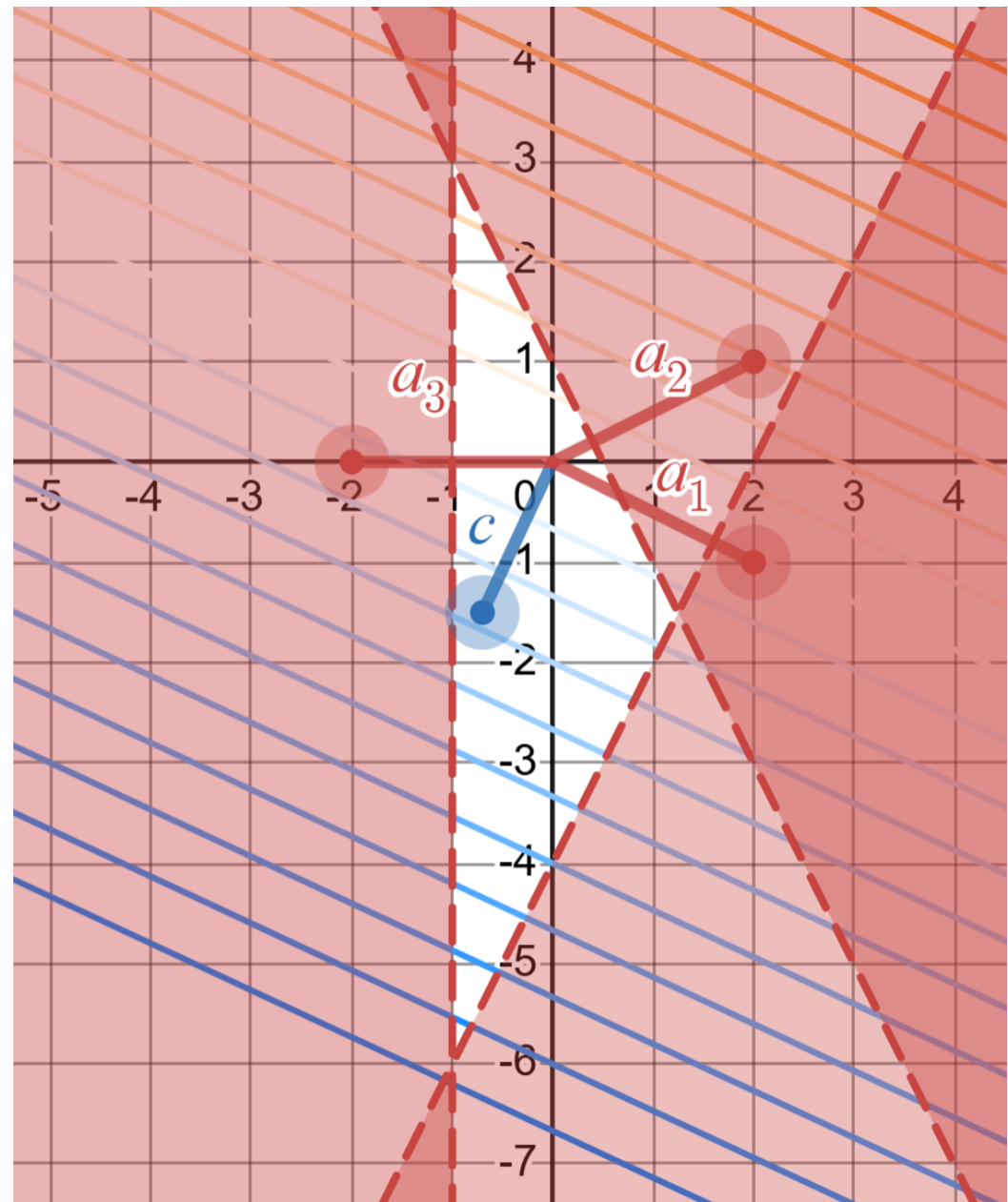


Poll 1

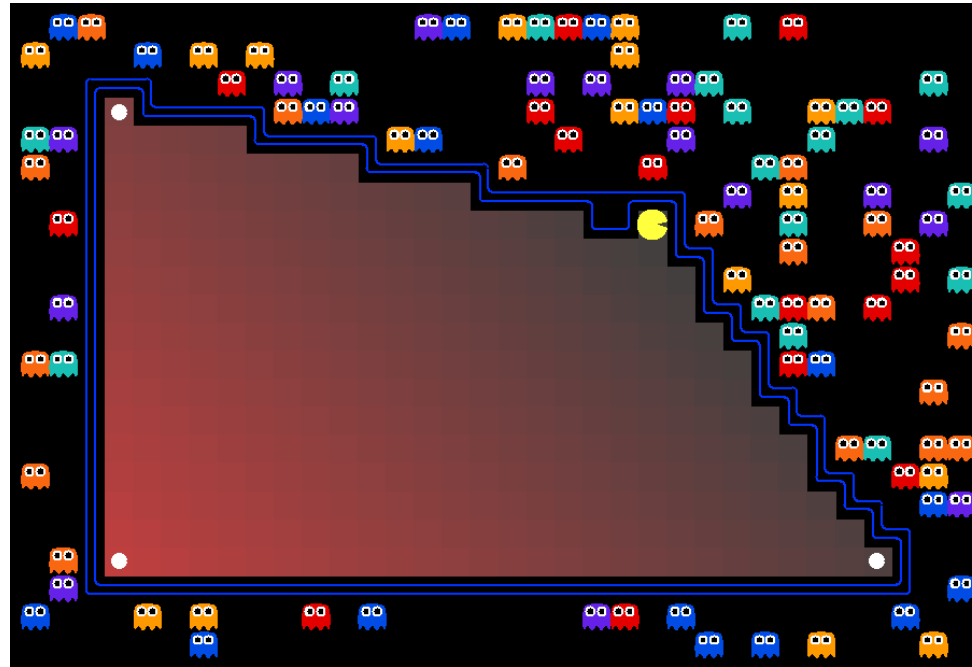
What is the solution to this LP?



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

07-380 AI & ML II

Linear and Integer Programming



Instructors: Pat Virtue & Mohammad Salameh

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

Integer Programming

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).

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

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

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

We could also do:

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

Notation Alert!

Integer Programming: Graphical Representation

Just add a grid of integer points onto our LP representation

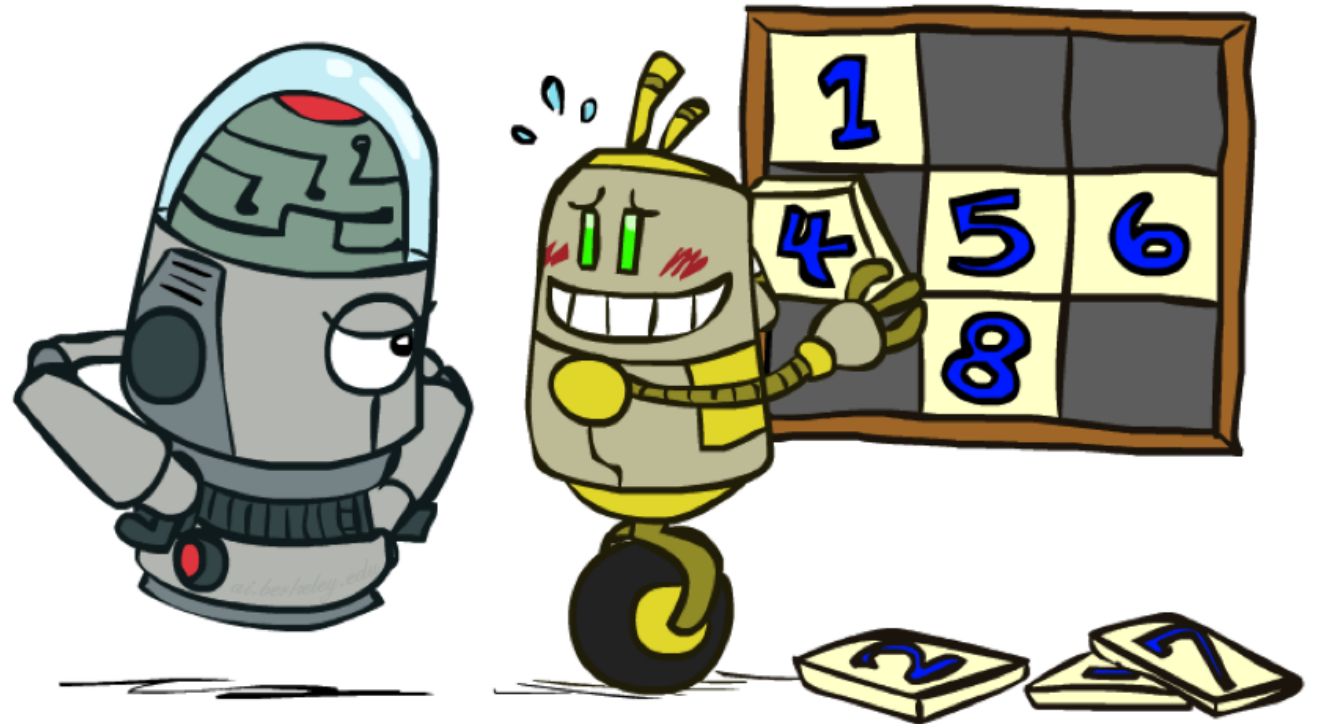
$$\begin{array}{ll}\min. & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^N\end{array}$$

Relaxation

Relax IP to LP by dropping integer constraints

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

Remember heuristics?



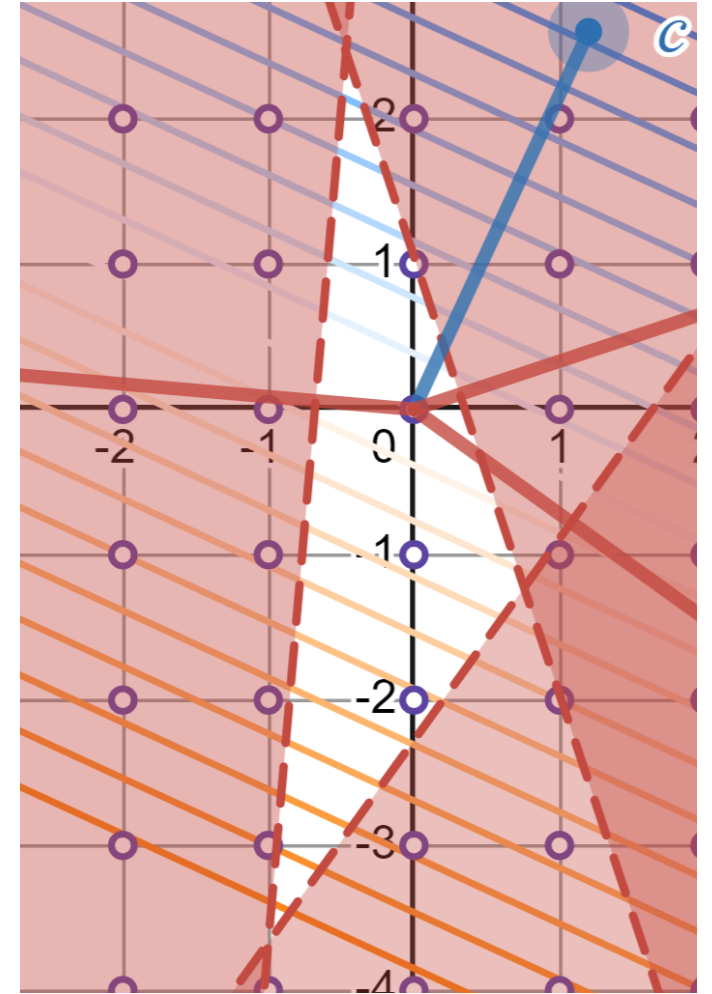
Notation Alert

Let y represent the objective value (total cost), $y = \mathbf{c}^\top \mathbf{x}$

Pay attention to argmin. vs just min.

$$\begin{array}{ll} x_{IP}^* = \underset{\mathbf{x}}{\operatorname{argmin.}} & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^N \end{array}$$

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



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 problem.

Which of the following are true?

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

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

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

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

$$\begin{array}{ll} y_{LP}^* = \min_{\mathbf{x}} & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \end{array}$$

Poll 3:

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

Solving an IP

Branch and Bound algorithm

Core steps:

Use LP solver to find $\mathbf{x}_{LP}^*$

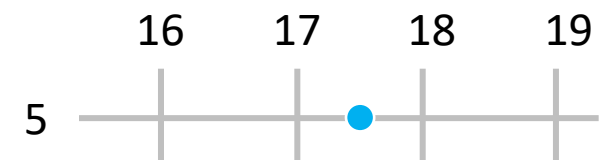
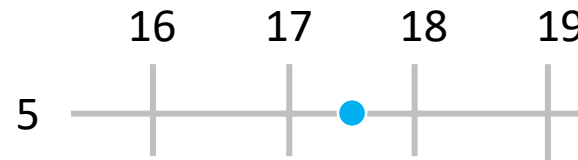
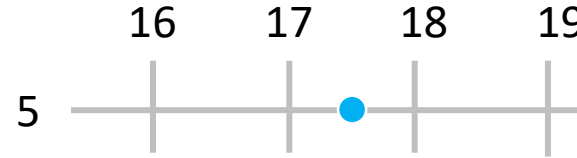
If $\mathbf{x}_{LP}^*$ is all integer valued,
return solution

Otherwise:

Create two branches

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

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



Solving an IP

Branch and Bound algorithm

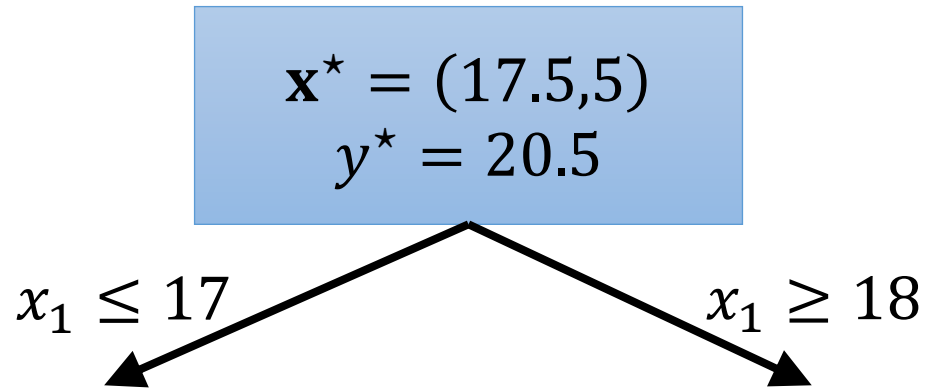
1. Push LP solution of problem into priority queue,
ordered by objective value of LP solution
2. Repeat:
 - If queue is empty, return IP is infeasible
 - Pop candidate solution $\mathbf{x}_{LP}^*$ from priority queue ()
 - If $\mathbf{x}_{LP}^*$ is all integer valued, we are done; return solution
 -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

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

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

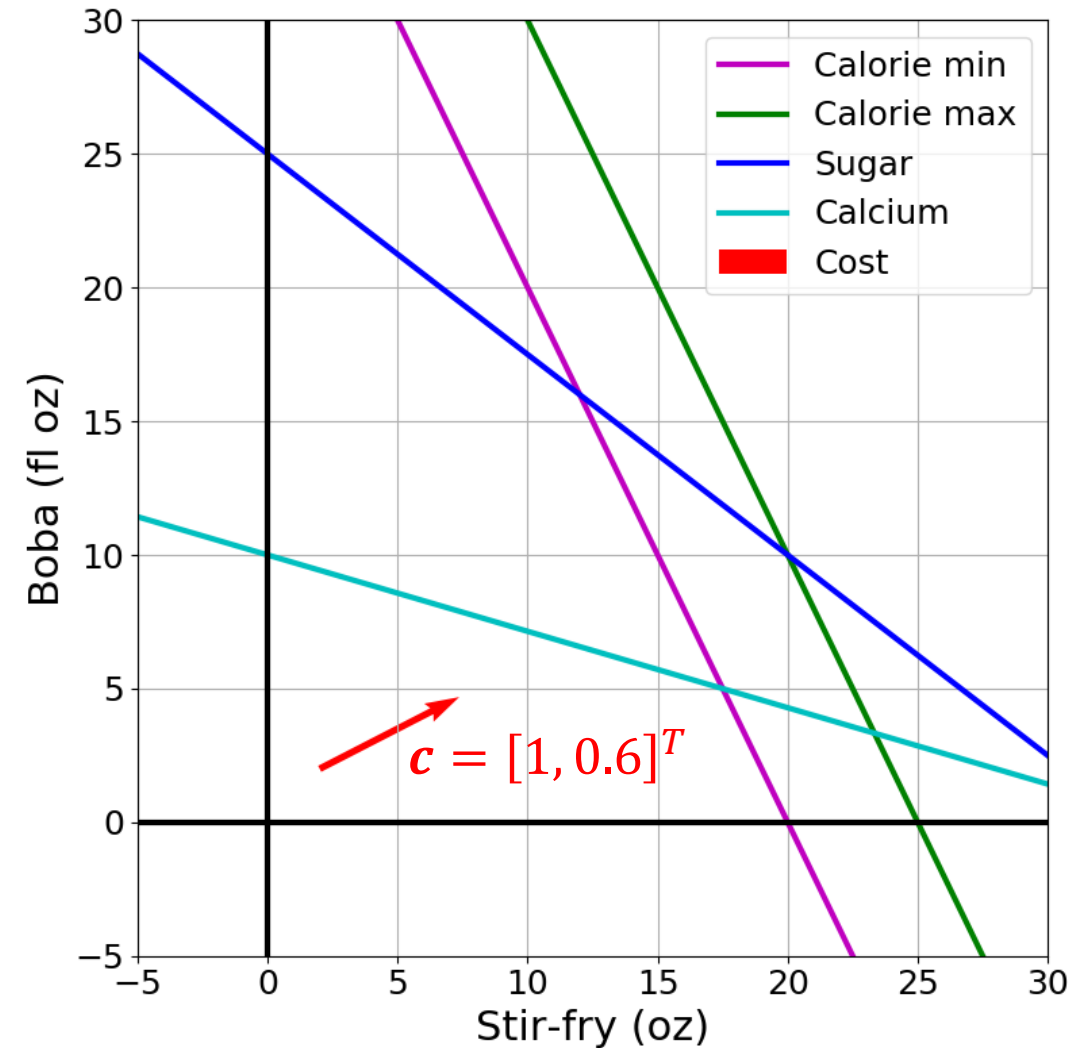
Note: Only add LPs to the queue if they are feasible

Branch and Bound Example

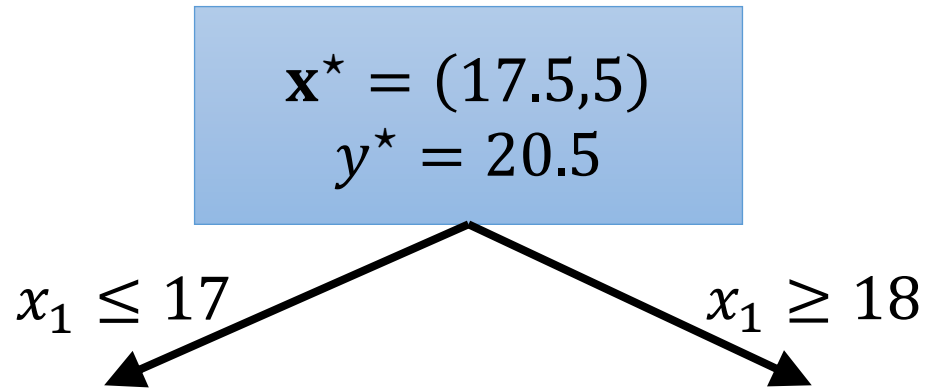


Priority Queue:

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

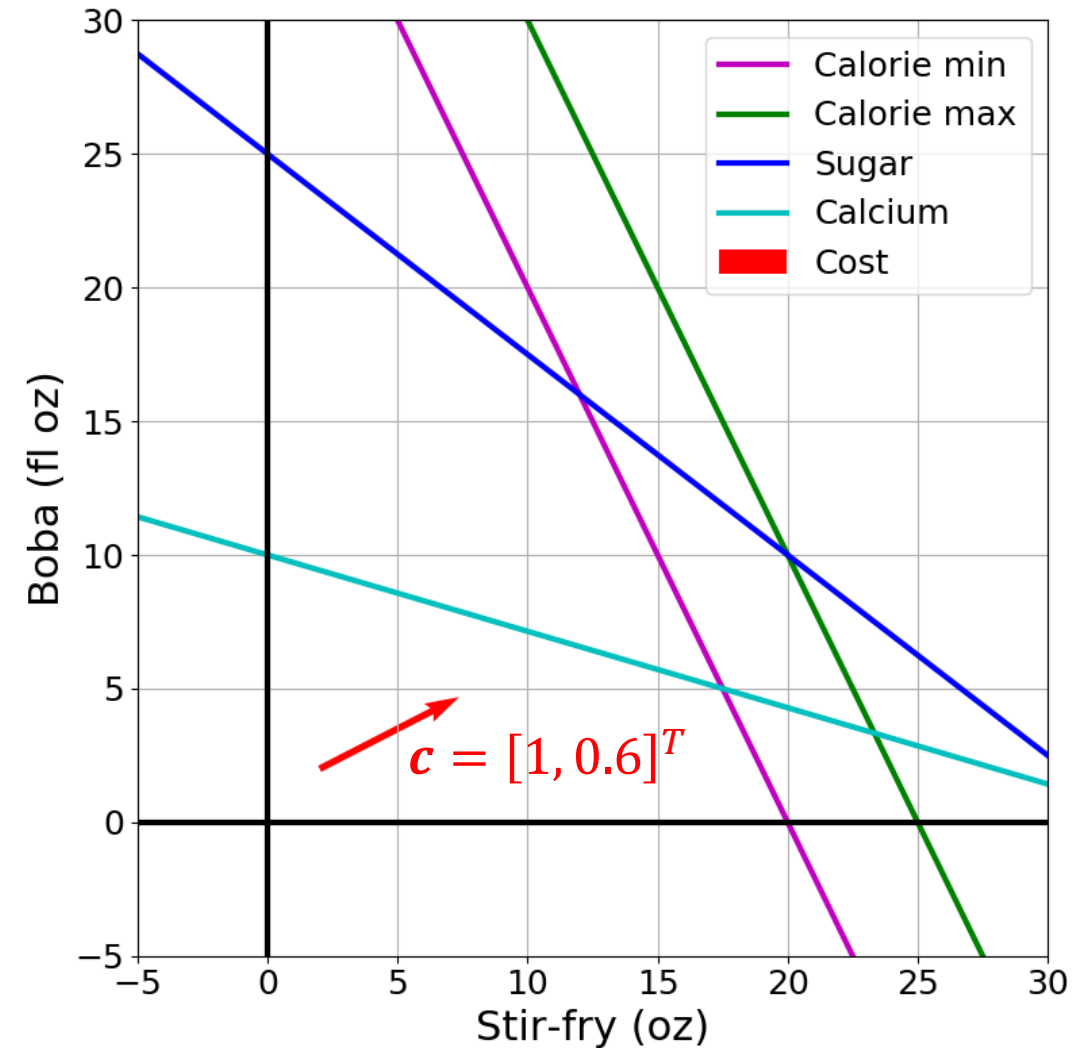


Branch and Bound Example

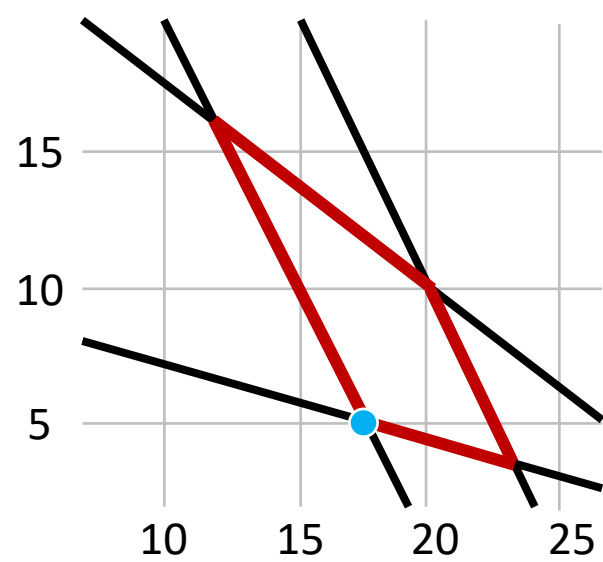


Priority Queue:

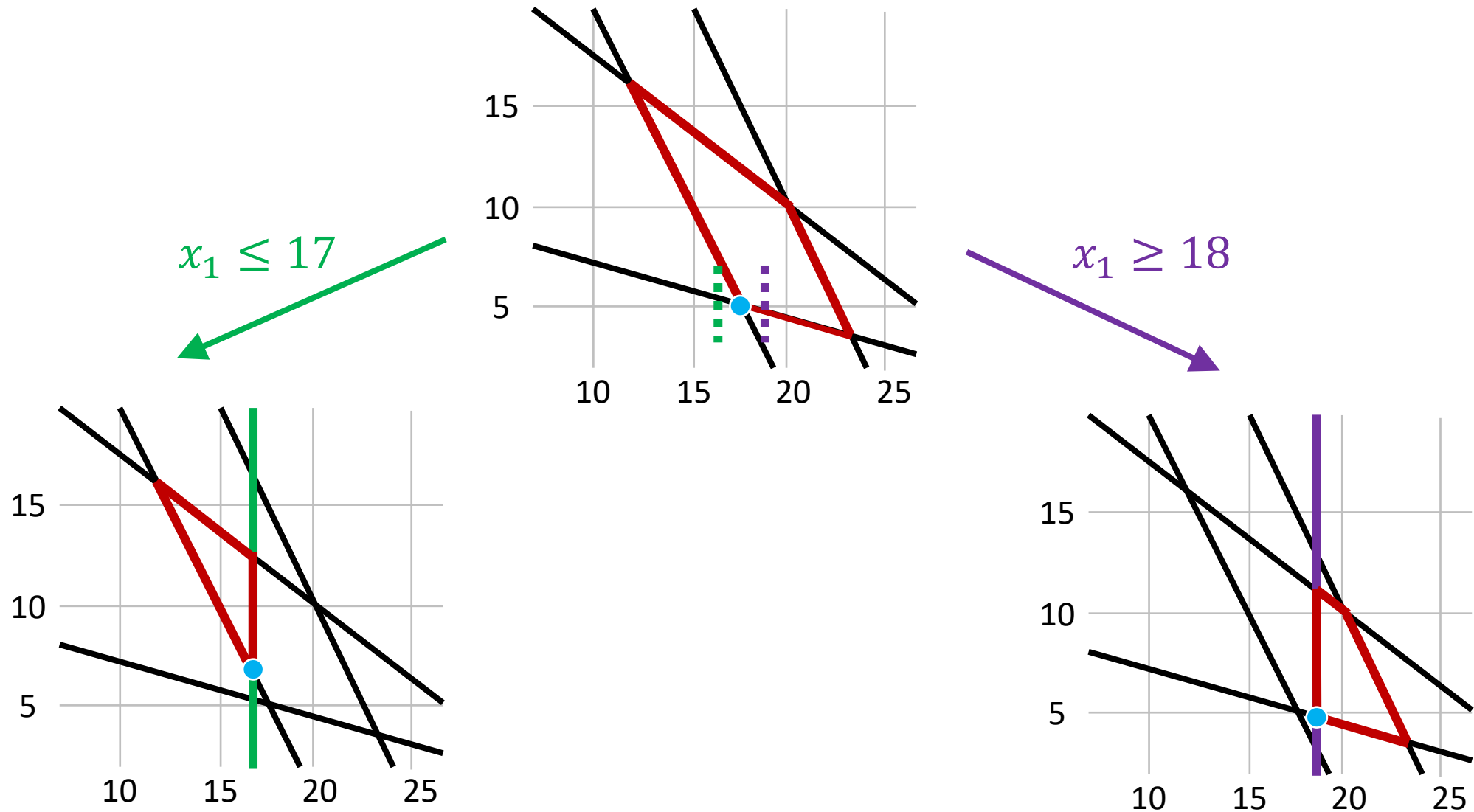
1. $\mathbf{x}^* = (17.5, 5), y^* = 20.5$



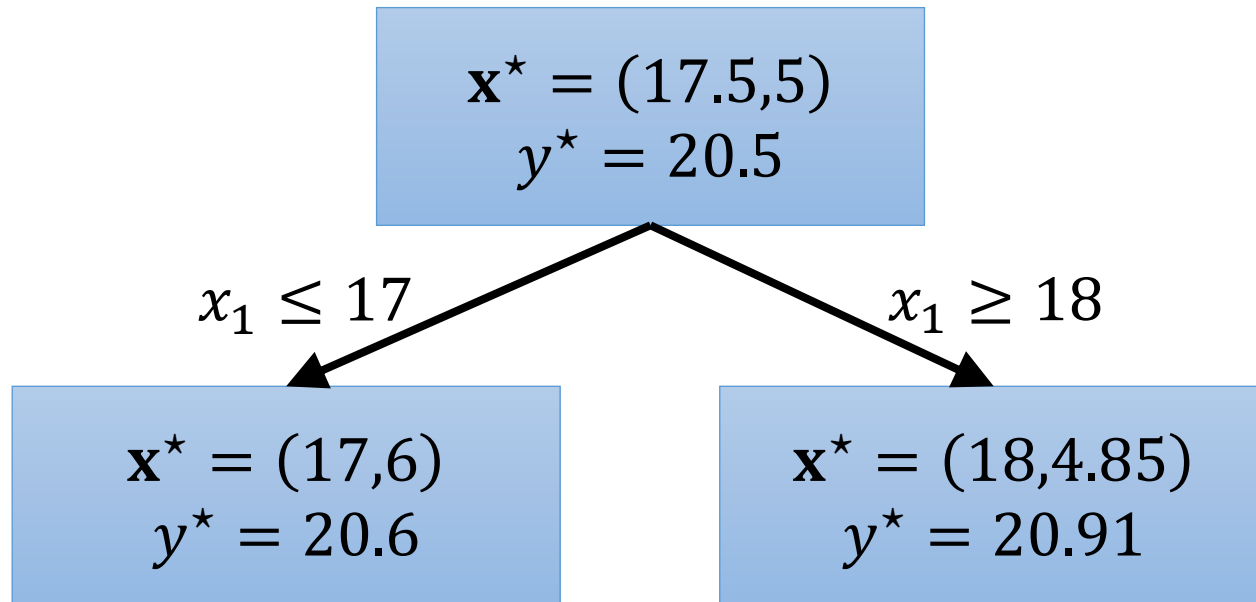
Branch and Bound Example



Branch and Bound Example



Branch and Bound Example



Priority Queue:

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

