Announcements

Midterm1

- Grade Released

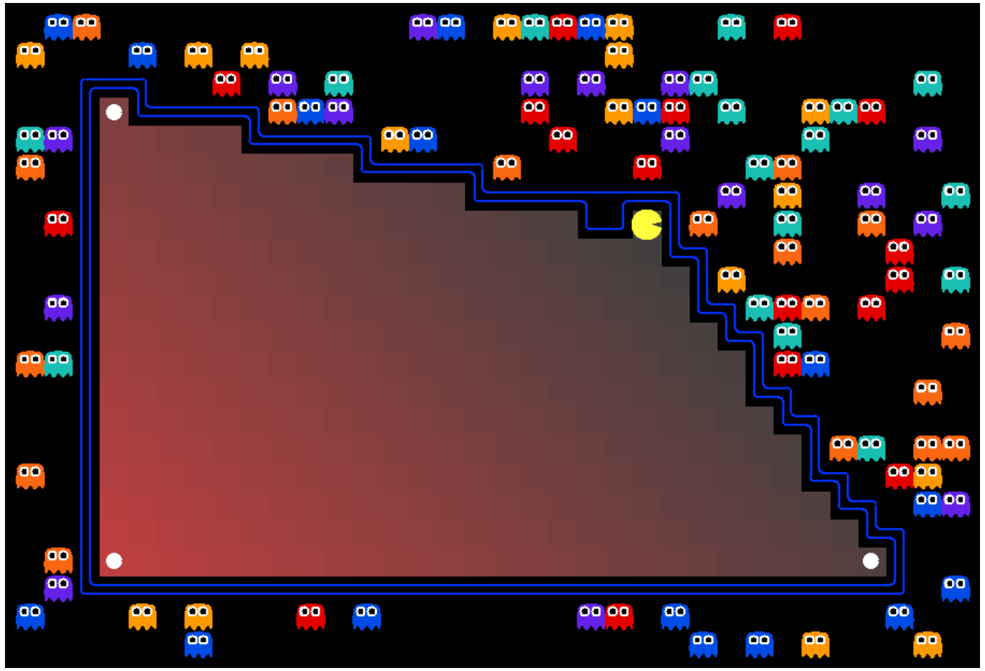
Assignments:

- HW5 (written)
  - Due 10/8 Tue, 10 pm
- HW6 (online)
  - Will be released today, Due 10/15 Tue, 10 pm
- P3: Optimization – expected average completion time < P2
  - Due 10/17 Thu, 10 pm

Final exam: Thursday, Dec 12, 1-4pm, location TBD

AI: Representation and Problem Solving

Integer Programming



Instructors: Fei Fang & Pat Virtue

Slide credits: CMU AI, http://ai.berkeley.edu

Learning Objectives

- Formulate a problem as a Integer (Linear) Program (IP or ILP)
- Write down the Linear Program (LP) relaxation of an IP
- Plot the graphical representation of an IP and find the optimal solution
- Understand the relationship between optimal solution of an IP and the optimal solution of the relaxed LP
- Describe and implement branch-and-bound algorithm

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

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

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

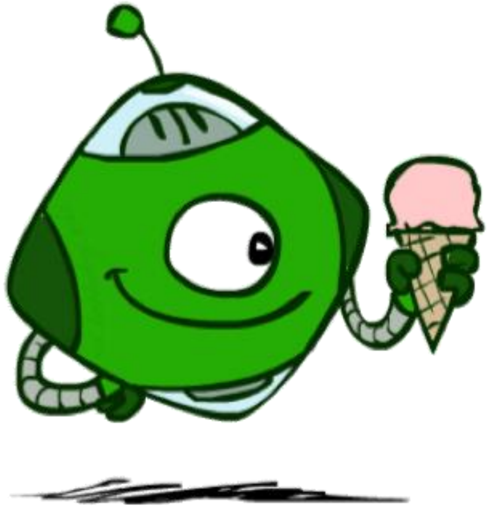
Food	Cost	Calories	Sugar	Calcium
Stir-fry (per bowl)	1	100	3	20
Boba (per glass)	0.5	50	4	70

What is the cheapest way to stay "healthy" with this menu?

How much stir-fry (ounce) and boba (fluid ounces) should we buy?

Problem Formulation

Formulate Diet Problem with integer constraints as an optimization problem



$$\min_{x_1, x_2} \quad 1\,x_1 + 0.5\,x_2$$

$$\text{s.t.} \quad 100\,x_1 + 50\,x_2 \geq 2000$$

$$100\,x_1 + 50\,x_2 \leq 2500$$

$$3\,x_1 + 4\,x_2 \leq 100$$

$$20\,x_1 + 70\,x_2 \geq 700$$

$$x_1, x_2 \in \mathbb{Z}$$

Healthy Squad Goals

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

Food	Cost	Calories	Sugar	Calcium
Stir-fry (per oz)	1	100	3	20
Boba (per fl oz)	0.5	50	4	70

Linear Programming vs Integer Programming

Linear objective with linear constraints, but now with additional constraint that all values in $\boldsymbol{x}$ must be integers

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

We could also do:

- Even more constrained: Binary Integer Programming (BIP)
- A hybrid: Mixed Integer Linear Programming (MIP or MILP)

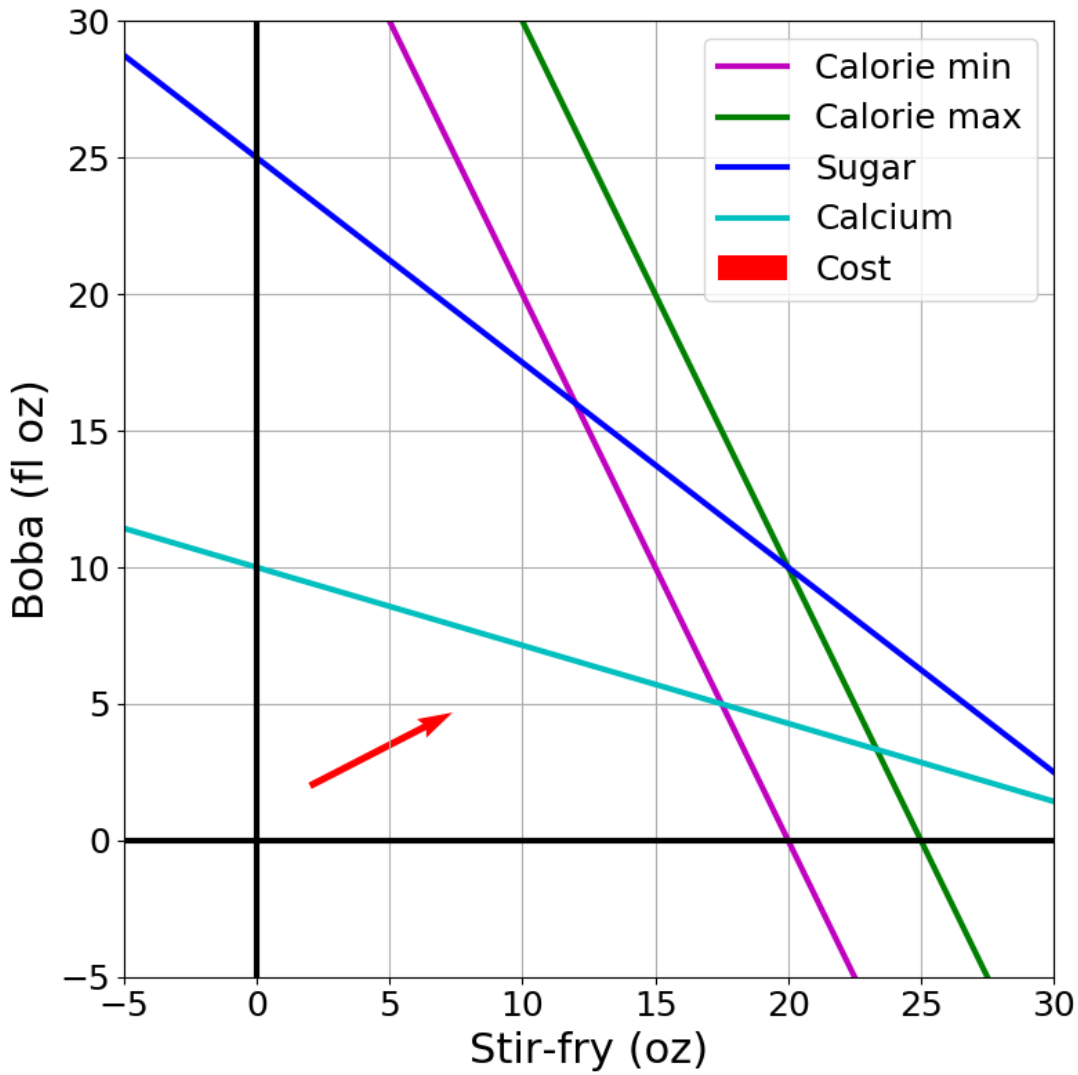
Notation Alert!

Integer Programming: Graphical Representation

Just add a grid of integer points onto our LP representation

$$\min_{\boldsymbol{x}} \quad \boldsymbol{c}^T \boldsymbol{x}$$
$$\text{s.t.} \quad A\boldsymbol{x} \preceq \boldsymbol{b}$$
$$\boldsymbol{x} \in \mathbb{Z}^N$$

$$\min_{x_1, x_2} \quad 1\, x_1 + 0.5\, x_2$$
$$\text{s.t.} \quad 100\, x_1 + 50\, x_2 \geq 2000$$
$$100\, x_1 + 50\, x_2 \leq 2500$$
$$3\, x_1 + 4\, x_2 \leq 100$$
$$20\, x_1 + 70\, x_2 \geq 700$$
$$x_1, x_2 \in \mathbb{Z}$$

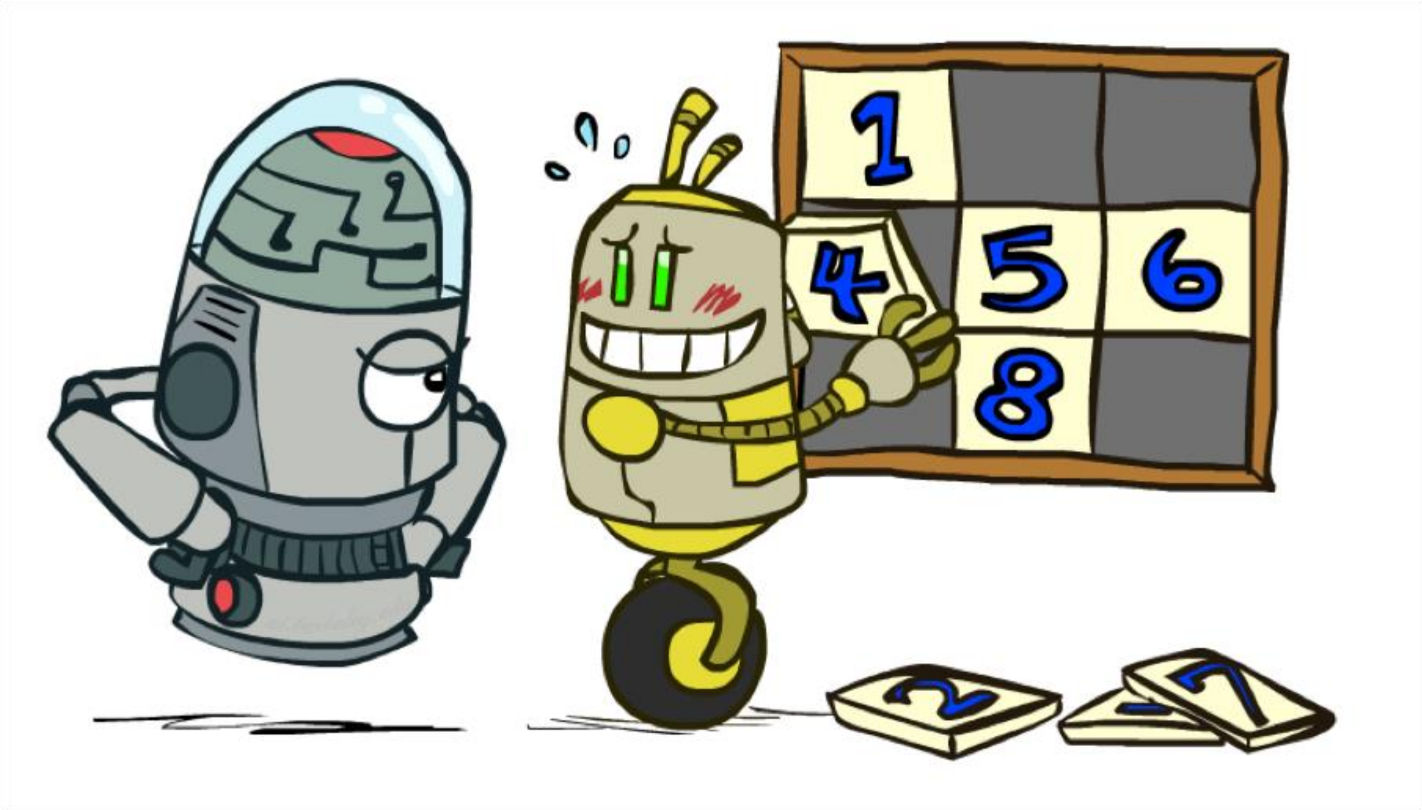


LP Relaxation

Relax IP to LP by dropping integer constraints

$$\min_{\boldsymbol{x}} \quad \boldsymbol{c}^T \boldsymbol{x}$$
$$\text{s.t.} \quad A\boldsymbol{x} \preceq \boldsymbol{b}$$
$$\sout{\boldsymbol{x} \in \mathbb{Z}^N}$$

Remember heuristics?



Piazza Poll 1:

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

Let $\boldsymbol{x}_{IP}^*$ be an optimal point of an integer program P.

Let y_{LP}^* be the optimal objective of the relaxed LP of P.

Let $\boldsymbol{x}_{LP}^*$ be an optimal point of the relaxed LP of P.

Assume that P is a minimization problem.

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

$$\begin{aligned} y_{LP}^* = \min_{\boldsymbol{x}}. \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \preceq \boldsymbol{b} \end{aligned}$$

Which of the following are true?

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

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

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

Question

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

Let $\boldsymbol{x}_{IP}^*$ be an optimal point of an integer program P.

Let $\boldsymbol{x}_0$ be a feasible point of P.

Let y_0 be the objective value of P at $\boldsymbol{x}_0$

Assume that P is a minimization problem.

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

$$
\begin{aligned}
y_{LP}^* = \min_{\boldsymbol{x}}. \quad & \boldsymbol{c}^T\boldsymbol{x} \\
\text{s.t.} \quad & A\boldsymbol{x} \preceq \boldsymbol{b}
\end{aligned}
$$

Which of the following are true?

A) $\boldsymbol{x}_{IP}^* = \boldsymbol{x}_0$

B) $y_{IP}^* \leq y_0$

C) $y_{IP}^* \geq y_0$

Question

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

Let $\boldsymbol{x}_{IP}^*$ be an optimal point of an integer program P.

Let $\boldsymbol{x}_0$ be a feasible point of P.

Let y_0 be the objective value of P at $\boldsymbol{x}_0$

Assume that P is a minimization problem.

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

Which of the following are true?

A) $\boldsymbol{x}_{IP}^* = \boldsymbol{x}_0$

B) $y_{IP}^* \leq y_0$

C) $y_{IP}^* \geq y_0$

Upper bound

$$y_{LP}^* \leq y_{IP}^* \leq y_0$$

Lower bound

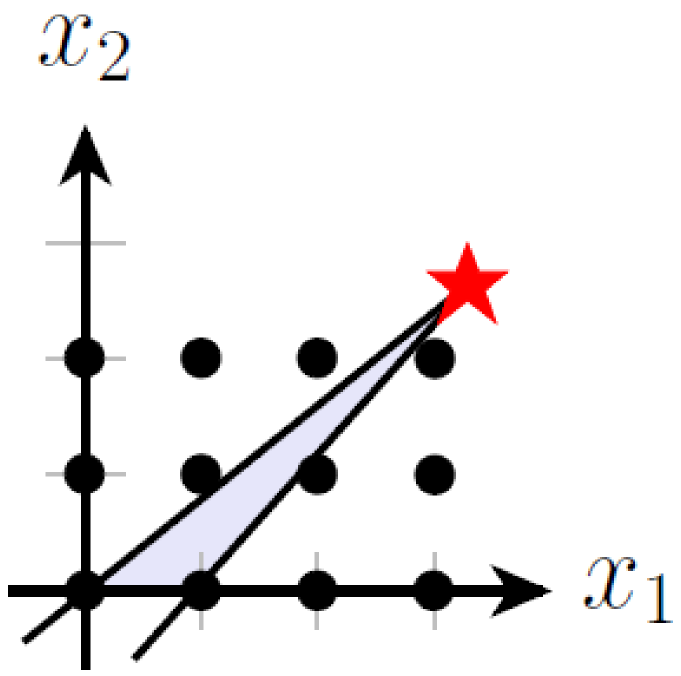
$$y_{LP}^* = \min_{\boldsymbol{x}}. \quad \boldsymbol{c}^T\boldsymbol{x} \quad \text{s.t.} \quad A\boldsymbol{x} \preceq \boldsymbol{b}$$

Piazza Poll 2:

True/False: It is sufficient to consider the integer points that are the closest to an optimal solution of the LP relaxation?

Piazza Poll 2:

True/False: It is sufficient to consider the integer points that are the closest to an optimal solution of the LP relaxation?



1. An LP can have infinite number of optimal solutions
2. So as IP
3. Integer points that are closest to an optimal solution of LP relaxation of IP may not be feasible
4. Integer points that are closest to an optimal solution of LP relaxation of IP may not be optimal (depends on the objective function)

Solving an IP

Basic Branch and Bound algorithm (essentially a search algorithm)

Assuming a minimization IP

1. Build the root node, which is the original IP. Set $\boldsymbol{x}_{IP}^{*}$=null, $y_{IP}^{*} = +\infty$
2. Solve the relaxed LP of the node
3. If relaxed LP is feasible, get solution $\boldsymbol{x}_{LP}^{*}$ and optimal objective value y_{LP}^{*}

   (Update) If integer($\boldsymbol{x}_{LP}^{*}$), update $\boldsymbol{x}_{IP}^{*}$=best($\boldsymbol{x}_{IP}^{*}, \boldsymbol{x}_{LP}^{*}$) and go to step 4

   (Prune) If $y_{LP}^{*} \geq y_{IP}^{*}$, go to step 4

   (Branch) Choose a variable x_i that has non-integer value in $\boldsymbol{x}_{LP}^{*}$, branch and construct two new nodes each representing a more constrained IP:

   New node at left branch: Add constraint $x_i \leq floor(x_i)$

   New node at right branch: Add constraint $x_i \geq ceil(x_i)$
4. (Recursion) Pick an unexplored node and go to step 2. Stop if all nodes explored.
5. Return $\boldsymbol{x}_{IP}^{*}$

What if it is a maximization IP?

Solving an IP

Basic Branch and Bound algorithm (essentially a search algorithm)

Assuming a minimization IP

(Upper bound of IP)

1. Build the root node, which is the original IP. Set $\boldsymbol{x}_{IP}^{*}$=null, $y_{IP}^{*} = +\infty$
2. Solve the relaxed LP of the node

(Lower bound for the IP of the node)

3. If relaxed LP is feasible, get solution $\boldsymbol{x}_{LP}^{*}$ and optimal objective value y_{LP}^{*}

   (Update) If integer($\boldsymbol{x}_{LP}^{*}$), update $\boldsymbol{x}_{IP}^{*}$=best($\boldsymbol{x}_{IP}^{*}, \boldsymbol{x}_{LP}^{*}$) and go to step 4

   (Prune) If $y_{LP}^{*} \geq y_{IP}^{*}$, go to step 4 **(Update upper bound)**

   (Branch) Choose a variable x_i that has non-integer value in $\boldsymbol{x}_{LP}^{*}$, branch and construct two new nodes each representing a more constrained IP:

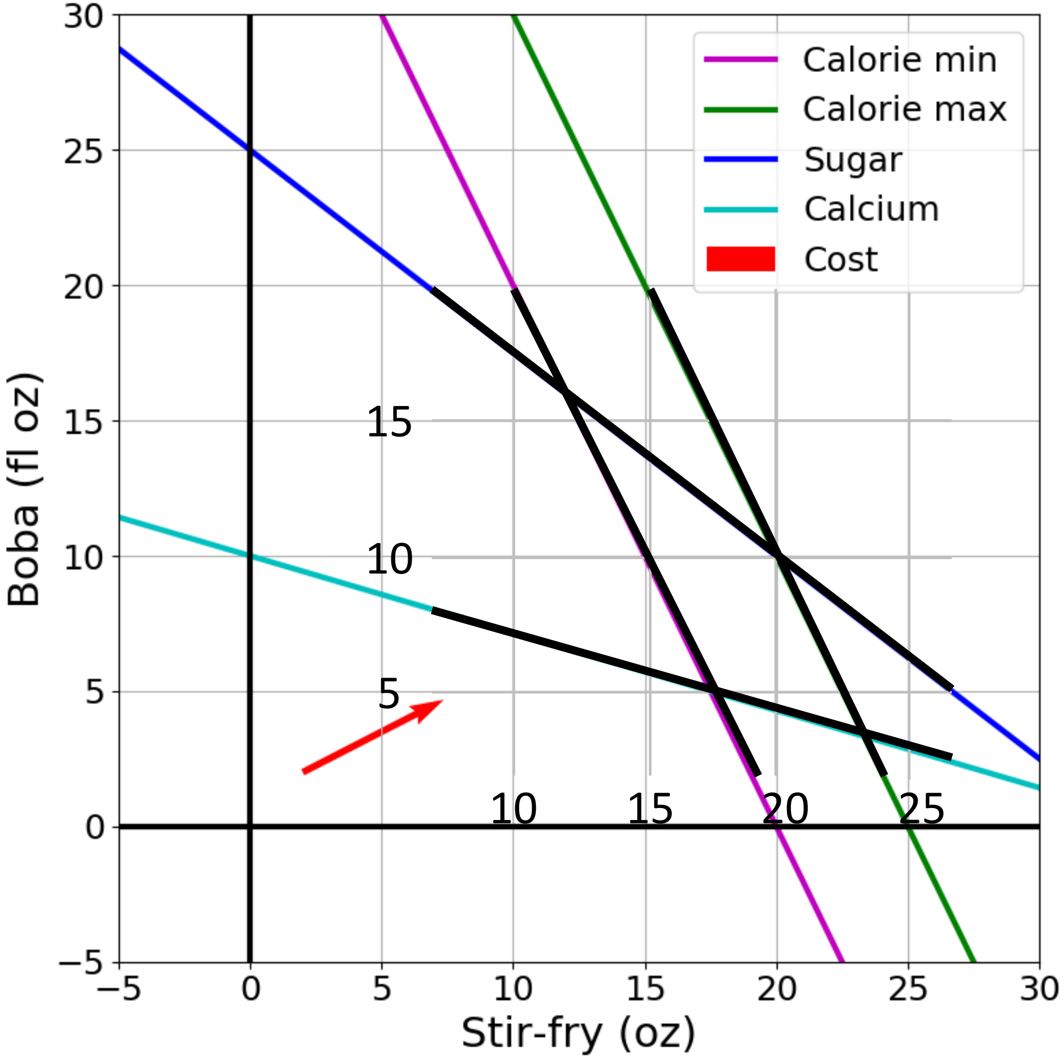
   New node at left branch: Add constraint $x_i \leq floor(x_i)$

   New node at right branch: Add constraint $x_i \geq ceil(x_i)$

4. (Recursion) Pick an unexplored node and go to step 2. Stop if all nodes explored.
5. Return $\boldsymbol{x}_{IP}^{*}$

What if it is a maximization IP?

Branch and Bound Example

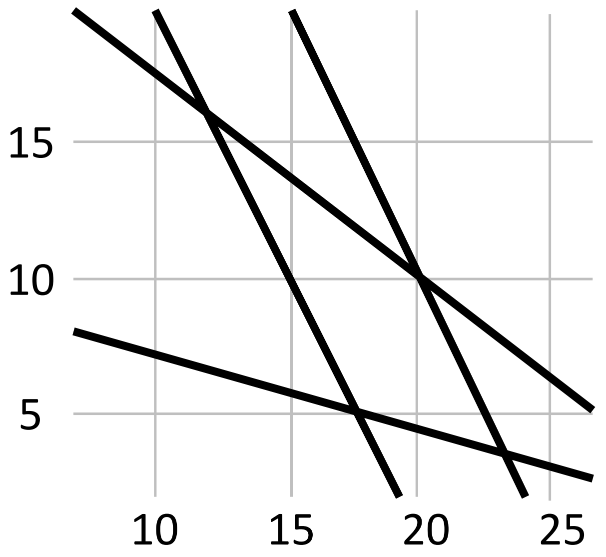


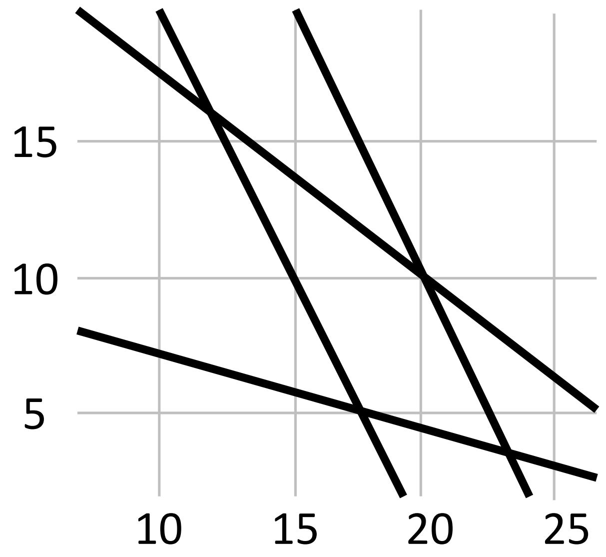
Branch and Bound Example

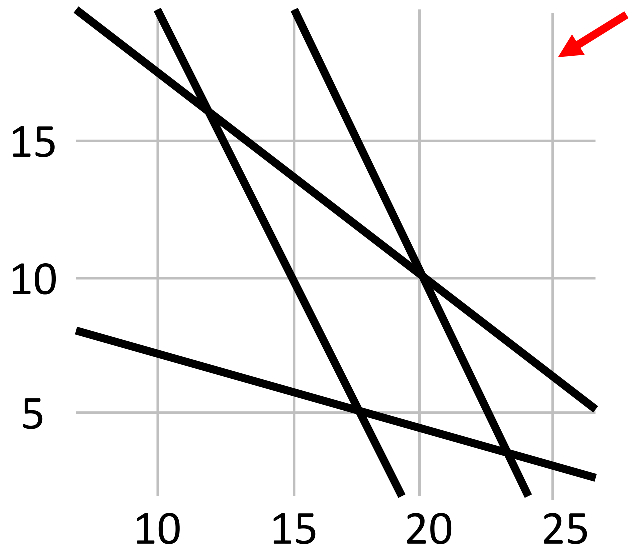
$$\min_{x_1, x_2} \quad 1\,x_1 + 0.5\,x_2$$

s.t.

$$100\,x_1 + 50\,x_2 \geq 2000$$
$$100\,x_1 + 50\,x_2 \leq 2500$$
$$3\,x_1 + 4\,x_2 \leq 100$$
$$20\,x_1 + 70\,x_2 \geq 700$$
$$x_1, x_2 \in \mathbb{Z}$$





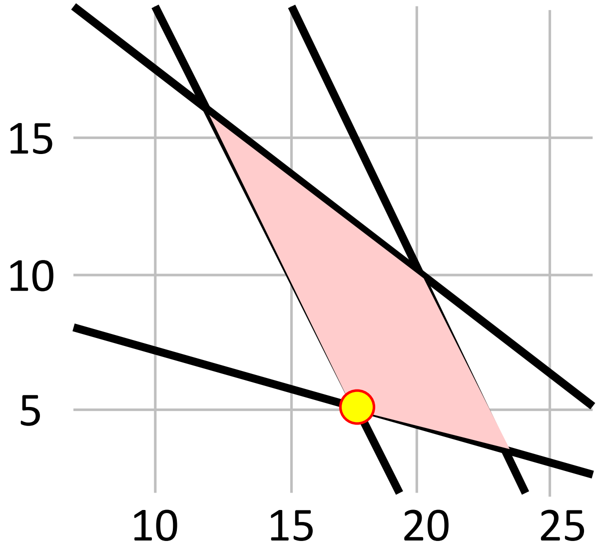


Branch and Bound Example

$$\min_{x_1, x_2} \quad 1\,x_1 + 0.5\,x_2$$

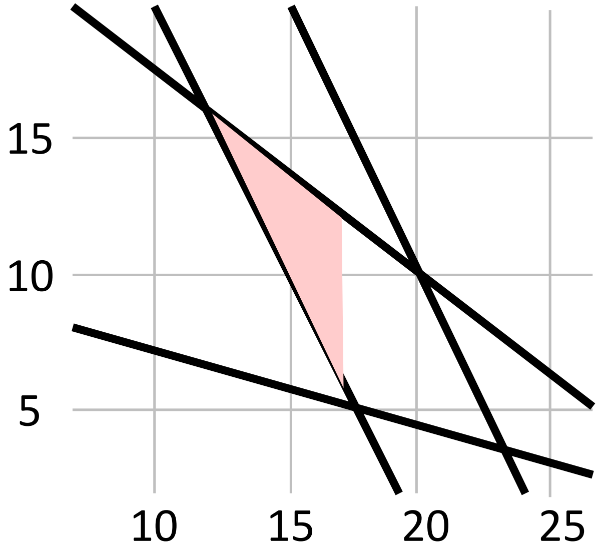
s.t.

$$100\,x_1 + 50\,x_2 \geq 2000$$
$$100\,x_1 + 50\,x_2 \leq 2500$$
$$3\,x_1 + \quad 4\,x_2 \leq 100$$
$$20\,x_1 + 70\,x_2 \geq 700$$
$$x_1, x_2 \in \mathbb{Z}$$



Optimal solution for LP relaxation: (17.5,5)

Optimal value (lower bound for IP): 20



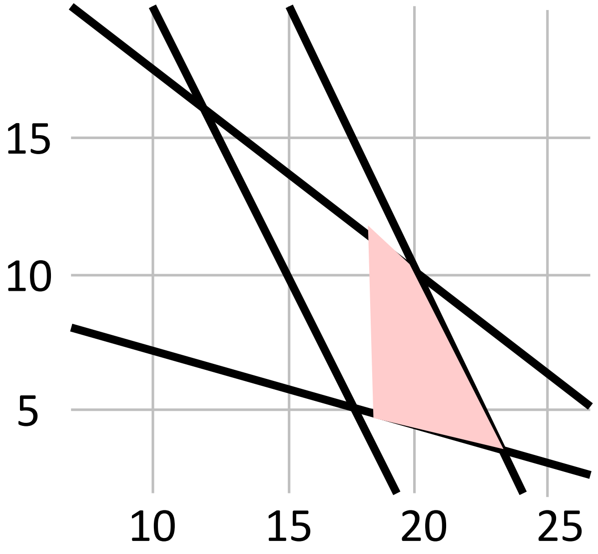
Add constraint $x_1 \leq 17$

Optimal solution for relaxed LP: (17,6) Optimal solution

Optimal value (lower bound for this branch): 20

An integer solution found (update upper bound of IP)

Stop



Add $x_1 \geq 18$

Optimal solution for relaxed LP: (18,4.86)

Optimal value (lower bound for this branch): 20.43

Worse than upper bound

Stop

Piazza Poll 3:

When does the branch-and-bound algorithm choose not to branch the current node? (Select all that apply)

A. When the LP returns an equal or worse objective value than the best feasible IP objective value you have seen before

B: When you hit an integer result from the LP

C: When LP is infeasible

D: When the LP returns a better objective value than the best feasible IP objective value you have seen before

Solving an IP

Basic Branch and Bound algorithm (essentially a search algorithm)

Assuming a minimization IP

1. Build the root node, which is the original IP. Set $\boldsymbol{x}_{IP}^*$=null, $y_{IP}^* = +\infty$
2. Solve the relaxed LP of the node
3. If relaxed LP is feasible, get solution $\boldsymbol{x}_{LP}^*$ and optimal objective value y_{LP}^*

   (Update) If integer($\boldsymbol{x}_{LP}^*$), update $\boldsymbol{x}_{IP}^*$=best($\boldsymbol{x}_{IP}^*, \boldsymbol{x}_{LP}^*$) and go to step 4

   (Prune) If $y_{LP}^* \geq y_{IP}^*$, go to step 4

   (Branch) Choose a variable x_i that has non-integer value in $\boldsymbol{x}_{LP}^*$, branch and construct two new nodes each representing a more constrained IP:

   New node at left branch: Add constraint $x_i \leq floor(x_i)$

   New node at right branch: Add constraint $x_i \geq ceil(x_i)$
4. (Recursion) Pick an unexplored node and go to step 2. Stop if all nodes explored.
5. Return $\boldsymbol{x}_{IP}^*$

Anything we can do to make the search more efficient?

Recall: Informed Search

function BEST-FIRST-SEARCH (*problem*, EVAL-FN) **returns** a solution sequence
inputs: *problem*, a problem
EVAL-FN, an evaluation function

Queuing-Fn ← a function that orders nodes by EVAL-FN
return GENERAL-SEARCH (*problem*, *Queuing-Fn)*

function GREEDY-SEARCH (*problem*) **returns** a solution or failure
return BEST-FIRST-SEARCH (*problem*, *h)*

Solving an IP

Basic Branch and Bound algorithm (essentially a search algorithm)

Assuming a minimization IP

1. Build the root node, which is the original IP. Set $\boldsymbol{x}_{IP}^{*}$=null, $y_{IP}^{*} = +\infty$
2. Solve the relaxed LP of the node
3. If relaxed LP is feasible, get solution $\boldsymbol{x}_{LP}^{*}$ and optimal objective value y_{LP}^{*}

   (Update) If integer($\boldsymbol{x}_{LP}^{*}$), update $\boldsymbol{x}_{IP}^{*}$=best($\boldsymbol{x}_{IP}^{*}, \boldsymbol{x}_{LP}^{*}$) and go to step 4

   (Prune) If $y_{LP}^{*} \geq y_{IP}^{*}$, go to step 4

   (Branch) Choose a variable x_i that has non-integer value in $\boldsymbol{x}_{LP}^{*}$, branch and construct two new nodes each representing a more constrained IP:

   New node at left branch: Add constraint $x_i \leq floor(x_i)$

   New node at right branch: Add constraint $x_i \geq ceil(x_i)$
4. (Recursion) Pick an unexplored node and go to step 2. Stop if all nodes explored.
5. Return $\boldsymbol{x}_{IP}^{*}$

What can be used as an evaluation function?

Optimal value of the LP relaxation!

Piazza Poll 4:

True or False: Node A and B are two nodes in the search tree. If

(1) Node A is not a descendent of B

(2) Node A's LP relaxation has better optimal objective value than node B, i.e., $y_{LP}^{A} < y_{LP}^{B}$ for a minimization problem

(3) The optimal solution of LP relaxation at node A is an integer solution, i.e., integer($\boldsymbol{x}_{LP}^{A}$)

then it is impossible that the optimal solution of the original IP is found in the subtree rooted at node B

Piazza Poll 4:

True or False: Node A and B are two nodes in the search tree. If

(1) Node A is not a descendent of B

(2) Node A's LP relaxation has better optimal objective value than node B, i.e., $y_{LP}^{A} < y_{LP}^{B}$ for a minimization problem

(3) The optimal solution of LP relaxation at node A is an integer solution, i.e., integer($\boldsymbol{x}_{LP}^{A}$)

then it is impossible that the optimal solution of the original IP is found in the subtree rooted at node B

Since $\boldsymbol{x}_{LP}^{\mathrm{A}}$ is integer, its objective value upper bounds the optimal value of the original LP, i.e., $y_{LP}^{A} \geq y_{IP}^{*}$.
y_{LP}^{B} is the lower bound of the IP at node B. Any node C in the subtree of node B has no less constraints than node B, so $y_{LP}^{\mathrm{B}} \leq y_{LP}^{\mathrm{C}}$. In addition, $y_{LP}^{\mathrm{C}} \leq y_{IP}^{C}$.
So $y_{IP}^{\mathrm{C}} > y_{IP}^{*}$. The optimal solution of the original IP cannot be found at node C.

Solving an IP

Improved Branch and Bound algorithm (essentially a best-first-search algorithm)

Assuming a minimization IP

1. Build the root node, which is the original IP.
2. Solve the relaxed LP of the node. **Return null if infeasible.**
3. Given solution $\boldsymbol{x}_{LP}^*$ and optimal objective value y_{LP}^* of the relaxed LP

   (Update) If integer($\boldsymbol{x}_{LP}^*$), **return $\boldsymbol{x}_{LP}^*$**

   (Branch) Choose a variable x_i that has non-integer value in $\boldsymbol{x}_{LP}^*$, branch and construct two new nodes each representing a more constrained IP:

   New node at left branch: Add constraint $x_i \leq floor(x_i)$

   New node at right branch: Add constraint $x_i \geq ceil(x_i)$
4. (Recursion) Pick an unexplored node **whose LP relaxation is feasible and has the best optimal objective value** and go to step 2. Stop if all nodes explored/infeasible.
5. Return **null**

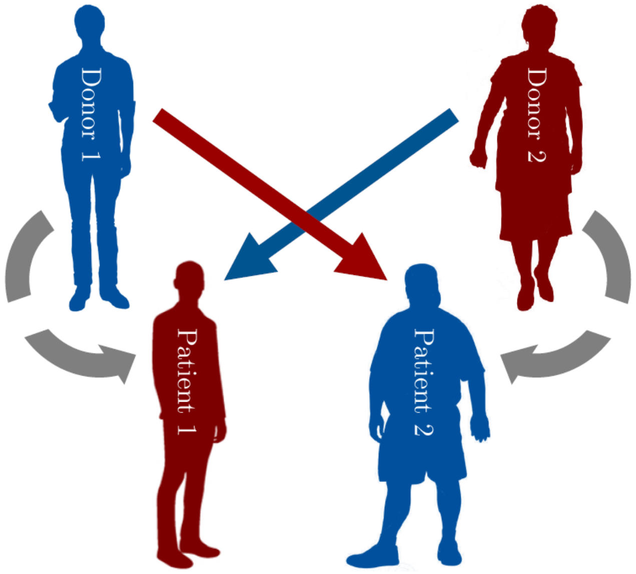
Why still optimal? All the unexplored nodes have no better LP relaxation values!

Formulate a Problem as an IP

Kidney Exchange

UNOS DONATE LIFE ®

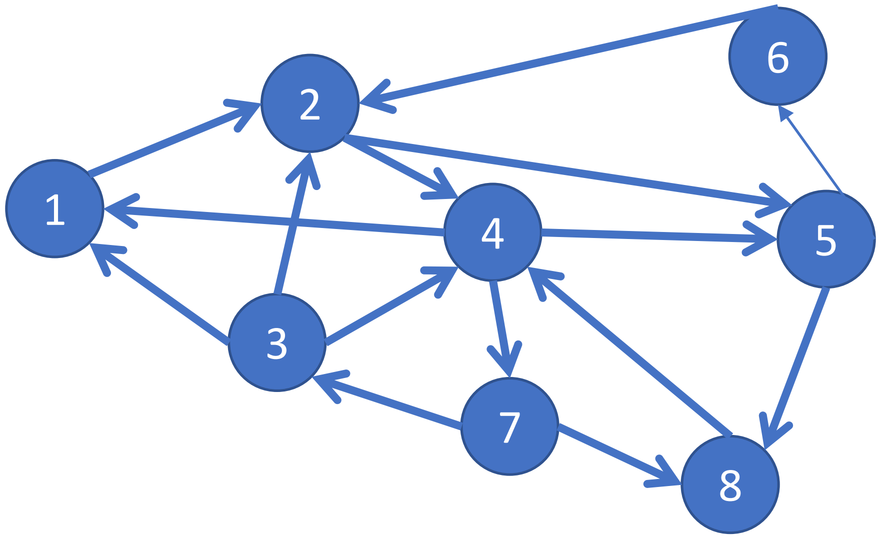
UNITED NETWORK FOR ORGAN SHARING



Formulate a Problem as an IP

Kidney Exchange

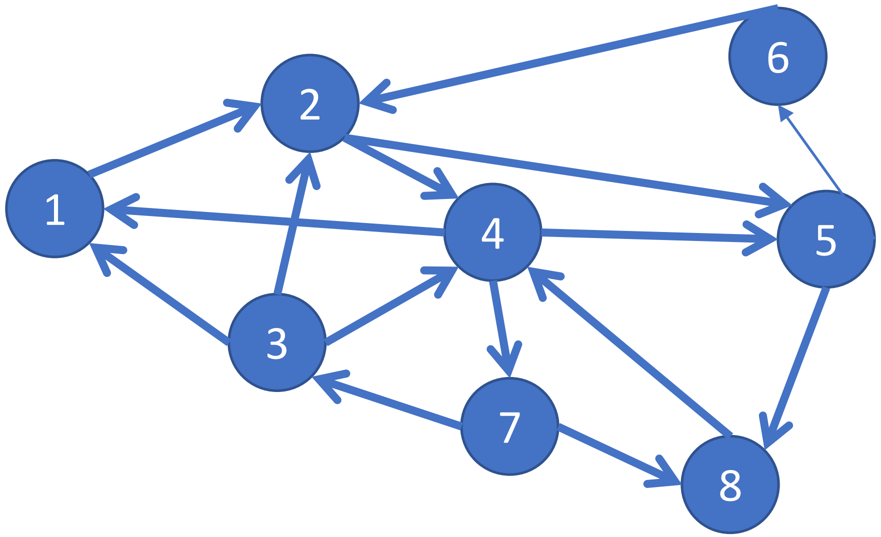
- Given directed graph $G = (V, E)$, where each node represent a patient-donor pair, and an edge $\langle u, v \rangle$ means donor of node u can give one kidney to patient of node v
- Find a set of disjoint cycles so as to maximize the number of nodes covered



Piazza Poll 5

Given the graph below, what is the maximum number of patients that can get a kidney through kidney exchange assuming the length of each cycle should be less than or equal to 3?

- A: 3
- B: 6
- C: 7
- D: 8

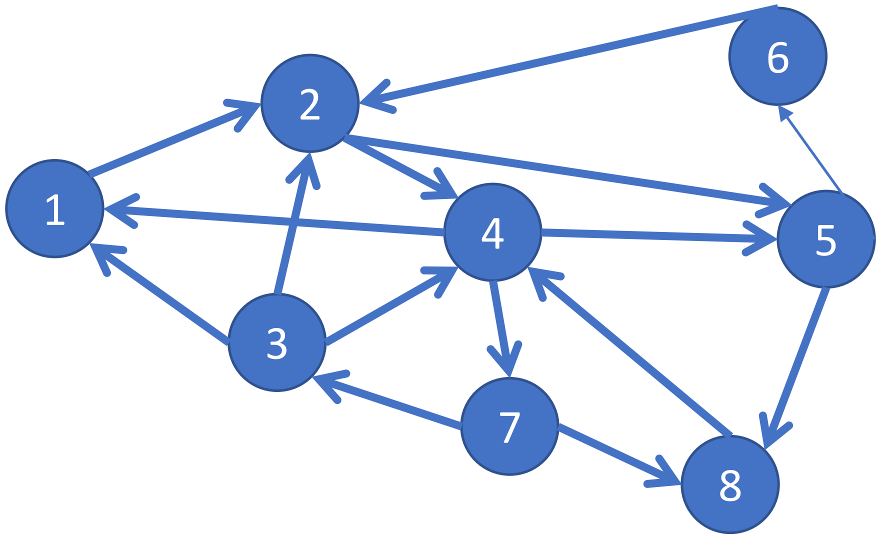


Formulate a Problem as an IP

Kidney Exchange

- Given directed graph $G = (V, E)$, where each node represent a patient-donor pair, and an edge $\langle u, v \rangle$ means donor of node u can give one kidney to patient of node v
- Find a set of disjoint cycles so as to maximize the number of nodes covered

Hint: enumerate all the cycles

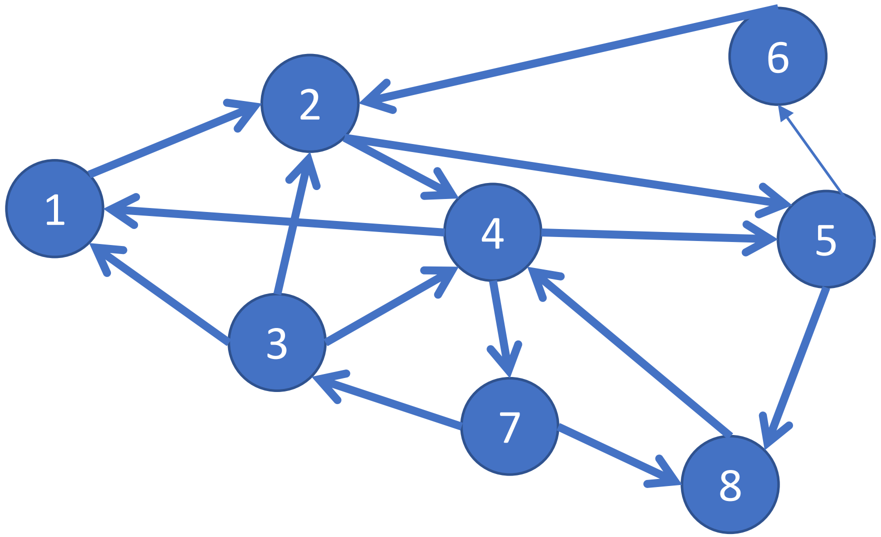


Formulate a Problem as an IP

Kidney Exchange

- Given directed graph $G = (V, E)$, where each node represent a patient-donor pair, and an edge $\langle u, v \rangle$ means donor of node u can give one kidney to patient of node v
- Find a set of disjoint cycles so as to maximize the number of nodes covered

Hint: enumerate all the cycles



$$\max_x \sum_c x_c l_c$$

$$\text{s.t.} \sum_{c: v \in c} x_c \leq 1, \forall v \in V$$

$$x_c \in \{0,1\}, \forall c$$

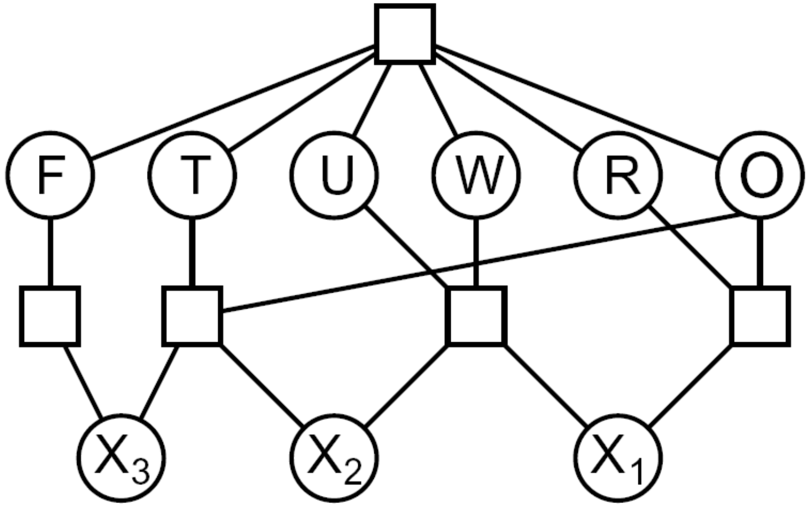
Formulate a Problem as an IP

Cryptarithmetic

Variables:

IP:

```
   T W O
 + T W O
 -------
 F O U R
```



Formulate a Problem as an IP

Cryptarithmetic

Variables: $x_T, x_W, x_O, x_F, x_U, x_R, c_1, c_2, c_3$

IP:

$$\min_{x,c} 1$$
$$\text{s.t. } x_O + x_O = x_R + 10c_1$$
$$x_W + x_W + c_1 = x_U + 10c_2$$
$$x_T + x_T + c_2 = x_O + 10c_3$$
$$0 + c_3 = x_F$$
$$x_T, x_F \in \{1, .., 9\}$$
$$x_W, x_O, x_U, x_R \in \{0, .., 9\}$$
$$c_1, c_2, c_3 \in \{0,1\}$$

Can also write as: $x_T, x_F, x_W, x_O, x_U, x_R \leq 9$,
$x_T, x_F \geq 1, x_W, x_O, x_U, x_R \geq 0$,
$c_1, c_2, c_3 \leq 1, c_1, c_2, c_3 \geq 0$
$x, c \in \mathbb{Z}^N$

```
    T W O
+   T W O
---------
  F O U R
```