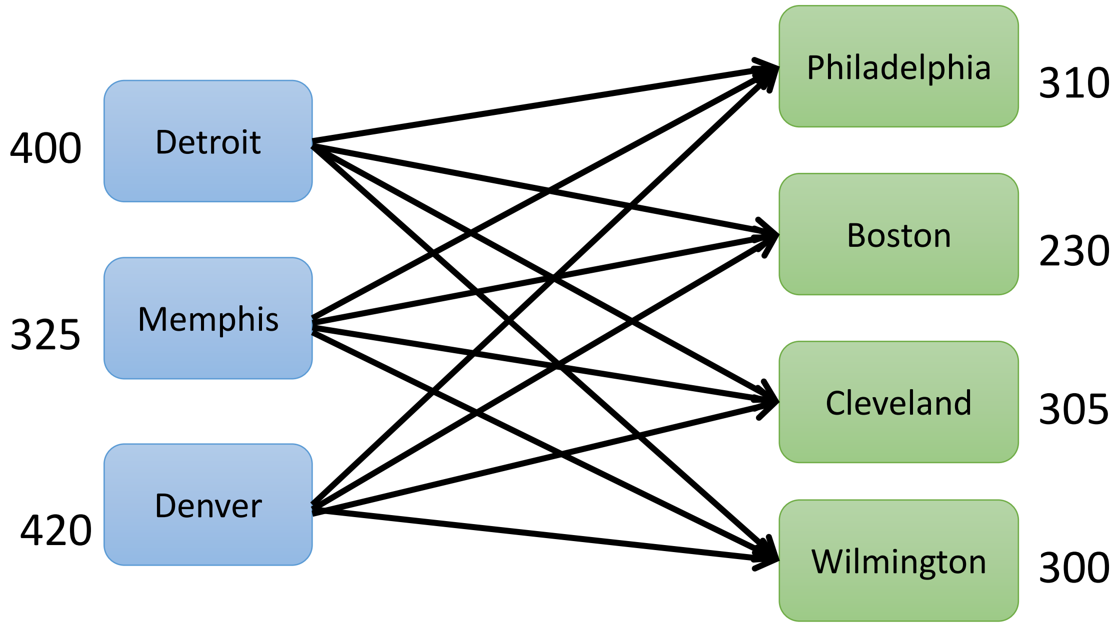
Distribution and Network Models

October 3rd, 2022

Outline (for today and possibly next class)

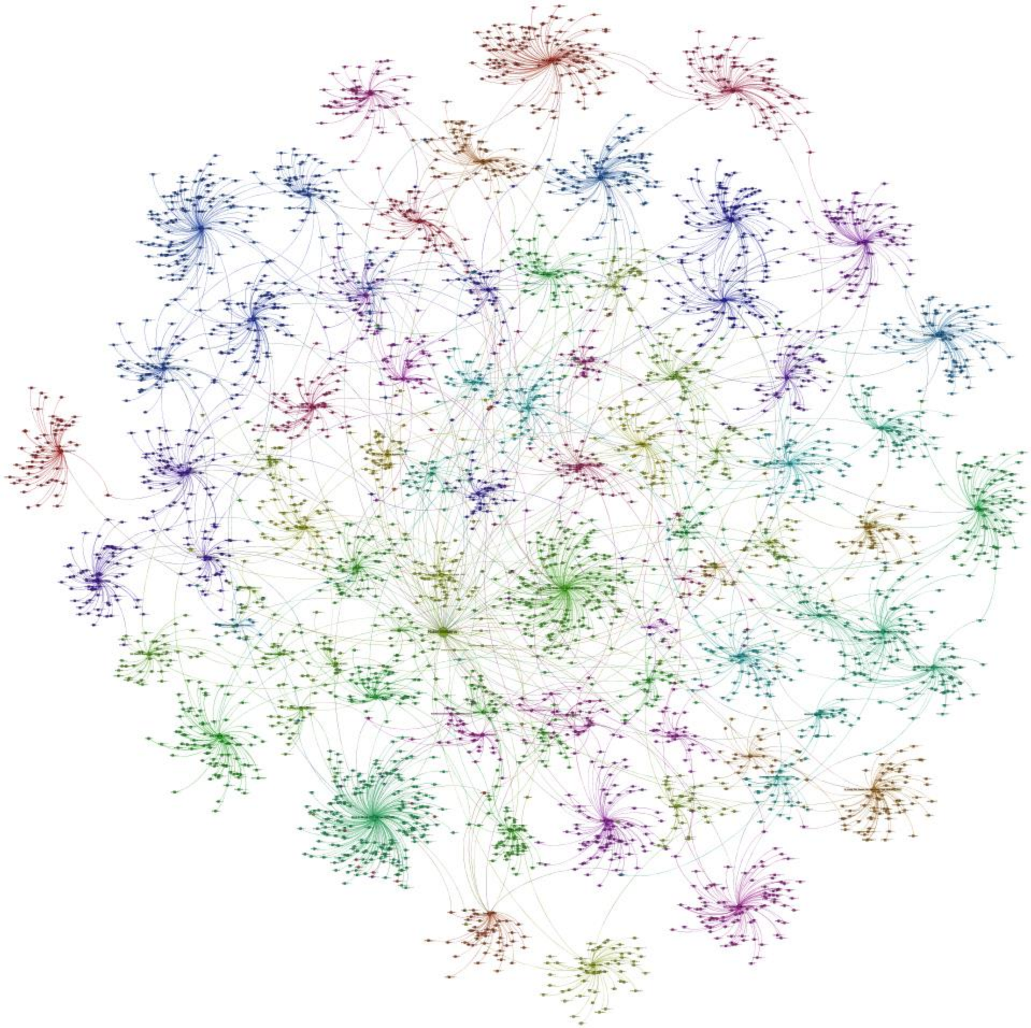
- 5 Major Classes/Applications
  - Transportation
  - Assignment
  - Transshipment
  - Shortest Route
  - Maximal Flow
- 2 Adjacent Applications
  - Inventory Across Time
  - Service Coverage

Network Models:



- A large class of problems that can be solved as linear programs whose formulation has special properties that lend themselves to large scale problems requiring integer solutions
- Basic Components:
  - Nodes
    - Represent a fixed position (a place, a person)
  - Arcs
    - Represent direct connections between two nodes (a street, a relationship)
    - Can be non-directional, bi-directional, or single directional
  - Supply/Demand
    - Represents endowment or need at a node
  - Capacity
    - Represents a constraint of an arc

Social Networks



http://allthingsgraphed.com/2014/08/15/the-lives-he-touched-network-of-robin-williams/

- Nodes:
  - Individuals or Groups
- Arcs:
  - Associations or shared connection
- Labels:
  - Types of connections or nodes

Maps

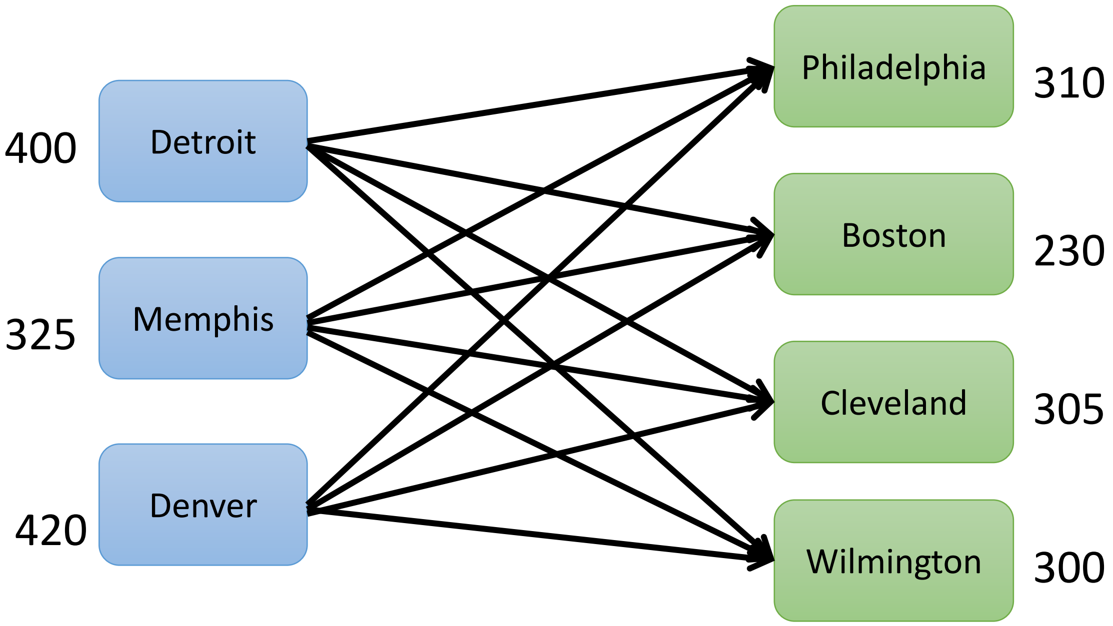
https://www.behance.net/gallery/42818427/Georgetown-Caricature-Map

- Nodes:
  - Locations or intersections
- Arcs:
  - Roads or paths
- Labels:
  - Type of Node
  - Type of Path

Integer Solutions

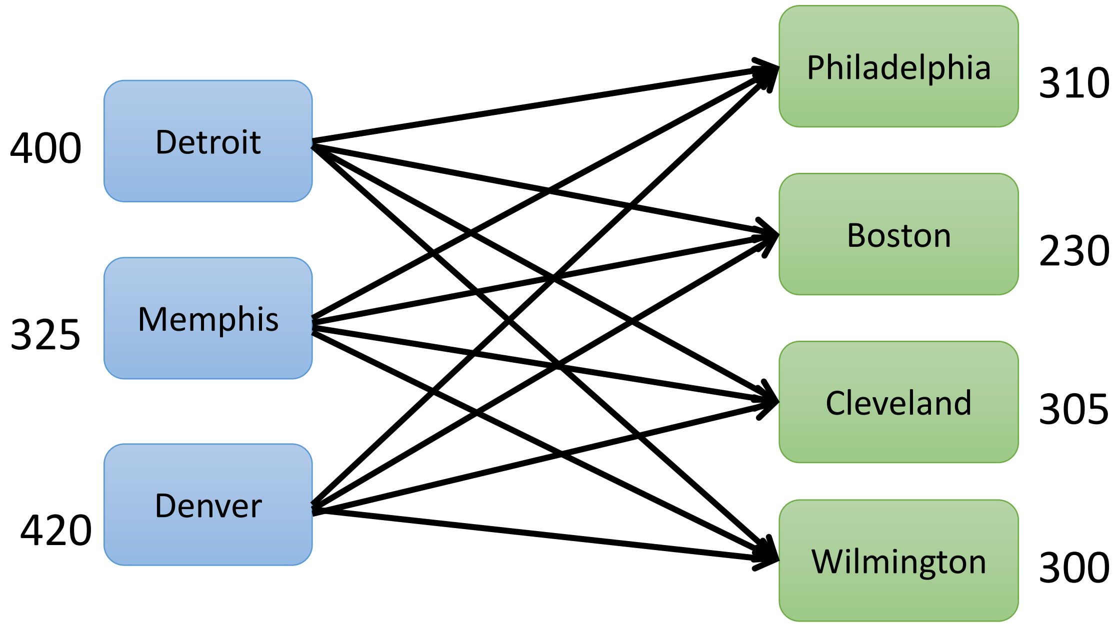
- Node-Arc incidence matrices (such as those that show up in the transportation problem, shortest path, and max flow problems) have a property known as Total Unimodularity
- Any problem formulated as a network model will solve with integer solutions so long as the inputs are all integers (demand and supply).

The Transportation Problem



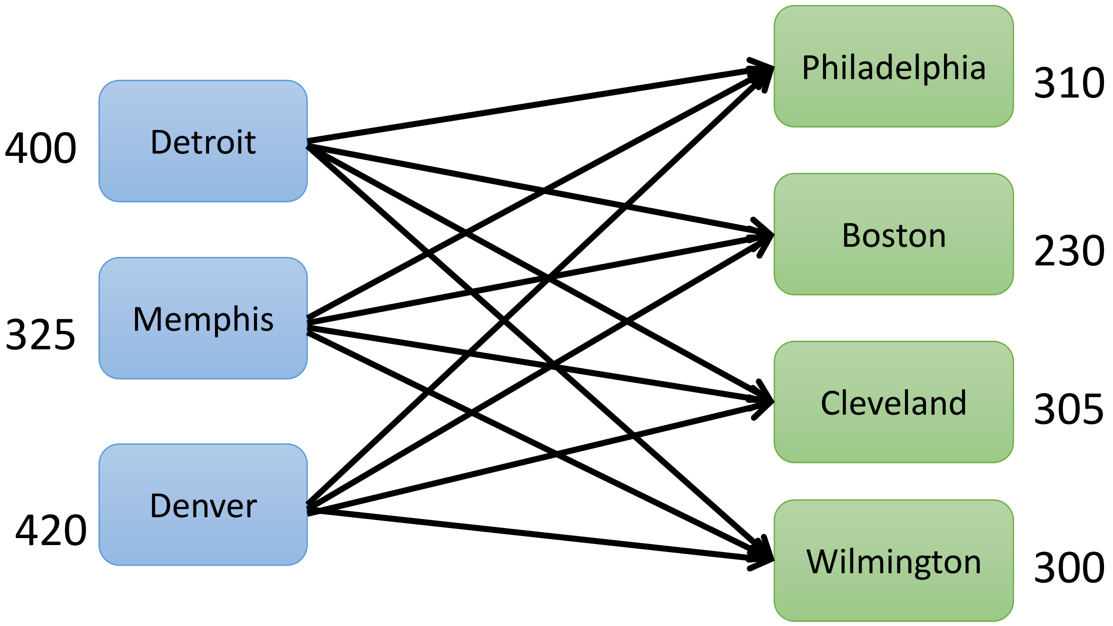
- **Objective**:
  - Minimize the cost of shipping from multiple supply locations to multiple demand locations

The Transportation Problem



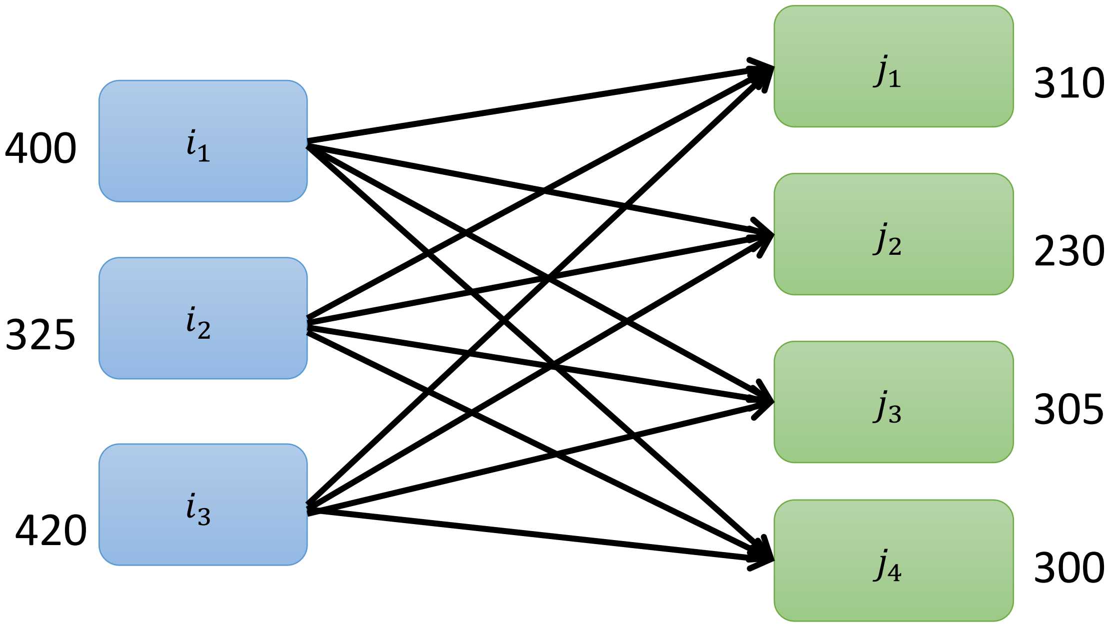
- **Nodes**:
  - Type 1: Supply nodes (places where products originate)
  - Type 2: Demand nodes (places where the products must end up)
- **Arcs**:
  - Directional connections from a demand to a supply node, each with its own cost
- **Supply/Demand**:
  - Type 1 nodes begin with a specific supply
  - Type 2 nodes begin with a specific demand

Transportation Information



	Philadelphia	Boston	Cleveland	Wilmington
Detroit	15	9	8	14
Memphis	12	24	11	18
Denver	21	47	9	21

Let's say that…



- Supply locations:
  - Detroit is city 1
  - Memphis is city 2
  - Denver is city 3
- Demand locations:
  - Philadelphia is city 1
  - Boston is city 2
  - Cleveland is city 3
  - Wilmington is city 4

Decision Variables:

$x_{i,j}$ where x is the quantity shipped from
- i the origin (1,n), to
- j the destination (1,m)

- $X_{1,1}$=# of units shipped from Detroit to Philadelphia
- $X_{1,2}$=# of units shipped from Detroit to Boston
- $X_{1,3}$=# of units shipped from Detroit to Cleveland
- $X_{1,4}$=# of units shipped from Detroit to Wilmington
- $X_{2,1}$=# of units shipped from Memphis to Philadelphia
- $X_{2,2}$=# of units shipped from Memphis to Boston
- $X_{2,3}$=# of units shipped from Memphis to Cleveland
- $X_{2,4}$=# of units shipped from Memphis to Wilmington
- $X_{3,1}$=# of units shipped from Denver to Philadelphia
- $X_{3,2}$=# of units shipped from Denver to Boston
- $X_{3,3}$=# of units shipped from Denver to Cleveland
- $X_{3,4}$=# of units shipped from Denver to Wilmington

Objective Function:

$$\text{Min} \quad \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} x_{i,j}$$

where $c_{i,j}$ is the cost to ship one unit from i to j

- Minimize Cost=
  - $15x_{1,1} + 9x_{1,2} + 8x_{1,3} + 14x_{1,4} + 12x_{2,1} + 24x_{2,2} + 11x_{2,3} + 18x_{2,4} + 21x_{3,1} + 47x_{3,2} + 9x_{3,3} + 21x_{3,4}$

Constraints:

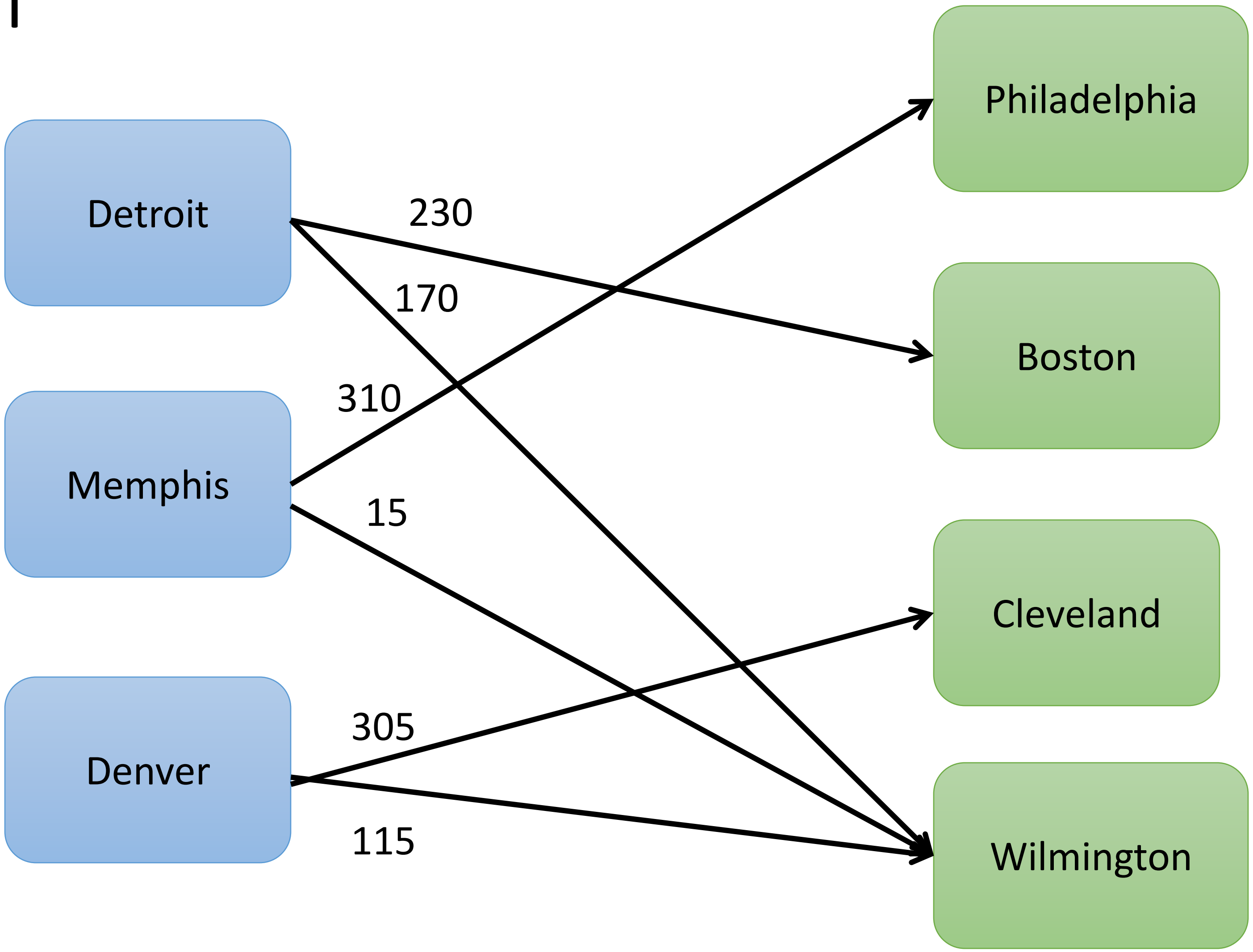
Supply $\sum_{j=1}^{m} x_{i,j} \leq s_i$ **for** $i = 1, 2, \ldots, n$

Demand $\sum_{i=1}^{n} x_{i,j} = d_j$ **for** $j = 1, 2, \ldots, m$

$x_{i,j} \geq 0 \quad \forall i, j$

- Must not exceed supply:
  - $X_{1,1}+ X_{1,2}+ X_{1,3} +X_{1,4} \leq 400$
  - $X_{2,1}+ X_{2,2}+ X_{2,3} +X_{2,4} \leq 325$
  - $X_{3,1}+ X_{3,2}+ X_{3,3} +X_{3,4} \leq 420$
- Must meet demand:
  - $X_{1,1}+ X_{2,1}+ X_{3,1}=310$
  - $X_{1,2}+ X_{2,2}+ X_{3,2}=230$
  - $X_{1,3}+ X_{2,3}+ X_{3,3}=305$
  - $X_{1,4}+ X_{2,4}+ X_{3,4}=300$

Solution



Detroit
Memphis
Denver
Philadelphia
Boston
Cleveland
Wilmington
230
170
310
15
305
115

The Transportation Problem General Form

- Decision Variables:

  $X_{i,j}=x_{i,j}$ where x is the quantity shipped from

  - i the origin (1,n), to
  - j the destination (1,m)

$$\text{Min} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} x_{i,j}$$

where $c_{i,j}$ **is the cost to ship one unit from** i **to** j

Supply $\sum_{j=1}^{m} x_{i,j} \leq s_i$ **for** $i = 1, 2, \ldots, n$

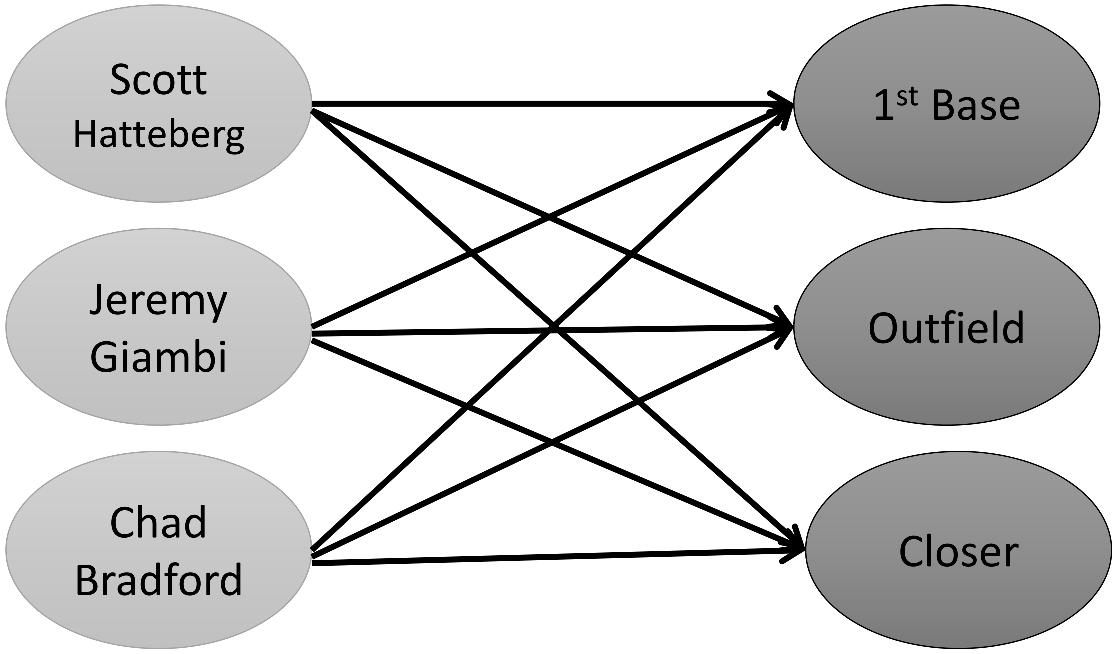
Demand $\sum_{i=1}^{n} x_{i,j} = d_j$ **for** $j = 1, 2, \ldots, m$

$$x_{i,j} \geq 0 \quad \forall i, j$$

Variations:

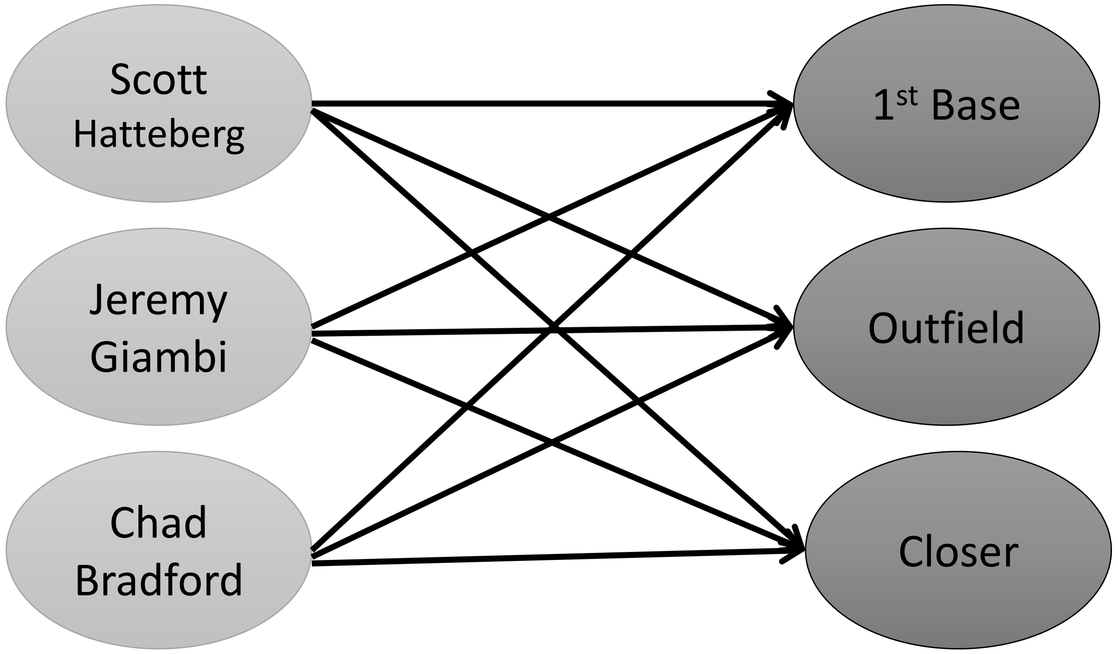
- Maximize the objective
  - Instead of minimize cost
- Some assignments are unacceptable
- Route capacities or route minimums

The Assignment Problem



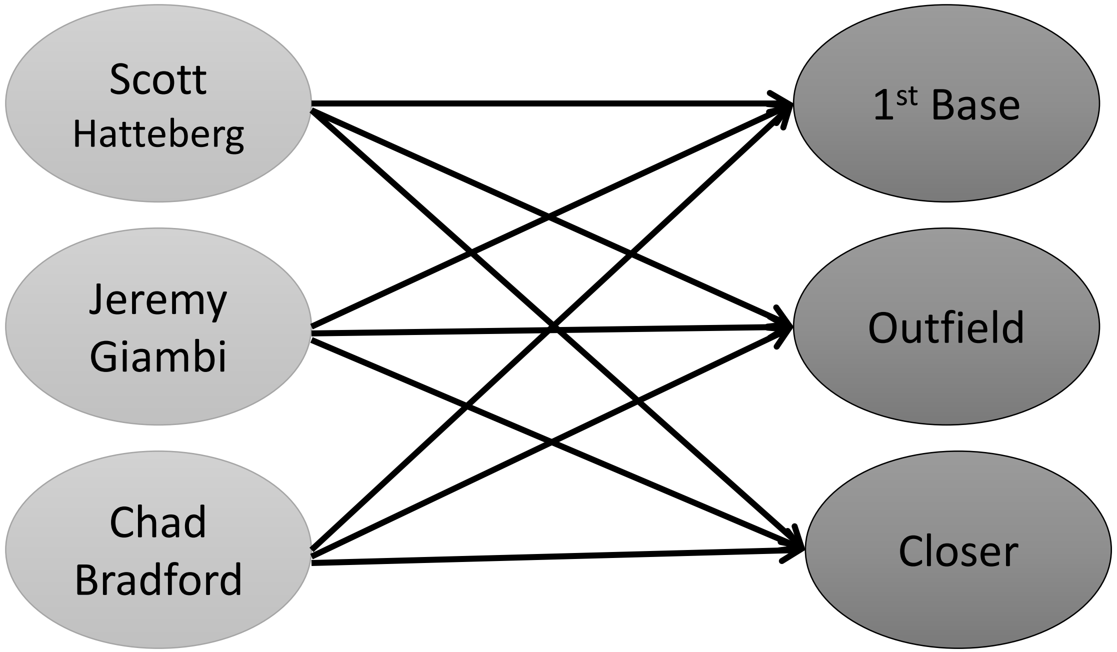
- Deals with the problem of assigning people to tasks/jobs in a cost effective or time effective manner

The Assignment Problem



- **Nodes**:
  - Type 1: People who can perform the task
  - Type 2: The jobs that need to be completed
- **Arcs**:
  - Represent how much each matching cost or how long each matching takes
- **Demand/Supply**:
  - The supply at Type 1 nodes is 1
  - The demand at Type 2 nodes is 1

The Assignment Problem



	Jason Giambi (1st Base)	Outfielder (Johnny Damon)	Closer (Jason Isringhausen)
Scott Hatteberg	.678	.324	.127
Jeremy Giambi	.271	.867	.143
Chad Bradford	.384	.499	.531

Model Formulation

$x_{i,j}$ where x is 1 if we assign

- i the person (1,n), to
- j the job (1,m)

- Decision Variables:
  - $X_{1,1}$= 1 if we assign Hatteberg to 1st base, 0 otherwise
  - $X_{1,2}$= 1 if we assign Hatteberg to outfield, 0 otherwise
  - $X_{1,3}$= 1 if we assign Hatteberg to Closer, 0 otherwise
  - $X_{2,1}$= 1 if we assign Giambi to 1st base, 0 otherwise
  - $X_{2,2}$= 1 if we assign Giambi to outfield, 0 otherwise
  - $X_{2,3}$= 1 if we assign Giambi to Closer, 0 otherwise
  - $X_{3,1}$= 1 if we assign Bradford to 1st base, 0 otherwise
  - $X_{3,2}$= 1 if we assign Bradford to outfield, 0 otherwise
  - $X_{3,3}$= 1 if we assign Bradford to Closer, 0 otherwise

Model Formulation

$$\text{Max} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} x_{i,j}$$

where $c_{i,j}$ is the affinity of person i to job j

- Objective Function:
  - Maximize efficiency=
    - $0.678\,X_{1,1} + 0.324X_{1,2} + 0.127X_{1,3} + 0.271X_{2,1} + 0.867X_{2,2} + 0.143X_{2,3} + 0.384X_{3,1} + 0.499X_{3,2} + 0.531X_{3,3}$

Model Formulation

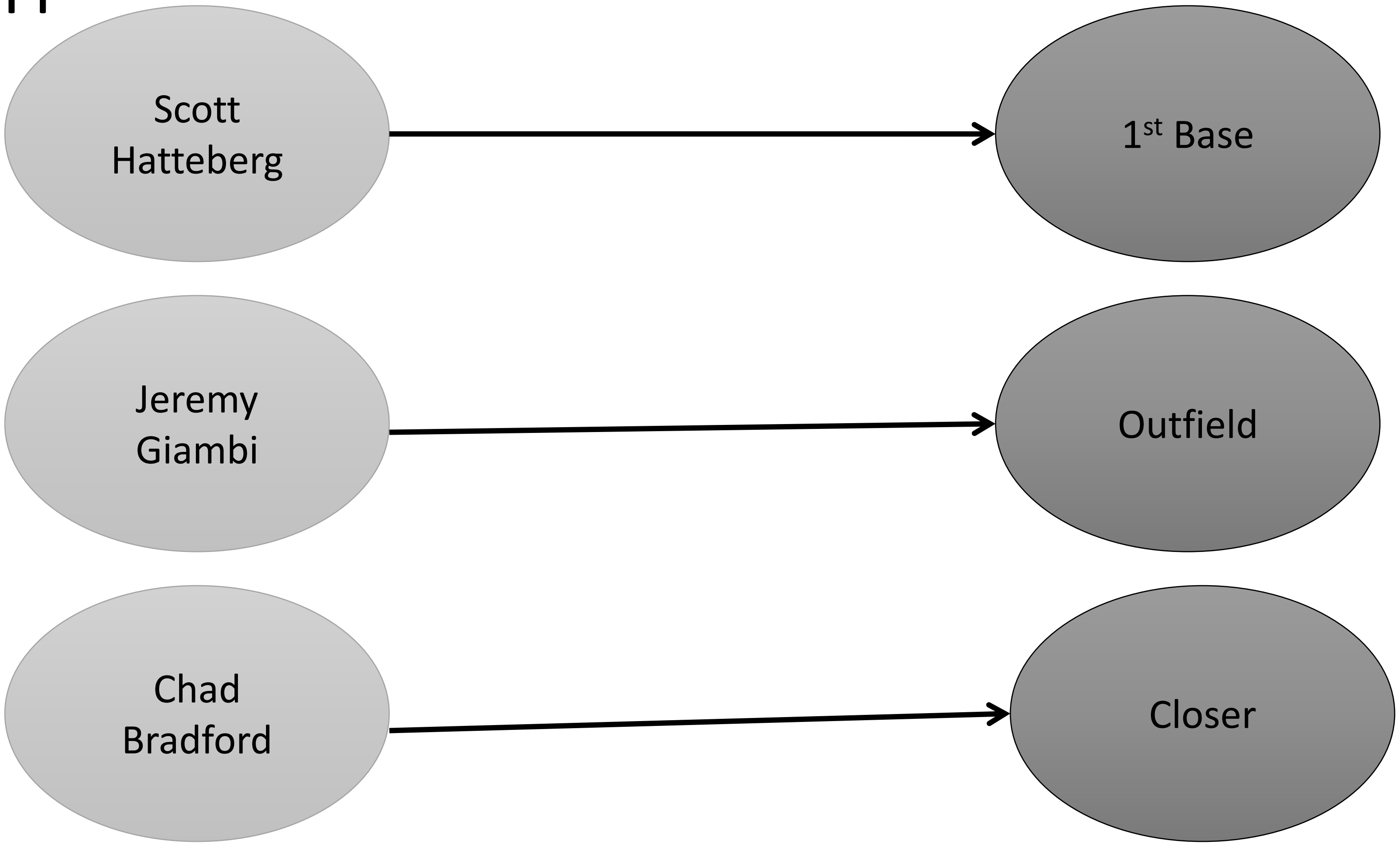
Employment $\sum_{j=1}^{m} x_{i,j} \leq 1$ **for person** $i = 1, 2, \ldots, n$

Fulfillment $\sum_{i=1}^{n} x_{i,j} = d_j$ **for job** $j = 1, 2, \ldots, m$

$x_{i,j} \geq 0 \quad \forall i, j$

- Constraints:
  - Every player gets assigned at most one position:
    - $X_{1,1}+ X_{1,2}+ X_{1,3}=1$
    - $X_{2,1}+ X_{2,2}+ X_{2,3}=1$
    - $X_{3,1}+ X_{3,2}+ X_{3,3}=1$
  - Every position gets assigned to one player
    - $X_{1,1}+ X_{2,1}+ X_{3,1}=1$
    - $X_{1,2}+ X_{2,2}+ X_{3,2}=1$
    - $X_{1,3}+ X_{2,3}+ X_{3,3}=1$

Solution



Scott Hatteberg
Jeremy Giambi
Chad Bradford
1st Base
Outfield
Closer

The Assignment Problem General Form

$x_{i,j}$ where x is 1 if we assign

- i the person (1,n), to
- j the job (1,m)

$$\text{Max} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} x_{i,j}$$

where $c_{i,j}$ **is the affinity of person** i **to job** j

Employment $\sum_{j=1}^{m} x_{i,j} \leq 1$ **for person** $i = 1, 2, \ldots, n$

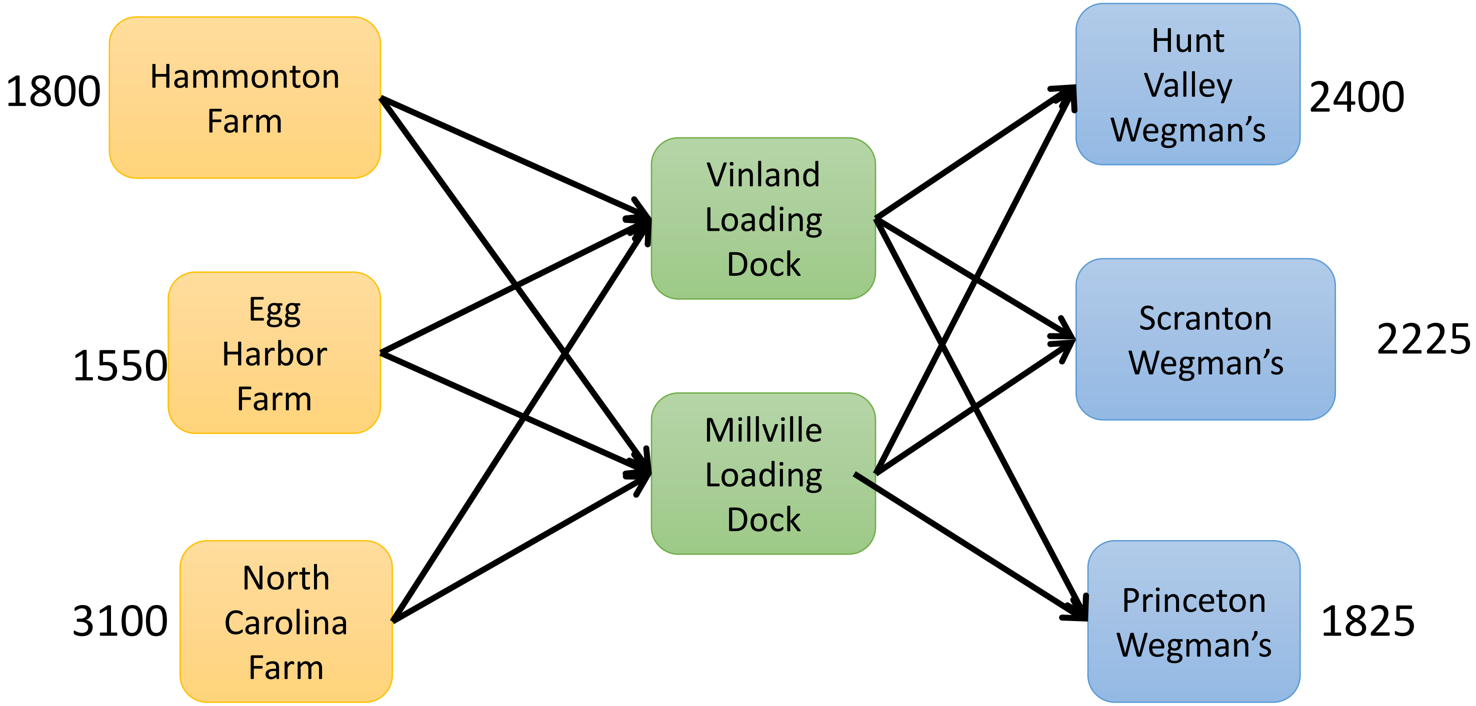
Fulfillment $\sum_{i=1}^{n} x_{i,j} = d_j$ **for job** $j = 1, 2, \ldots, m$

$$x_{i,j} \geq 0 \quad \forall i, j$$

Variations:

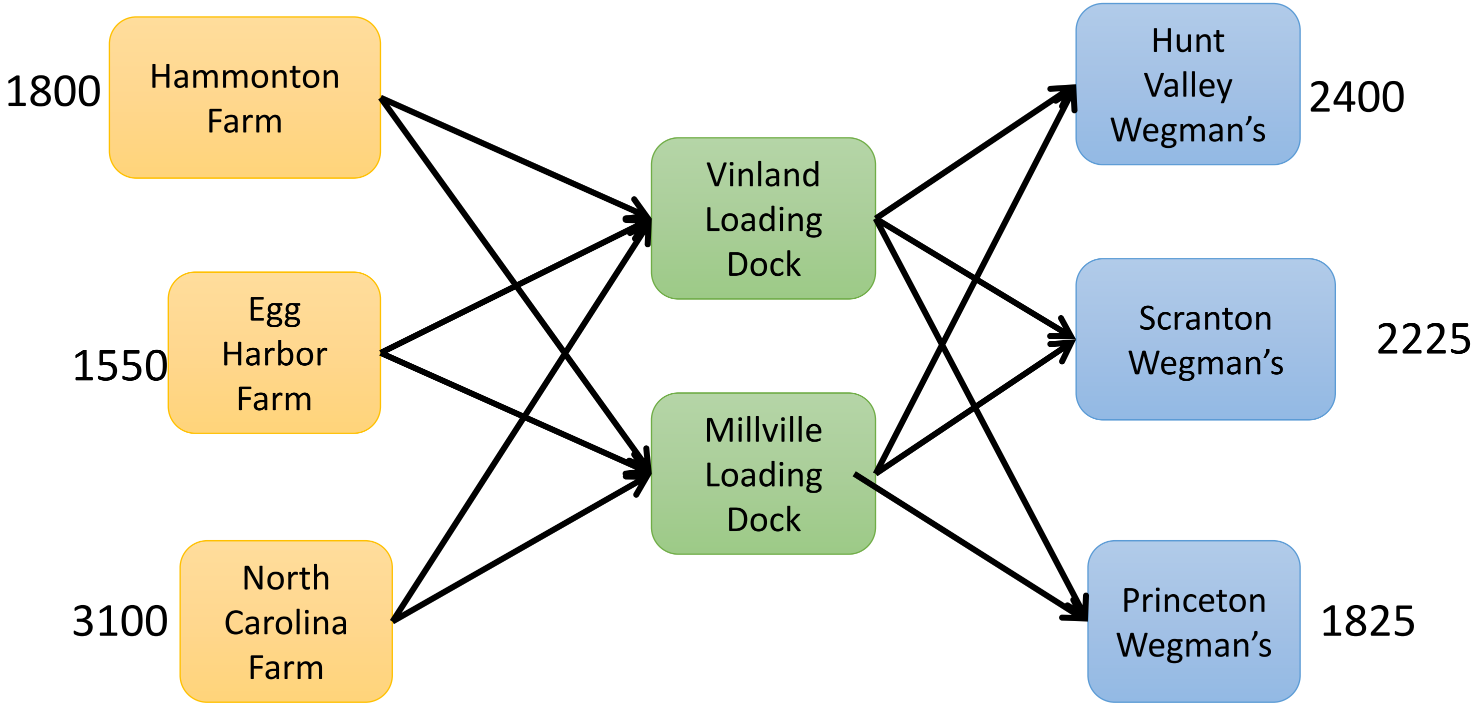
- Unacceptable assignments
- Total number of people does not equal the number of jobs
- Different objective function forms

The Transshipment Problem



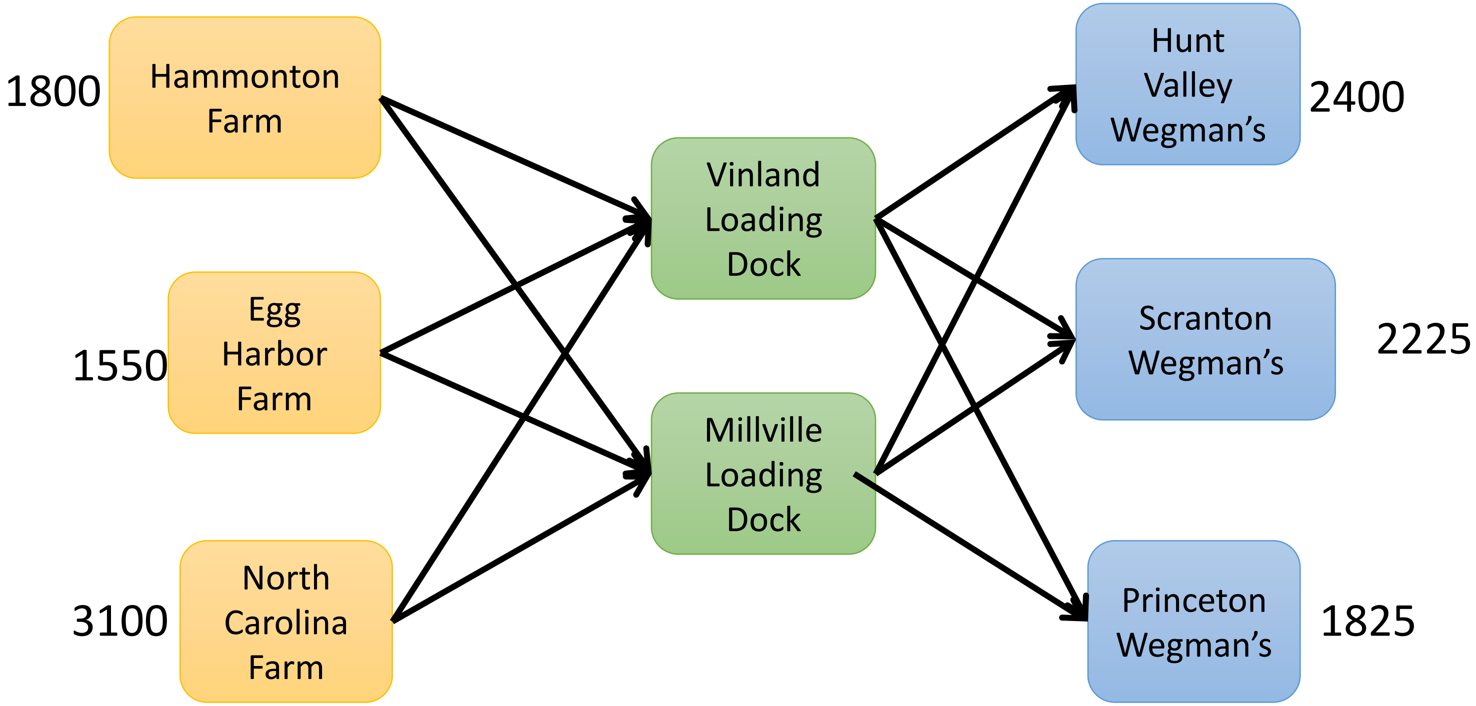
- Deals with problems where products need to be moved through intermediary nodes before continuing on to their final destination

The Transshipment Problem



- **Nodes**:
  - Type 1: nodes where products are produced
  - Type 2: nodes where products must end up
  - Type 3: nodes that the products must pass through
- **Arcs**
  - Directional connections between nodes that the products can move along
- **Supply/Demand**:
  - At Type 1 or 2 nodes: Same as for Transportation
  - At Type 3 nodes: The supply/demand here is zero (everything that flows in, must flow out)

The Transshipment Problem



	Vinland	Millville	Hunt Valley	Scranton	Princeton
Hammonton	10	7			
Egg Harbor	8	6			
North Carolina	51	48			
Vineland			32	60	41
Millville			30	65	37

Model Formulation

Decision Variables:

$X_{i,j}=x_{i,j}$ where x is the quantity shipped from

- i the origin (1,n), to
- j the destination (1,m)

- Decision Variables:
  - $X_{1,4}$=# pallets shipped from Hammonton to Vineland loading dock
  - $X_{1,5}$=# pallets shipped from Hammonton to Millville loading dock
  - $X_{2,4}$=# pallets shipped from Egg Harbor to Vinland loading dock
  - $X_{2,5}$=# pallets shipped from Egg Harbor Millville loading dock
  - $X_{3,4}$=# pallets shipped from North Carolina to Vinland loading dock
  - $X_{3,5}$=# pallets shipped from North Carolina to Millville loading dock
  - $X_{4,6}$=# pallets shipped from Vinland to Hunt Valley loading dock
  - $X_{4,7}$=# pallets shipped from Vinland to Scranton loading dock
  - $X_{4,8}$=# pallets shipped from Vinland to Princeton loading dock
  - $X_{5,6}$=# pallets shipped from Millville to Hunt Valley loading dock
  - $X_{5,7}$=# pallets shipped from Millville to Scranton loading dock
  - $X_{5,8}$=# pallets shipped from Millville to Princeton loading dock

Model Formulation

$$\sum_{\text{all arcs}} c_{i,j} x_{i,j}$$

where $c_{i,j}$ is the cost to ship one unit from i to j

- Objective Function:
  - Minimize Cost=
    - $10X_{1,4} + 7X_{1,5} + 8X_{2,4} + 6X_{2,5} + 51X_{3,4} + 48X_{3,5} + 32X_{4,6} + 60X_{4,7} + 41X_{4,8} + 30X_{5,6} + 65X_{5,7} + 37X_{5,8}$

Model Formulation

Supply Nodes $\sum_{\text{arcs out}} x_{i,j} - \sum_{\text{arcs out}} x_{i,j} \leq s_i$

Transshipment Nodes $\sum_{\text{arcs out}} x_{i,j} - \sum_{\text{arcs out}} x_{i,j} = 0$

Demand Nodes $\sum_{\text{arcs out}} x_{i,j} - \sum_{\text{arcs out}} x_{i,j} = d_j$

$x_{i,j} \geq 0 \quad \forall i, j$

- Constraints:
  - Supply nodes
    - $X_{1,4} + X_{1,5} \leq 1800$
    - $X_{2,4} + X_{2,5} \leq 1550$
    - $X_{3,4} + X_{3,5} \leq 3100$
  - Transshipment Nodes
    - $(X_{1,4} + X_{2,4} + X_{3,4}) - (X_{4,6} + X_{4,7} + X_{4,8}) = 0$
    - $(X_{1,5} + X_{2,5} + X_{3,5}) - (X_{5,6} + X_{5,7} + X_{5,8}) = 0$
  - Demand Nodes
    - $X_{4,6} + X_{5,6} = 2400$
    - $X_{4,7} + X_{5,7} = 2225$
    - $X_{4,8} + X_{5,8} = 1825$

The Transshipment Problem General Form

- Decision Variables:

  $X_{i,j}=x_{i,j}$ where x is the quantity shipped from
  - i the origin (1,n), to
  - j the destination (1,m)

$$\sum_{\text{all arcs}} c_{i,j}x_{i,j}$$

Supply Nodes $\sum_{\text{arcs out}} x_{i,j} - \sum_{\text{arcs out}} x_{i,j} \leq s_i$

Transshipment Nodes $\sum_{\text{arcs out}} x_{i,j} - \sum_{\text{arcs out}} x_{i,j} = 0$

Demand Nodes $\sum_{\text{arcs out}} x_{i,j} - \sum_{\text{arcs out}} x_{i,j} = d_j$

$x_{i,j} \geq 0 \quad \forall i, j$

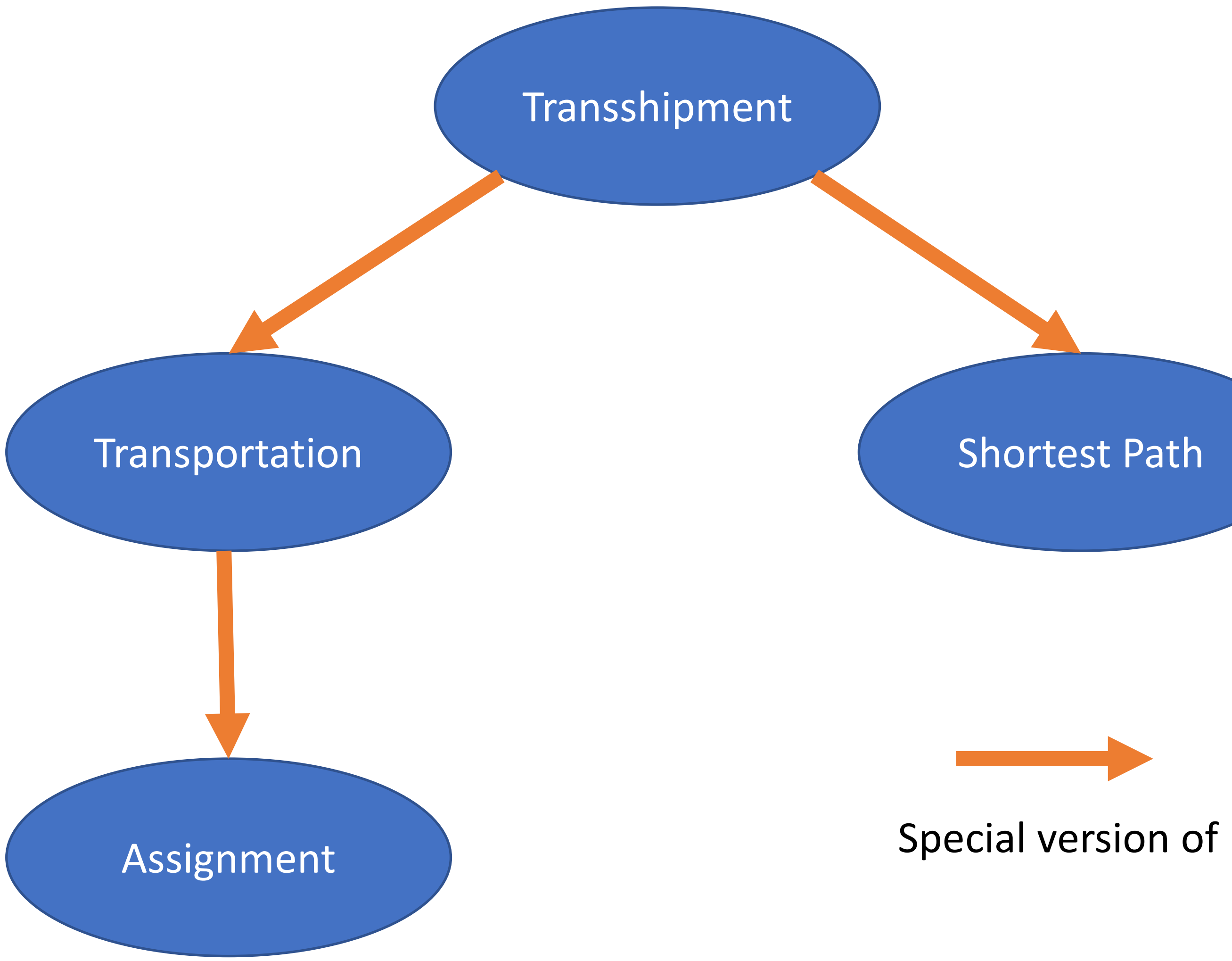
where $c_{i,j}$ is the cost to ship one unit from i to j

Notice the Similarities/Differences

- Similarities
  - The objective function
  - The variables
  - The supply/demand constraints
- Difference
  - The network setup
  - The transshipment constraints

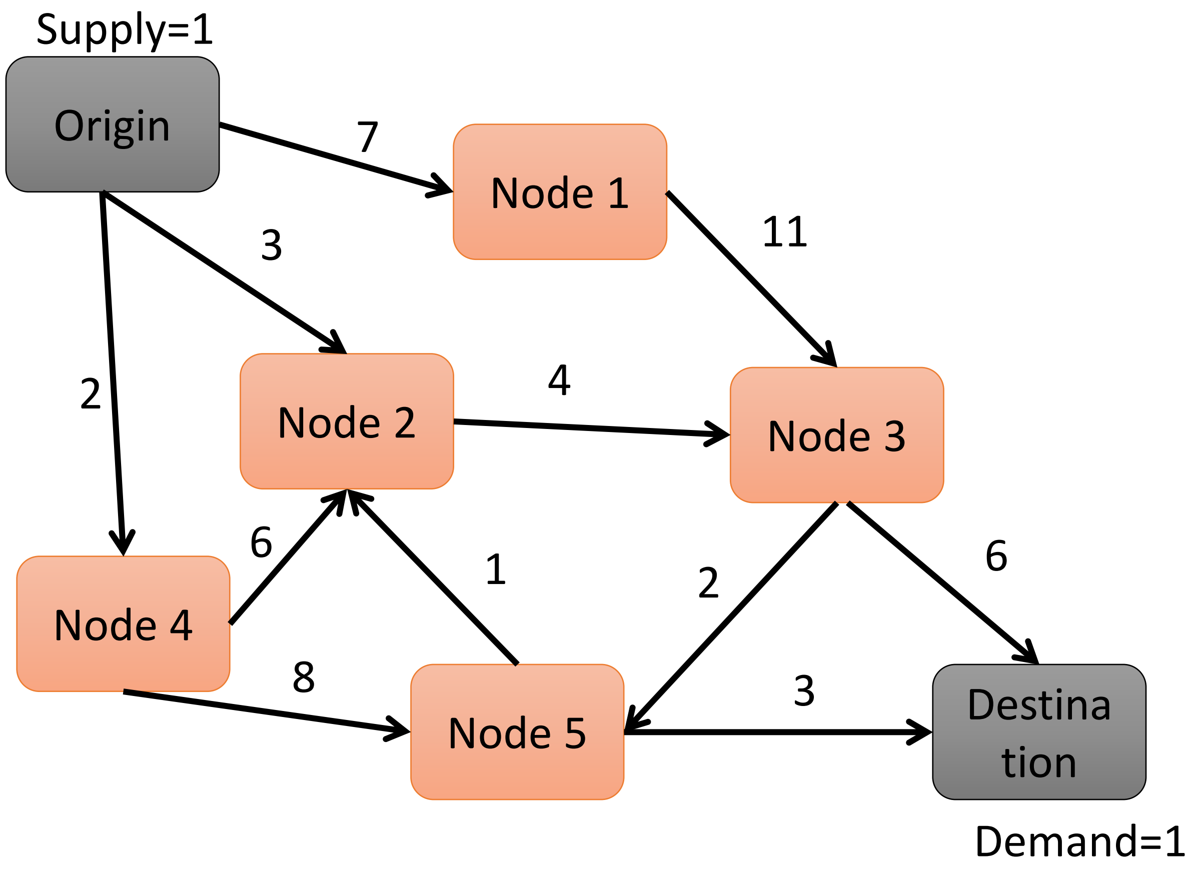
Variations

- Supply doesn’t equal demand
- Transshipment nodes have starting inventory/need inventory to be left
- Route capacities/minimum
- Alternative objective functions
- Unacceptable routes



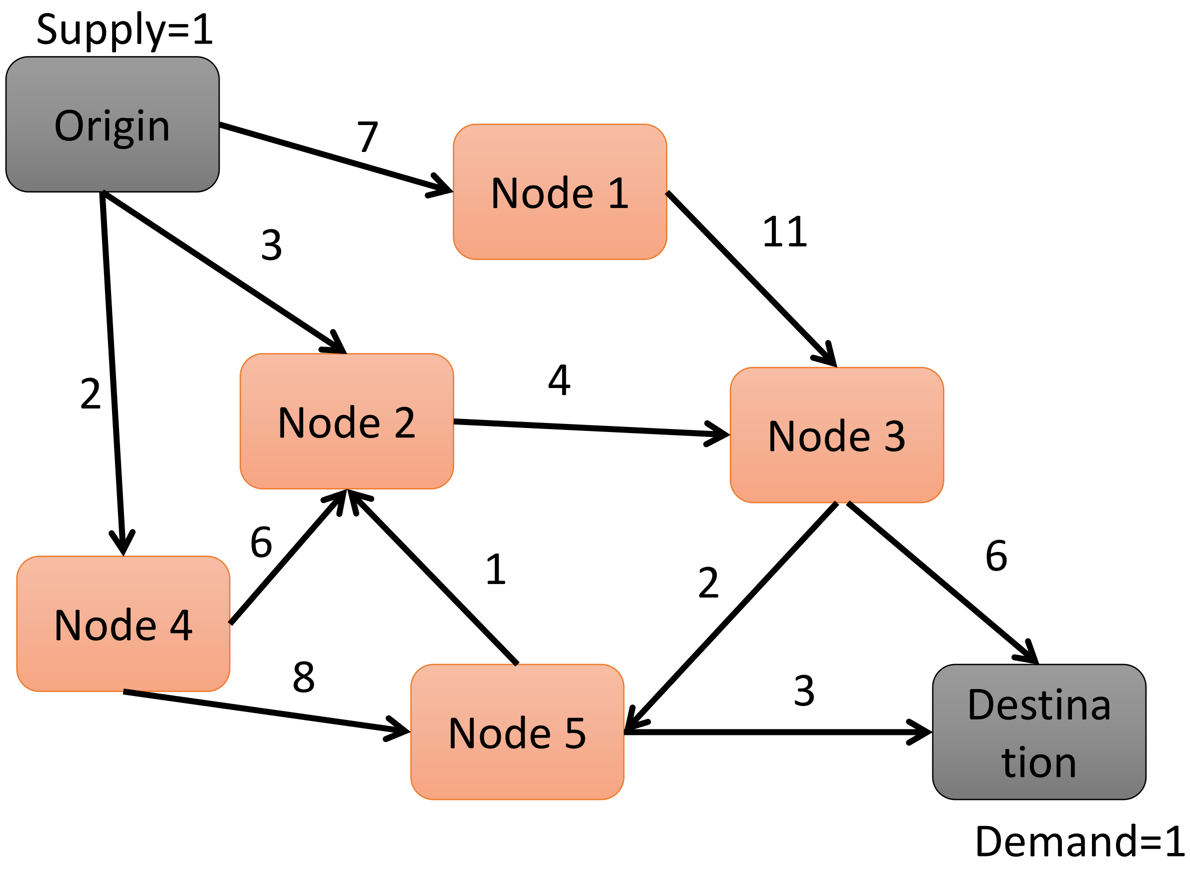
- **The transshipment problem is the most general form allowing for generalized:**
  - **Arbitrary supply amounts**
  - **Arbitrary demand amounts**
  - **Transshipment nodes**

The Shortest Path Problem



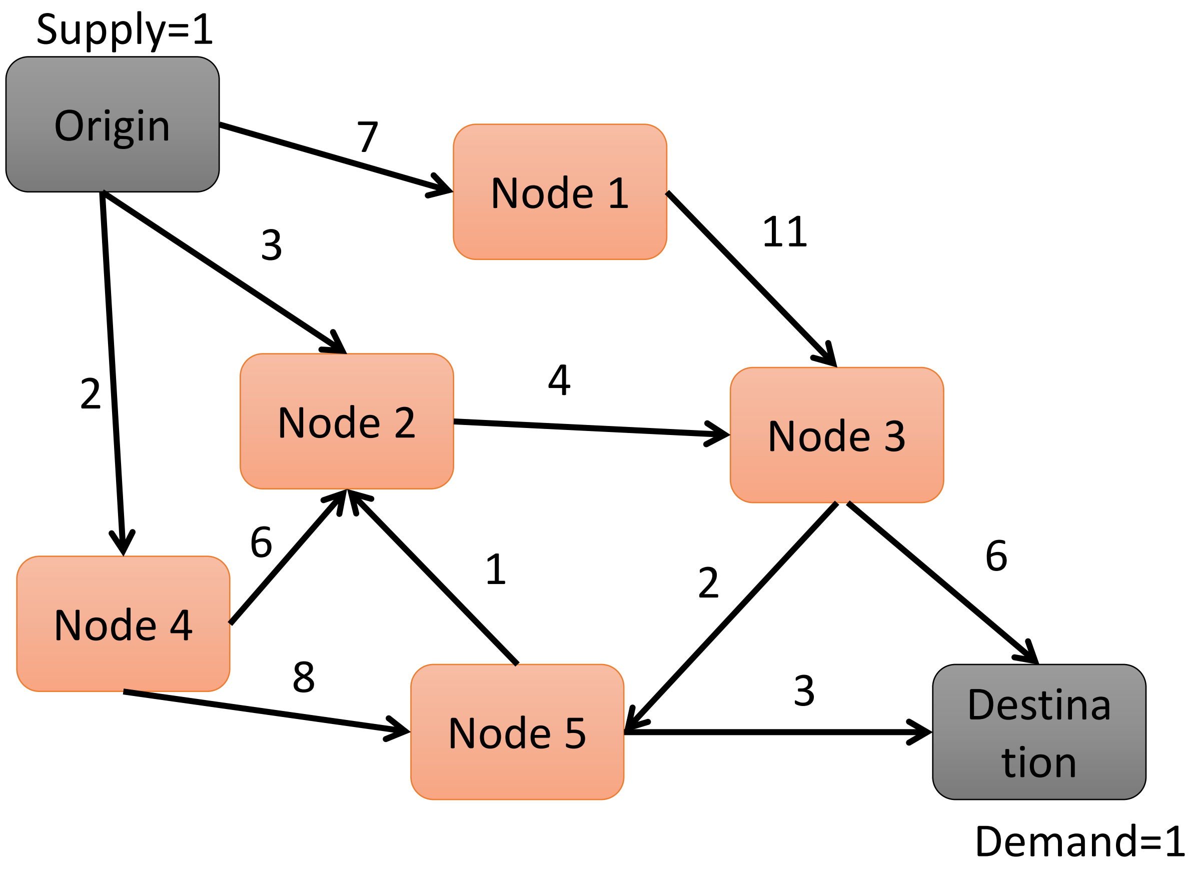
- **Objective**:
  - Find the shortest route from an origin to a destination when multiple routes/combinations of routes are possible

The Shortest Path Problem



- Nodes
  - Type 1: the origin
  - Type 2: the destination
  - Type 3: mid-points (similar to transshipment nodes)
- Arcs
  - Directional connections from one node to another that represent feasible paths
  - Costs can be assessed in multiple ways
    - Distance
    - Time
- Supply/Demand
  - Type 1 Node: Has a supply of one
  - Type 2 Node: Has a demand of one
  - Type 3 Nodes: Have a supply/demand of zero

The Shortest Path Problem



- $X_{origin, 1}$= 1 if you take path from origin to Node 1, 0 otherwise
- $X_{origin, 2}$= 1 if you take path from origin to Node 1, 0 otherwise
- $X_{1, 3}$= 1 if you take path from Node 1 to Node , 0 otherwise
- $X_{2,3}$ = 1 if you take path from Node 2 to Node 3, 0 otherwise
- $X_{3,5}$= 1 if you take path from Node 3 to Node 5, 0 otherwise
- $X_{4,2}$= 1 if you take path from Node 4 to Node 2, 0 otherwise
- $X_{4,5}$= 1 if you take path from Node 4 to Node 5, 0 otherwise
- $X_{3,5}$= 1 if you take path from Node 3 to Node 5, 0 otherwise
- $X_{5,2}$= 1 if you take path from Node 5 to Node 2, 0 otherwise
- $X_{3, Destination}$= 1 if you take path from Node 3 to Destination, 0 otherwise
- $X_{5, Destination}$= 1 if you take path from Node 5 to Destination, 0 otherwise

Objective Function

$$\text{Min} \sum_{\text{all arcs}} c_{i,j} x_{i,j}$$

where $c_{i,j}$ is the travel time/distance from i to j
where $x_{i,j} \in 0, 1$ is the decision to travel from i to j

- Minimize Travel Time=
  - $7X_{origin,1} + 3X_{origin,2} + 2X_{origin,4} + 11X_{1,3} + 4X_{2,3} + 2X_{3,5} + 6X_{3,Destination} + 6X_{4,2} + 8X_{4,5} + X_{5,2} + 3X_{5,Destination}$

Constraints

Source Node $\sum_{\text{arcs out}} x_{i,j} = 1$

Transshipment Nodes $\sum_{\text{arcs out}} x_{i,j} - \sum_{\text{arcs in}} x_{i,j} = 0$

Destination Node $\sum_{\text{arcs in}} x_{i,j} = 1$

$x_{i,j} \geq 0 \quad \forall i, j$

- Origin (Source Node)
  - $X_{origin,1} + X_{origin,2} + X_{origin,4} = 1$
- Mid-Points (Transshipment Nodes)
  - $(X_{origin,1}) - (X_{1,3}) = 0$
  - $(X_{origin,2} + X_{4,2} + X_{5,2}) - (X_{2,3}) = 0$
  - $(X_{1,3} + X_{2,3}) - (X_{3,5} + X_{3,Destination}) = 0$
  - $(X_{origin,4}) - (X_{4,2} + X_{4,5}) = 0$
  - $(X_{4,5} + X_{3,5}) - (X_{5,2} + X_{5,Destination}) = 0$
- Destination (Demand Node)
  - $X_{3,Destination} + X_{5,Destination} = 1$
- Non-negativity

Shortest Path Problem

Decision Variables

$$(x_{0,1}, x_{0,2}, x_{0,4}, x_{1,3}, x_{2,3}, x_{3,5}, x_{3,6}, x_{4,2}, x_{4,5}, x_{5,2}, x_{5,6})$$

In [13]:

```
obj <- c(7,3,2,11,4,2,6,6,8,1,3)
mat <- matrix(c(1,1,1,0,0,0,0,0,0,0,0, # Origin node
                1,0,0,-1,0,0,0,0,0,0,0, # Transshipment Node 1
                0,1,0,0,-1,0,0,1,0,1,0, # Transshipment Node 2
                0,0,0,1,1,-1,-1,0,0,0,0, # Transshipment Node 3
                0,0,1,0,0,0,0,-1,-1,0,0, # Transshipment Node 4
                0,0,0,0,0,1,0,0,1,-1,-1, # Transshipment Node 5
                0,0,0,0,0,0,1,0,0,0,1), nrow = 7, byrow=TRUE) # Destination Node
dir <- c("=", "=", "=", "=", "=","=","=")
rhs <- c(1, 0, 0, 0, 0, 0, 1)
```

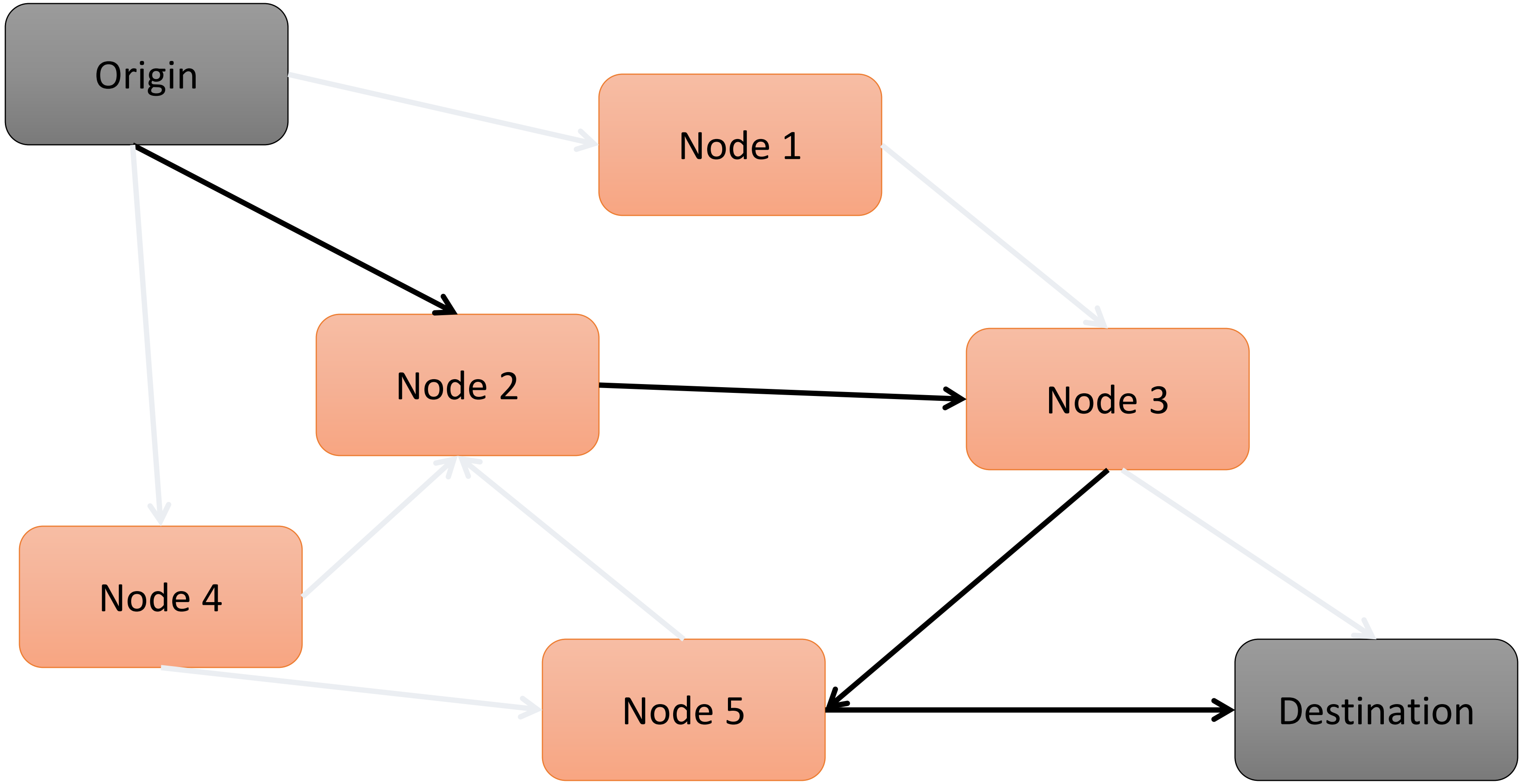
In [14]:

```
lp_shpp <- lp("min", obj, mat, dir, rhs, compute.sens = 1)
sensitivity_report(lp_shpp)
```

```
Objective Function Value:
12

 Objective Function
   variable final_value reduced_cost coefficient min_coef max_coef
1       x 1           0            4           7        3      Inf
2       x 2           1            0           3     -Inf        4
3       x 3           0            0           2        1      Inf
4       x 4           0            7          11        4      Inf
5       x 5           1            0           4       -3        5
6       x 6           1            0           2       -5        3
7       x 7           0            1           6        5      Inf
8       x 8           0            5           6        1      Inf
9       x 9           0            1           8        7      Inf
10     x 10           0            7           1       -6      Inf
11     x 11           1            0           3     -Inf        4
```

Solution: Travel Time=12 minutes



The Shortest Path Problem General Form

$$\text{Min} \sum_{\text{all arcs}} c_{i,j} x_{i,j}$$

where $c_{i,j}$ is the travel time/distance from i to j

where $x_{i,j} \in 0, 1$ is the decision to travel from i to j

Source Node $\sum_{\text{arcs out}} x_{i,j} = 1$

Transshipment Nodes $\sum_{\text{arcs out}} x_{i,j} - \sum_{\text{arcs in}} x_{i,j} = 0$

Destination Node $\sum_{\text{arcs in}} x_{i,j} = 1$

$x_{i,j} \geq 0 \quad \forall i, j$

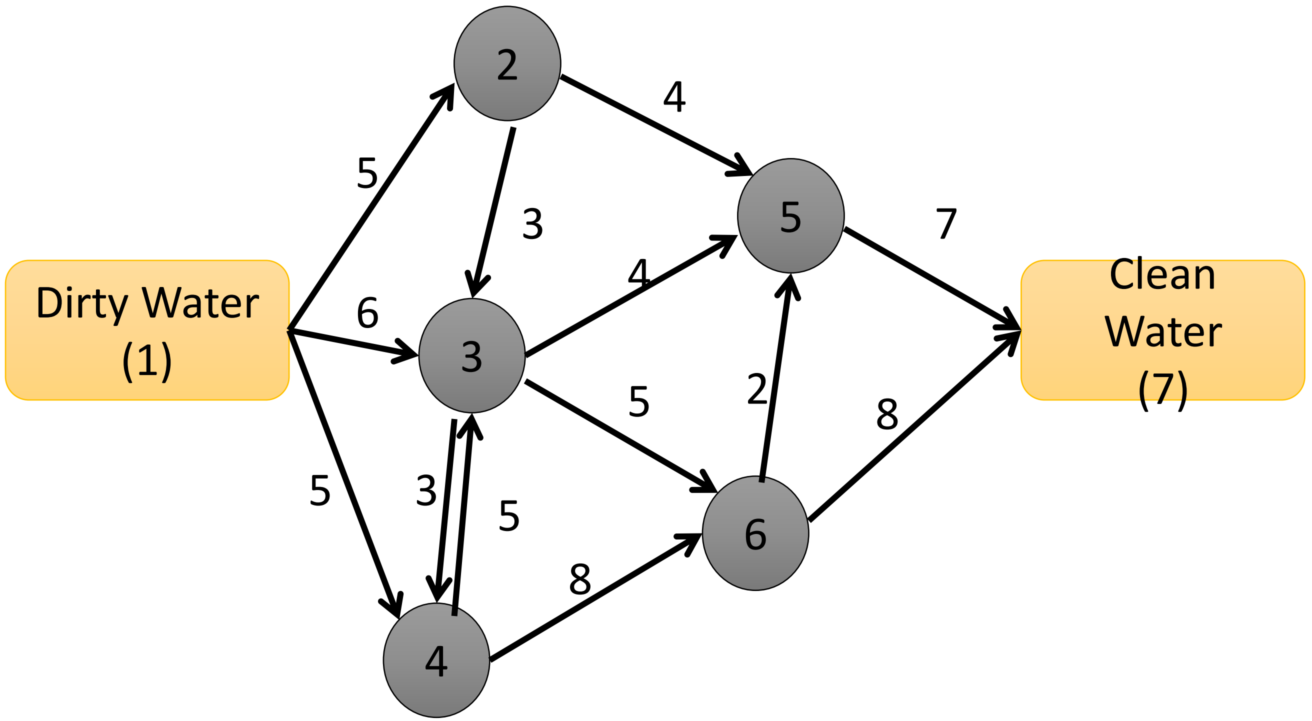
The Max Flow Problem

- **Objective**:
  - Find the maximum flow possible (people, vehicles, fluids, etc) through a network within a given amount of time

The Max Flow Problem

- Nodes:
  - Node Type 1: Source
  - Node Type 2: Sink
  - Node Type 3: Mid-Points (Transshipment Nodes)
- Arcs:
  - Directional connections between nodes with CAPACITIES
- Supply/Demand
  - Flow out of Node Type 1 equals flow into Node Type 2
  - Zero demand/supply at mid-points

Decision Variables:



- $X_{1,2}$=gallons of water that flow from node 1 to node 2 per hour
- $X_{1,3}$=gallons of water that flow from node 1 to node 3 per hour
- $X_{1,4}$=gallons of water that flow from node 1 to node 4 per hour
- $X_{2,3}$=gallons of water that flow from node 2 to node 3 per hour
- $X_{2,5}$=gallons of water that flow from node 2 to node 5 per hour
- $X_{3,4}$=gallons of water that flow from node 3 to node 4 per hour
- $X_{3,5}$=gallons of water that flow from node 3 to node 5 per hour
- $X_{3,6}$=gallons of water that flow from node 3 to node 6 per hour
- $X_{4,3}$=gallons of water that flow from node 4 to node 3 per hour
- $X_{4,6}$=gallons of water that flow from node 4 to node 6 per hour
- $X_{5,7}$=gallons of water that flow from node 5 to node 7 per hour
- $X_{6,5}$=gallons of water that flow from node 6 to node 5 per hour
- $X_{6,7}$=gallons of water that flow from node 6 to node 7 per hour

Constraints

Arc Capacities $x_{i,j} \le c_{i,j}$

Transshipment Nodes $\sum_{\text{arcs out}} x_{i,j} - \sum_{\text{arcs in}} x_{i,j} = 0$

Match Sinks and Sources $\sum_{i \in \text{Sources}} x_{i,j} - \sum_{j \in \text{Sinks}} x_{i,j} = 0$

$x_{i,j} \ge 0 \quad \forall i, j$

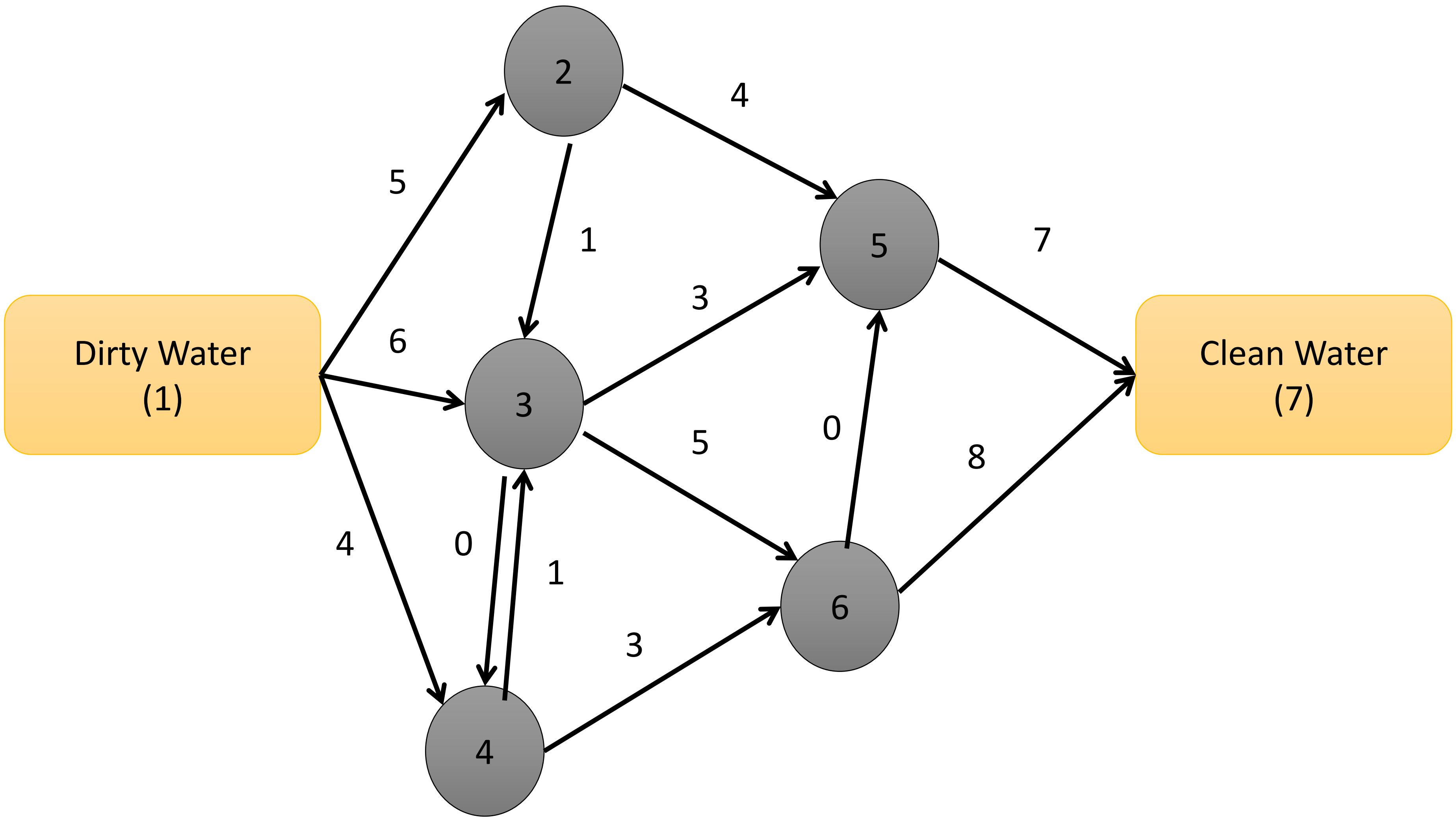
- Arc Capacity Constraints
  - Limits the flow along arcs
- Transshipment Node Constraints
  - Ensures that outside of Sinks and Sources all flow into nodes balances flow out
- Match flow from Sources to flow into Sinks
  - Balances flow through the whole system

In [10]:
```
obj <- c(1,1,1,0,0,0,0,0,0,0,0,0,0)
cap_A <- diag(13) # Capacity Constraints
cap_rhs <- c(5,6,5,3,4,3,4,5,5,8,7,2,8)
cap_dir <- rep('<=',13)
tr_A <- matrix(c(1,1,1,0,0,0,0,0,0,0,-1,0,-1, # Match Sink and Source
                1,0,0,-1,-1,0,0,0,0,0,0,0,0, # Transshipment Node 2
                0,1,0,1,0,-1,-1,-1,1,0,0,0,0, # Transshipment Node 3
                0,0,1,0,0,1,0,0,-1,-1,0,0,0, # Transshipment Node 4
                0,0,0,0,1,0,1,0,0,0,-1,1,0, # Transshipment Node 5
                0,0,0,0,0,0,0,1,0,1,0,-1,-1), nrow = 6, byrow=TRUE) # Transshipment Node 6
mat <- rbind(tr_A, cap_A)
dir <- c(rep("=", 6), cap_dir)
rhs <- c(rep(0,6), cap_rhs)
```

In [11]:
```
lp_maxf <- lp("max", obj, mat, dir, rhs, compute.sens = 1)
sensitivity_report(lp_maxf)
```

```
Objective Function Value:
15

 Objective Function
   variable final_value reduced_cost coefficient min_coef max_coef
1       x 1           5            0           1        1      Inf
2       x 2           6            0           1        1      Inf
3       x 3           4            0           1        0        1
4       x 4           1            0           0        0        0
5       x 5           4            0           0        0      Inf
6       x 6           0            0           0     -Inf        0
7       x 7           3            0           0        0        0
8       x 8           5            0           0        0      Inf
9       x 9           1            0           0        0        0
10     x 10           3            0           0       -1        0
11     x 11           7            0           0       -1      Inf
12     x 12           0            0           0     -Inf        0
13     x 13           8            0           0       -1      Inf
```


Dirty Water
(1)
2
3
4
5
6
Clean Water
(7)
5
6
4
1
4
3
5
0
1
3
0
7
8

The Max Flow Problem General Form

$$\text{Max} \sum_{i \in \text{Sources}} x_{i,j}$$

where $x_{i,j}$ **is the flow from** i **to** j

Arc Capacities $x_{i,j} \leq c_{i,j}$

Transshipment Nodes $\sum_{\text{arcs out}} x_{i,j} - \sum_{\text{arcs in}} x_{i,j} = 0$

Match Sinks and Sources $\sum_{i \in \text{Sources}} x_{i,j} - \sum_{j \in \text{Sinks}} x_{i,j} = 0$

$x_{i,j} \geq 0 \quad \forall i, j$

Alternative Algorithms

- Linear programming is a general-purpose solver and so is rarely if ever the fastest algorithm to solve a specific problem such as shortest path or max flow.
- https://www.quantamagazine.org/researchers-achieve-absurdly-fast-algorithm-for-network-flow-20220608/

	Nodes	Variables	Objective Function	Constraints
Transportation	Supply, Demand	One for every arc	Min Cost (or Max)	Can't exceed supply, must meet demand (one for every node)
Assignment	People, Tasks	One for every assign. (binary)	Min Cost (or Max)	All jobs filled, all people assigned (one for every node)
Transshipment	Supply, Demand, Trans.	One for every arc	Min Cost (or Max)	Can't exceed supply, must meet demand, trans. Flow=0 (one for every node)
Shortest Path	Origin, Dest., Mid-points	One for every arc (binary)	Min Dist/Time (Max NOT always ok)	Flow out of origin=1, flow into dest.=1, Trans. Flow=0 (one for every node)
Max Flow	Source, Sink, Mid-points	One for every arc	Max Flow into destination	Can't exceed arc capacity, trans. Flow=0 (one for every arc and trans. Node)