

Announcements

Midterm1

- Grade Released

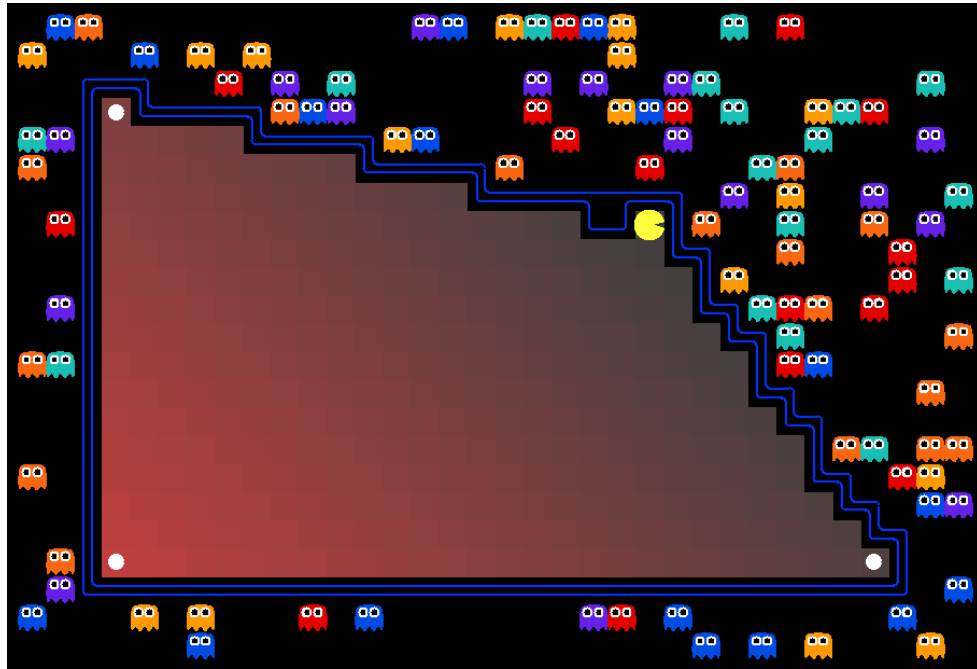
Assignments:

- 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).

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

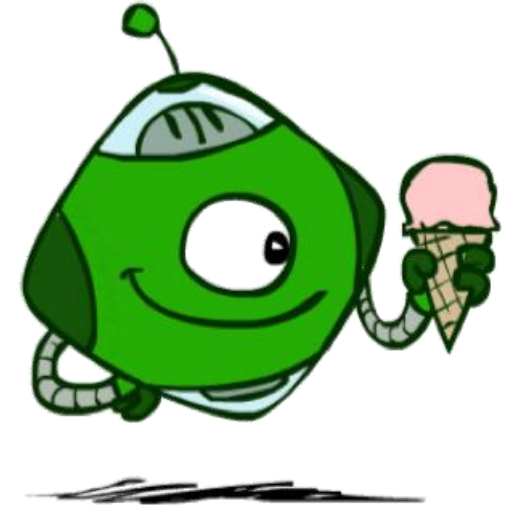
What is the cheapest way to stay “healthy” with this menu?

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

Problem Formulation

Formulate Diet Problem with integer constraints as an optimization problem

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



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

Linear Programming vs Integer Programming

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

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

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

We could also do:

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

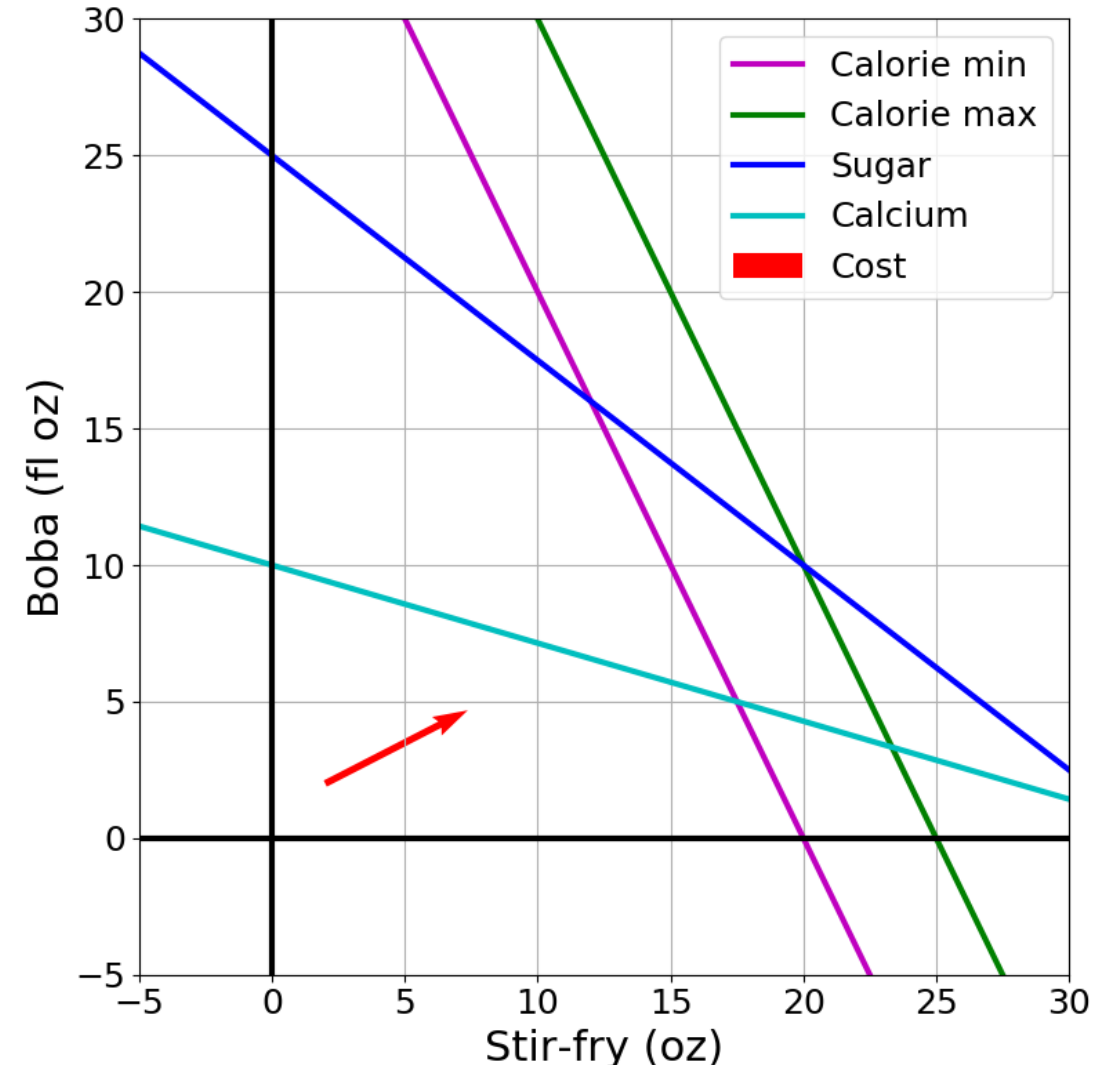
Notation Alert!

Integer Programming: Graphical Representation

Just add a grid of integer points onto our LP representation

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

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

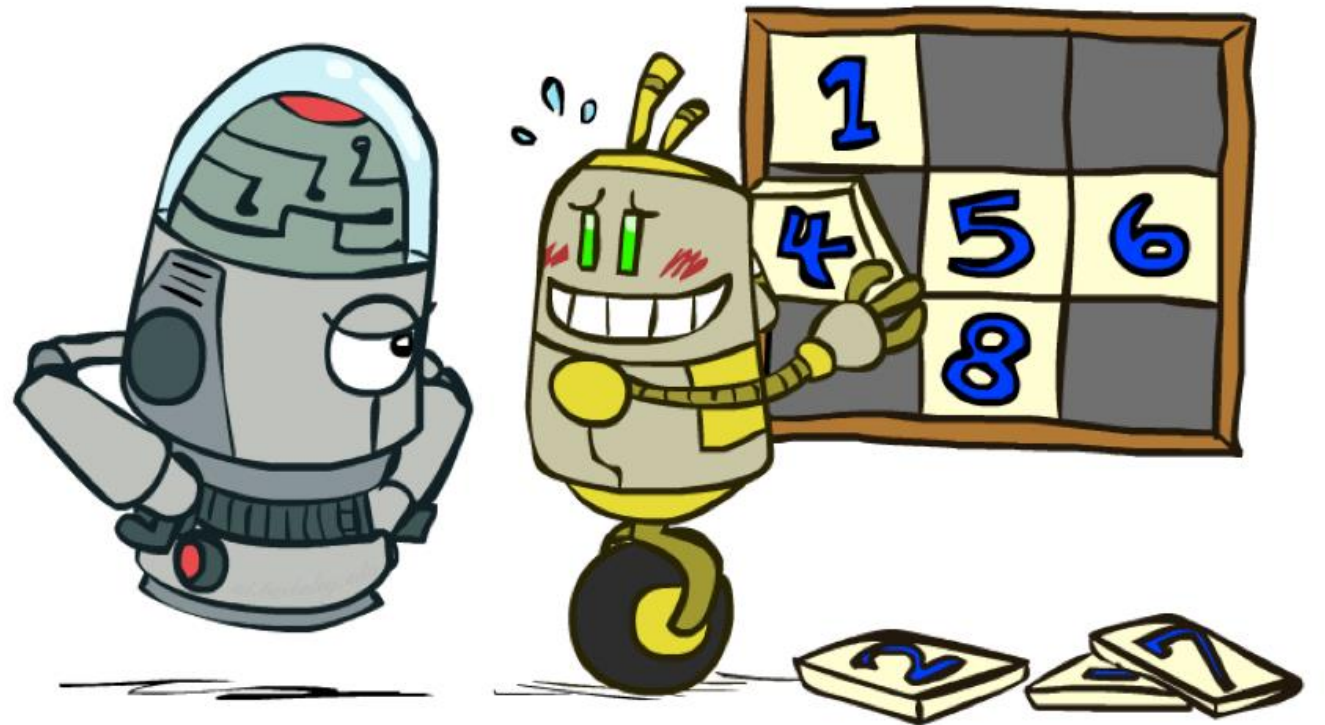


LP Relaxation

Relax IP to LP by dropping integer constraints

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

Remember heuristics?



Piazza Poll 1:

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 relaxed LP of P .

Let $\mathbf{x}_{LP}^*$ be an optimal point of the relaxed LP 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) $y_{IP}^* \geq y_{LP}^*$

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

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

Question

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

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

Assume that P is a minimization problem.

Which of the following are true?

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

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

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

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

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

Question

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

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

Assume that P is a minimization problem.

Which of the following are true?

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

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

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

Upper bound

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

Lower bound

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

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

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?

Solving an IP

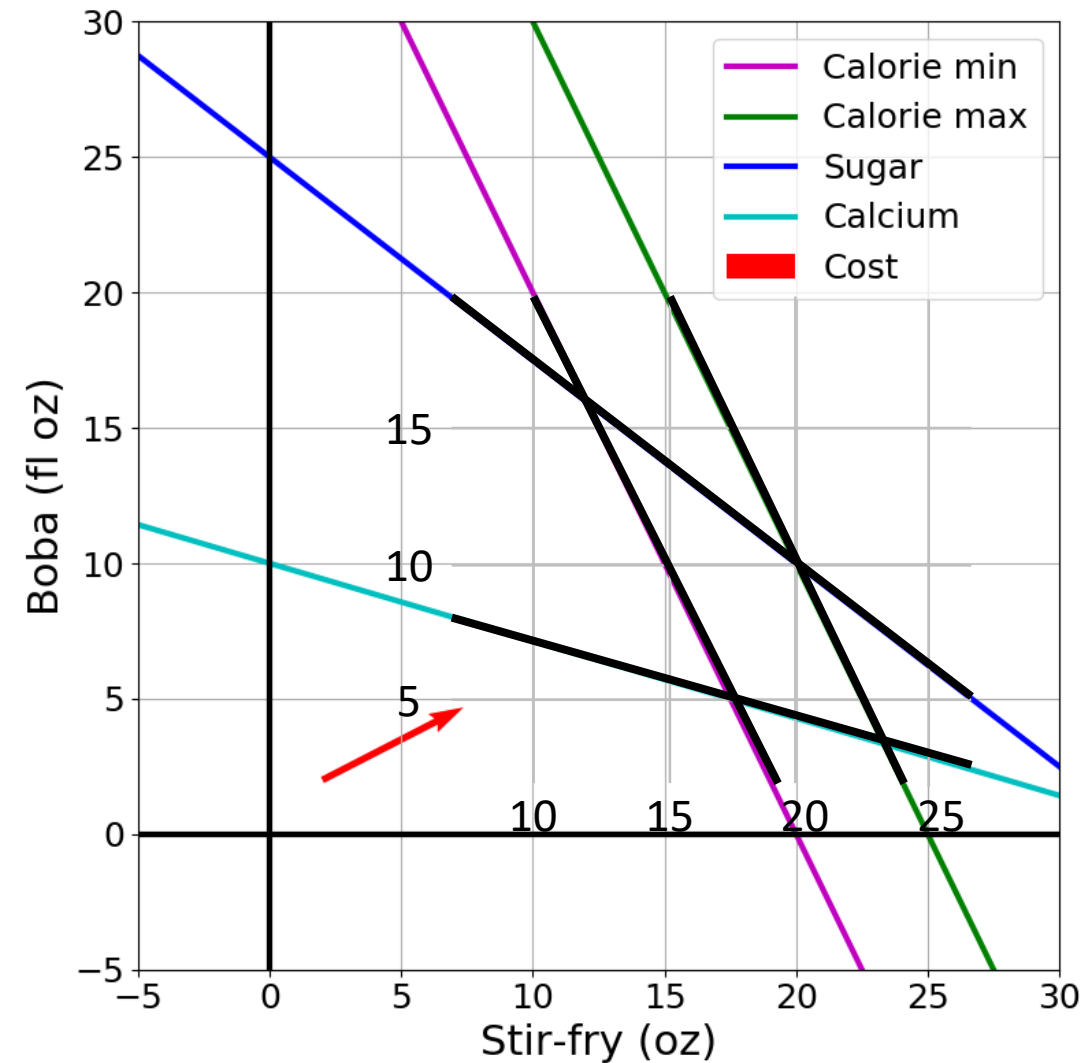
Basic Branch and Bound algorithm (essentially a search algorithm)

Assuming a minimization IP

1. Build the root node, which is the original IP. Set $\mathbf{x}_{IP}^* = \text{null}$, $y_{IP}^* = +\infty$
2. Solve the relaxed LP of the node
3. If relaxed LP is feasible, get solution $\mathbf{x}_{LP}^*$ and optimal objective value y_{LP}^*
 - (Update) If $\text{integer}(\mathbf{x}_{LP}^*)$, update $\mathbf{x}_{IP}^* = \text{best}(\mathbf{x}_{IP}^*, \mathbf{x}_{LP}^*)$ 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 $\mathbf{x}_{LP}^*$, branch and construct two new nodes each representing a more constrained IP:
 - New node at left branch: Add constraint $x_i \leq \text{floor}(x_i)$*
 - New node at right branch: Add constraint $x_i \geq \text{ceil}(x_i)$*
4. (Recursion) Pick an unexplored node and go to step 2. Stop if all nodes explored.
5. Return $\mathbf{x}_{IP}^*$

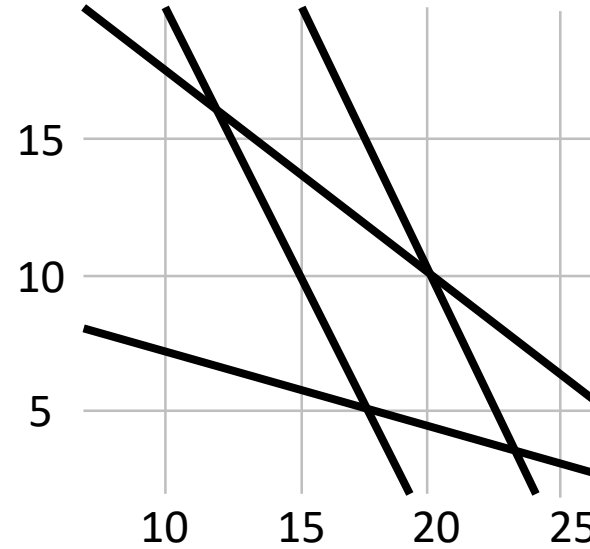
What if it is a maximization IP?

Branch and Bound Example

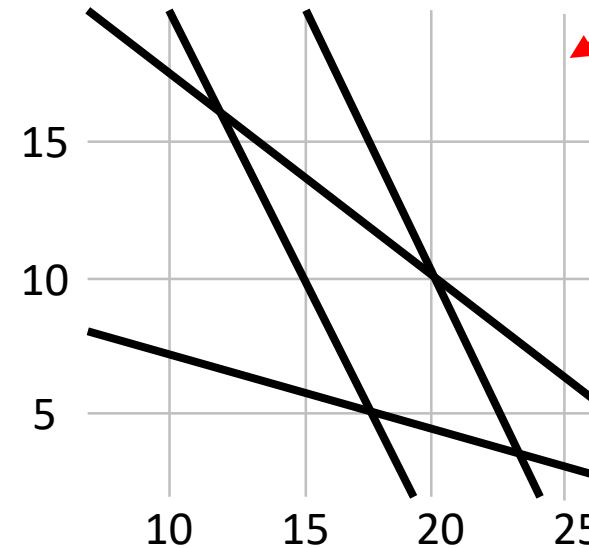
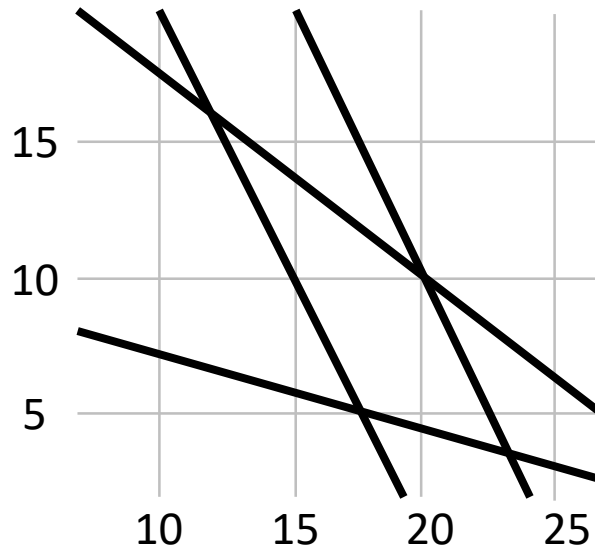


Branch and Bound Example

$$\begin{array}{ll}\min & 1x_1 + 0.5x_2 \\ \text{s.t.} & 100x_1 + 50x_2 \geq 2000 \\ & 100x_1 + 50x_2 \leq 2500 \\ & 3x_1 + 4x_2 \leq 100 \\ & 20x_1 + 70x_2 \geq 700 \\ & x_1, x_2 \in \mathbb{Z}\end{array}$$



Is it necessary to solve this branch?



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 $\mathbf{x}_{IP}^* = \text{null}$, $y_{IP}^* = +\infty$
2. Solve the relaxed LP of the node
3. If relaxed LP is feasible, get solution $\mathbf{x}_{LP}^*$ and optimal objective value y_{LP}^*
 - (Update) If $\text{integer}(\mathbf{x}_{LP}^*)$, update $\mathbf{x}_{IP}^* = \text{best}(\mathbf{x}_{IP}^*, \mathbf{x}_{LP}^*)$ 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 $\mathbf{x}_{LP}^*$, branch and construct two new nodes each representing a more constrained IP:
 - New node at left branch: Add constraint $x_i \leq \text{floor}(x_i)$*
 - New node at right branch: Add constraint $x_i \geq \text{ceil}(x_i)$*
4. (Recursion) Pick an unexplored node and go to step 2. Stop if all nodes explored.
5. Return $\mathbf{x}_{IP}^*$

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 $\leftarrow$ 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 $\mathbf{x}_{IP}^* = \text{null}$, $y_{IP}^* = +\infty$
2. Solve the relaxed LP of the node
3. If relaxed LP is feasible, get solution $\mathbf{x}_{LP}^*$ and optimal objective value y_{LP}^*
(Update) If $\text{integer}(\mathbf{x}_{LP}^*)$, update $\mathbf{x}_{IP}^* = \text{best}(\mathbf{x}_{IP}^*, \mathbf{x}_{LP}^*)$ 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 $\mathbf{x}_{LP}^*$, branch and construct two new nodes each representing a more constrained IP:
New node at left branch: Add constraint $x_i \leq \text{floor}(x_i)$
New node at right branch: Add constraint $x_i \geq \text{ceil}(x_i)$
4. (Recursion) Pick an unexplored node and go to step 2. Stop if all nodes explored.
5. Return $\mathbf{x}_{IP}^*$

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., $\text{integer}(x_{LP}^A)$

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

Solving an IP

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

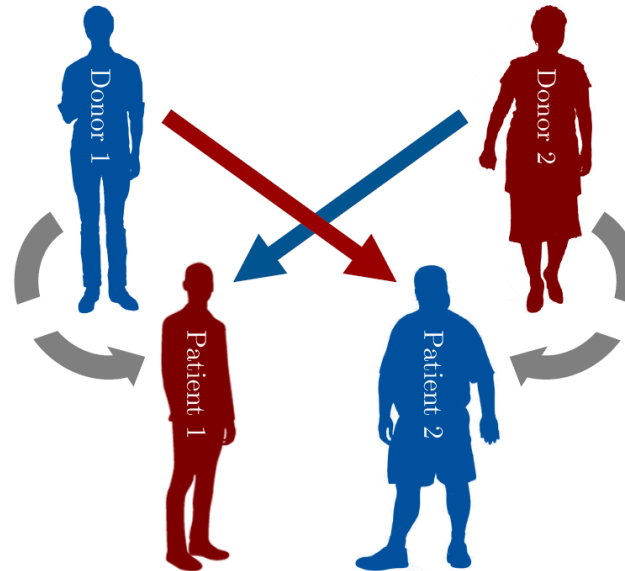
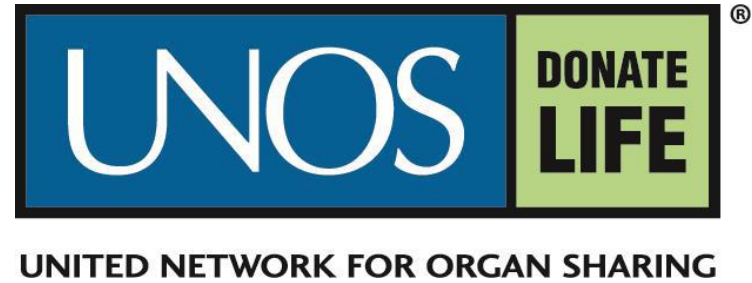
Assuming a minimization IP

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 x_{LP}^* and optimal objective value y_{LP}^* of the relaxed LP
(Update) If $\text{integer}(x_{LP}^*)$, **return x_{LP}^***
(Branch) Choose a variable x_i that has **non-integer value** in x_{LP}^* , branch and construct two new nodes each representing a more constrained IP:
New node at left branch: Add constraint $x_i \leq \text{floor}(x_i)$
New node at right branch: Add constraint $x_i \geq \text{ceil}(x_i)$
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

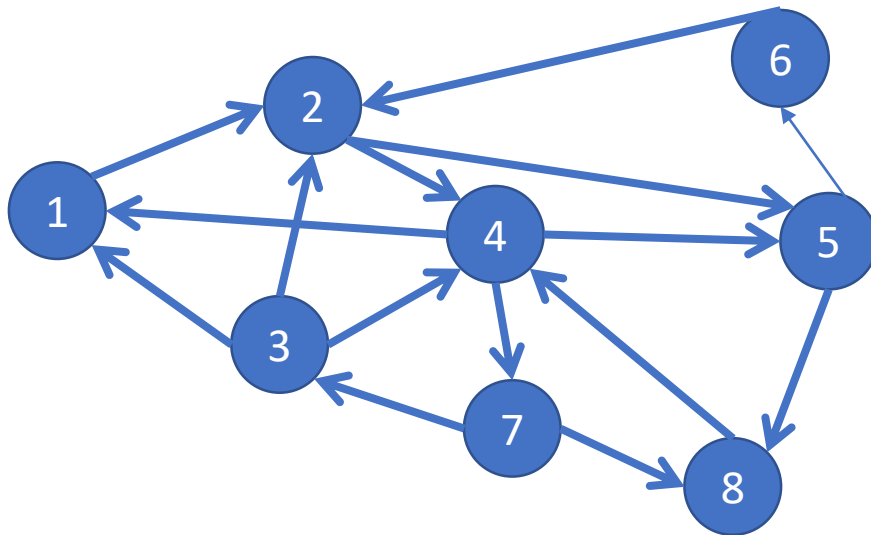
Kidney Exchange



Formulate a Problem as an IP

Kidney Exchange

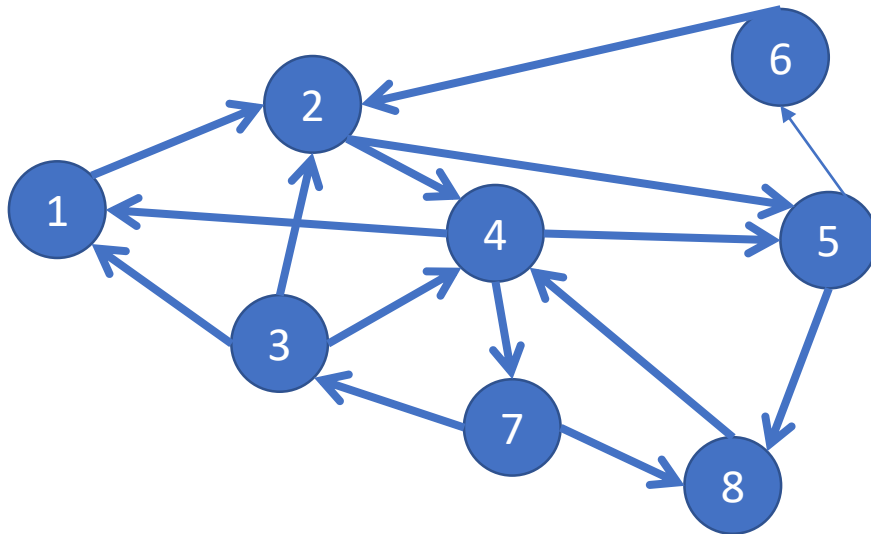
- Given directed graph $G = (V, E)$, where each node represent a patient-donor pair, and an edge $\langle u, v \rangle$ means donor of node u can give one kidney to patient of node v
- Find a set of disjoint cycles so as to maximize the number of nodes covered



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

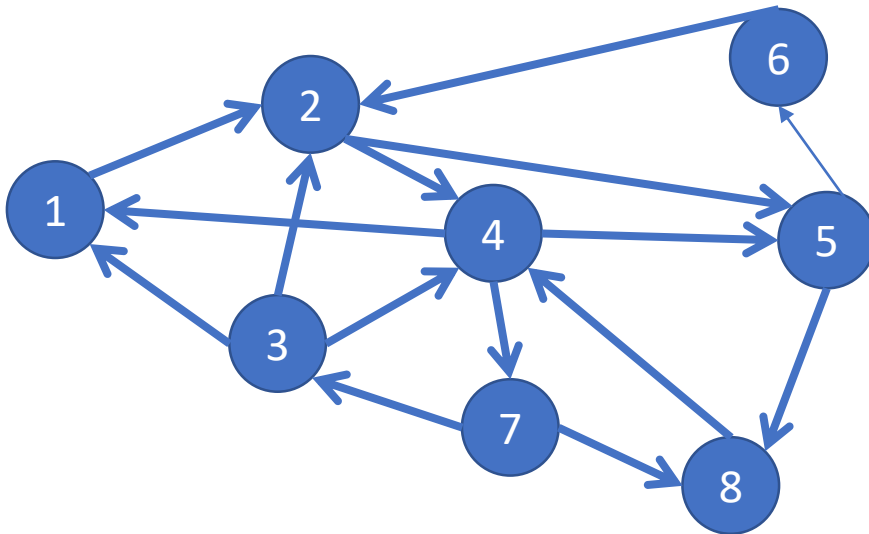


Formulate a Problem as an IP

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

Cryptarithmic

Variables:

IP:

$$\begin{array}{r} \text{TWO} \\ + \text{TWO} \\ \hline \text{FOUR} \end{array}$$

