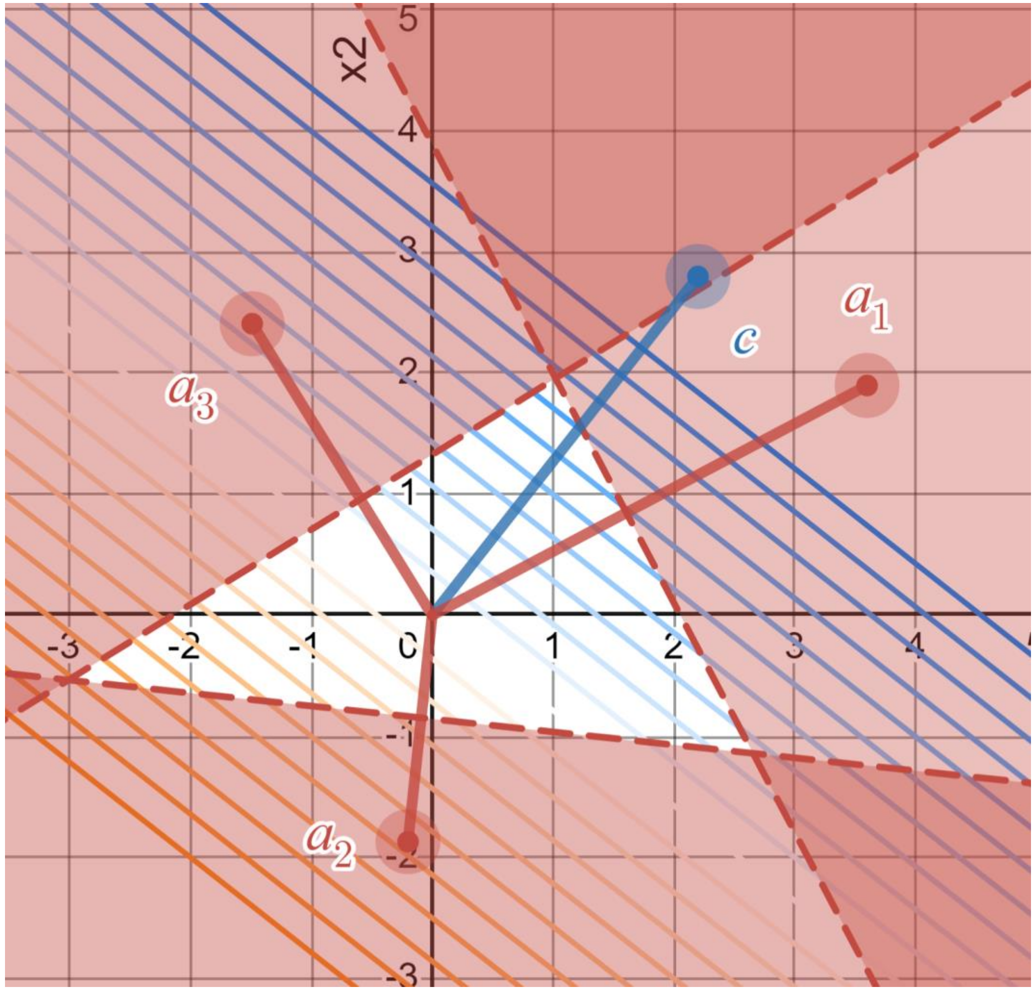
Warm-up as you walk in

What is the solution to this LP?



https://www.desmos.com/calculator/tnlo7p5plp

Plan

Last Time

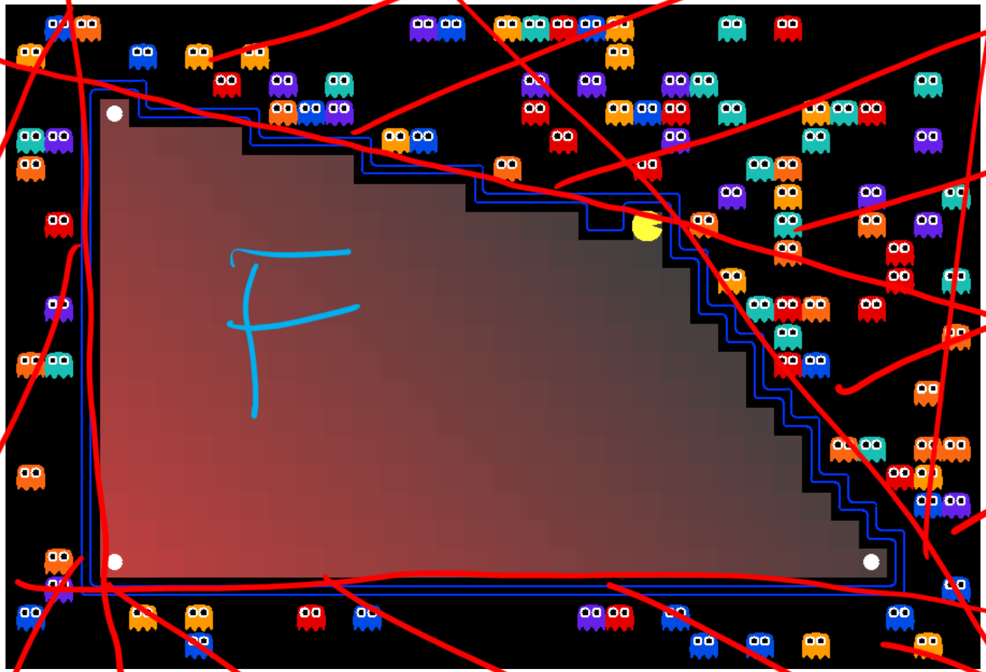
- Linear programming formulation
  - Problem description
  - Graphical representation
  - Optimization representation

Today

- Solving linear programs
- Higher dimensions than just 2 $\rightarrow$ 3+
- Integer programs

AI: Representation and Problem Solving

Integer Programming



Instructor: Pat Virtue

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

Reminder: Cost Contours

Given the cost vector $[c_1, c_2]^\top$ where will

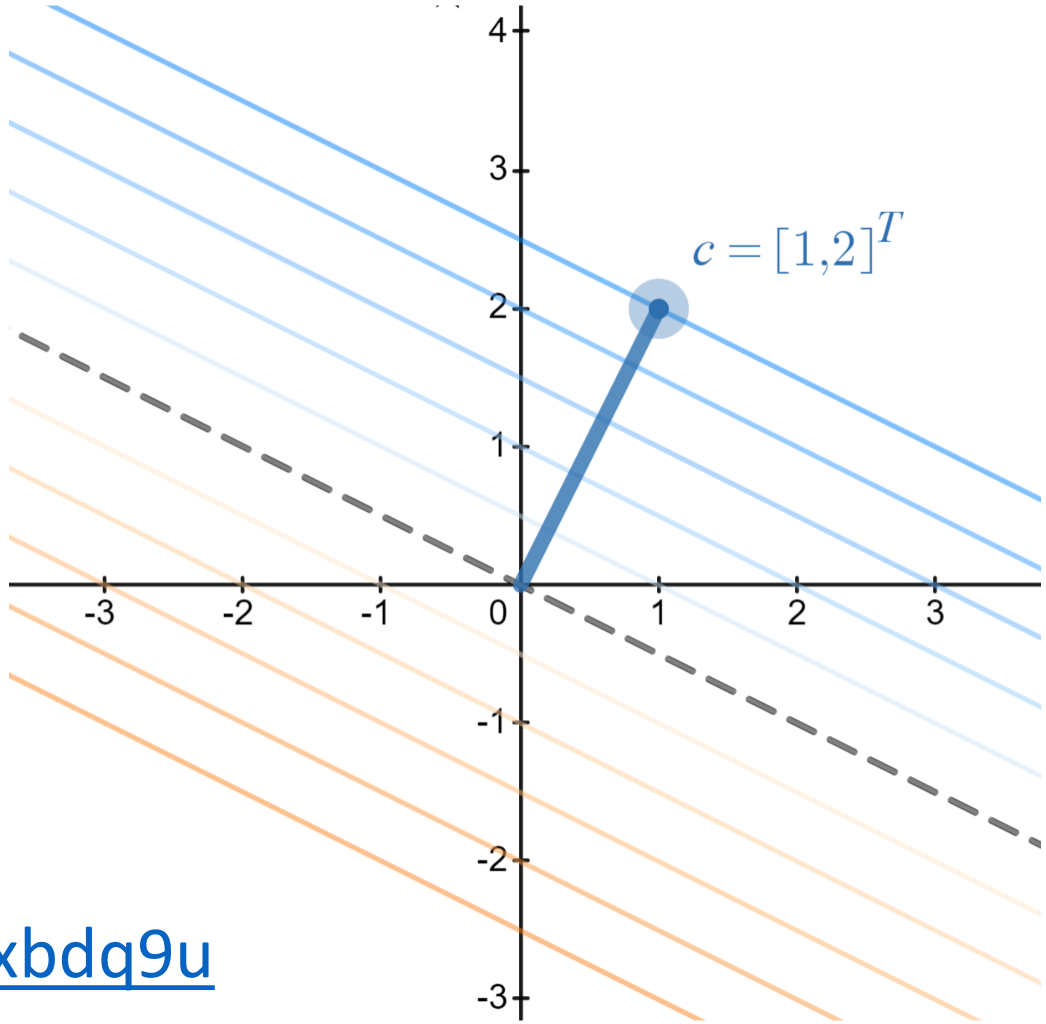
$\mathbf{c}^\top\mathbf{x} = 0$?

$\mathbf{c}^\top\mathbf{x} = 1$?

$\mathbf{c}^\top\mathbf{x} = 2$?

$\mathbf{c}^\top\mathbf{x} = -1$?

$\mathbf{c}^\top\mathbf{x} = -2$?



https://www.desmos.com/calculator/8d9kxbdq9u

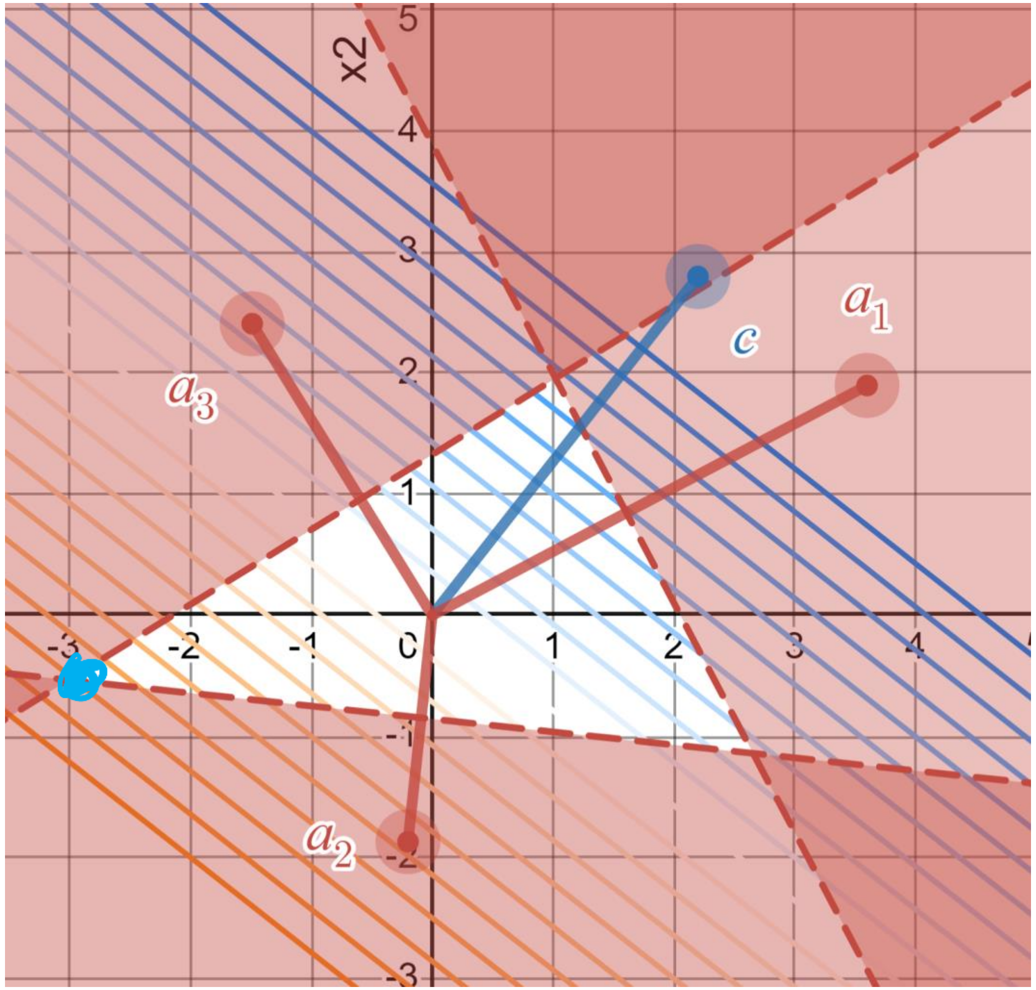
Solving a Linear Program

What is the solution to this LP?

$$y^* = c^T x^* \approx -8$$

$$y^* = \min_{\vec{x}} c^T x \quad \text{st } Ax \le b$$

$$\vec{x}^* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -3 \\ -0.5 \end{bmatrix}$$



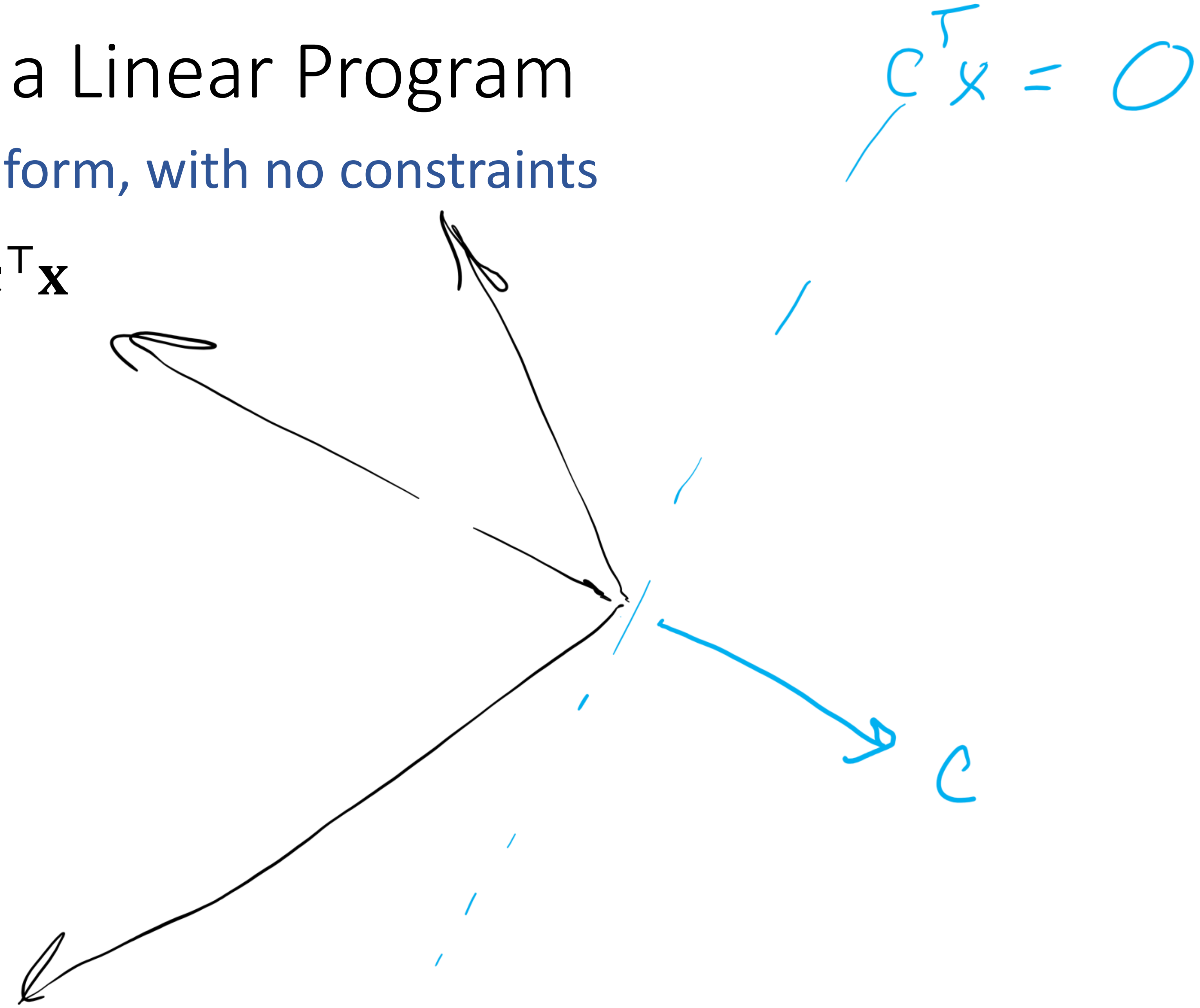
$$\vec{x}^* = \underline{\arg}\min_{\vec{x}} \vec{c}^T \vec{x} \quad \text{s.t. } A\vec{x} \le \vec{b}$$

https://www.desmos.com/calculator/tnlo7p5plp

Solving a Linear Program

Inequality form, with no constraints

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

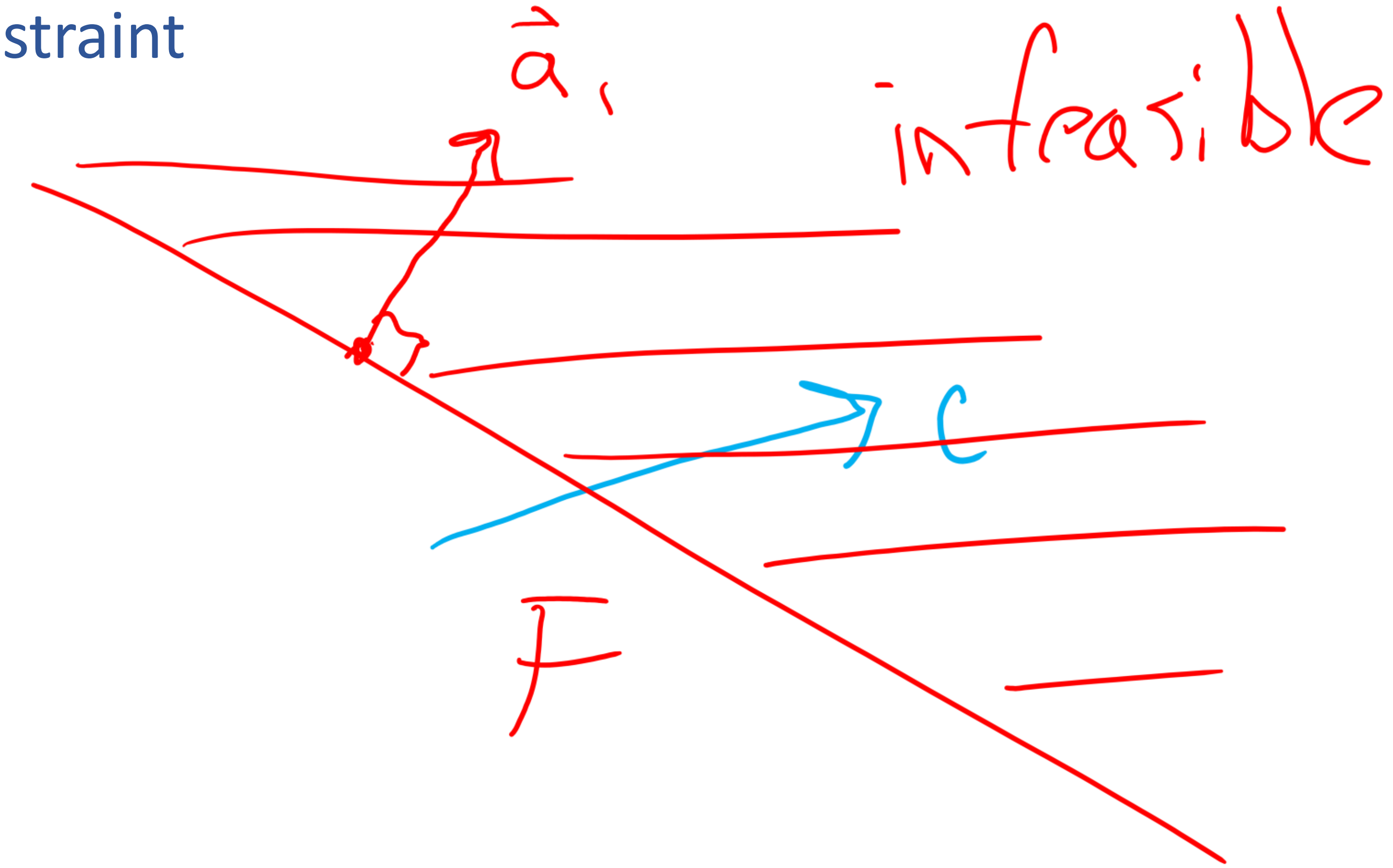


Solving a Linear Program

Inequality form, with one constraint

$$\begin{aligned} \min_{\mathbf{x}}. \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & a_1 x_1 + a_2 x_2 \leq b \end{aligned}$$

$A = [a_1 \quad a_2]$



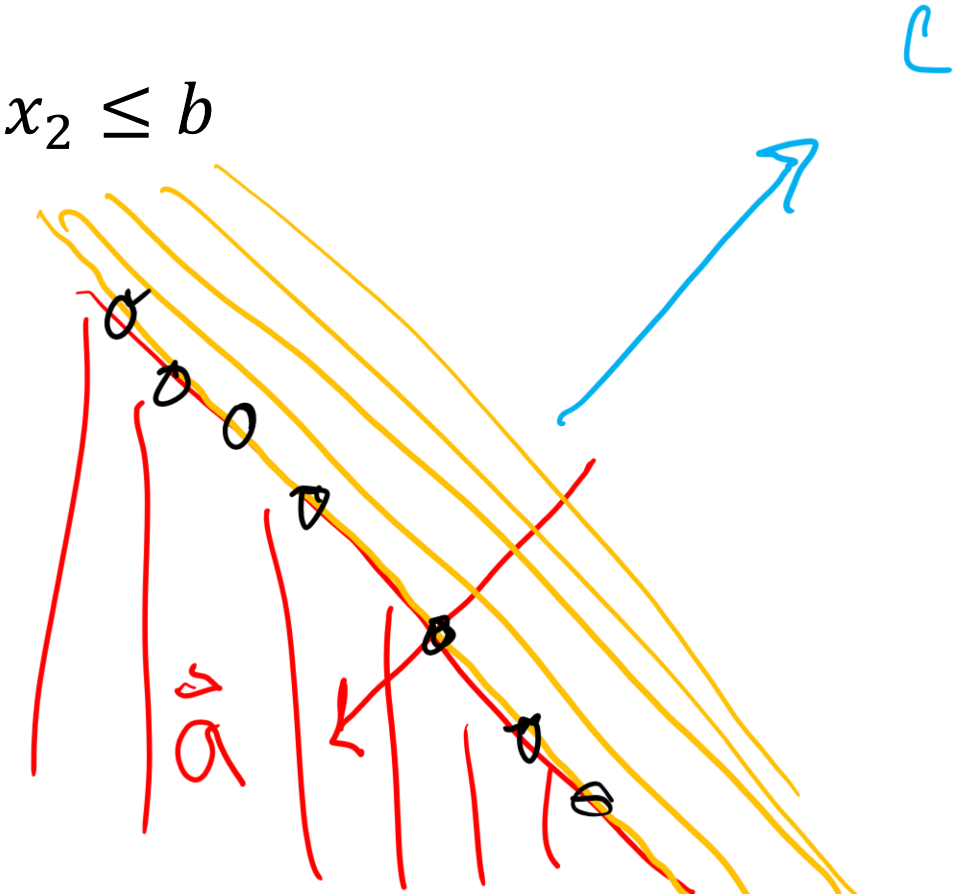
Poll 1

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

y^*

$$\min_{\mathbf{x}} \quad \mathbf{c}^\top \mathbf{x}$$
$$\text{s.t.} \quad a_1 x_1 + a_2 x_2 \leq b$$

x^*



Solving an LP

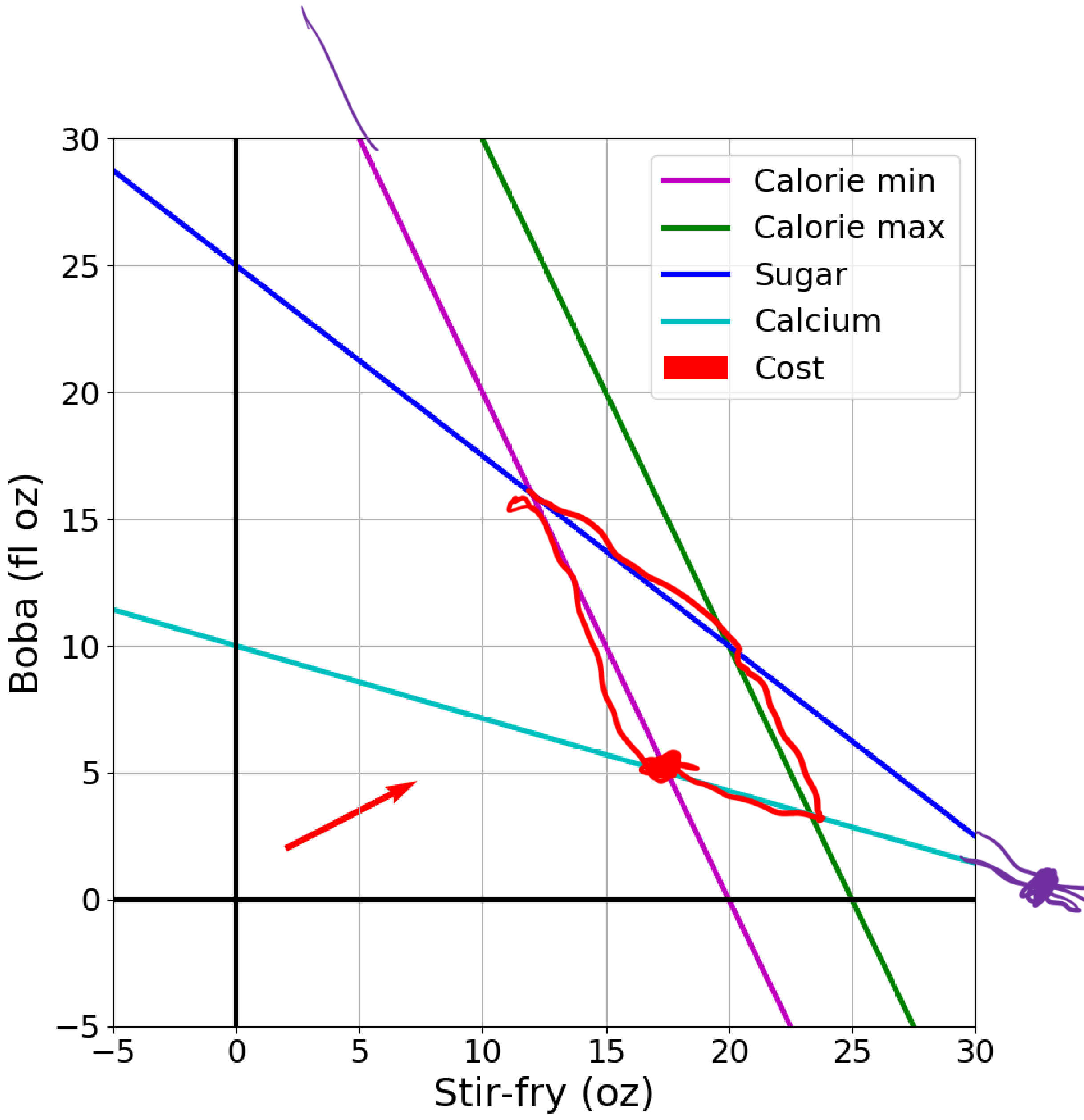
Solutions are at feasible intersections of constraint boundaries!!

Algorithms

- 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

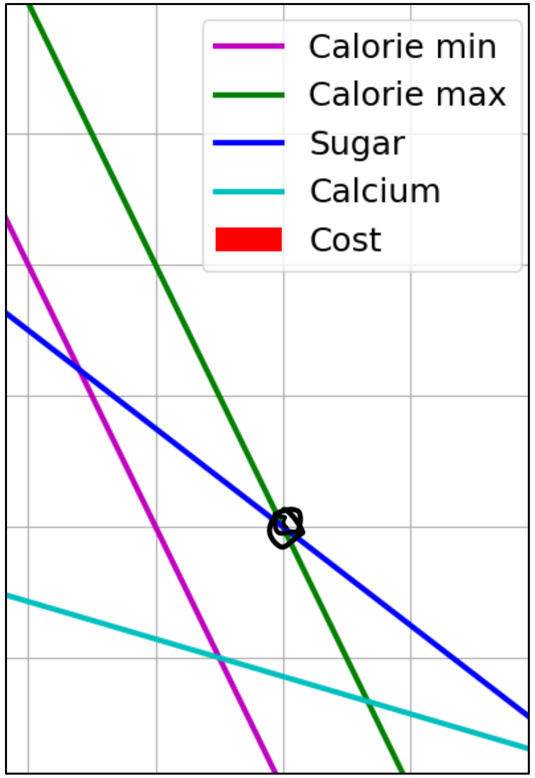


Solving an LP

But, how do we find the intersection between boundaries?

$$\min_{\mathbf{x}} \quad \mathbf{c}^{\mathsf{T}}\mathbf{x}$$
$$\text{s.t.} \quad A\mathbf{x} \preceq \mathbf{b}$$

$$A = \begin{bmatrix} -100 & -50 \\ 100 & 50 \\ 3 & 4 \\ -20 & -70 \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} -2000 \\ 2500 \\ 100 \\ -700 \end{bmatrix} \begin{matrix} \text{Calorie min} \\ \text{Calorie max} \\ \text{Sugar} \\ \text{Calcium} \end{matrix}$$



$$A' \begin{bmatrix} 100 & 50 \\ 3 & 4 \end{bmatrix} \qquad b' = \begin{bmatrix} 2500 \\ 100 \end{bmatrix}$$

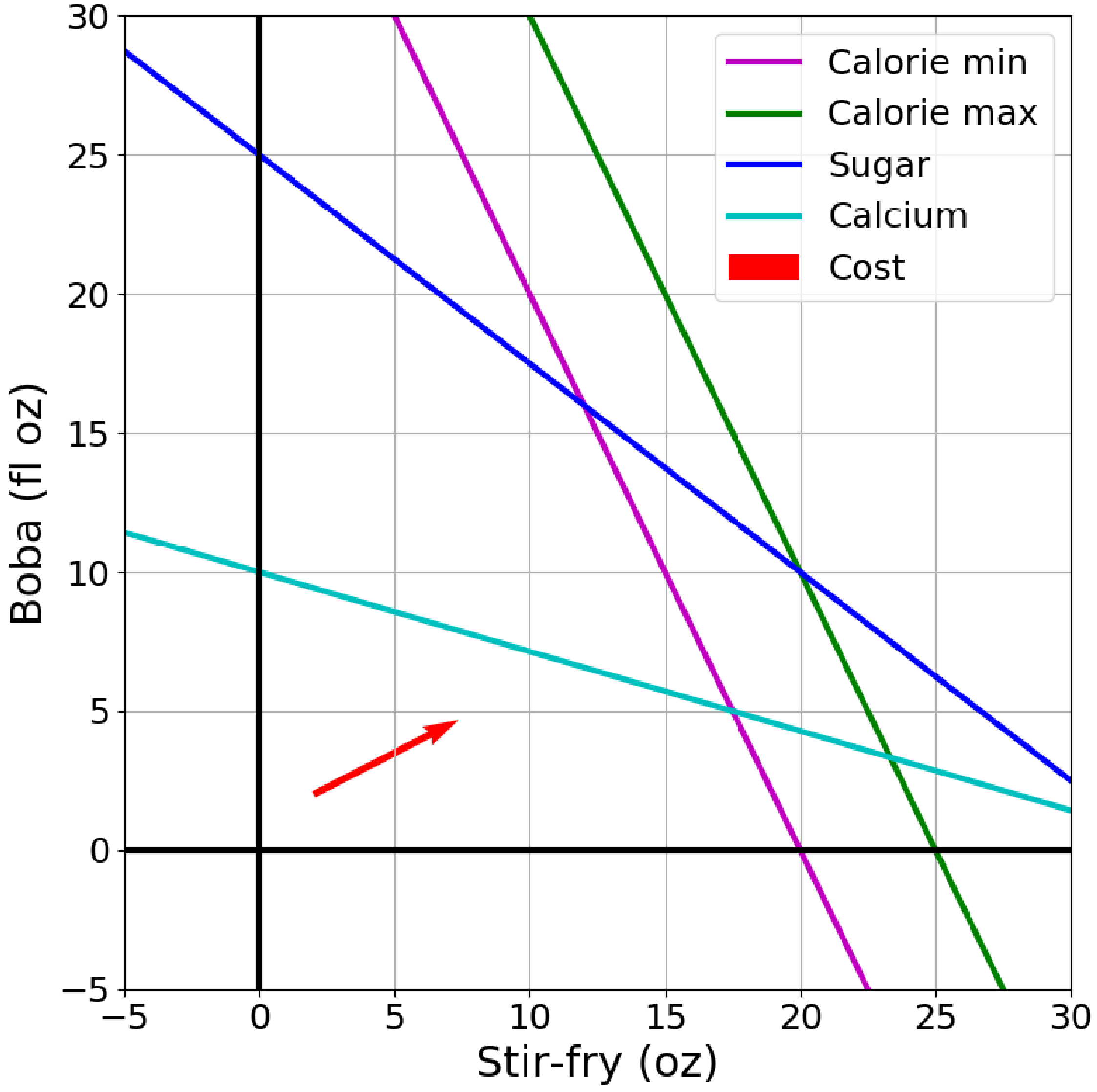
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad A'x = b'$$

Solving an LP

Solutions are at feasible intersections of constraint boundaries!!

Algorithms

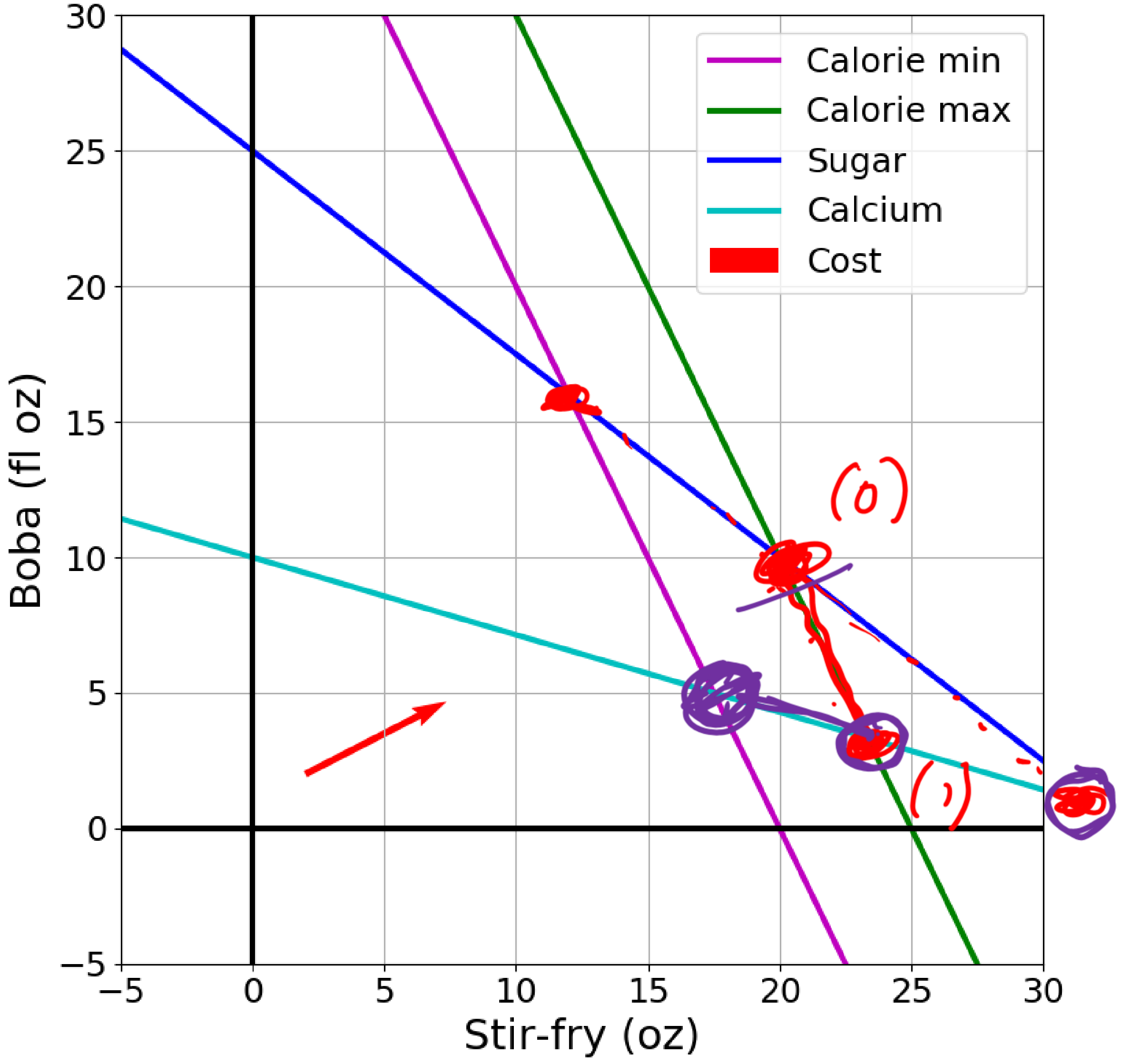
- Check objective at all feasible intersections
- Simplex



Solving an LP

Simplex algorithm

- 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

Solving an LP

Solutions are at feasible intersections of constraint boundaries!!

Algorithms

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

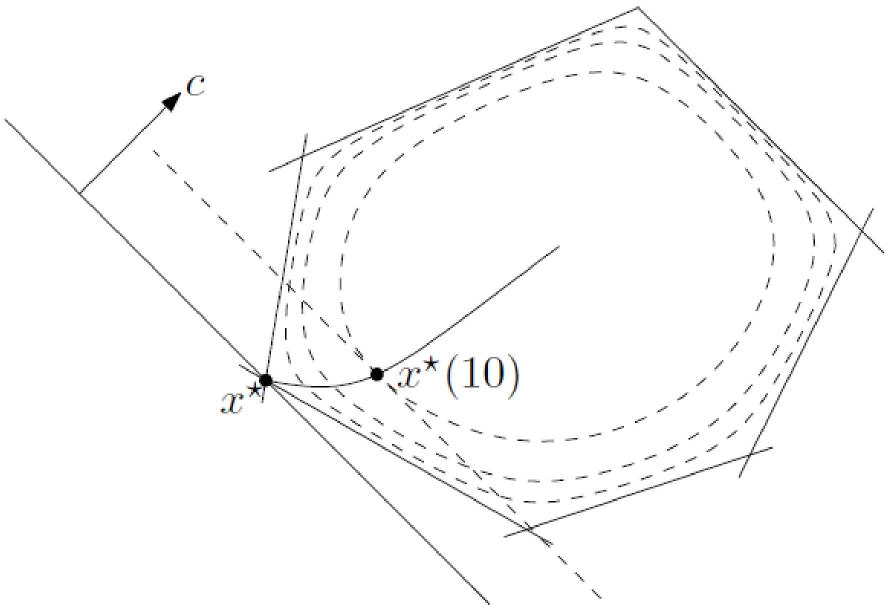


Figure 11.2 from Boyd and Vandenberghe, *Convex Optimization*

425

10-725

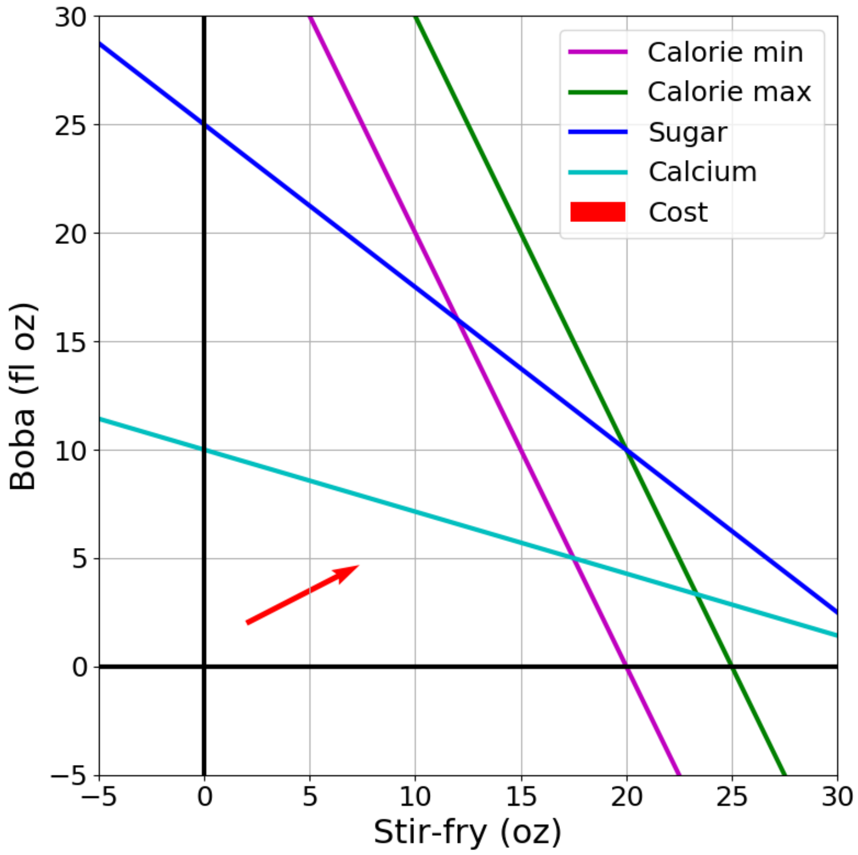
What about higher dimensions?

Problem Description

Optimization Representation

$$\min_{\mathbf{x}} \quad \mathbf{c}^\top \mathbf{x}$$
$$\text{s.t.} \quad A\mathbf{x} \preceq \mathbf{b}$$

Graphical Representation



“Marty, you’re not thinking fourth-dimensionally”

https://www.youtube.com/watch?v=CUcNM7OsdsY

Shapes in higher dimensions

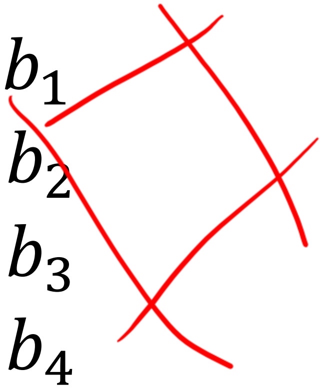
How do these linear shapes extend to 3-D, N-D?

$+a_3 x_3 +$

$$a_1\, x_1 + a_2\, x_2 = b_1$$

$$a_1\, x_1 + a_2\, x_2 \le b_1$$

$$\begin{aligned} a_{1,1}\, x_1 + a_{1,2}\, x_2 &\le b_1 \\ a_{2,1}\, x_1 + a_{2,2}\, x_2 &\le b_2 \\ a_{3,1}\, x_1 + a_{3,2}\, x_2 &\le b_3 \\ a_{4,1}\, x_1 + a_{4,2}\, x_2 &\le b_4 \end{aligned}$$



2-D	3-D	N-D
line	plane	hyperplane
halfplane	half space	half space
polygon	polyhedron	polytope

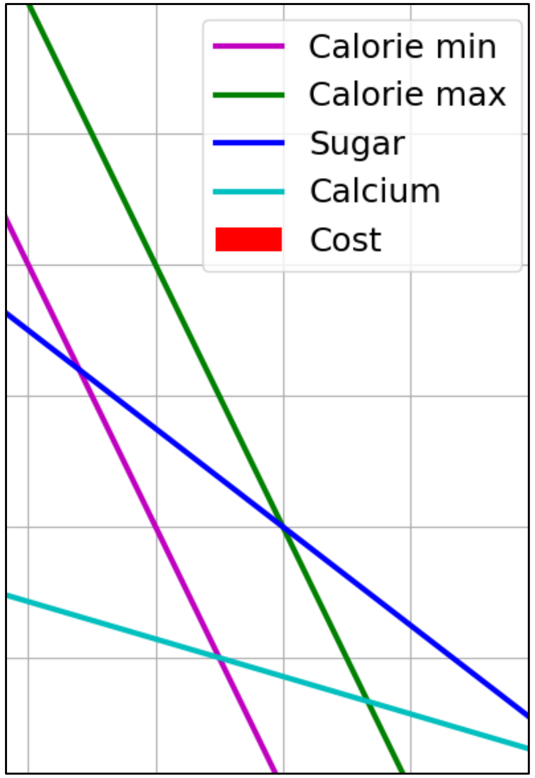
What are intersections in higher dimensions?

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

$$\min_{\mathbf{x}} \quad \mathbf{c}^{\top}\mathbf{x}$$
$$\text{s.t.} \quad A\mathbf{x} \preceq \mathbf{b}$$

$$A = \begin{bmatrix} -100 & -50 & 2 \\ 100 & 50 & 3 \\ 3 & 4 & 11 \\ -20 & -70 & 33 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -2000 \\ 2500 \\ 100 \\ -700 \end{bmatrix}$$

Calorie min
Calorie max
Sugar
Calcium



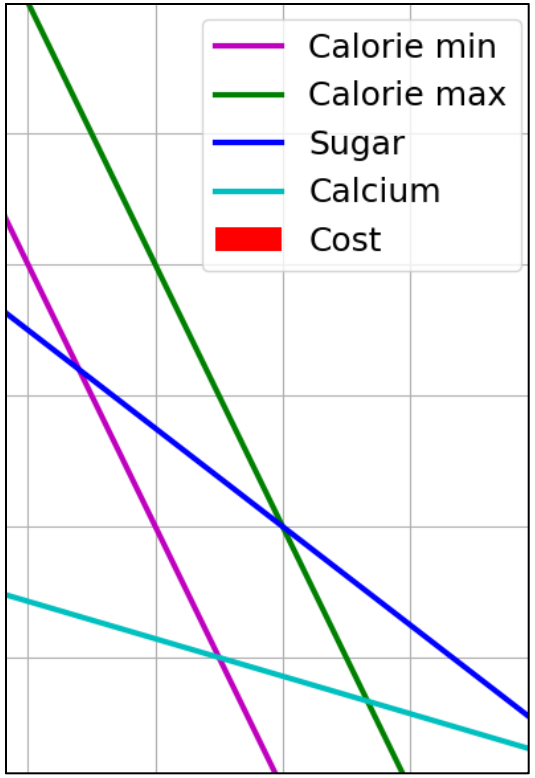
$$A' \begin{bmatrix} - & - & - \\ - & - & - \\ - & - & - \end{bmatrix} \qquad b' \begin{bmatrix} - \\ - \\ - \end{bmatrix}$$

How do we find intersections in higher dimensions?

Still looking at subsets of A matrix

$$\min_{\mathbf{x}} \quad \mathbf{c}^{\top}\mathbf{x}$$
$$\text{s.t.} \quad A\mathbf{x} \preceq \mathbf{b}$$

$$A = \begin{bmatrix} -100 & -50 \\ 100 & 50 \\ 3 & 4 \\ -20 & -70 \end{bmatrix} \qquad \mathbf{b} = \begin{array}{c} \begin{bmatrix} -2000 \\ 2500 \\ 100 \\ -700 \end{bmatrix} \end{array} \begin{array}{l} \text{Calorie min} \\ \text{Calorie max} \\ \text{Sugar} \\ \text{Calcium} \end{array}$$



Linear Programming

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

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

<u>Healthy Squad Goals</u>

- $2000 \leq$ Calories ≤ 2500
- Sugar ≤ 100 g
- Calcium ≥ 700 mg

x_1

x_2

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 ≤ Calories ≤ 2500
- Sugar ≤ 100 g
- Calcium ≥ 700 mg

	Food	Cost	Calories	Sugar	Calcium
x_1	Stir-fry (per bowl)	1	100	3	20
x_2	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 $\boldsymbol{x}$ must be integers

LP

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \preceq \mathbf{b} \end{aligned}$$

$\longrightarrow$

IP

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \preceq \mathbf{b} \\ & \boxed{\mathbf{x} \in \mathbb{Z}^N} \end{aligned}$$

We could also do:

- Even more constrained: Binary Integer Programming
- A hybrid: Mixed Integer Linear Programming

Notation Alert!

Integer Programming: Graphical Representation

Just add a grid of integer points onto our LP representation

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

Relaxation

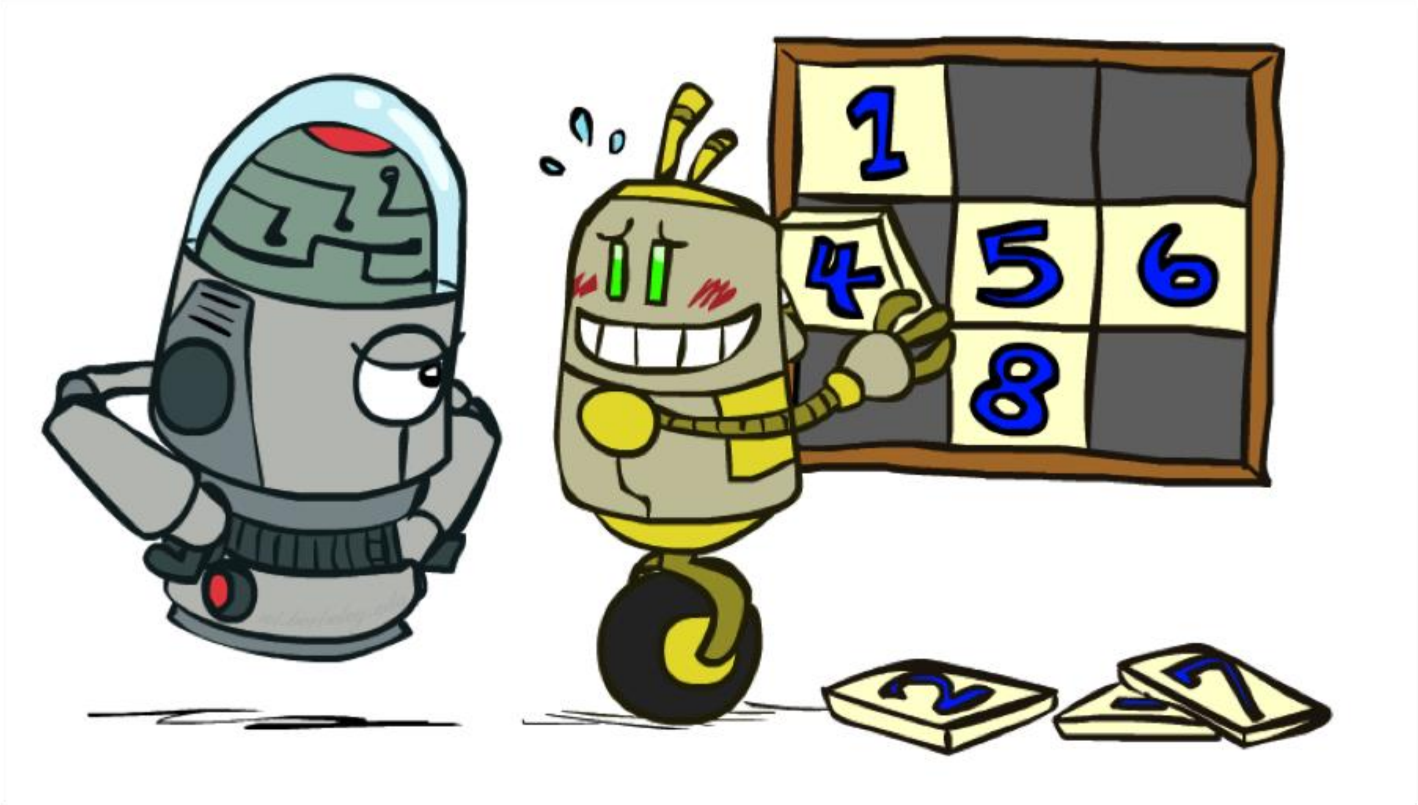
Relax IP to LP by dropping integer constraints

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

$LP \rightarrow \vec{x}^*_{LP},\ y^*_{LP}$

$IP \rightarrow \vec{x}^*_{IP},\ y^*_{IP}$

Remember heuristics?



Poll 2:

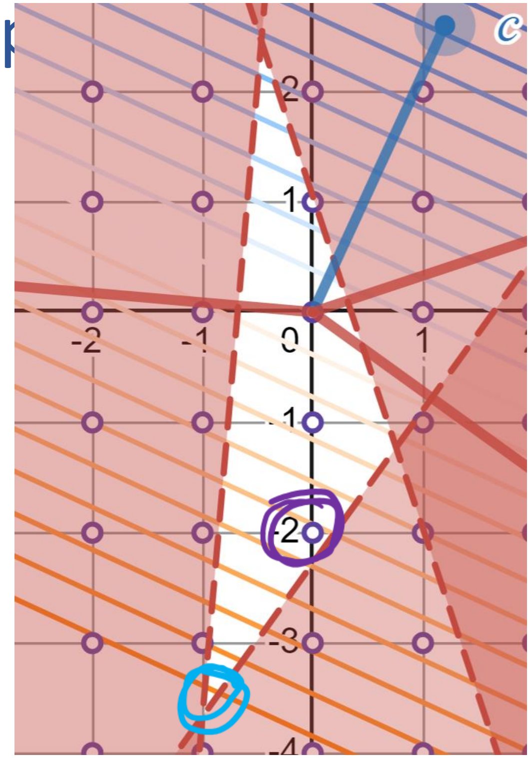
Let y_{IP}^* be the optimal objective of an integer program P.

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 p



$$y_{IP}^* = \min_{\mathbf{x}} \quad \mathbf{c}^\top \mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} \preceq \mathbf{b}, \quad \mathbf{x} \in \mathbb{Z}^N$$

$$y_{LP}^* = \min_{\mathbf{x}} \quad \mathbf{c}^\top \mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} \preceq \mathbf{b}$$

Which of the following are true?

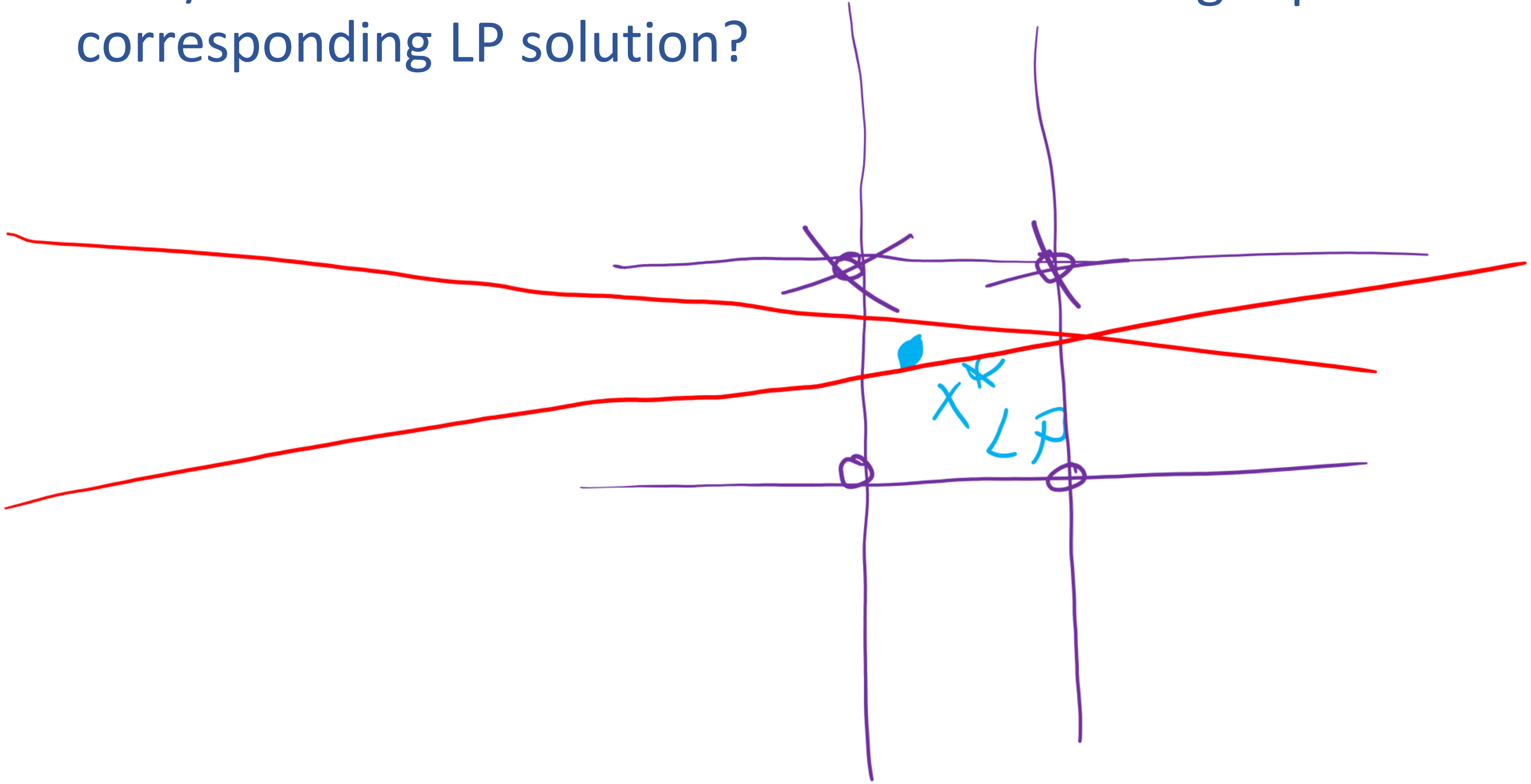
A) $\mathbf{x}_{IP}^* = \mathbf{x}_{\mathrm{LP}}^*$

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

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

Poll 3:

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



Solving an IP

Branch and Bound algorithm

1. Push LP solution of problem into priority queue,

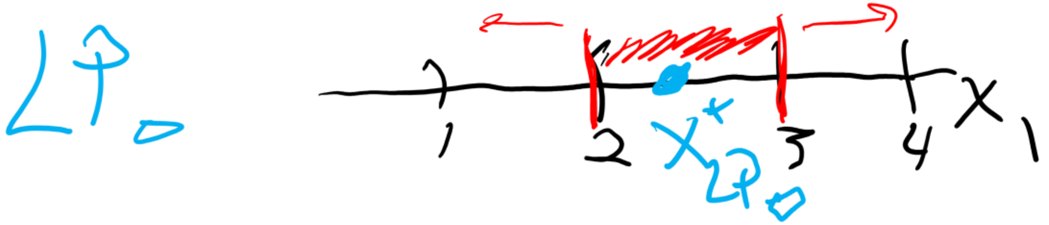
   ordered by objective value of LP solution

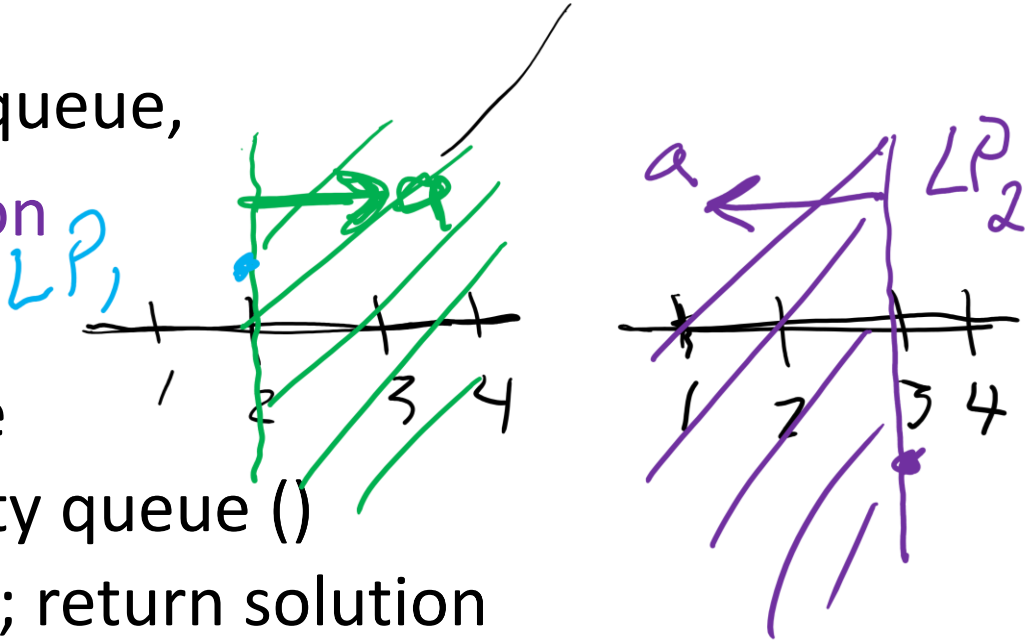
2. Repeat:
   - If queue is empty, return IP is infeasible
   - Pop candidate solution $\mathbf{x}_{LP}^{\star}$ from priority queue ()
   - If $\mathbf{x}_{LP}^{\star}$ is all integer valued, we are done; return solution
   - Otherwise, select a coordinate x_i that is not integer valued, and add two additional LPs to the priority queue:

     Left branch: Added constraint $x_i \leq floor(x_i)$

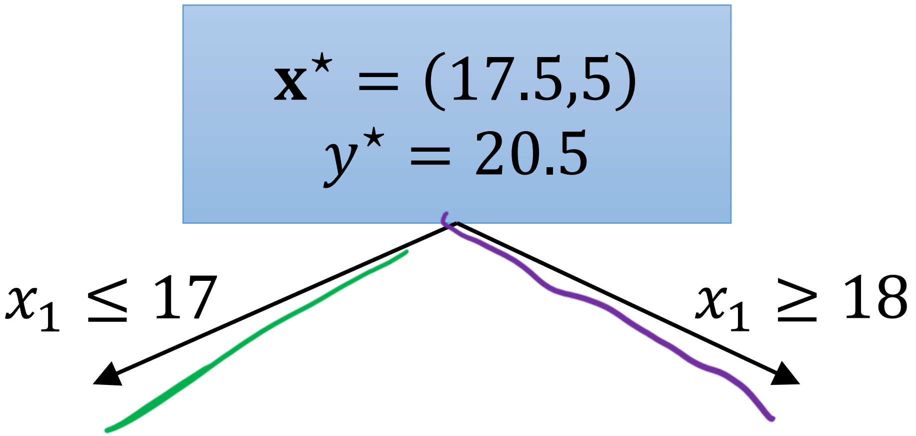
     Right branch: Added constraint $x_i \geq ceil(x_i)$

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



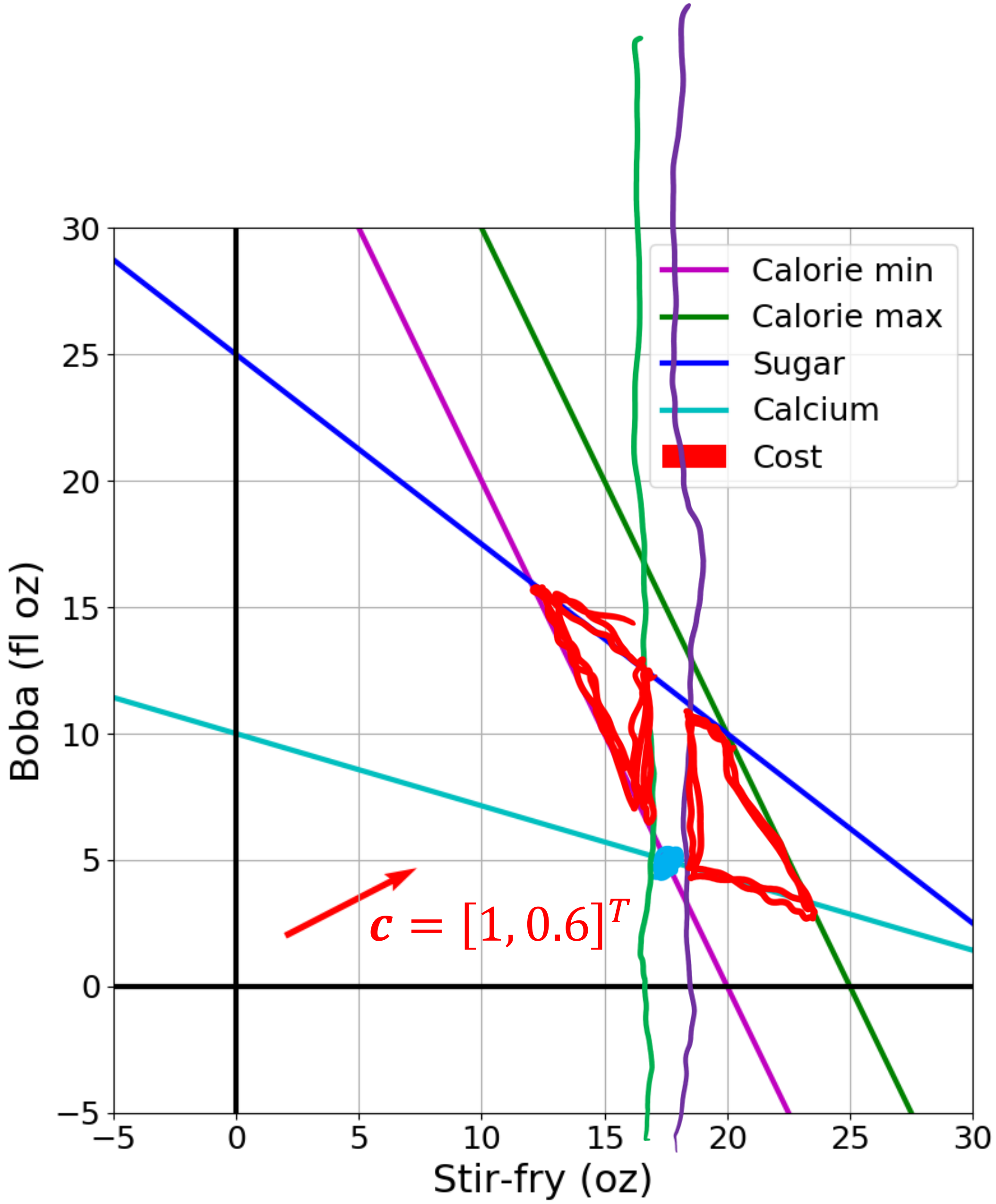


Branch and Bound Example

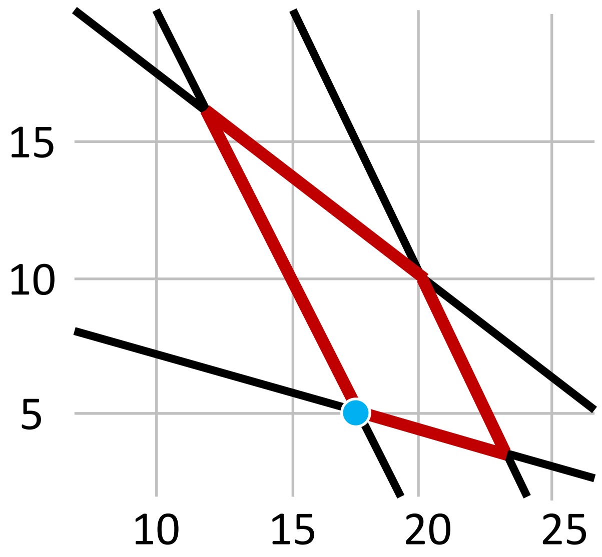


Priority Queue:

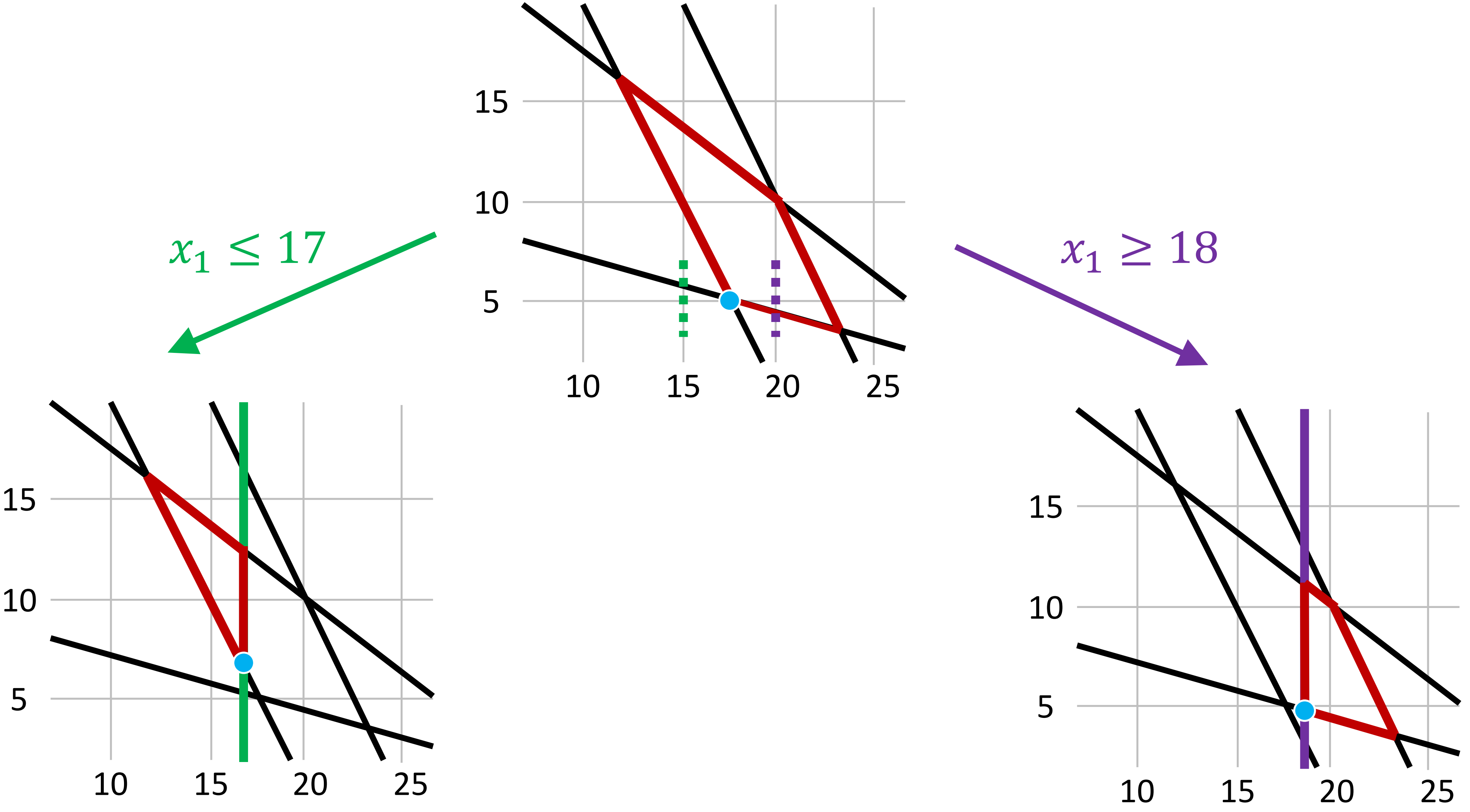
1. $\mathbf{x}^\star = (17.5,5)$, $y^\star = 20.5$



Branch and Bound Example



Branch and Bound Example



Branch and Bound Example

$$\mathbf{x}^\star = (17.5,5)$$
$$y^\star = 20.5$$

$x_1 \leq 17$ | $x_1 \geq 18$

$$\mathbf{x}^\star = (17,6)$$
$$y^\star = 20.6$$

$$\mathbf{x}^\star = (18,4.85)$$
$$y^\star = 20.91$$

?

Priority Queue:

1. $\mathbf{x}^\star = (17,6), \quad y^\star = 20.6$
2. $\mathbf{x}^\star = (18,4.85), \quad y^\star = 20.91$