DSC3214
Introduction to Optimization
Lecture 3

Schedule

- Lecture 1: Linear Algebra Basics
- Lecture 2: Introduction to Python
- Lecture 3: Linear Optimization Models
- Lecture 4: Hands-on with Gurobi
- Lecture 5: Geometry of Linear Optimization
- Lecture 6: Sensitivity Analysis and Duality
- Lecture 7: Network Optimization
- Lecture 8: Discrete Optimization
- Lecture 9: Quadratic Optimization
- Lecture 10: Stochastic Optimization
- Lecture 11: Robust Optimization
- Lecture 12: Students' presentations

Manufacturing Problem

- Data
  - n products, m raw materials
  - c_j: profit of j
  - b_i: available units of material i
  - a_{ij}: # units of material i product j needs in order to be produced

Manufacturing Problem

- Formulation

$$x_j = \text{amount of product } j \text{ produced}$$

$$\begin{aligned} \text{max} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & a_{11}x_1 + \ldots + a_{1n}x_n \le b_1 \\ & \vdots \\ & a_{m1}x_1 + \ldots + a_{mn}x_n \le b_m \\ & x_1, \ldots, x_n \ge 0 \end{aligned}$$

Linear Optimization

- Also called *Linear Programming (LP)*
- The objective function and all constraints are linear functions of the decision variables
- Real-world LPs are solved which contain hundreds of thousands to millions of variables
- Examples:
  – Product Mix
  – Scheduling/Allocation
  – Routing/Logistics
  – Supply Chain Optimization
  – Facility Location
  – Financial Planning/Asset Management

Basic LP Assumptions

- Linearity
  - The objective function and constraints are linear functions of the decision variables.
- Divisibility
  - Each variable is allowed to assume fractional values.
- Data certainty
  - Coefficients are not random variables.

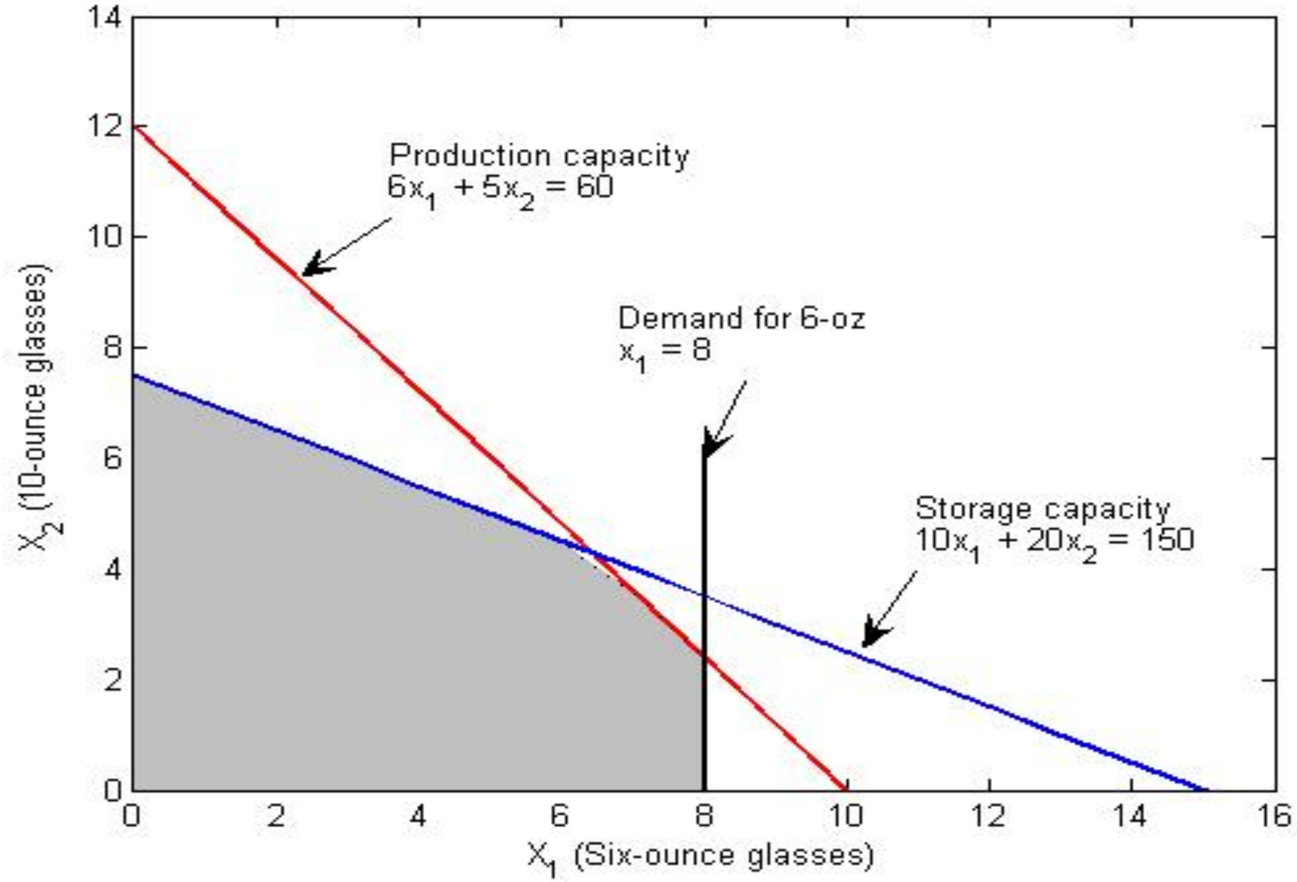
An Example Problem

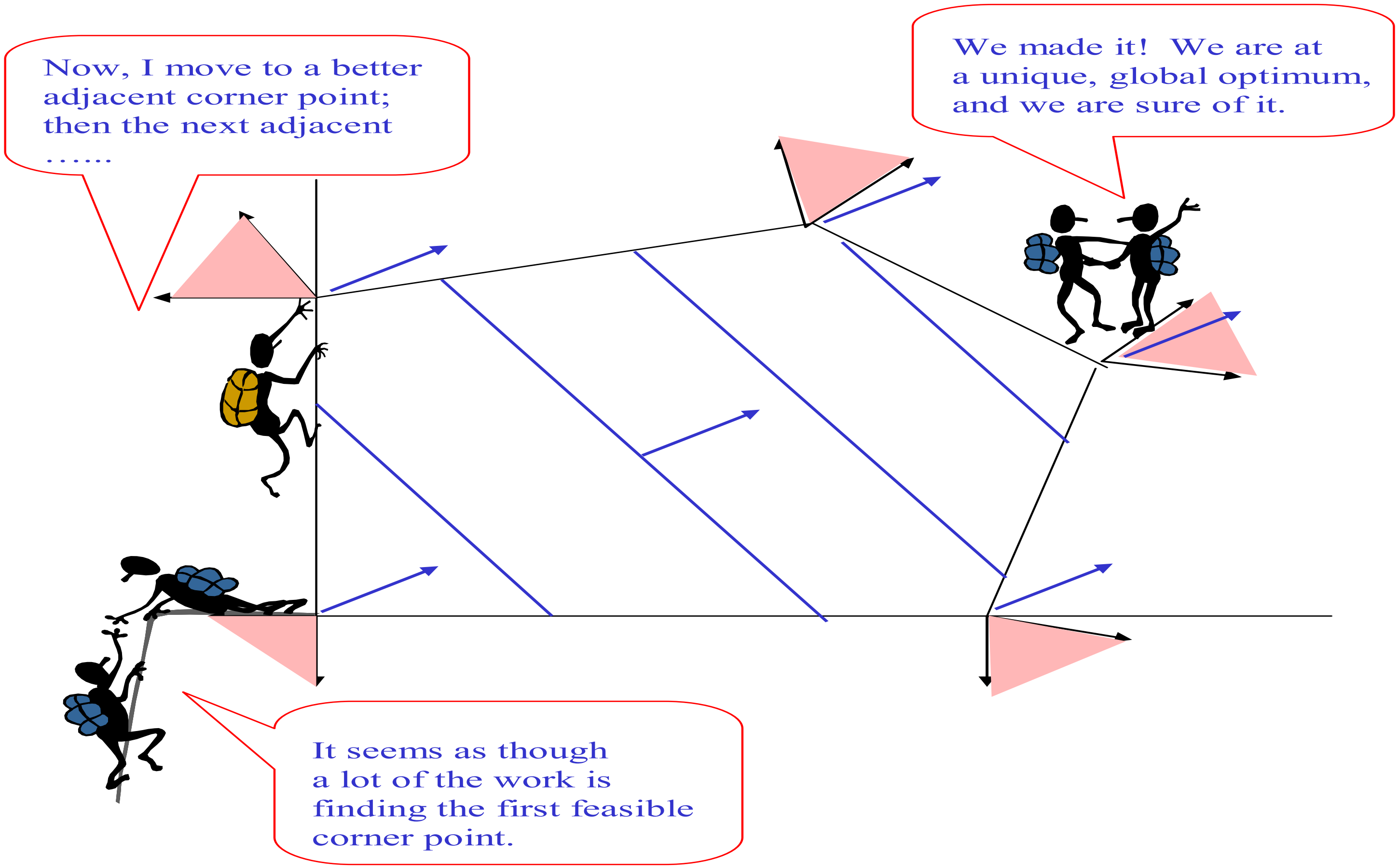
- The Custom Molder Problem
  - Suppose that a custom molder has one injection-molding machine with two different dies to fit the machine. Due to differences in number of cavities and cycle times, with the first die he can produce 100 cases of six-ounce juice glasses in six hours, while with the second die he can produce 100 cases of ten-ounce fancy cocktail glasses in five hours. He prefers to operate only on a schedule of 60 hours of production per week. He stores the week's production in his own stockroom where he has an effective capacity of 15,000 cubic feet. A case of six-ounce juice glasses requires 10 cubic feet of storage space, while a case of ten-ounce cocktail glasses requires 20 cubic feet due to special packaging. The contribution of the six-ounce glasses is $5.00 per case; however, the only customer available will not accept more than 800 cases per week. The contribution of ten-ounce cocktail glasses is $4.50 per case and there is no limit on the amount that can be sold.

LP Formulation

- Decision variables
  - x_1 : Number of six-oz glass cases to produce (in 100s)
  - x_2 : Number of ten-oz glass cases to produce (in 100s)
- Objective function
  - $500\ x_1 + 450\ x_2$
- Constraints
  - Production capacity : $6\ x_1 + 5\ x_2 \leq 60$
  - Storage capacity : $10\ x_1 + 20\ x_2 \leq 150$
  - 6-oz Demand : $x_1 \leq 8$
  - Non-negativity : $x_1, x_2 \geq 0$

LP feasible region





Linear programming optimality

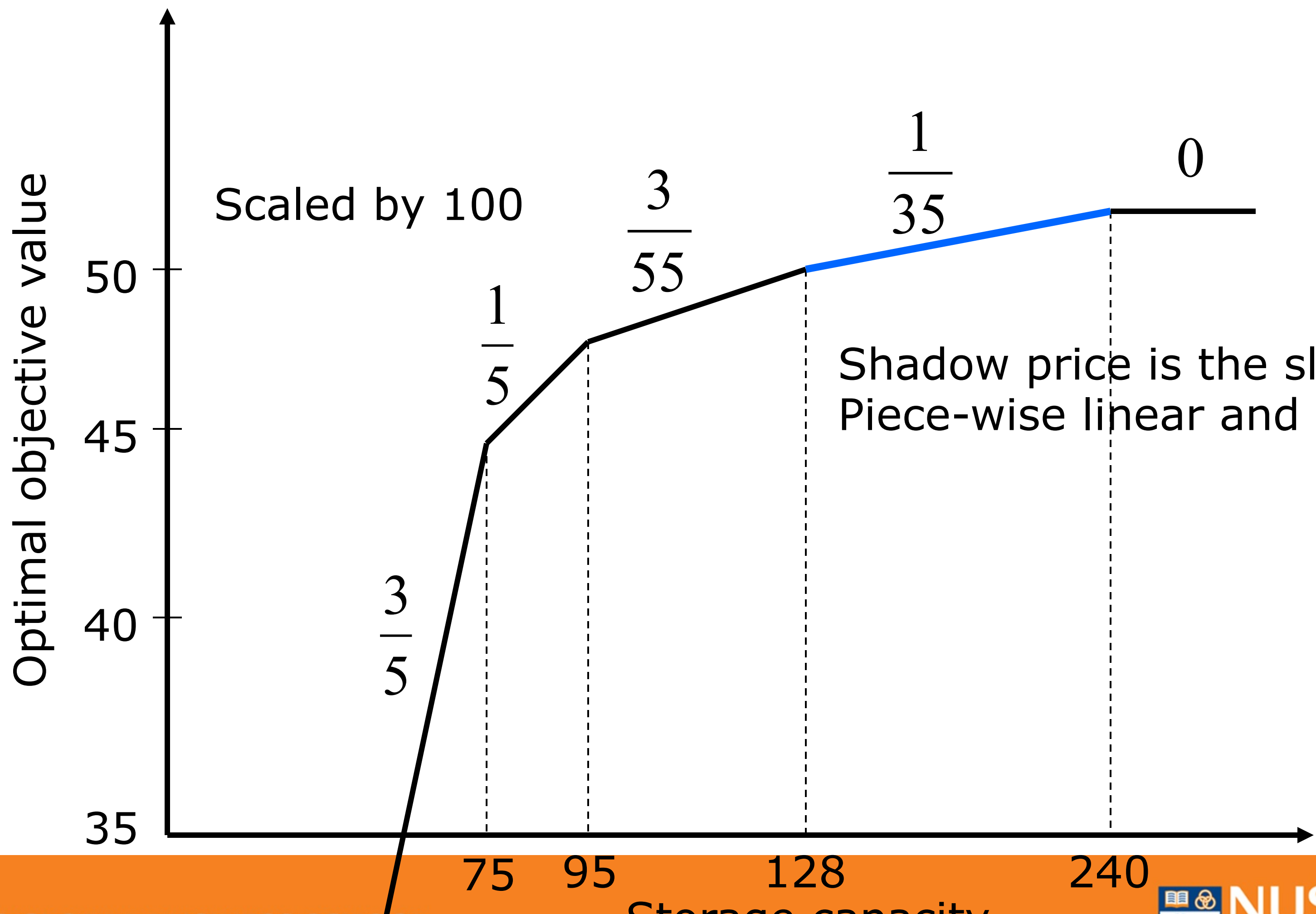
Perturbing the rhs value of a constraint

- Suppose the number of hours of molding-machine capacity was increased from 60 hours to 61 hours
  - Can you compute the new co-ordinates of the optimal vertex?
  - What is the optimal objective value?
  - What is the improvement in the objective function?
- Can you perform a similar calculation for the storage-capacity constraint

Shadow prices and validity range

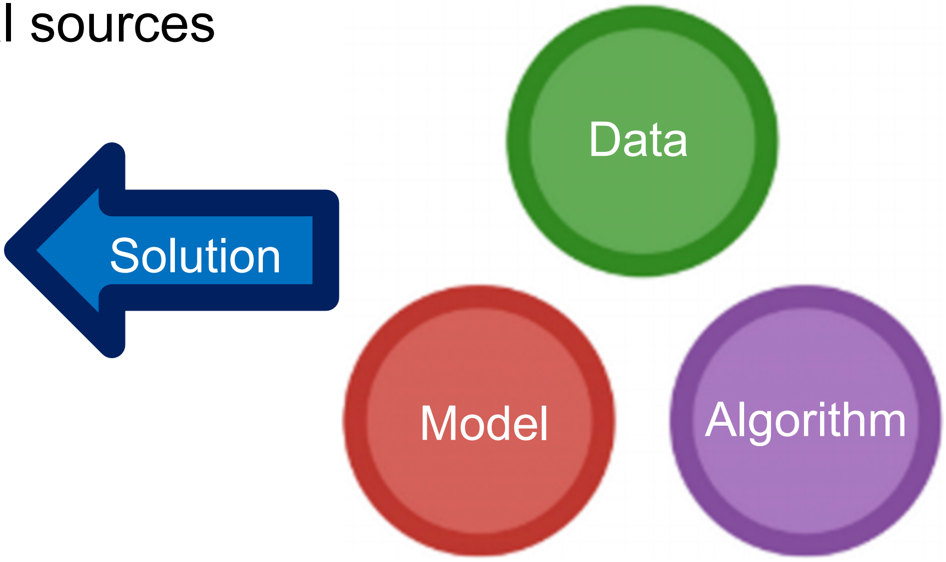
- Relaxing the right-hand side (rhs) of a binding constraint improves the optimal objective function value
  - Relaxing the constraint enlarges the feasible region
  - As the optimal vertex moves, its objective function value improves
  - The marginal improvement in the objective function value caused by relaxing the rhs one unit is called the constraint's ***shadow price***
- The shadow price is the derivative! It is defined based on the math formulation of the LP
  - The shadow price of production capacity is $G'(60) = 78\ 4/7$.
- How do we compute the range of validity of this shadow price?

Shadow prices always show "diminishing returns to scale"



Optimization Software

- Why we need optimization software?
  - Large-scale real world optimization problem
  - Concise and readable syntax to the mathematical notation
  - Easy model development, debug and maintenance
  - Clear output presentation
  - Exchange data with external sources



Optimization Software

- Algebraic Modeling Language for Optimization
  - A high-level computer programming languages for modeling optimization problems
  - E.g. AIMMS, EXCEL Solver, AMPL, ROME or programming languages: **Python**, C, C++, Julia
- Solver
  - Well-established algorithms to solve the program
  - Cplex, **Gurobi**, Mosek, SDPT3, etc.

Gurobi Package in Python

- Python + Gurobi
- Code in programming language Python
- Call external solver
  - Gurobi

Gurobi Implementation

$$
\begin{array}{lll}
\text{min(max)} & c_1x_1 + \ldots + c_nx_n & \\
\text{s.t.} & a_{11}x_1 + \ldots + a_{1n}x_n = b_1 & \text{Equality constraints} \\
& \vdots & \\
& a_{41}x_1 + \ldots + a_{4n}x_n \leq b_4 & \text{Inequality constraints} \\
& \vdots & \\
& a_{81}x_1 + \ldots + a_{8n}x_n \geq b_8 & \text{Inequality constraints} \\
& \vdots & \\
& x_1, \ldots, x_n \geq 0, & \text{Nonnegative constraints}
\end{array}
$$

- Decision variables
- Parameters
- Constraints
- Objective function

Gurobi Example

- Import “gurobipy” package

```
from gurobipy import *
from math import sqrt
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import xlrd
```

Gurobi Example

- Create model and optimize

```
# Create a new model
m = Model("simplelp")

# Create variables
x1 = m.addVar(ub = 10, name="x1")
x2 = m.addVar(name="x2")
x3 = m.addVar(name="x3")

# Set objective
m.setObjective(x1 + 2 * x2 + 5 * x3, GRB.MAXIMIZE)

# Add constraint:
m.addConstr(-x1 + x2 + 3*x3 <= -5, "c0")

# Add constraint:
m.addConstr(x1 + 3*x2 - 7*x3 >= 10, "c1")

m.optimize()
```

Gurobi Example

- An alternative way

```
# Create a new model
m = Model("simplelp")

# Create variables
ubd = [10, GRB.INFINITY, GRB.INFINITY]
x = m.addVars(3, ub = ubd, name="x")

# Set objective
c = [1, 2, 5]

m.setObjective(  sum(x[i] * c[i] for i in range(3)) , GRB.MAXIMIZE)

# Add constraints:
b = [-5, -10]
A = np.array([[-1, 1, 3], [-1, -3, 7]])

m.addConstrs( quicksum(A[i,j] * x[j] for j in range(3)) <= b[i] for i in range(2))

m.optimize()
```

Gurobi Example

- Print the optimal solution, objective value and dual values

```
# print optimal solutions
for v in m.getVars():
    print('%s %g' % (v.varName, v.x))

# print optimal value
print('Obj: %g' % m.objVal)

# print dual values to all constraints
print(m.getAttr("Pi", m.getConstrs()))
```

Production Scheduling Problem

- Regular production: $190/unit up to 160 units per week
- Overtime production: $260/unit up to 50 units per week
- Surplus units can be carried over to next week. Holding cost is $10/unit per week.
- Find the minimum cost production schedule.

Week	Orders
1	105
2	170
3	230
4	180
5	150
6	250

Production Scheduling Problem

Decision variables:

r_i: Number of units produced under regular hours in week $i = 1, \ldots, 6$

v_i: Number of units produced under overtime hours in week $i = 1, \ldots, 6$

s_i: Number of units brought over from week i to week $i+1$, $i = 1, \ldots, 5$

Production Scheduling Problem

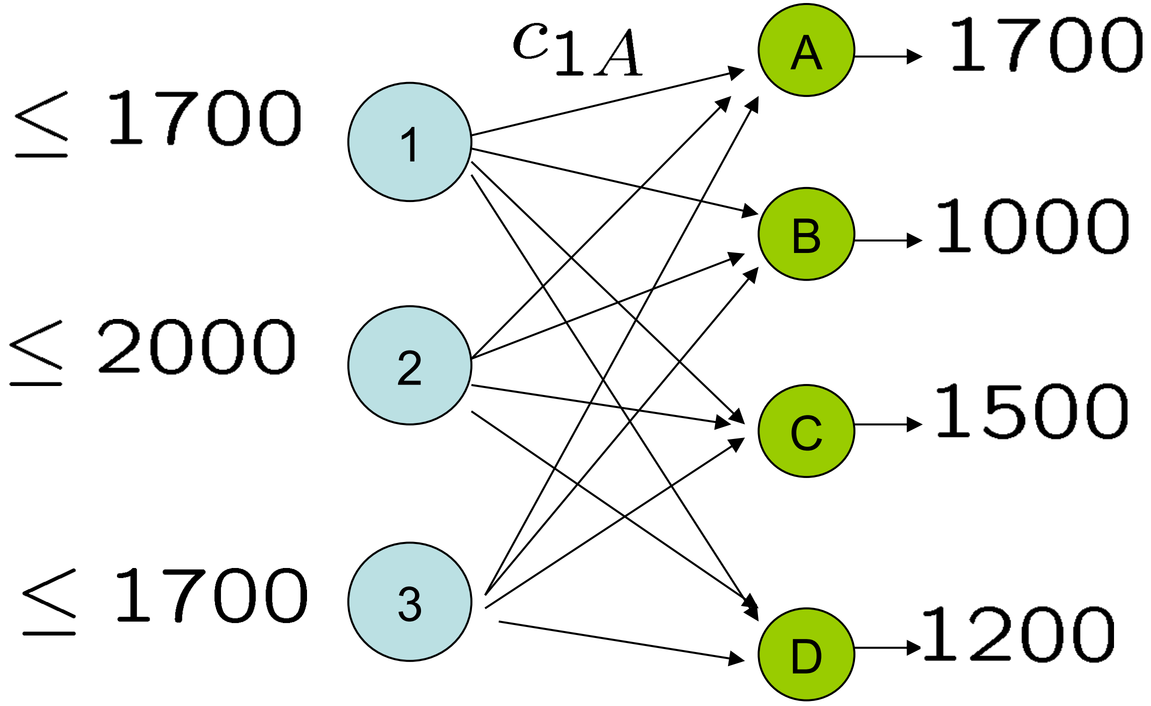
$$
\begin{array}{ll}
\text{min} & 190(r_1+\ldots+r_6)+260(v_1+\ldots+v_6)+10(s_1+\ldots+s_5) \\
\text{s.t.} & r_1+v_1=s_1+105 \\
 & s_1+r_2+v_2=s_2+170 \\
 & s_2+r_3+v_3=s_3+230 \\
 & s_3+r_4+v_4=s_4+180 \\
 & s_4+r_5+v_5=s_5+150 \\
 & s_5+r_6+v_6=250 \\
 & r_1,\ldots,r_6 \le 160 \\
 & v_1,\ldots,v_6 \le 50 \\
 & r_1,\ldots,r_6,v_1,\ldots,v_6,s_1,\ldots,s_5 \ge 0
\end{array}
$$

Transportation Problem

A company has three PC assembly plants at locations, 1, 2 and 3, with monthly production capacity of 1700 units, 2000 units, and 1700 units, respectively. Their PC's are sold through four retail outlets in locations A, B, C,D, with monthly orders of 1700 units, 1000 units, 1500 units, and 1200 units respectively.

Shipping costs ($/unit)	A	B	C	D
1	5	3	2	6
2	7	7	8	10
3	6	5	3	8

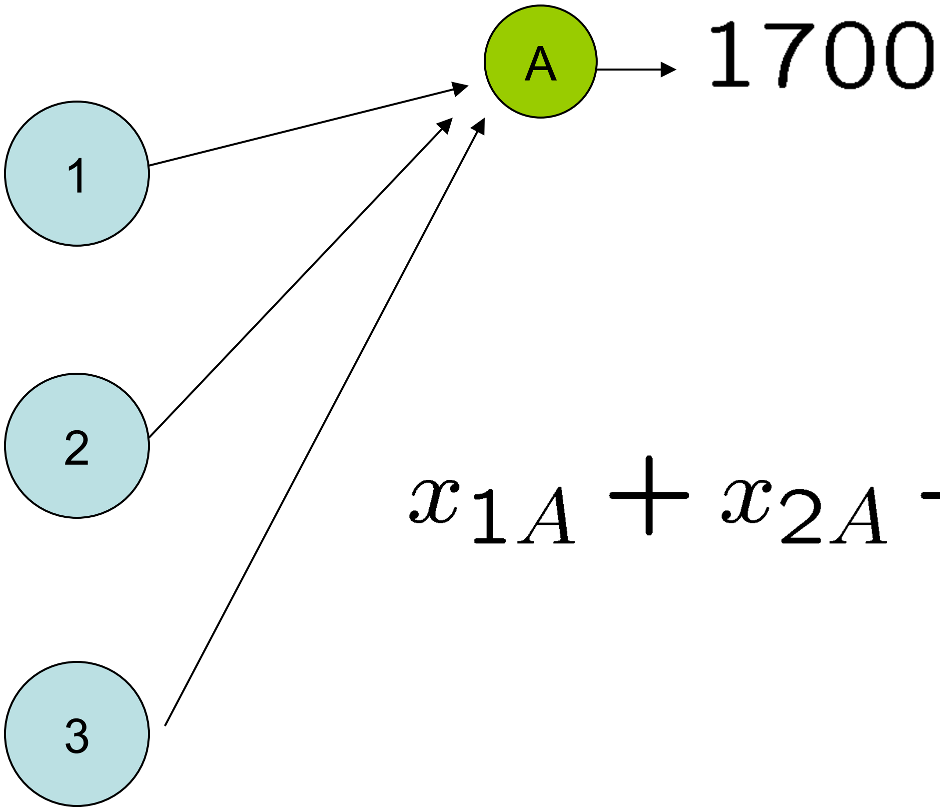
Transportation Problem



Decision Variables

$x_{ij} =$ number of units to send $i \to j$

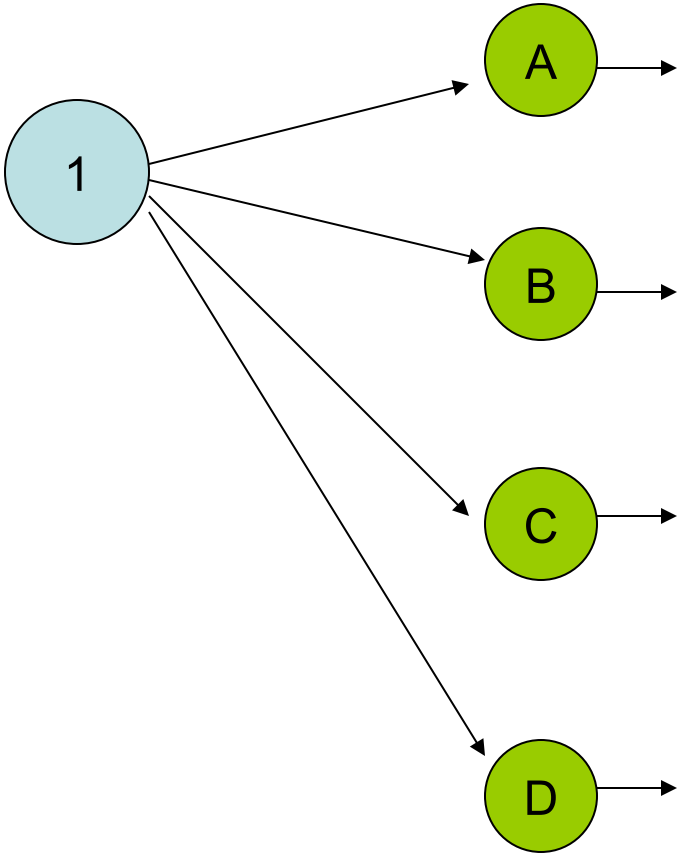
Transportation Problem



$$x_{1A} + x_{2A} + x_{3A} = 1700$$

Transportation Problem

≤ 1700



$$x_{1A} + x_{1B} + x_{1C} + x_{1D} \leq 1700$$

Transportation Problem

$$\begin{aligned}
\text{min} \quad & c_{1A}x_{1A} + c_{1B}x_{1B} + c_{1C}x_{1C} + c_{1D}x_{1D} + \\
& c_{2A}x_{2A} + c_{2B}x_{2B} + c_{2C}x_{2C} + c_{2D}x_{2D} + \\
& c_{3A}x_{3A} + c_{3B}x_{3B} + c_{3C}x_{3C} + c_{3D}x_{3D} \\
\text{s.t.} \quad & x_{1A} + x_{2A} + x_{3A} = d_A \\
& x_{1B} + x_{2B} + x_{3B} = d_B \\
& x_{1C} + x_{2C} + x_{3C} = d_C \\
& x_{1D} + x_{2D} + x_{3D} = d_D \\
& x_{1A} + x_{1B} + x_{1C} + x_{1D} \leq s_1 \\
& x_{2A} + x_{2B} + x_{2C} + x_{2D} \leq s_2 \\
& x_{3A} + x_{3B} + x_{3C} + x_{3D} \leq s_3 \\
& x_{1A}, x_{1B}, x_{1C}, x_{1D}, x_{2A}, x_{2B}, x_{2C}, x_{2D}, \\
& x_{3A}, x_{3B}, x_{3C}, x_{3D} \geq 0
\end{aligned}$$

Transportation Problem

- General representation

Sets:

- P: set of plants
- R: set of retail outlets

Parameters:

- s_i: supply of i plant, $i \in P$
- d_j: demand from outlet j, $j \in R$
- c_{ij}: unit transportation cost from i to j, $i \in P, j \in R$

Variables:

- x_{ij}: num. items transport from i to j, $i \in P, j \in R$

Transportation Problem

- Formulation

$$
\begin{aligned}
\min \quad & \sum_{i \in P} \sum_{j \in R} c_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{i \in P} x_{ij} = d_j & \forall j \in R \\
& \sum_{j \in R} x_{ij} \leq s_i & \forall i \in P \\
& x_{ij} \geq 0 & \forall i \in P, j \in R
\end{aligned}
$$

Scheduling Problem

- Hospital want to make weekly shift for its nurses
- d_j: demand for nurses in day j, $j \in \{1, \ldots, 7\}$
- Every nurse works 5 days in a row
- Objective: hire minimum number of nurses

Scheduling Problem

Mon	Tue	Wed	Thu	Fri	Sat	Sun
x_1	x_1	x_1	x_1	x_1		
	x_2	x_2	x_2	x_2	x_2	
		x_3	x_3	x_3	x_3	x_3
x_4			x_4	x_4	x_4	x_4
x_5	x_5			x_5	x_5	x_5
x_6	x_6	x_6			x_6	x_6
x_7	x_7	x_7	x_7			x_7

Scheduling Problem

x_i: # of nurses starting their week on day i.

$$
\begin{array}{rl}
\min & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \\
\text{s.t} & x_1 + x_4 + x_5 + x_6 + x_7 \geq d_1 \\
& x_1 + x_2 + x_5 + x_6 + x_7 \geq d_2 \\
& x_1 + x_2 + x_3 + x_6 + x_7 \geq d_3 \\
& x_1 + x_2 + x_3 + x_4 + x_7 \geq d_4 \\
& x_1 + x_2 + x_3 + x_4 + x_5 \geq d_5 \\
& x_2 + x_3 + x_4 + x_5 + x_6 \geq d_6 \\
& x_3 + x_4 + x_5 + x_6 + x_7 \geq d_7 \\
& x_i \geq 0
\end{array}
$$

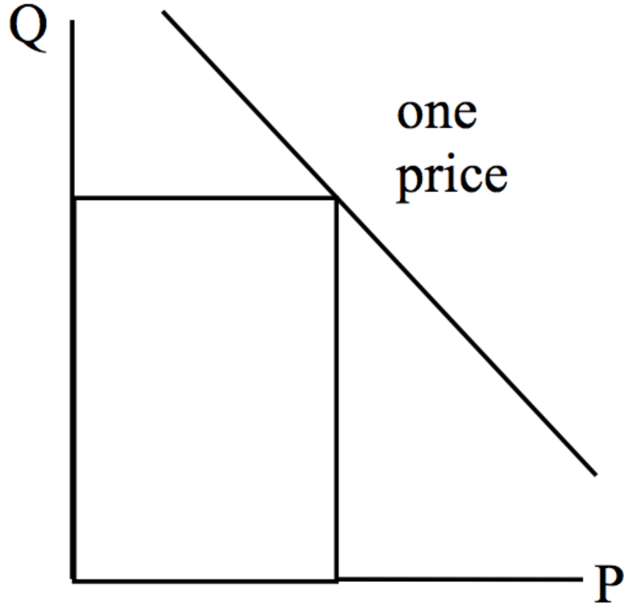
Revenue Management

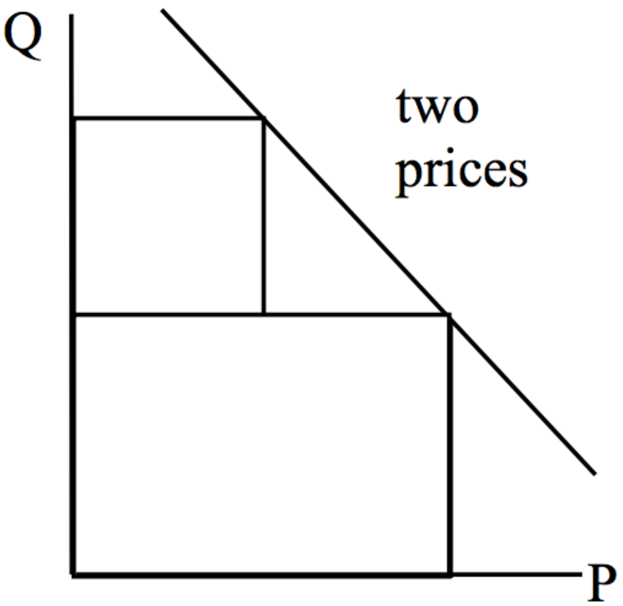
- The Airline industry in US
  - Deregulation in 1978.
    - Prior to Deregulation: carriers only allowed to certain routes. Hence airlines such as Northwest, Eastern Southwest etc.
    - Fares determined by Civil Aeronautics Board (CAB) based on mileage and other costs - (CAB no longer exists)
  - Post Deregulation
    - anyone can fly anywhere
    - fares determined by carrier and the market

Revenue Management

- Economics
  - Huge sunk and fixed costs
  - Very low variable costs per passenger
    - $10 per passenger or less
  - Strong economically competitive environment
  - Near perfect information and negligible cost of information
  - Highly perishable inventory
  - **Result:** Multiple fares

Multiple fare classes: a monopolist's perspective

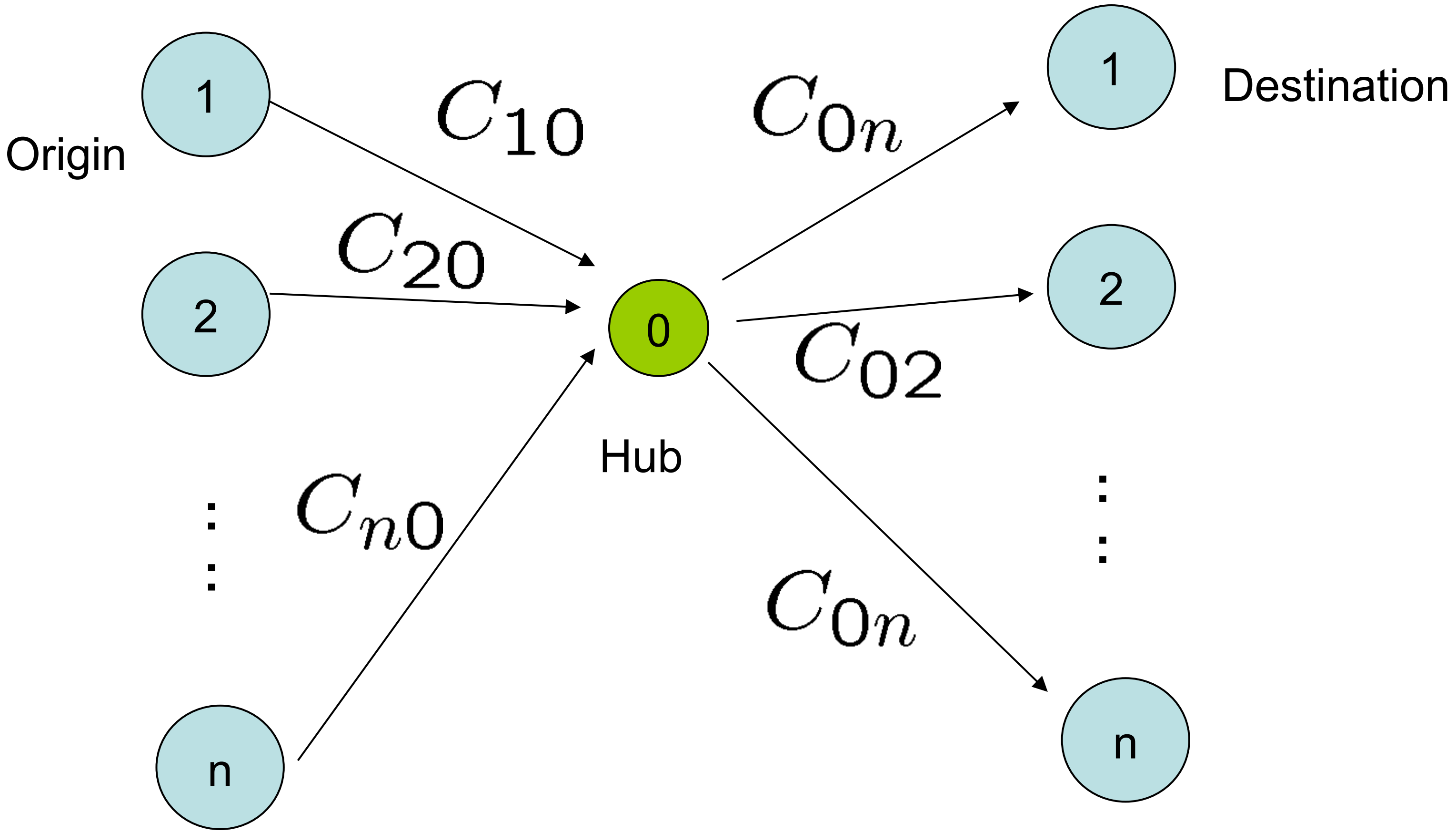




- The two fare model presumes that customers are willing to pay the higher price, even if the lower price is available. How did airlines achieve this?

Revenue Management

- Origin, Destination and Hub



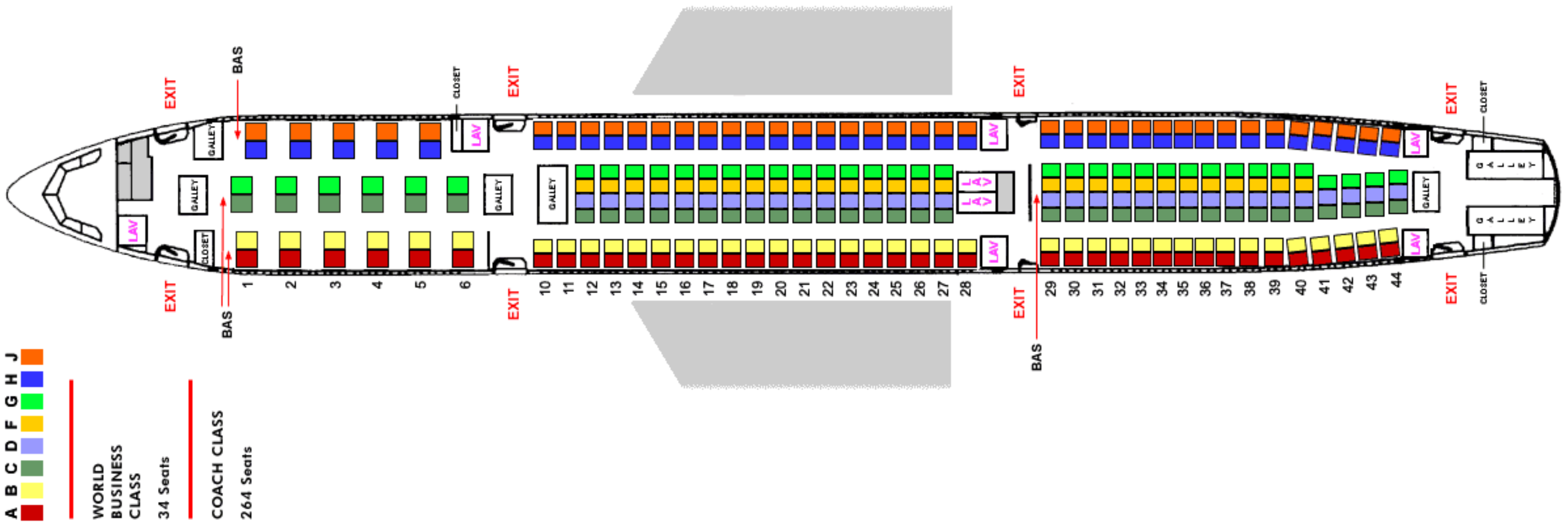
Revenue Management

- 2 class (for simplicity) Q and Y

Revenues: r_{ij}^{Q}, r_{ij}^{Y}

Capacities: $C_{i0}, C_{0j},\ i = 1, \ldots, n, j = 1, \ldots, n$

Expected Demand: D_{ij}^{Q}, D_{ij}^{Y}



Revenue Management

- The right question asked… how many class Q and Y customers should we accept in order to maximize revenue?

Revenue Management

Q_{ij} : Num. of Q class customers to accept from i to j

Y_{ij} : Num. of Y class customers to accept from i to j

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} r_{ij}^{Q} Q_{ij} + r_{ij}^{Y} Y_{ij} & \\
\text{s.t.} \quad & \sum_{j=1}^{n} (Q_{ij} + Y_{ij}) \leq C_{i0} & i = 1, \ldots, n \\
& \sum_{i=1}^{n} (Q_{ij} + Y_{ij}) \leq C_{0j} & j = 1, \ldots, n \\
& 0 \leq Q_{ij} \leq D_{ij}^{Q}, 0 \leq Y_{ij} \leq D_{ij}^{Y} &
\end{aligned}
$$

Revenue Management

"We estimate that RM has generated $1.4 billion in incremental revenue for American Airlines in the last three years alone. This is not a one-time benefit. We expect RM to generate at least $500 million annually for the foreseeable future…"

Robert Crandall, former CEO of American Airlines