Making Linear Decisions

Linear Programming Part II

Google Colab

- If you are using
  - https://colab.research.google.com
- Please install CVXOPT and CVXPY
  - Just type (in the notebook on Colab)

```
!pip install cvxopt
!pip install cvxpy
```

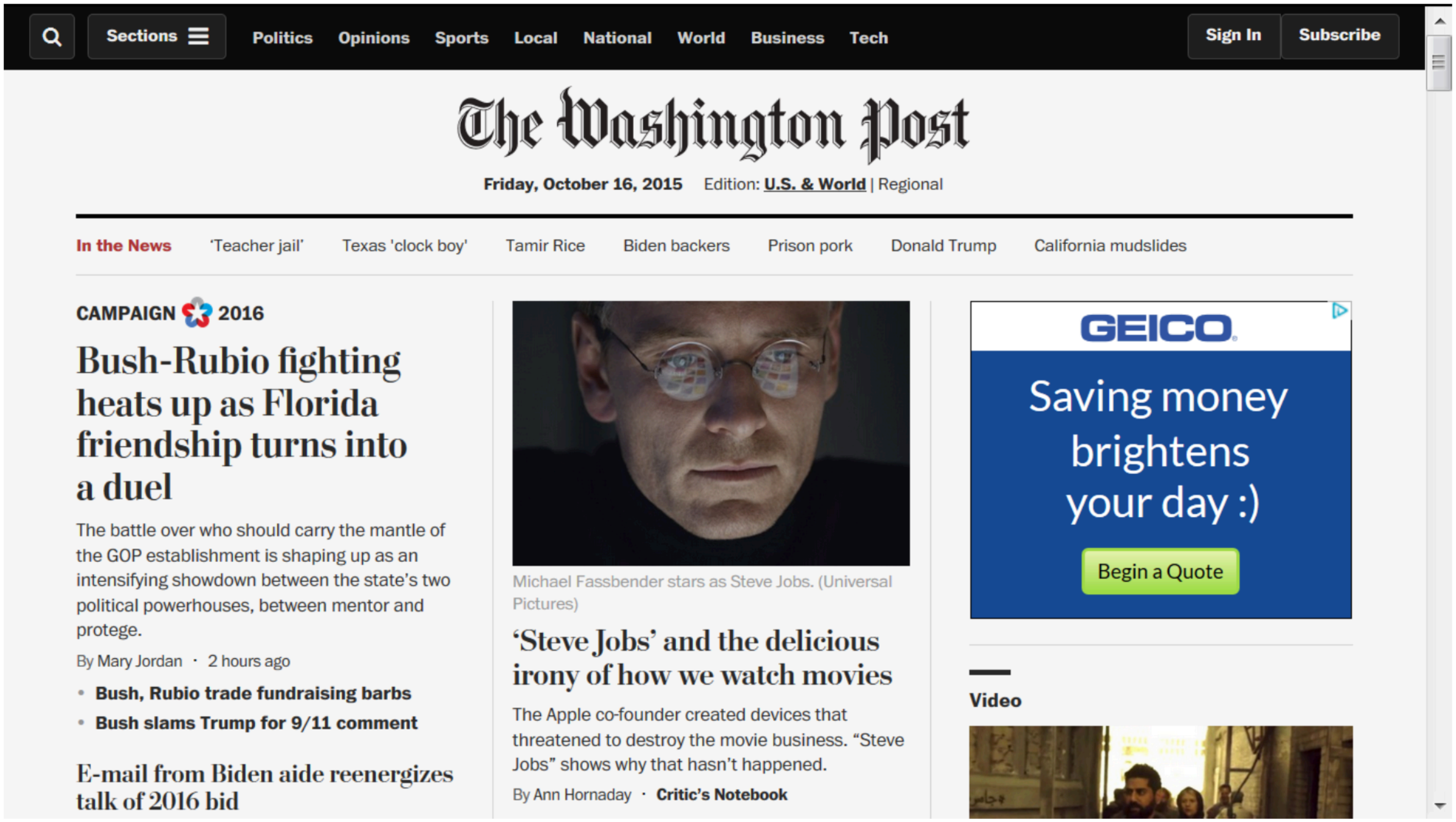
Agenda

- Digital Advertising
- Linear **Integer** Programming
  - Lego Part II
  - Warehouse Location
    - You code
  - Supply Chain Strategy for Online Retailing
    - You model and code

Business Trivia

- Why do firms advertise?

Display Advertising



Display Advertising

- Content publishers such as The New York Times, The Washington Post and The Wall Street Journal generate revenue by using display advertisements.
- The Washington Post's website contains several different sections including Sports and National, as can be seen above. The number of views each section gets per day can be estimated by analyzing historical data. Assume that the Sports section gets six million views per day and the National section get five million views per day.
- Assume four companies, GEICO, Delta, T-Mobile and Capital One, wish to advertise on the Sports and National sections of the Washington Post and they contract directly with the newspaper. For each company, the contract specifies the number of times its display ads are shown in these two sections. The contracts sometimes also specify a total number of page views that can originate from any section of the newspaper. The page views promised by The Washington Post to each advertiser are summarized in the left table on the next slide.

- Assume that the contract also specifies that The Washington Post receives \$2.30 per click-through from each of the four companies. However, not every page view leads to a click. If every 1000 views leads to 5 clicks, the click-through rate is 0.5%. Newspapers use historical data and tracking technologies to determine click-through rates. Assume that the relevant click-through rates are given in the right table below.

Company	Sports	National	Total
GEICO	2 million	1 million	-
Delta	-	1 million	2 million
T-Mobile	1 million	1 million	3 million
Capital One	-	-	2 million

Company	Sports	National
GEICO	2.5%	0.8%
Delta	2.0%	1.0%
T-Mobile	1.0%	3.0%
Capital One	1.5%	2.0%

What is the optimal ad placement policy?

Modeling

This is not an advertising problem

- What are the decisions?
- What is the objective function?
- What are the constraints?
- http://cvxopt.org/userguide/modeling.html

```
import math
import numpy
from cvxopt import matrix
from cvxopt.modeling import variable
from cvxopt.modeling import op
```

Decisions

```
# Definition of the Variables
x11 = variable(1,'Geico Sports')
x12 = variable(1,'Geico National')
x21 = variable(1,'Delta Sports')
x22 = variable(1,'Delta National')
x31 = variable(1,'T-Mobile Sports')
x32 = variable(1,'T-Mobile National')
x41 = variable(1,'Capital One Sports')
x42 = variable(1,'Capital One National')
```

Modeling

- Assume that the contract also specifies that The Washington Post receives $2.30 per click-through from each of the four companies. However, not every page view leads to a click. If every 1000 views leads to 5 clicks, the click-through rate is 0.5%. Newspapers use historical data and tracking technologies to determine click-through rates. Assume that the relevant click-through rates are given in the right table below.

Maximize

$$2.3 \times \sum_{i=1}^{4} \sum_{j}^{2} \kappa_{ij} \times x_{ij}$$

Subject to ???

Company	Sports	National
GEICO	2.5%	0.8%
Delta	2.0%	1.0%
T-Mobile	1.0%	3.0%
Capital One	1.5%	2.0%

Objective Function

```
kappa=matrix([[2.5,2.0,1.0,1.5],[0.8,1.0,3.0,2.0]])
kappa1=kappa[0]
kappa2=kappa[1]
x1=cvx.Int(4,1)
x2=cvx.Int(4,1)

#Objective Function
Z1=sum_entries(kappa1*x1)*2.3/100
Z2=sum_entries(kappa2*x2)*2.3/100
objective = cvx.Maximize((Z1+Z2))
```

Maximize

$$2.3 \times \sum_{i=1}^{4} \sum_{j}^{2} \kappa_{ij} \times x_{ij}$$

Constraints

```
#Constraints
c1=(sum_entries(x1))<=6*1000000#Capacity on Sports
c2=(sum_entries(x2))<=5*1000000#Capacity on Sports
c3=x1[0]>=2*1000000 # Geico must have at least 2m in sports
c4=x2[0]>=1*1000000 # Geico must have at least 1m in sports
c5=x2[1]>=1*1000000 # Delta must have at least 1m in sports
c6=x1[1]+x2[1]==2*1000000 # Delta's total # of impressions must be equal to 2m total
c7=x1[2]>=1*1000000# Tmobile must have at least 1m impression in sports
c8=x2[2]>=1*1000000# Tmobile must have at least 1m impression in national
c9=x1[2]+x2[2]==3*1000000 # Tmobile's total # of impressions must be equal 3m total
c10=x1[3]+x2[3]==2*1000000# Capital Obe's impression equal 2m
c=[c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,x1>=0,x2>=0]
```

Lego Last Week

- You are the manager of a Lego Furniture production facility. The current resources available in your factory are 6 large orange Lego blocks and 8 small green Lego blocks.
- You can make two kinds of furniture with your available Lego blocks. The first kind is a chair, which retails for $10 and takes 2 green blocks and 1 orange block to manufacture. The second one is a table, which retails for $16 and takes 2 green blocks and 2 orange blocks to manufacture. Both pieces are good sellers and you'll be able to sell anything you produce.
- What should you build in order to maximize revenues?

$$
\begin{array}{lll}
\text{Maximize} & 10x_1 + 16x_2 & \\
\text{Subject to} & 2x_1 + 2x_2 & \leq 8 \\
 & x_1 + 2x_2 & \leq 6 \\
 & x_1, x_2 & \geq 0
\end{array}
$$

- # of Chairs (x1): 2
- # of Table (x2): 2
- Optimal Revenue: $52

Lego This Week

- You are the manager of a Lego Furniture production facility. The current resources available in your factory are 6 large orange Lego blocks and 11 small green Lego blocks.
- You can make two kinds of furniture with your available Lego blocks. The first kind is a chair, which retails for $10 and takes 2 green blocks and 1 orange block to manufacture. The second one is a table, which retails for $16 and takes 2 green blocks and 2 orange blocks to manufacture. Both pieces are good sellers and you'll be able to sell anything you produce.
- What should you build in order to maximize revenues?

Maximize $10x_1 + 16x_2$

Subject to

$$
\begin{aligned}
2x_1 + 2x_2 &\le \mathbf{11} \\
x_1 + 2x_2 &\le 6 \\
x_1, x_2 &\ge 0
\end{aligned}
$$

- # of Chairs (x1): ?
- # of Table (x2): ?
- Optimal Revenue: $?

What do you think?

```
from cvxopt import matrix, solvers
import numpy
A = matrix([ [2.0, 1.0,-1.0,0.0], [2.0,2.0,0.0,-1.0] ])
b = matrix([ 11.0, 6.0,0.0,0.0 ])
c = matrix([ -10.0, -16.0 ])
sol=solvers.lp(c,A,b)
print(sol['x'])
print(-sol['primal objective'])
print(-numpy.matmul(numpy.transpose(c),sol['x']))
```

```
     pcost       dcost       gap    pres   dres   k/t
 0: -5.8667e+01 -1.0967e+02  5e+01  3e-01  6e-01  1e+00
 1: -5.6001e+01 -6.3734e+01  5e+00  4e-02  9e-02  4e-01
 2: -5.8012e+01 -5.8384e+01  2e-01  2e-03  4e-03  3e-02
 3: -5.8000e+01 -5.8004e+01  2e-03  2e-05  5e-05  4e-04
 4: -5.8000e+01 -5.8000e+01  2e-05  2e-07  5e-07  4e-06
 5: -5.8000e+01 -5.8000e+01  2e-07  2e-09  5e-09  4e-08
Optimal solution found.
[ 5.00e+00]
[ 5.00e-01]

58.000000007119844
[[ 58.00000001]]
```

Better Modeling

- Solution by brute force yields a solution that does not make sense
- Rounding up the solutions will give a point that is outside of the feasibility set
- We need to take into account the integer nature of the decision variables

Integer Programming

Tool	Decision $\boldsymbol{x}$	Objective $f(.)$	Constraint $g(.)$
Linear Programming I	$x \in R^+$	$c_1x_1 + c_2x_2$	$a_1x_1 + a_2x_2 < b$
Linear Programming II	$x = \{0,1\}$	$c_1x_1 + c_2x_2$	$a_1x_1 + a_2x_2 < b$
Non-Linear Programming	$x \in R^+$	$f'' \neq 0$	$g(x)$
Dynamic Optimization	$x_1, x_2, \dots x_T$	$\Sigma_t \Pi_t(x_t)$	$\Pi_t = g(\Pi_{t-1})$
Stochastic Optimization	$x_1, x_2, \dots x_T$	$\Sigma_t \mathbb{E}[\Pi_t(x_t)]$	$\Pi_t = g(\Pi_{t-1}) + \epsilon$
Game Theory and incentives Design	My decision (x) vs. your decision (y)	$f(x, y)$	$g(x, y)$

Solution to by Integer Programming

```python
import cvxopt
import numpy as np
from   cvxopt import glpk

c=cvxopt.matrix([ -10.0, -16.0 ],tc='d')
A=cvxopt.matrix([ [2.0, 1.0,-1.0,0.0], [2.0,2.0,0.0,-1.0] ],tc='d')
b=cvxopt.matrix([ 11.0, 6.0,0.0,0.0 ],tc='d')
(status, x)=cvxopt.glpk.ilp(c,A,b,I=set([0,1]))
print (status)
print (x[0],x[1] )
print (-sum(c.T*x))
```

```
optimal
4.0 1.0
56.0
```

Solution with CVXOPT

```python
import cvxpy as cvx
import numpy
# Problem data.

# Construct the problem.
#x1 = cvx.Variable(1)
#x2 = cvx.Variable(1)

x1=cvx.Int()
x2=cvx.Int()

#objective
objective = cvx.Maximize(10*x1+16*x2)
#constraint
con = [2*x1+2*x2<=11,x1+2*x2<=6,0 <= x1,0 <= x2]

#solving
prob = cvx.Problem(objective, con)
result = prob.solve()

print (x1.value)
print (x2.value)
print (objective.value)
```

```
3.99999998824
1.00000000668
55.9999999892
```

Solution with CVXPY

Solution to by Integer Programming

```python
import cvxopt
import numpy as np
from   cvxopt import glpk

c=cvxopt.matrix([ -10.0, -16.0 ],tc='d')
A=cvxopt.matrix([ [2.0, 1.0,-1.0,0.0], [2.0,2.0,0.0,-1.0] ],tc='d')
b=cvxopt.matrix([ 11.0, 6.0,0.0,0.0 ],tc='d'))
(status, x)=cvxopt.glpk.ilp(c,A,b,I=set([0,1]))
print (status)
print (x[0],x[1] )
print (-sum(c.T*x))
```

I declare in the solvers which are the variables that are integers

```
optimal
4.0 1.0
56.0
```

Solution with CVXOPT

```python
import cvxpy as cvx
import numpy
# Problem data.

# Construct the problem.
#x1 = cvx.Variable(1)
#x2 = cvx.Variable(1)

x1=cvx.Int()
x2=cvx.Int()

#objective
objective = cvx.Maximize(10*x1+16*x2)
#constraint
con = [2*x1+2*x2<=11,x1+2*x2<=6,0 <= x1,0 <= x2]

#solving
prob = cvx.Problem(objective, con)
result = prob.solve()

print (x1.value)
print (x2.value)
print (objective.value)
```

```
3.99999998824
1.00000000668
55.9999999892
```

Solution with CVXPY

Solution to by Integer Programming

```
import cvxopt
import numpy as np
from   cvxopt import glpk

c=cvxopt.matrix([ -10.0, -16.0 ],tc='d')
A=cvxopt.matrix([ [2.0, 1.0,-1.0,0.0], [2.0,2.0,0.0,-1.0] ],tc='d')
b=cvxopt.matrix([ 11.0, 6.0,0.0,0.0 ],tc='d')
(status, x)=cvxopt.glpk.ilp(c,A,b,I=set([0,1]))
print (status)
print (x[0],x[1] )
print (-sum(c.T*x))
```

```
optimal
4.0 1.0
56.0
```

Solution with CVXOPT

```
import cvxpy as cvx
import numpy
# Problem data.

# Construct the problem.
#x1 = cvx.Variable(1)
#x2 = cvx.Variable(1)

x1=cvx.Int()
x2=cvx.Int()
```

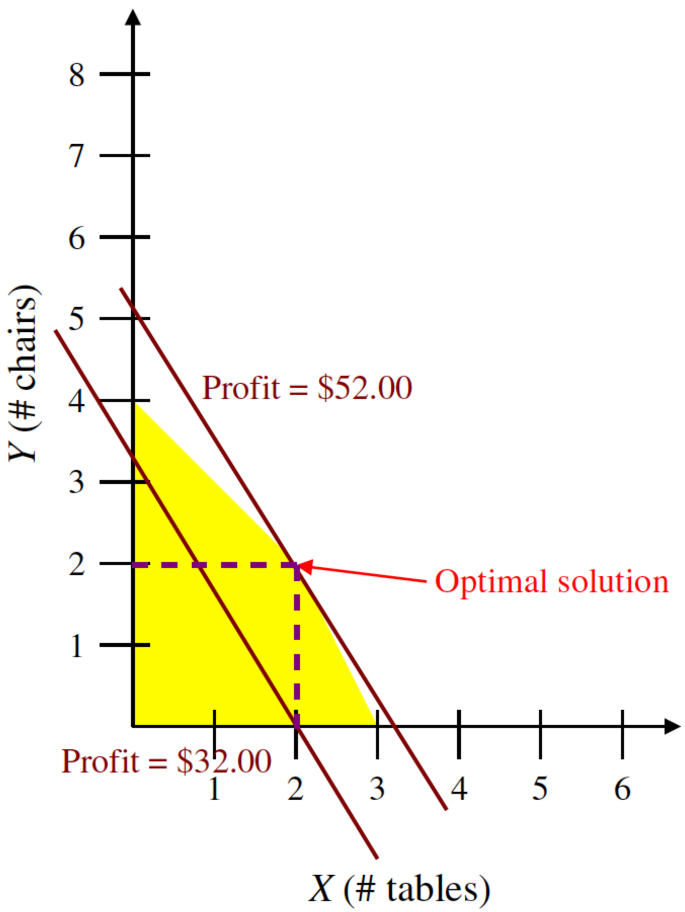
The decision variables are Integers (in cvx, i.e., "Int( )")

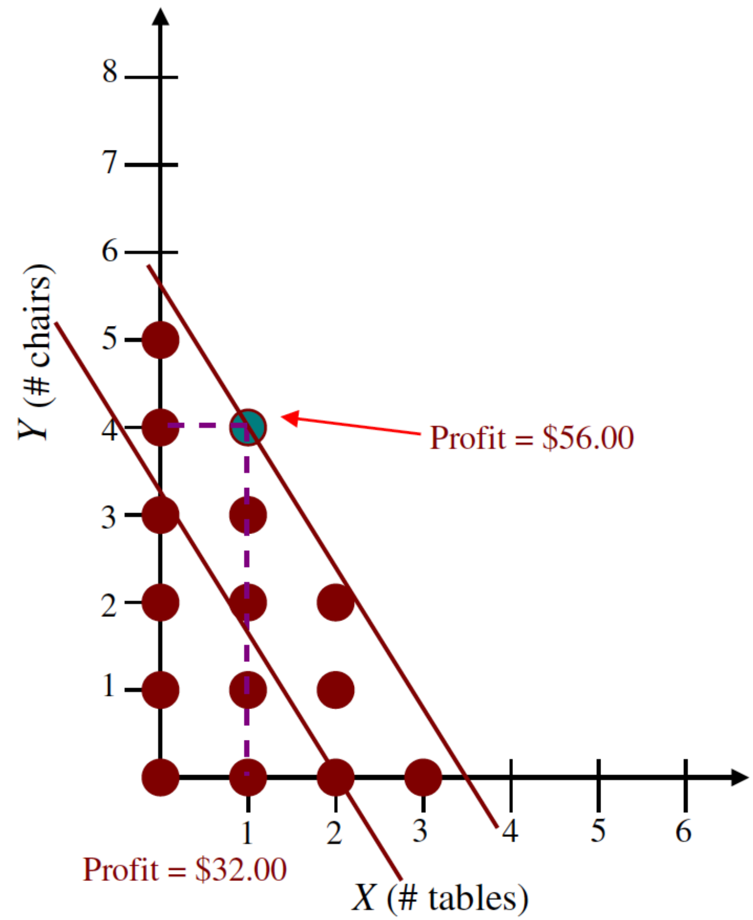
```
#objective
objective = cvx.Maximize(10*x1+16*x2)
#constraint
con = [2*x1+2*x2<=11,x1+2*x2<=6,0 <= x1,0 <= x2]

#solving
prob = cvx.Problem(objective, con)
result = prob.solve()

print (x1.value)
print (x2.value)
print (objective.value)
```

```
3.99999998824
1.00000000668
55.9999999892
```

Solution with CVXPY

Visually


Profit = $52.00
Optimal solution
Profit = $32.00
Y (# chairs)
X (# tables)
1
2
3
4
5
6
7
8


Profit = $56.00
Profit = $32.00
Y (# chairs)
X (# tables)
1
2
3
4
5
6
7
8

Integer Programming

- Decision variables are not continuous but discrete
- Mathematically, shadow price cannot be computed
  - Resort to “sensitivity analysis”

Binary Variables and Binary Choice

- A binary variable, which takes on the values zero or one, can be used to represent a "go / no-go" decision.
  - $X=\{0,1\}$
- We can think in terms of discrete projects, where the decision to accept the project is represented by the value 1, and the decision to reject the project is represented by the value 0.

0-1 Constraints

- When x_i and x_j represent binary variables designating whether projects i and j have been chosen, the following special constraints may be formulated:
  - At most <u>k out of n</u> projects will be chosen:

    $$\sum_j x_j \leq k$$

  - Project j is <u>conditional</u> on project i:

    $$x_j - x_i \leq 0$$

  - Project i is a <u>corequisite</u> for project j:

    $$x_j - x_i = 0$$

  - Projects i and j are <u>mutually exclusive</u>:

    $$x_i + x_j \leq 1$$

Warehouse Location Problem

- You have to decide where to open warehouses to serve a customers located in four possible locations. The three possible locations are Harlingen, Memphis and Ashland. The four customer locations are NYC, LA, Chitown and Houston. The distance (cost) between the candidate locations and customer locations are given in the following table

	NYC	LA	Chicago	Houston
Harlingen	1956	1606	1410	330
Memphis	1096	1792	531	567
Ashland	485	2322	324	1236

Warehouse Location Problem

- You have the decide where to open warehouses such that to minimize the cost of shipping while respecting some constraints. For instance, all customers must be served and you cannot open more than two warehouses.

- Questions:
  – How many warehouses do you open?
  – Where are you going to location your warehouse(s)?
  – Who ships to whom?
  – What is the optimum?

p-Median Problem

Minimize

$$\sum_{n=1}^{3}\sum_{m=1}^{4} d_{n,m}x_{n,m}$$

Subject to

$$\sum_{n=1}^{3} x_{n,m} = 1 \; for \; All \; m$$

$$x_{n,m} \leq y_n \; for \; All \; n \; and \; m$$

$$\sum_{n=1}^{3} y_n \leq 2$$

$$0 \leq x \leq 1$$

$$y \in \{0,1\}$$

You code

Linking Constraints with Fixed Costs

- In many situations, activity costs are composed of fixed costs and variable costs, with only the variable costs being proportional to activity level.

- Integer programming model, we can be helpful in integrating fixed cost components.

Incorporating Fixed Costs

- We separate the fixed and variable components of cost.
- In algebraic terms, we write cost as

$$Cost = f\,y + cx$$

Where f represents the fixed cost, and c represents the linear variable cost.

- The variables x and y are decision variables, where x is a normal (continuous) variable, and y is a binary variable.

Linking Constraint

- To achieve consistent linking of the two variables, we add the following generic **linking constraint** to the model:

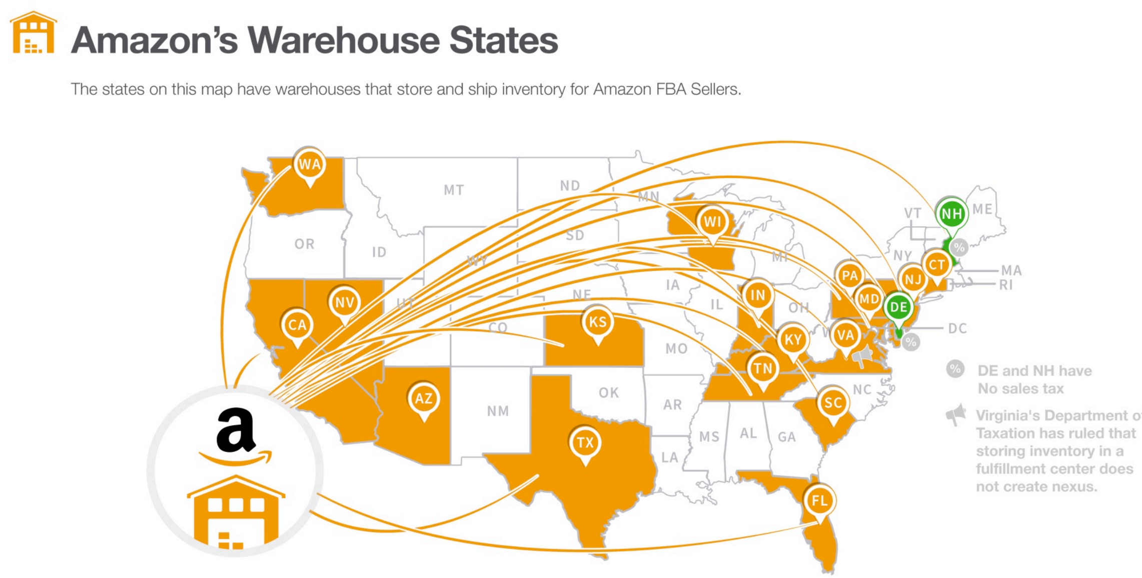
  $$x <= My$$

  where the number M represents an upper bound on the variable x.

- In other words, M is at least as large as any value we can feasibly choose for x.

Amazon

- In the last few years, due to the expansion of the services such as *Amazon Prime* and *Amazon Prime Now*, the importance of effectively selecting the location of new warehouses and/or choosing the right local warehouse to work with has dramatically increased



- Because of the emerging demand, Amazon decided to start new warehouses in the Gulf Coast. Based on their studies, four potential locations for warehouses have been identified, each of which having a specific capacity and also a specific average cost-per-unit for shipping to any of the five Gulf Coast states. The capacity of each warehouse and its operating cost, in addition to the average cost-per-unit for shipping a product to any of the states are given in below, as well as demand from each state.

	TX	LA	MS	AL	FL	Ops. Cost	Capacity
WH 1	$2.00	$4.00	$3.00	$2.50	$1.50	$10,000	14000
WH 2	$3.00	$2.50	$3.75	$2.00	$1.75	$8,000	18000
WH 3	$2.75	$2.00	$3.25	$3.00	$4.00	$7,000	16500
WH 4	$1.25	$1.75	$4.00	$3.50	$3.00	$12,000	16000

	TX	LA	MS	AL	FL
Demand	8000 units	3500 units	4000 units	2000 units	7000 units

Questions

- What is the minimal total cost for Amazon's supply chain?
- Which warehouses should Amazon operate?

- Additional information on such problems:
  – http://examples.gurobi.com/facility-location/
  – https://www.sciencedirect.com/science/article/abs/pii/S0377221703008191
  – https://optimization.mccormick.northwestern.edu/index.php/Facility_location_problems

Modeling

- What are the decision variables?
- What are the constraints?
- What is the objective function?

```
import math
import numpy as np
import cvxpy as cvx
```

Business Trivia

- What are the differences between:
  - Fixed costs vs. variable costs
  - Economies of scale vs. Economies of scope