GEMS OF TCS

LINEAR PROGRAMMING

Sasha Golovnev
February 23, 2021
LINEAR PROGRAMMING

- Optimization problems: among all solutions satisfying certain constraint find optimal one
LINEAR PROGRAMMING

- Optimization problems: among all solutions satisfying certain constraints find optimal one
- Find shortest cycle through all vertices TSP
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 **maximum vertex color covers**
LINEAR PROGRAMMING

• Optimization problems: among all solutions satisfying certain constraints find optimal one
• Find shortest cycle through all vertices
• Find optimal coloring
• Find maximum vertex color
• 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 100gm of vitamin C, grapefruit has 150gm of vitamin C, maximize daily vitamin C intake.
Orange costs $1, grapefruit costs $1; we have budget of $2/day.

Orange weighs 100gm, grapefruit weighs 200gm, we can carry 300gm.

Orange has 100gm of vitamin C, grapefruit has 150gm of vitamin C, maximize daily vitamin C intake.

\[
\begin{align*}
\text{Constraints} \\
x \geq 0 \\
y \geq 0 \\
x + y \leq 2 \\
100x + 200y \leq 300 \\
x + 2y \leq 3
\end{align*}
\]

Maximize \(100x + 150y\)

Maximize \(2x + 3y\)

OPT SOL: \(x = y = 1\)

\[
\frac{2x + 3y}{100x + 150y} = 250 \text{ gm}
\]
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
• 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

\[
\begin{align*}
2m \text{ workers} & \quad \text{operate machines} \\
(w-2m) \text{ workers} & \quad \text{make chocolate without machines}
\end{align*}
\]

How many chocolates/hour?

\[
m \cdot 20 + (w-2m) \cdot 5 = 10m + 5w
\]

Profit?

\[
(10m+5w) \cdot 10 - 40w = 100m + 10w
\]
WORKERS AND MACHINES

\[ w > 0 \]
\[ m > 0 \]
\[ m \leq 6 \]
\[ w \leq 20 \]
\[ w \geq 2m \]
\[ 10m + 5w \leq 100 \]
TWO WORKERS OPERATE A MACHINE

\[ w \geq 0 \]
\[ m \geq 0 \]
\[ m \leq 6 \]
\[ w \leq 20 \]
\[ w \geq 2m \]
\[ 10m + 5w \leq 1 \]
Chocolate Demand

\[ \text{Max} \ 100m + 10w \]

Constraints:

\[ w > 0 \]
\[ m > 0 \]
\[ m \leq 6 \]
\[ w \leq 20 \]
\[ w \geq 2m \]
\[ 10m + 5w \leq 100 \]

Profit levels:

- Profit = 100
- Profit = 50
- Profit = 200

Graph showing the feasible region for \( w \) and \( m \) with different profit levels.
\[ w = 10 \quad m = 5 \]

\[ 100m + 10w = 600 \]

\[ w \leq 20 \]
\[ w \geq 2m \]
\[ 10m + 5w \leq 100 \]

\[ 6(10m + 5w) + 20 \cdot 2m \leq 100 \cdot 6 + 20 \cdot w \]

\[ 60m + 30w + 40m \leq 600 + 20w \]

\[ 100m + 10w \leq 600 \]
Linear Classifier
• Given \( n_1 \) spam emails, and \( n_2 \) ham emails as points in \( \mathbb{R}^d \)

\[
\begin{align*}
\text{each email} & \quad \text{- point in plane} \\
ax + by + c &= 0 & a, b, c \text{ are constants} \\
3x - 3.5y + 7 &= 0
\end{align*}
\]
Want to find $a, b, c$

define line $ax + by + c = 0$

\[
\begin{align*}
ax_1 + by_1 + c & \leq -\delta \\
ax_2 + by_2 + c & \geq \delta \\
ax_3 + by_3 + c & \leq -\delta \\
\end{align*}
\]

max $\delta$

emails in $\mathbb{R}^d$ where $d \approx 10^4$
**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

\[
\begin{align*}
\sum_{i=1}^{n} a_{1i} x_i + a_{12} x_2 + \ldots + a_{1n} x_n & \geq b_1 \\
\sum_{i=1}^{n} a_{2i} x_i + a_{22} x_2 + \ldots + a_{2n} x_n & \geq b_2 \\
& \quad \ldots \\
\sum_{i=1}^{n} a_{mi} x_i + a_{m2} x_2 + \ldots + a_{mn} x_n & \geq b_m
\end{align*}
\]

\( n \text{ variables } \quad x_1, \ldots, x_n \)
**Linear Programming**

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

\[
\begin{align*}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & \geq b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n & \geq b_2 \\
& \quad \ldots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n & \geq b_m
\end{align*}
\]

- So that linear objective is maximized

\[
c_1x_1 + c_2x_2 + \ldots + c_nx_n
\]

\[\rightarrow x_1 + 2x_2 \quad -10x_6\]
**EQUIVALENT FORMULATIONS**

- Turn **minimization** problem into **maximization** problem:

  \[
  \begin{align*}
  \text{min} & \quad c_1x_1 + c_2x_2 + \ldots - c_nx_n \\
  \text{max} & \quad -c_1x_1 - c_2x_2 - \ldots - c_nx_n
  \end{align*}
  \]
**EQUIVALENT FORMULATIONS**

- Turn minimization problem into maximization problem:

  \[
  \min c_1x_1 + c_2x_2 + \ldots - c_nx_n \\
  \max -c_1x_1 - c_2x_2 - \ldots - c_nx_n
  \]

- Turn \(\leq\) into \(\geq\):

  \[
  a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1 \\
  -a_{11}x_1 - a_{12}x_2 - \ldots - a_{1n}x_n \geq -b_1
  \]
Equivalent Formulations

- Turn $=$ into $\geq$:

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

\[ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \geq b_1 \]

\[ -a_{11}x_1 - a_{12}x_2 - \ldots - a_{1n}x_n \geq -b_1 \]

Max Flow - another example of LP
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 $c \in \mathbb{R}^n$

$$Ax = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 & \ldots & a_{1n}x_n \\ \vdots & \ddots & \vdots \\ a_{m1}x_1 & \ldots & a_{mn}x_n \end{bmatrix} \geq \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$
Matrix Formulation

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

$Ax = \begin{bmatrix}
a_{11} & \ldots & a_{1n} \\
\vdots & \ddots & \vdots \\
a_{m1} & \ldots & a_{mn}
\end{bmatrix} \begin{bmatrix}
x_1 \\
\vdots \\
x_n
\end{bmatrix} = \begin{bmatrix}
a_{11}x_1 & \ldots & a_{1n}x_n \\
\vdots & \ddots & \vdots \\
a_{m1}x_1 & \ldots & a_{mn}x_n
\end{bmatrix} \geq \begin{bmatrix}
b_1 \\
\vdots \\
b_m
\end{bmatrix}$
**Matrix Formulation**

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

\[
Ax = \begin{bmatrix}
a_{11} & \ldots & a_{1n} \\
\vdots & \ddots & \vdots \\
a_{m1} & \ldots & a_{mn}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
\vdots \\
x_n
\end{bmatrix}
= \begin{bmatrix}
a_{11}x_1 & \ldots & a_{1n}x_n \\
\vdots & \ddots & \vdots \\
a_{m1}x_1 & \ldots & a_{mn}x_n
\end{bmatrix}
\geq \begin{bmatrix}
b_1 \\
\vdots \\
b_m
\end{bmatrix}
\]

(i) \( Ax \geq b \)

(ii) maximize \( cx = \begin{bmatrix}
c_1 & \ldots & c_n
\end{bmatrix}
\begin{bmatrix}
x_1 \\
\vdots \\
x_n
\end{bmatrix}
= c_1x_1 + \ldots + c_nx_n \)
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
**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

\[
\begin{align*}
    x &> 2 \\
    y &> 1 \\
    x + y &\leq 3
\end{align*}
\]
CORNER CASES

- No solutions
- Unbounded profit

\[ x + y \geq 7 \]

\[ \text{max } y \]
**Algorithms for Simplex Method**

- quite efficient in practice

- Simplex method

\[
0 \leq x_1 \leq 1 \\
0 \leq x_2 \leq 1 \\
\vdots \\
0 \leq x_n \leq 1
\]

in n dimensions

\[2^n \text{ vertices}\]

Known examples where simple method will have to check \(\approx 2^n\) vertices
ALGORITHMS FOR SIMPLEX METHOD

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

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

Khachiyan, 1979

Run-time \( n^7 \)

non really practical
ALGORITHMS FOR SIMPLEX METHOD

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

Karmarkar
runtime $\approx n^{3.5}$
ALGORITHMS FOR SIMPLEX METHOD

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

- Last week! [JSWZ'21] essentially solves LP in $n^2$ time
ELLIPSOID METHOD