# Gems of TCS

## Linear Programming

Sasha Golovnev

September 25, 2023

# Linear Programming

- Optimization problems: among all solutions satisfying certain constraints find optimal one

# Linear Programming

- Optimization problems: among all solutions satisfying certain constraints find optimal one
- Find shortest cycle through all vertices

# Linear Programming

- Optimization problems: among all solutions satisfying certain constraints find optimal one
- Find shortest cycle through all vertices
- Find optimal coloring

# Linear Programming

- Optimization problems: among all solutions satisfying certain constraints find optimal one
- Find shortest cycle through all vertices
- Find optimal coloring
- Find minimum vertex cover

# Linear Programming

- Optimization problems: among all solutions satisfying certain constraints find optimal one
- Find shortest cycle through all vertices
- Find optimal coloring
- Find minimum vertex cover
- Linear programming: class of optimization problems where constraints and optimization criterion are linear functions

# Avoiding Scurvy

- Orange costs \$1,
  grapefruit costs \$1;
  we have budget of \$2/day

- Orange costs $1,
  grapefruit costs $1;
  we have budget of $2/day
- Orange weighs 100gm,
  grapefruit weighs 200gm,
  we can carry 300gm

- Orange costs \$1,
  grapefruit costs \$1;
  we have budget of \$2/day
- Orange weighs 100gm,
  grapefruit weighs 200gm,
  we can carry 300gm
- Orange has 50gm of vitamin C,
  grapefruit has 75gm of vitamin C,
  maximize daily vitamin C intake.

# Avoiding Scurvy. Plot

$$\max 2x + 3y$$

$$x + y \leq 2$$

$$x + 2y \leq 3$$

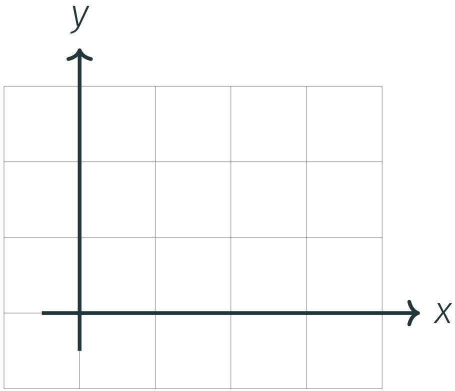
$$x \geq 0$$

$$y \geq 0$$

# Avoiding Scurvy. Plot

$$\max 2x + 3y$$

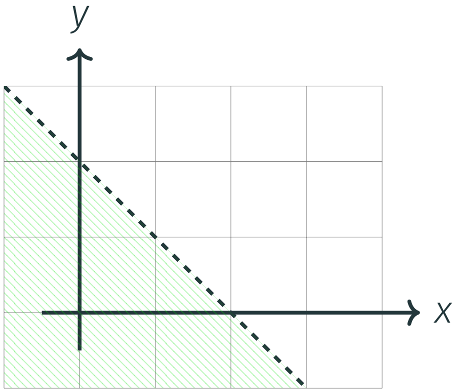
$$x + y \leq 2$$
$$x + 2y \leq 3$$
$$x \geq 0$$
$$y \geq 0$$



# Avoiding Scurvy. Plot

$$\max 2x + 3y$$

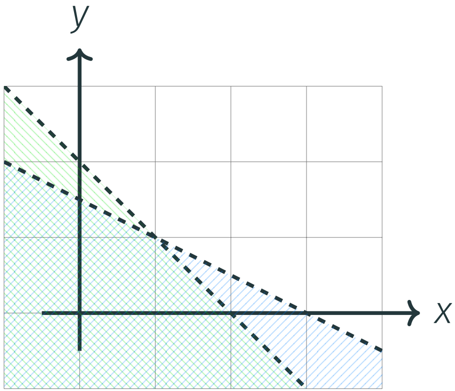
$$x + y \leq 2$$
$$x + 2y \leq 3$$
$$x \geq 0$$
$$y \geq 0$$



# Avoiding Scurvy. Plot

$$\max 2x + 3y$$

$$x + y \leq 2$$
$$x + 2y \leq 3$$
$$x \geq 0$$
$$y \geq 0$$



# Avoiding Scurvy. Plot

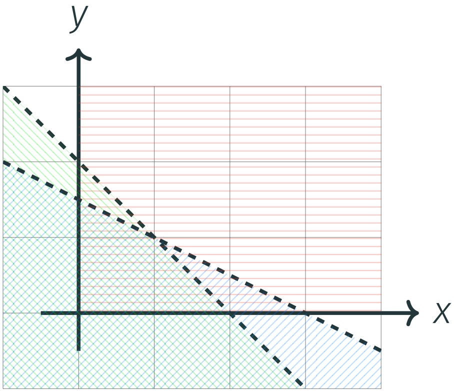
$\max 2x + 3y$

$x + y \leq 2$

$x + 2y \leq 3$

$x \geq 0$

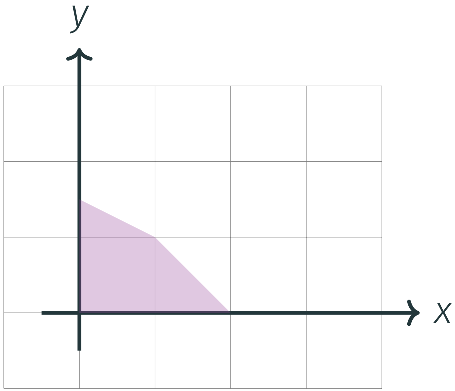
$y \geq 0$



# Avoiding Scurvy. Plot

$$\max 2x + 3y$$

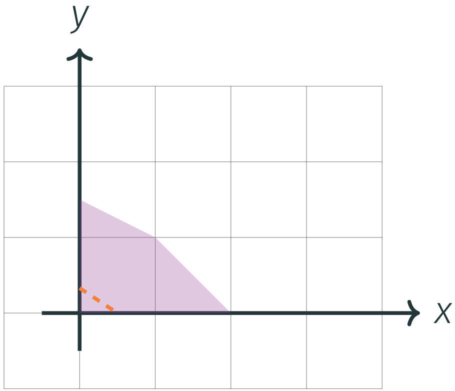
$$\begin{aligned} x + y &\leq 2 \\ x + 2y &\leq 3 \\ x &\geq 0 \\ y &\geq 0 \end{aligned}$$



# Avoiding Scurvy. Plot

$$\max 2x + 3y$$

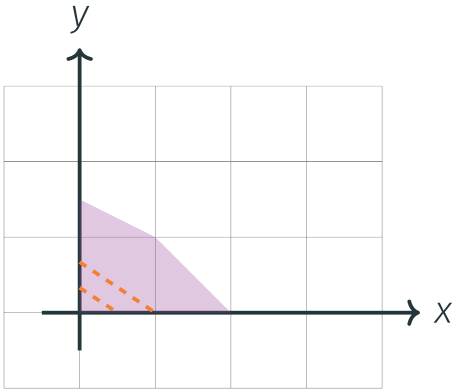
$$x + y \leq 2$$
$$x + 2y \leq 3$$
$$x \geq 0$$
$$y \geq 0$$



# Avoiding Scurvy. Plot

$$\max 2x + 3y$$

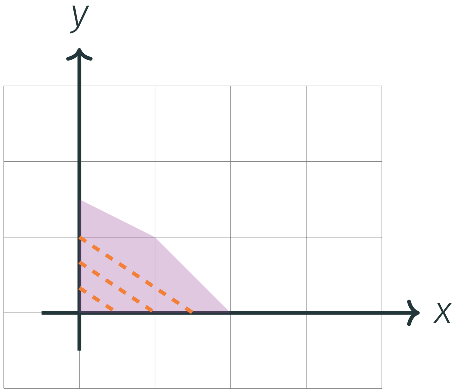
$$x + y \leq 2$$
$$x + 2y \leq 3$$
$$x \geq 0$$
$$y \geq 0$$



# Avoiding Scurvy. Plot

$$\max 2x + 3y$$

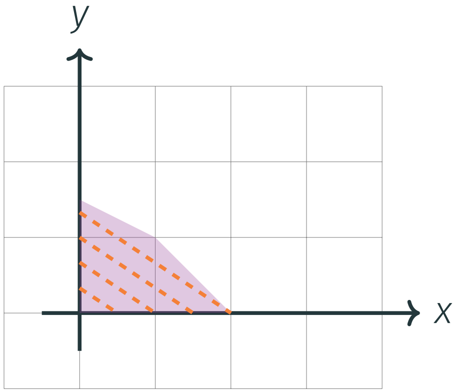
$$x + y \leq 2$$
$$x + 2y \leq 3$$
$$x \geq 0$$
$$y \geq 0$$



# Avoiding Scurvy. Plot

$$\max 2x + 3y$$

$$x + y \leq 2$$
$$x + 2y \leq 3$$
$$x \geq 0$$
$$y \geq 0$$



# Avoiding Scurvy. Plot

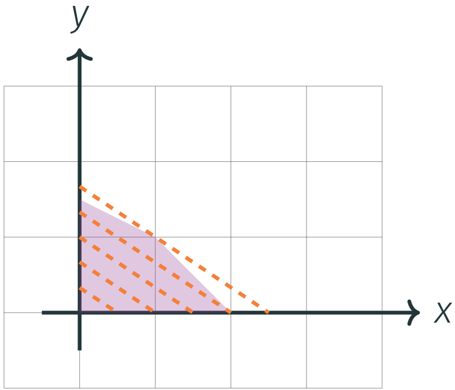
$$\max 2x + 3y$$

$$x + y \leq 2$$

$$x + 2y \leq 3$$

$$x \geq 0$$

$$y \geq 0$$



# Profit Maximization

# Profit Maximization

- We have 6 machines and 20 workers

# Profit Maximization

- We have 6 machines and 20 workers
- A machine takes two workers to operate

# Profit Maximization

- We have 6 machines and 20 workers
- A machine takes two workers to operate
- Each machine produces 20 chocolates/hour, each worker produces 5 chocolates/hour

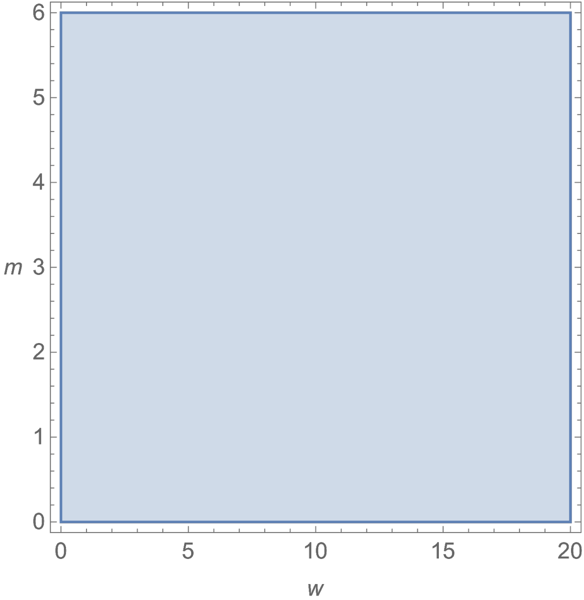
# Profit Maximization

- We have 6 machines and 20 workers
- A machine takes two workers to operate
- Each machine produces 20 chocolates/hour, each worker produces 5 chocolates/hour
- We need to produce at most 100 chocolates/hour

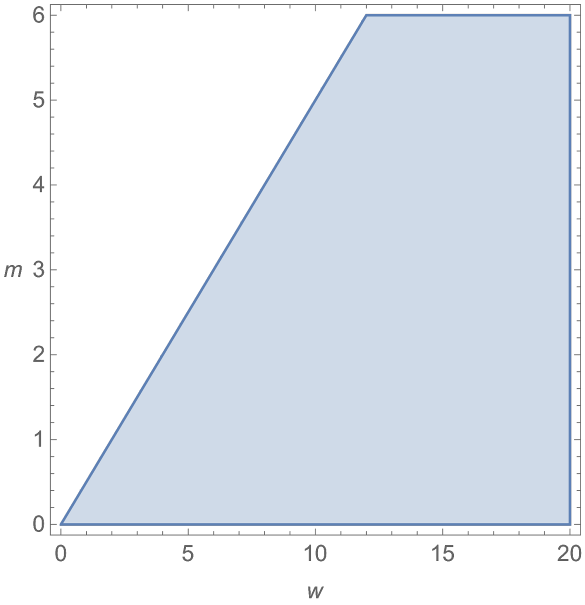
# Profit Maximization

- We have 6 machines and 20 workers
- A machine takes two workers to operate
- Each machine produces 20 chocolates/hour, each worker produces 5 chocolates/hour
- We need to produce at most 100 chocolates/hour
- Each chocolate costs \$10, each worker gets \$40 per hour

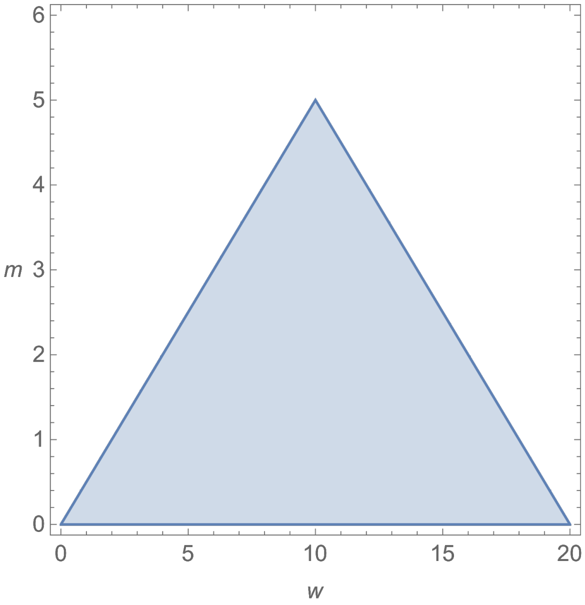
# Workers and Machines



# Two Workers Operate a Machine



# Chocolate Demand



# Linear Classifier

# Linear Classifier

- Given $n_1$ spam emails, and $n_2$ ham emails as points in $\mathbb{R}^d$

# Linear Classifier

- Given $n_1$ spam emails, and $n_2$ ham emails as points in $\mathbb{R}^d$
- Find a linear function $h(a_1, \ldots, a_d)$ s.t.

# Linear Classifier

- Given $n_1$ spam emails, and $n_2$ ham emails as points in $\mathbb{R}^d$

- Find a linear function $h(a_1, \ldots, a_d)$ s.t.
    - $h(a_1, \ldots, a_d) < 0$ for all spam emails
    - $h(a_1, \ldots, a_d) > 0$ for all ham emails

# Linear Programming

# Linear Programming

- Find real numbers $x_1, \ldots, x_n$ that satisfy linear constraints

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\geq b_2 \\ &\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\geq b_m \end{aligned}$$

# Linear Programming

- Find real numbers $x_1, \ldots, x_n$ that satisfy linear constraints

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\geq b_2 \\ &\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\geq b_m \end{aligned}$$

- So that linear objective is maximized

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

# Equivalent Formulations

- Turn minimization problem into maximization problem:

$$\min \quad c_1x_1 + c_2x_2 + \ldots - c_nx_n$$

# Equivalent Formulations

- Turn minimization problem into maximization problem:

$$\begin{aligned} \min &\quad c_1x_1 + c_2x_2 + \ldots - c_nx_n \\ \max &\quad -c_1x_1 - c_2x_2 - \ldots - c_nx_n \end{aligned}$$

# Equivalent Formulations

- Turn minimization problem into maximization problem:

$$\begin{aligned} \textbf{min} \quad & c_1x_1 + c_2x_2 + \ldots - c_nx_n \\ \textbf{max} \quad & -c_1x_1 - c_2x_2 - \ldots - c_nx_n \end{aligned}$$

- Turn $\leq$ into $\geq$:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1$$

# Equivalent Formulations

- Turn minimization problem into maximization problem:

$$\begin{aligned} \textbf{min} \quad & c_1x_1 + c_2x_2 + \ldots - c_nx_n \\ \textbf{max} \quad & -c_1x_1 - c_2x_2 - \ldots - c_nx_n \end{aligned}$$

- Turn $\leq$ into $\geq$:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\leq b_1 \\ -a_{11}x_1 - a_{12}x_2 - \ldots - a_{1n}x_n &\geq -b_1 \end{aligned}$$

# Equivalent Formulations

- Turn $=$ into $\geq$:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$

# Equivalent Formulations

- Turn $=$ into $\geq$:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\geq b_1 \\
-a_{11}x_1 - a_{12}x_2 - \ldots - a_{1n}x_n &\geq -b_1
\end{aligned}
$$

# Matrix Formulation

Input is a matrix $A \in \mathbb{R}^{m \times n}$, and vectors $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$

# Matrix Formulation

Input is a matrix $A \in \mathbb{R}^{m \times n}$, and vectors $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$

$$Ax = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \vdots & \dots \\ \dots & \vdots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + & \dots & +a_{1n}x_n \\ \dots & \vdots & \dots \\ \dots & \vdots & \dots \\ a_{m1}x_1 + & \dots & +a_{mn}x_n \end{bmatrix} \geq \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ b_m \end{bmatrix}$$

# Matrix Formulation

Input is a matrix $A \in \mathbb{R}^{m \times n}$, and vectors $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$

$$Ax = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \vdots & \dots \\ \dots & \vdots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + & \dots & +a_{1n}x_n \\ \dots & \vdots & \dots \\ \dots & \vdots & \dots \\ a_{m1}x_1 + & \dots & +a_{mn}x_n \end{bmatrix} \geq \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ b_m \end{bmatrix}$$

$$Ax \geq b$$

# Matrix Formulation

Input is a matrix $A \in \mathbb{R}^{m \times n}$, and vectors $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$

$$Ax = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \vdots & \dots \\ \dots & \vdots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + & \dots & +a_{1n}x_n \\ \dots & \vdots & \dots \\ \dots & \vdots & \dots \\ a_{m1}x_1 + & \dots & +a_{mn}x_n \end{bmatrix} \geq \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ b_m \end{bmatrix}$$

$$Ax \geq b$$

$$\text{maximize } cx = \begin{bmatrix} c_1 & \dots & c_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = c_1x_1 + \dots c_nx_n$$

# History of Linear Programming

- Kantorovich, 1939, started studying Linear Programming

# History of Linear Programming

- Kantorovich, 1939, started studying Linear Programming
- Dantzig, 1947, developed Simplex Method for US Air force planning problems

# History of Linear Programming

- Kantorovich, 1939, started studying Linear Programming
- Dantzig, 1947, developed Simplex Method for US Air force planning problems
- Koopmans, 1947, showed how to use LP for analysis of economic theories

# History of Linear Programming

- Kantorovich, 1939, started studying Linear Programming
- Dantzig, 1947, developed Simplex Method for US Air force planning problems
- Koopmans, 1947, showed how to use LP for analysis of economic theories
- Kantorovich and Koopmans won Nobel Prize in Economics in 1971

# History of Linear Programming

- Kantorovich, 1939, started studying Linear Programming
- Dantzig, 1947, developed Simplex Method for US Air force planning problems
- Koopmans, 1947, showed how to use LP for analysis of economic theories
- Kantorovich and Koopmans won Nobel Prize in Economics in 1971
- Dantzig's algorithm is "One of top 10 algorithms of the 20th century"

**Theorem**

A linear function takes its maximum and minimum values on vertices

# Simplex Method

**Theorem**

A linear function takes its maximum and minimum values on vertices

- Start at any vertex

# Simplex Method

**Theorem**

A linear function takes its maximum and minimum values on vertices

- Start at any vertex
- While there is an adjacent vertex with higher profit
    - Move to that vertex

# Corner Cases

- No solutions

# Corner Cases

- No solutions

- Unbounded profit

# Algorithms for Linear Programming

- Simplex method

# Algorithms for Linear Programming

- Simplex method
- Many professional packages that implement efficient algorithms for LP

# Algorithms for Linear Programming

- Simplex method
- Many professional packages that implement efficient algorithms for LP
- Ellipsoid method

# Algorithms for Linear Programming

- Simplex method
- Many professional packages that implement efficient algorithms for LP
- Ellipsoid method
- Projective algorithm

## Algorithms for Linear Programming

- Simplex method
- Many professional packages that implement efficient algorithms for LP
- Ellipsoid method
- Projective algorithm
- Recent results!

# Ellipsoid Method