq y_{LP}^*$

C) $y_{IP}^* \geq y_{LP}^*$

$$\begin{aligned} y_{IP}^* = \min_{\mathbf{x}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^N \end{aligned}$$

$$\begin{aligned} y_{LP}^* = \min_{\mathbf{x}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \leq \mathbf{b} \end{aligned}$$

Solving an IP

Branch and Bound algorithm

1. Push current LP (with its solution) into priority queue,
ordered by objective value of LP solution (heuristic)

2. Repeat:

- If queue is empty, return that the IP is infeasible
- Pop candidate solution \mathbf{x}_{LP}^* from priority queue
- If \mathbf{x}_{LP}^* is all integer valued, we are done; return solution
- Otherwise, select a coordinate \bar{x}_i that is not integer valued, and add two additional LPs (with their solutions) to the priority queue, in each case adding 1 constraint to current LP:

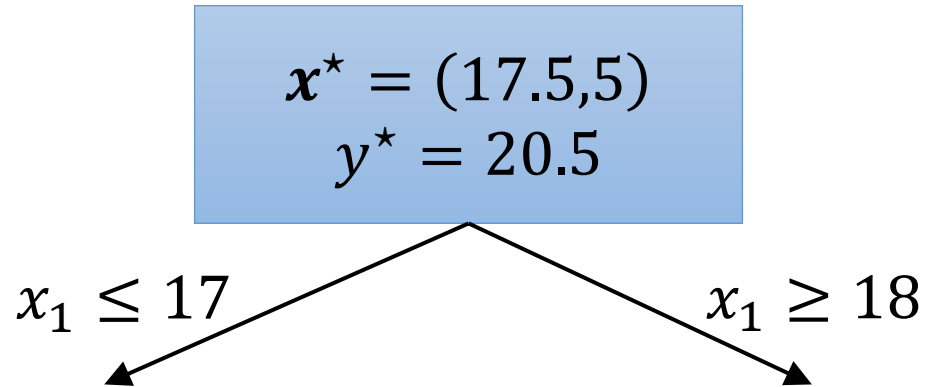
~~$x_i \leq 2$~~
or $x_i \geq 3$
 $\bar{x}_i = 2.5$

Left branch: Added constraint $x_i \leq \text{floor}(\bar{x}_i)$

Right branch: Added constraint $x_i \geq \text{ceil}(\bar{x}_i)$

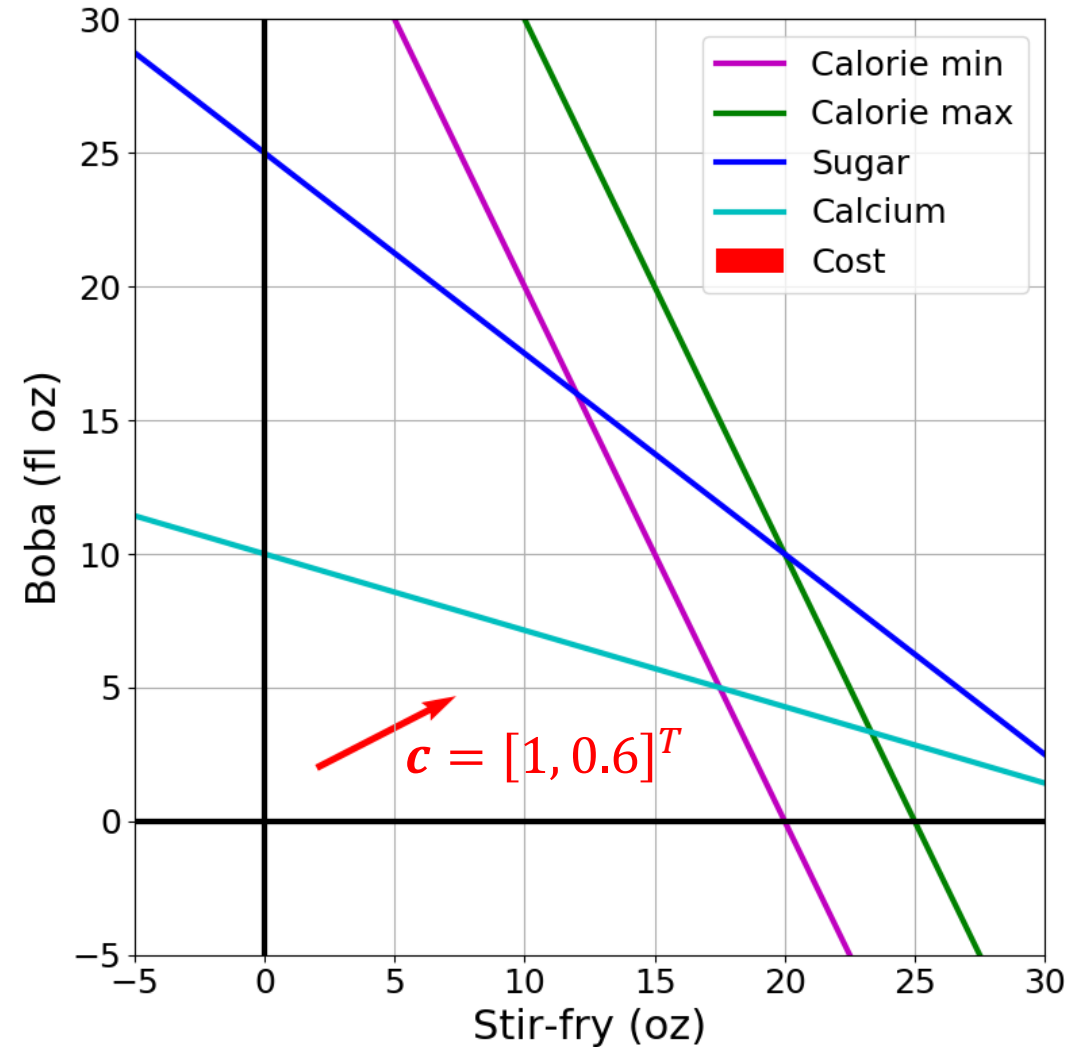
Note: Only add LPs to the queue if they are feasible

Branch and Bound Example

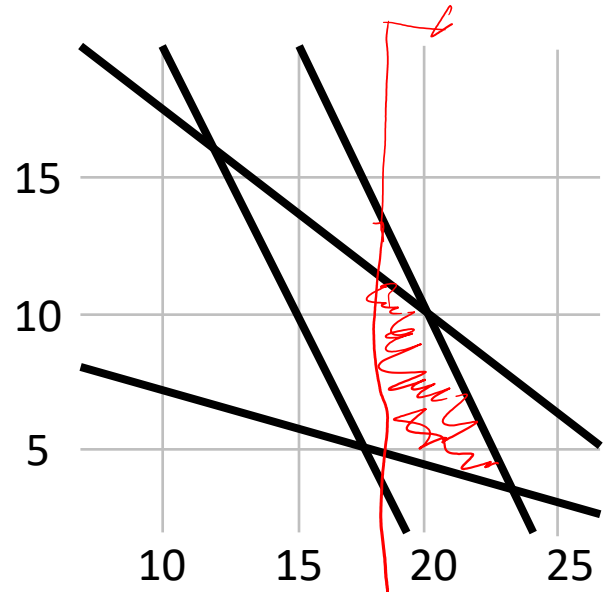
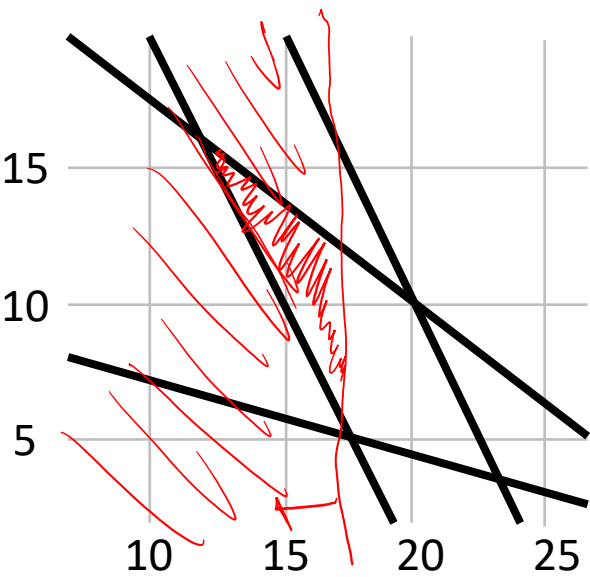
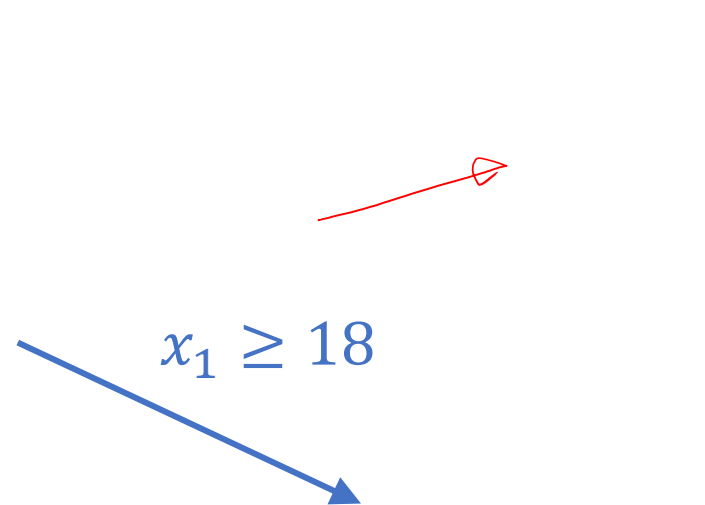
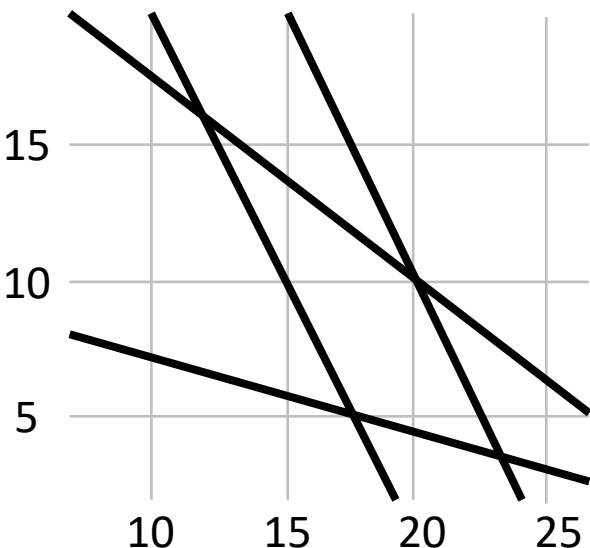
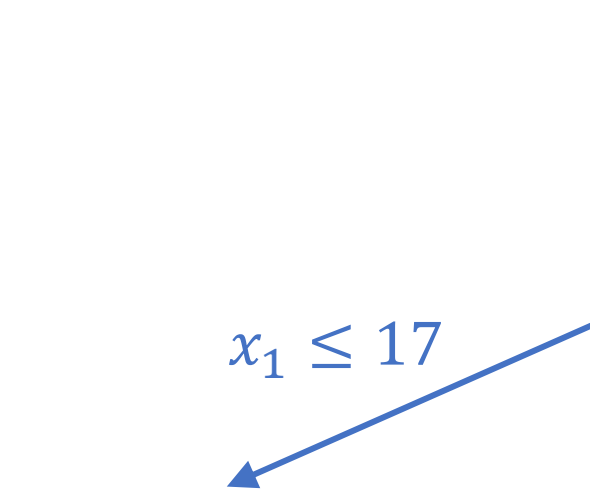


Priority Queue:

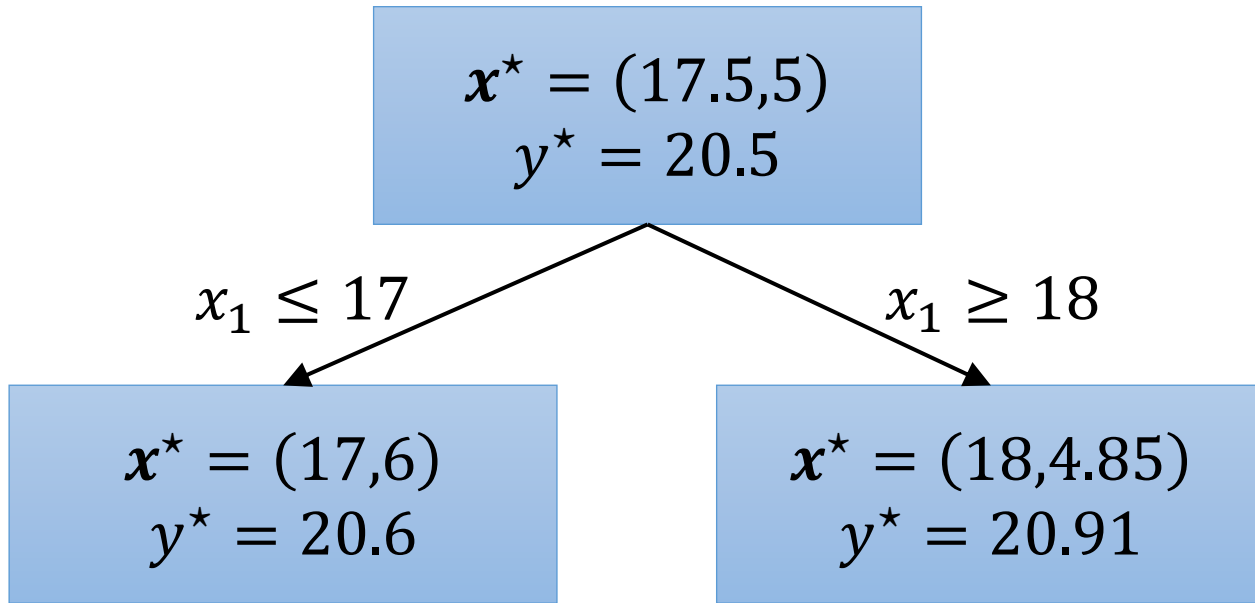
1. $x^* = (17.5, 5), y^* = 20.5$



Branch and Bound Example

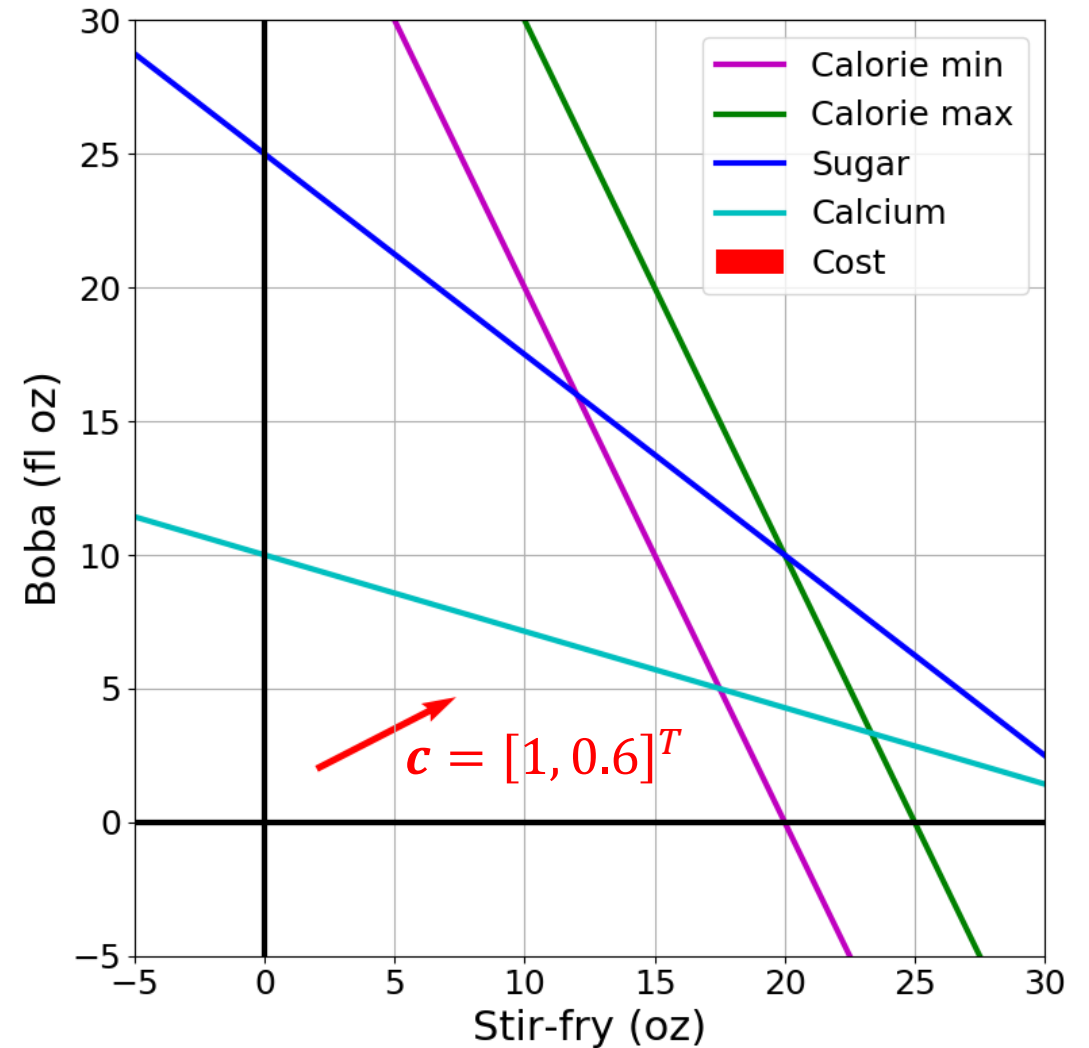


Branch and Bound Example



Priority Queue:

1. $x^* = (17, 6), y^* = 20.6$
2. $x^* = (18, 4.85), y^* = 20.91$



Activity + Poll

Constraints :

$$y \geq -1.4x + 4.58$$

$$y \geq 1.56x + 3.41$$

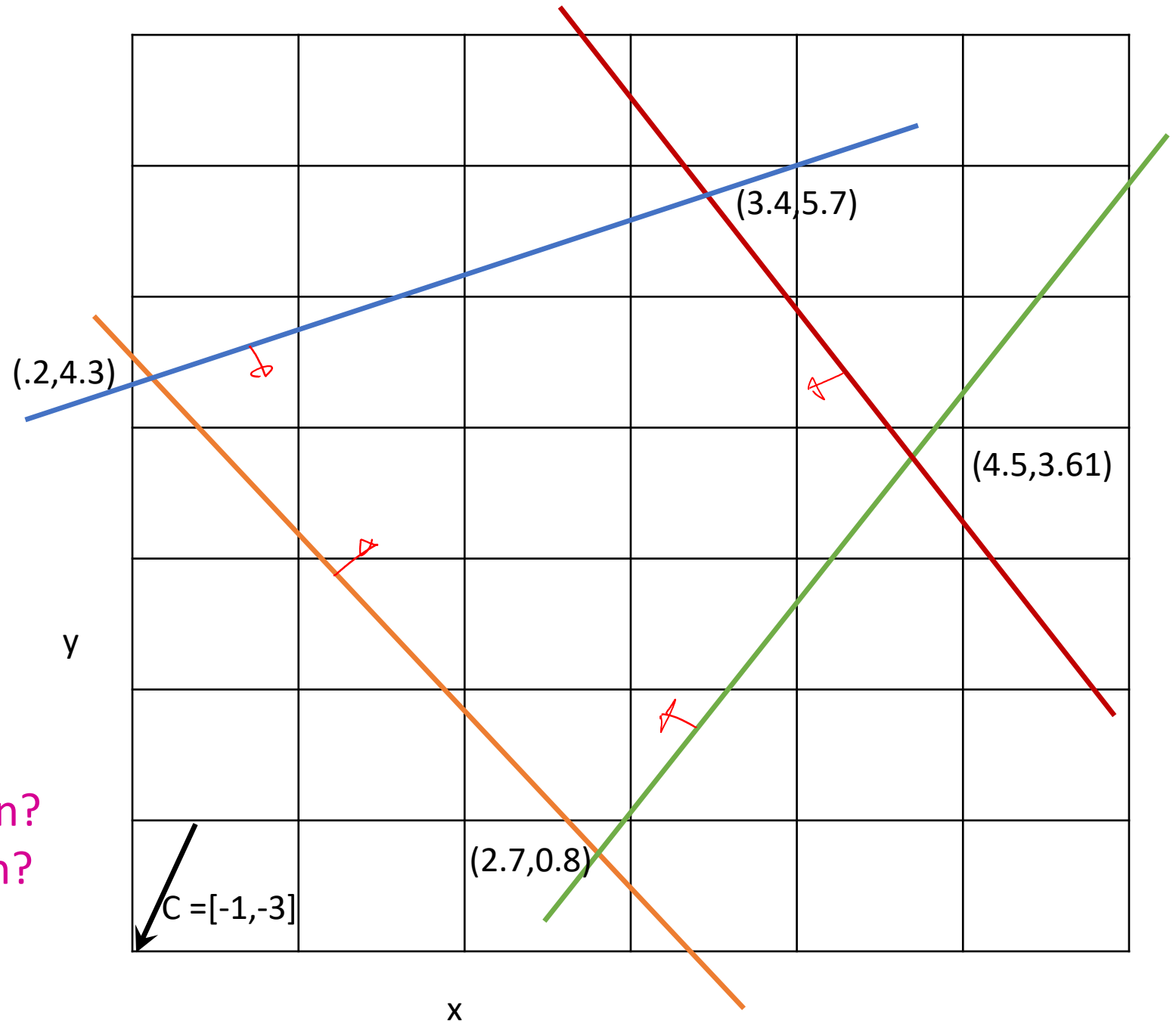
$$y \leq -1.9x + 12.16$$

$$y \leq .44x + 4.21$$

Priority Queue:

Poll 4: What is the LP solution?

Poll 5: What is the IP solution?



Activity

Constraints :

$$y = -1.4x + 4.58$$

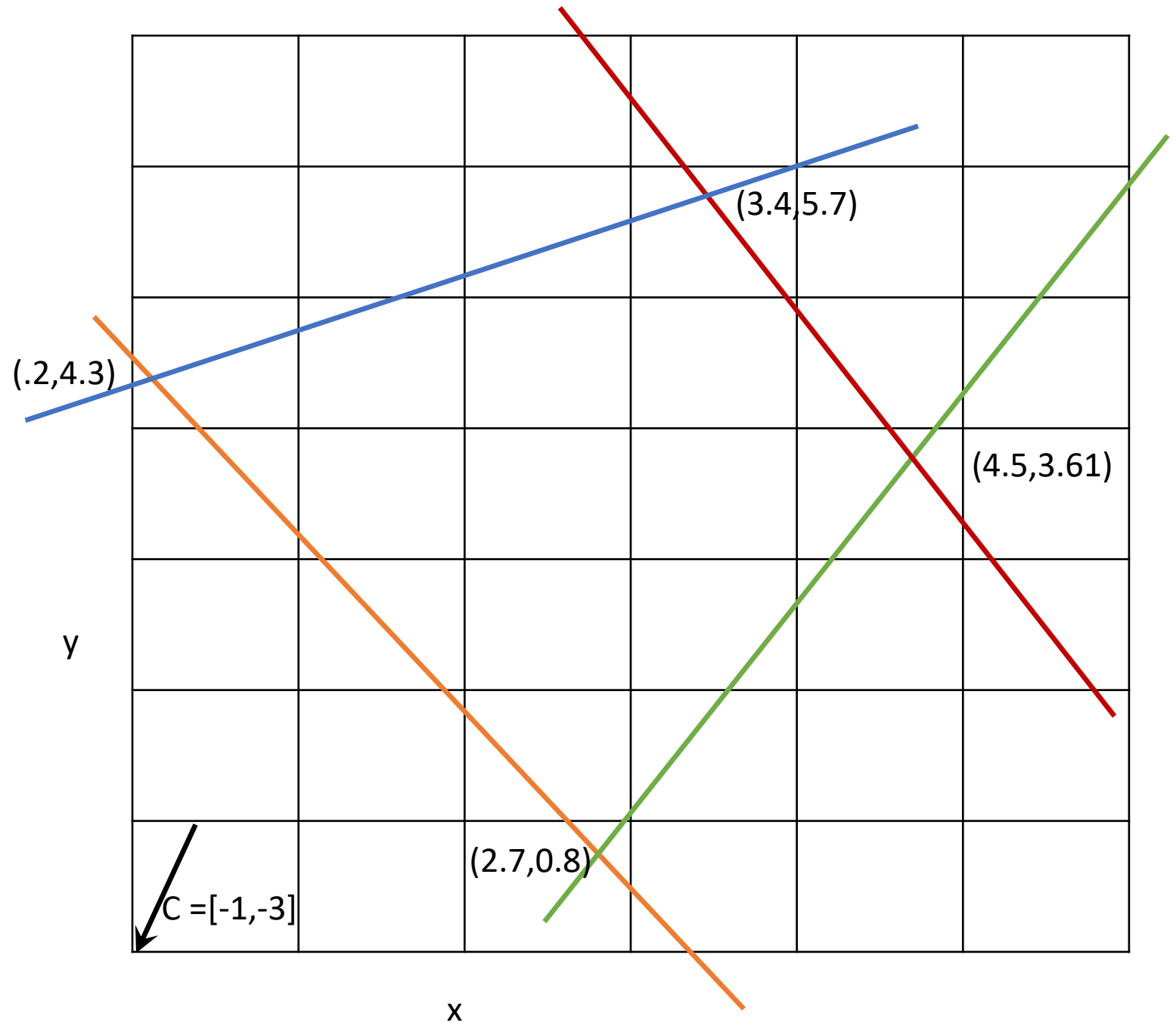
$$y = 1.56x + 3.41$$

$$y = -1.9x + 12.16$$

$$y = .44x + 4.21$$

Priority Queue:

-20.5: (3.4,5.7)



Activity

Constraints :

$$y = -1.4x + 4.58$$

$$y = 1.56x + 3.41$$

$$y = -1.9x + 12.16$$

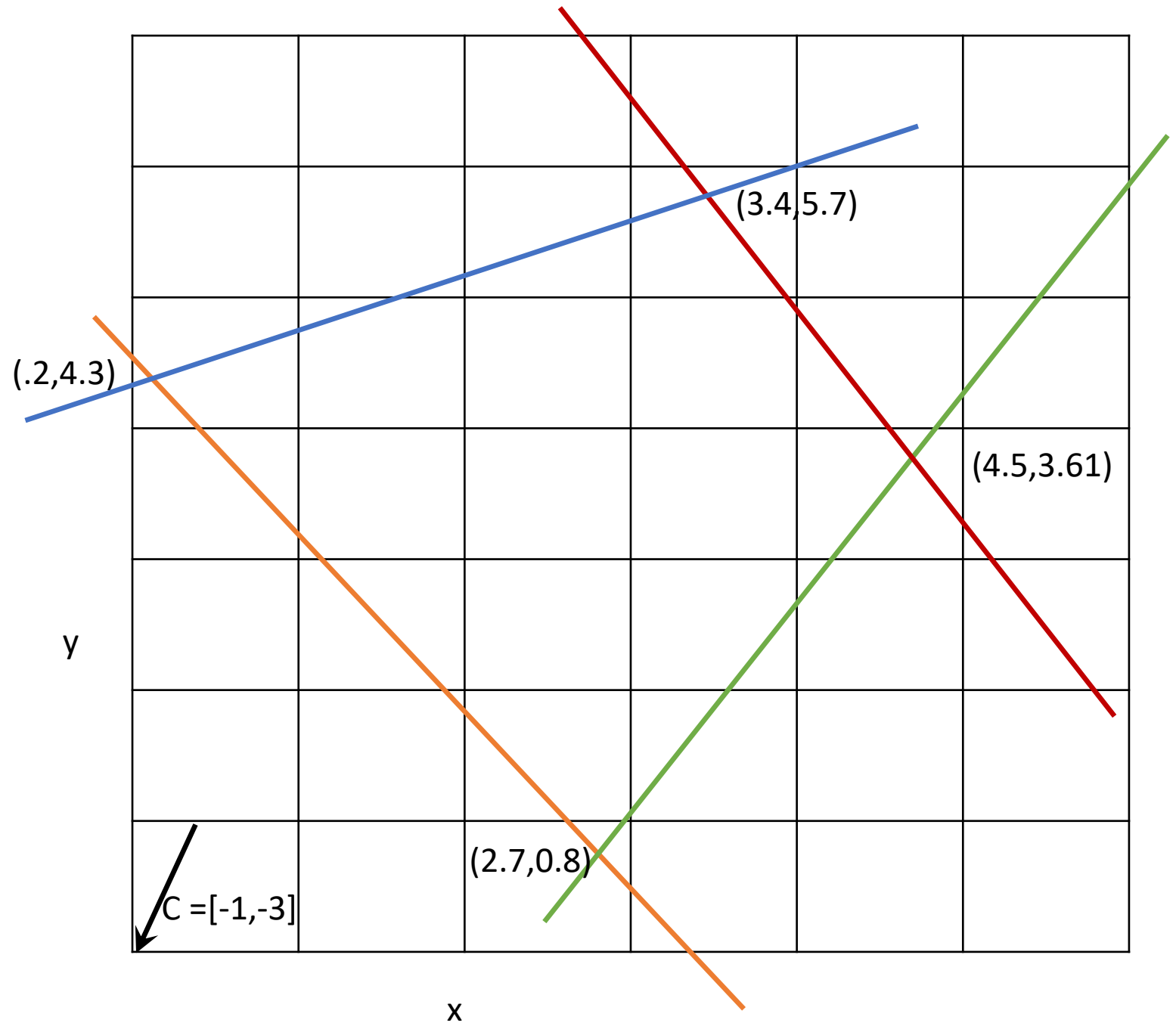
$$y = .44x + 4.21$$

Priority Queue:

~~-20.5: (3.4,5.7)~~

-19.6: (3,5.53) ($x \leq 3$)

-17.7: (4,4.56) ($x \geq 4$)



Activity

Constraints :

$$y = -1.4x + 4.58$$

$$y = 1.56x + 3.41$$

$$y = -1.9x + 12.16$$

$$y = .44x + 4.21$$

Priority Queue:

~~-20.5: (3.4,5.7)~~

~~-19.6: (3,5.53) (x ≤ 3)~~

-17.7: (4,4.56) (x ≥ 4)

-18.0: (3,5) (x ≤ 3, y ≤ 5)

Inf: (x ≤ 3, y ≥ 6)



Why do we not need to recurse on -17.7?