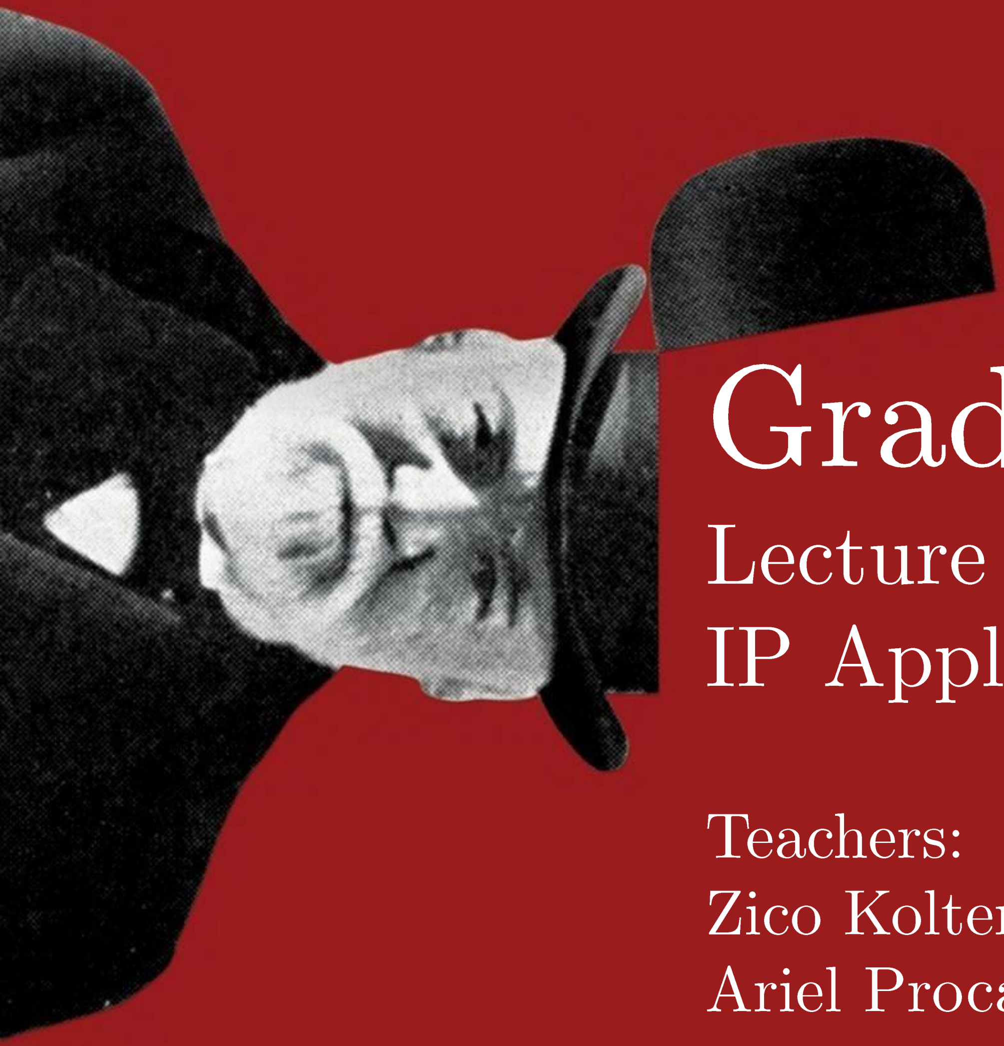


Graduate AI

Lecture 7: IP Applications

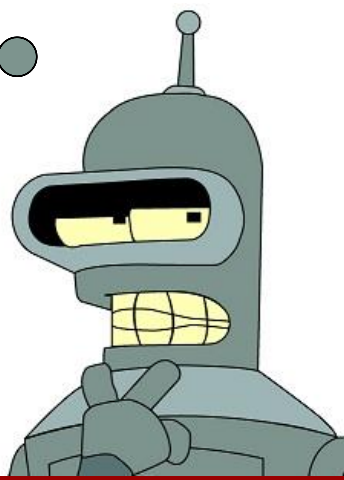
Teachers:
Zico Kolter
Ariel Procaccia (this time)

Integer Programming

- An integer programming (IP) problem:
  - $a_{ij} \in \mathbb{R}$ for $i \in [k] = \{1, \ldots, k\}, j \in [\ell]$
  - $b_i \in \mathbb{R}$ for $i \in [k]$
  - Variables x_j for $j \in [\ell]$
- The (feasibility) problem is:

$$\begin{aligned}
&\text{find} \quad x_1 \ldots, x_\ell \\
&\text{s.t.} \quad \forall i \in [k],\ \textstyle\sum_{j=1}^{\ell} a_{ij} x_j \le b_i \\
&\qquad\ \ \forall j \in [\ell],\ x_j \in \mathbb{Z}
\end{aligned}$$

How can we express $\geq$ constraints?
Equality constraints?
Restricted domains?



Example: Sudoku

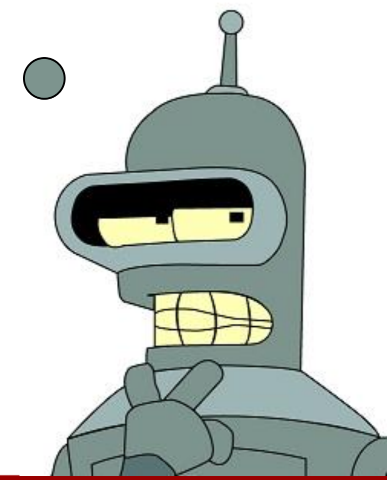
8			4		6			7
						4		
	1					6	5	
5		9		3		7	8	
				7				
	4	8		2		1		3
	5	2					9	
		1						
3			9		2			5

Example: Sudoku

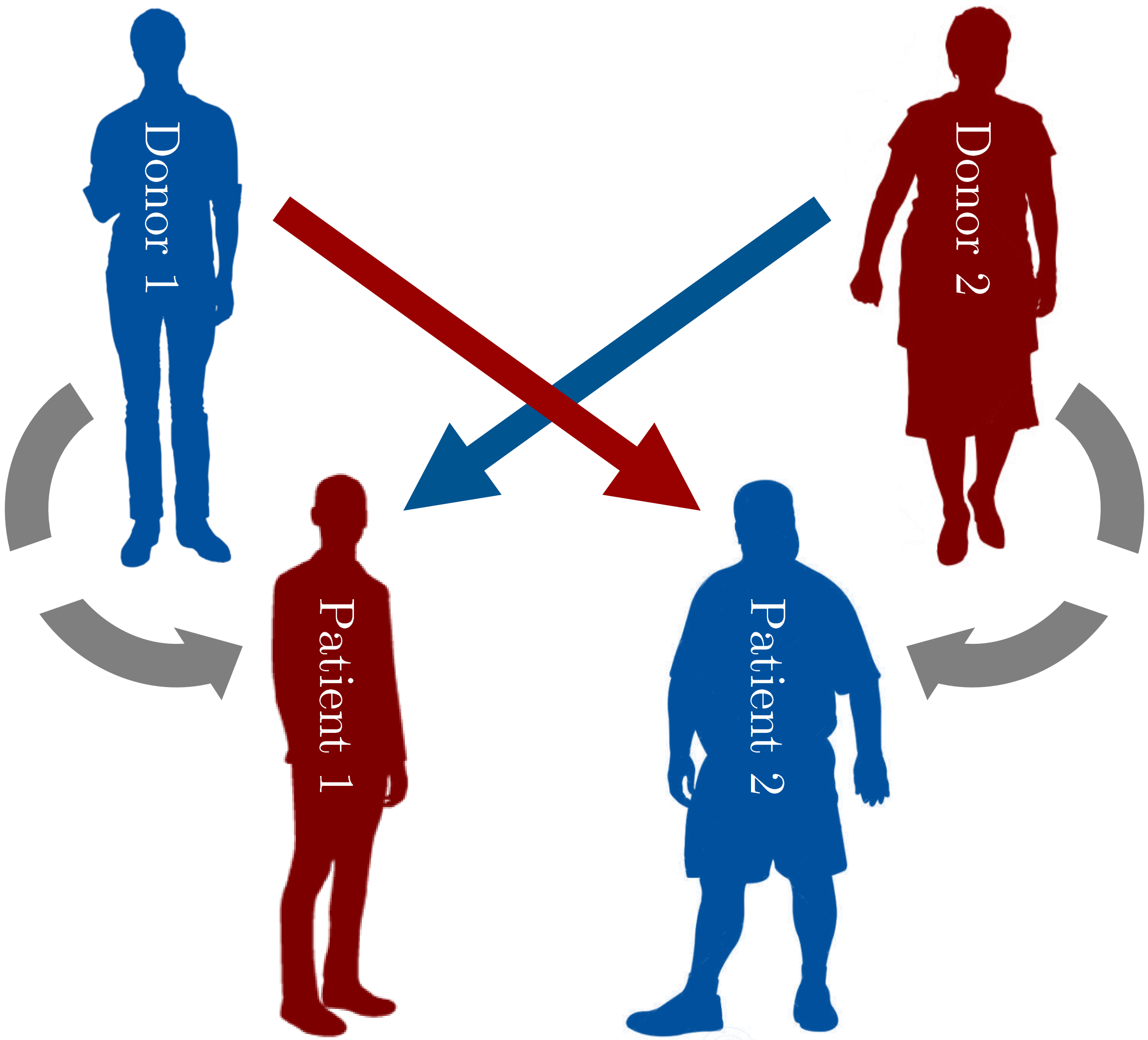
- For each $i, j, k \in [9]$, binary variable x_k^{ij} s.t. $x_k^{ij} = 1$ iff we put k in entry (i, j)
- For $\mathrm{t} = 1, \ldots, 27$, S_t is a row, column, or 3×3 square

$$
\begin{aligned}
\text{find}\quad & x_1^{11}, \ldots, x_9^{99} \\
\text{s.t.}\quad & \forall t \in [27], \forall k \in [9],\ \textstyle\sum_{(i,j)\in S_t} x_k^{ij} = 1 \\
& \forall i, j \in [9], \textstyle\sum_{k\in[9]} x_k^{ij} = 1 \\
& \forall i, j, k \in [9],\ x_k^{ij} \in \{0,1\}
\end{aligned}
$$

If you have a hard time expressing something as an IP, try using binary variables

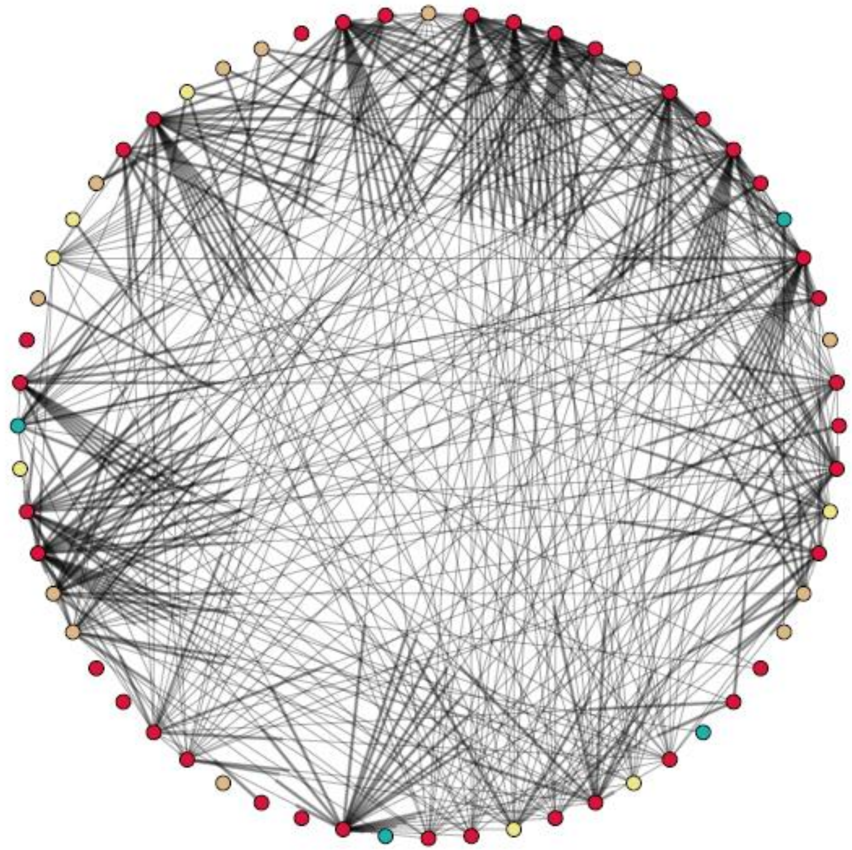


Example: Kidney Exchange



Example: Kidney Exchange

- Cycle-Cover: Given a directed graph G and $L \in \mathbb{N}$, find a collection of disjoint cycles of length $\leq L$ in G that maximizes the number of covered vertices
- The problem is:
  - Easy for $L = 2$ (why?)
  - Easy for unbounded L
  - NP-hard for a constant $L \geq 3$



UNOS pool, Dec 2010
[Courtesy John Dickerson]

Example: Kidney Exchange

- Variables: For each cycle c of length $\ell_c \leq L$, variable $x_c \in \{0,1\}$, $x_c = 1$ iff cycle c is included in the cover
- Cycle-Cover as an IP:

$$
\begin{aligned}
\max\ & \textstyle\sum_c x_c \ell_c \\
\text{s.t.}\ & \forall v \in V,\ \textstyle\sum_{c:v \in c} x_c \leq 1 \\
& \forall c,\ x_c \in \{0,1\}
\end{aligned}
$$

Application: UNOS

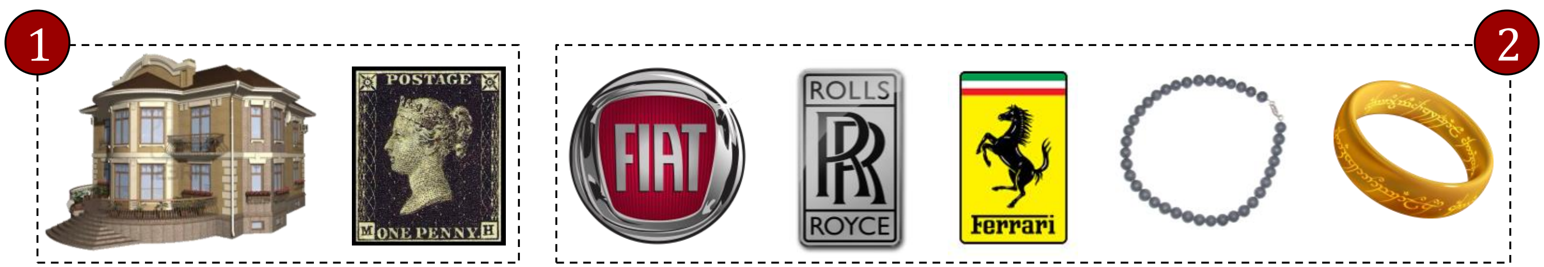
UNOS

DONATE LIFE

UNITED NETWORK FOR ORGAN SHARING

Example: Envy-Freeness

- Players $N = \{1, \ldots, n\}$ and items $M = \{1, \ldots, m\}$
- Player i has value v_{ij} for item j
- Partition items to bundles $A_1, \ldots, A_n$
- $A_1, \ldots, A_n$ is envy-free iff $\forall i, i', \sum_{j \in A_i} v_{ij} \geq \sum_{j \in A_{i'}} v_{ij}$



1	\$30	\$50	\$2	\$5	\$5	\$3	\$5
2	\$2	\$10	\$5	\$20	\$20	\$3	\$40

Example: Envy-Freeness

- Variables: $x_{ij} \in \{0,1\}$, $x_{ij} = 1$ iff $j \in A_i$
- Envy-Free as an IP:

$$
\begin{aligned}
&\text{find } x_{11}, \ldots, x_{nm} \\
&\text{s.t. } \forall i \in N, \forall i' \in N, \textstyle\sum_{j \in M} v_{ij} x_{ij} \geq \sum_{j \in M} v_{ij} x_{i'j} \\
&\qquad \forall j \in M, \ \textstyle\sum_{i \in N} x_{ij} = 1 \\
&\qquad \forall i \in N, j \in M, x_{ij} \in \{0,1\}
\end{aligned}
$$

- Problem: An EF allocation may not exist

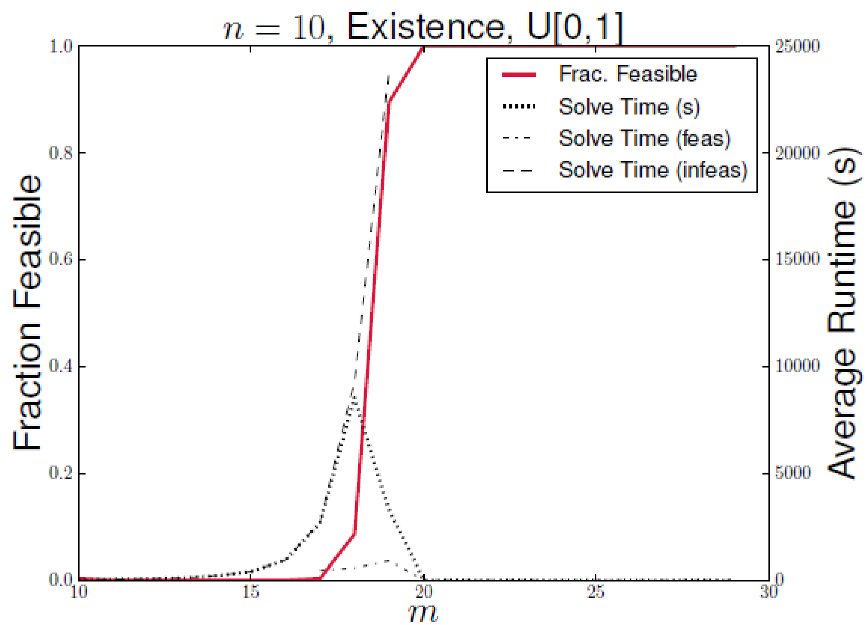
Phase Transition

- Imagine the v_{ij} are drawn independently and uniformly at random from [0,1]
- Poll 1: If $m = n/2$, what is the probability that an envy-free allocation exists?
  1. 0
  2. $2/n$
  3. $1/2$
  4. 1

Phase Transition

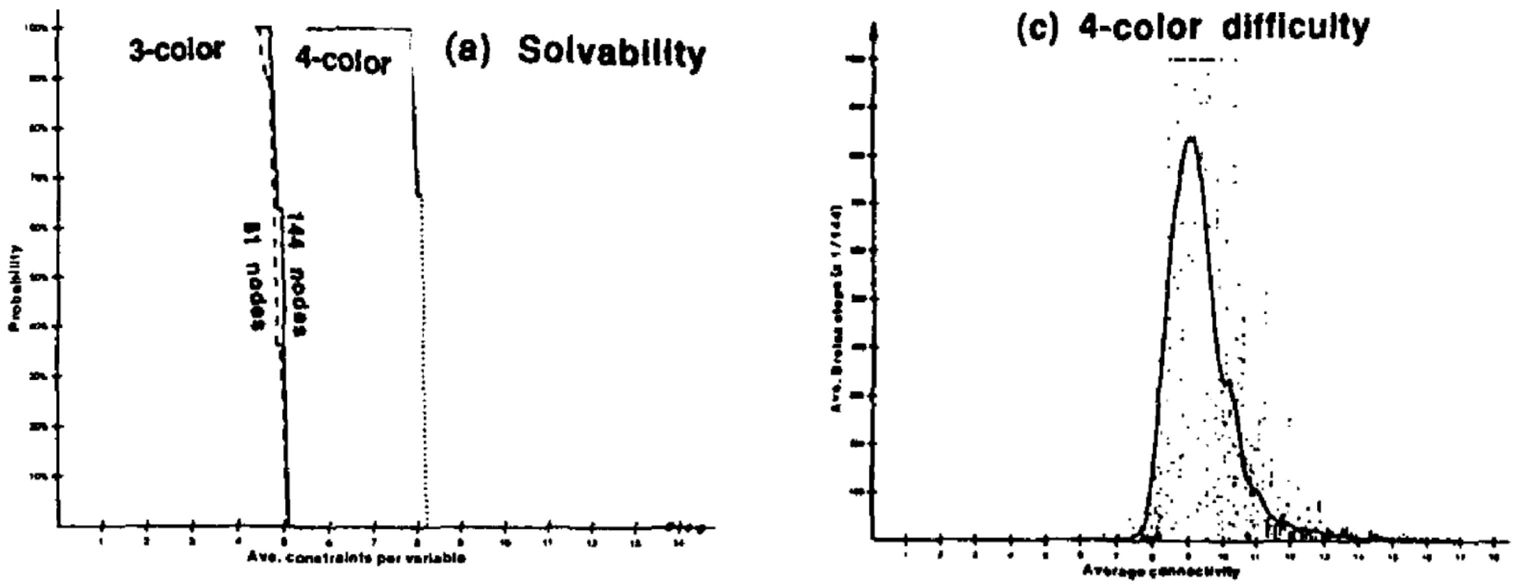
- Imagine the v_{ij} are drawn independently and uniformly at random from [0,1]
- Poll 2: If $m \gg n$, what is the probability that an envy-free allocation exists?
  1. Close to 0
  2. Close to 1/3
  3. Close to 1/2
  4. Close to 1

Sharp Transition



[Dickerson et al., AAAI 2014]

Sharp Transition



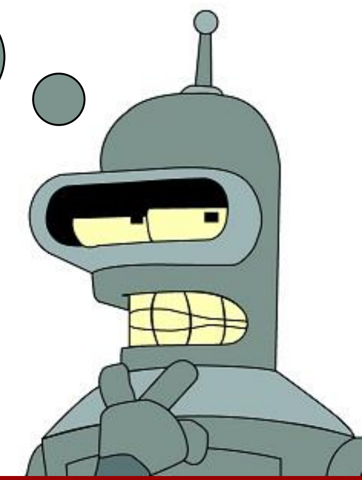
[Cheeseman et al., IJCAI 1993]

Example: MMS Guarantee

- Maximin share (MMS) guarantee [Budish 2011] of player i: $\max_{X_1,\ldots,X_n} \min_k v_i(X_k)$
- MMS guarantee of player i as IP:

$$
\begin{aligned}
&\max D \\
&\text{s.t.} \quad \forall k \in N,\ \ \textstyle\sum_{j \in M} v_{ij} y_{jk} \geq D \\
&\qquad\ \ \forall j \in M,\ \ \textstyle\sum_{k=1}^{n} y_{jk} = 1 \\
&\qquad\ \ \forall j \in M, k \in N,\ \ y_{jk} \in \{0,1\}
\end{aligned}
$$

Max min and min max can be expressed using a linear objective function and linear constraints!



Example: MMS Guarantee

- Suppose we computed $MMS(i)$ for each i
- Now finding an MMS allocation, where $v_i(A_i) \geq MMS(i)$ for all $i \in N$, is just another IP:

$$
\begin{aligned}
&\text{find } x_{11}, \dots, x_{nm} \\
&\text{s.t. } \forall i \in N,\ \textstyle\sum_{j \in M} v_{ij} x_{ij} \geq MMS(i) \\
&\qquad \forall j \in M,\ \textstyle\sum_{i \in N} x_{ij} = 1 \\
&\qquad \forall i \in N, j \in M, x_{ij} \in \{0,1\}
\end{aligned}
$$

Application: Spliddit

Other IPs: Coming Soon

Dodgson's
voting rule

Stackelberg
security games

Summary

- IP tricks:
  - Binary variables
  - Max min and min max
- Big ideas:
  - IP representation leads to "efficient" solutions
  - Phase transition ⇔ complexity