LP Modeling and Optimization of Networked Systems: Transportation, Assignment, Minimum Cost Flow, Shortest Path, and Max Flow

Andrés D. González

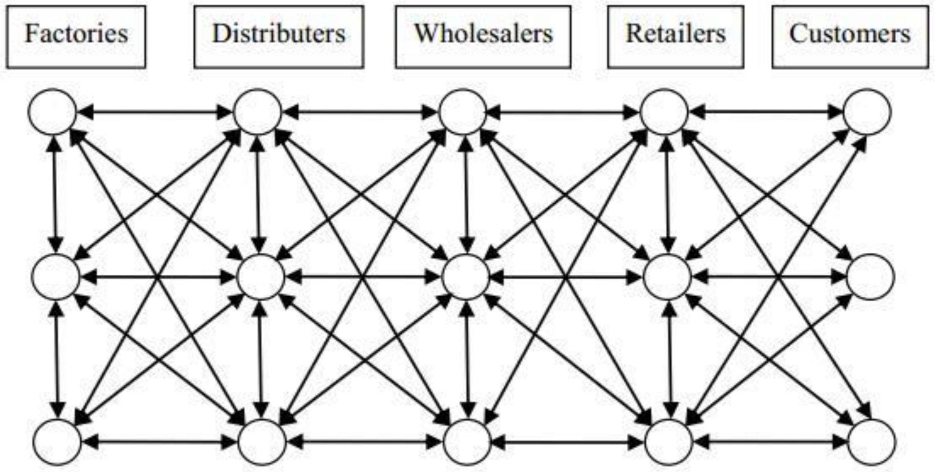
Assistant Professor
School of Industrial and Systems Engineering, The University of Oklahoma

ISE 4623/5023: Deterministic Systems Models / Systems Optimization

The University of Oklahoma, Norman, OK, USA

Why are networks of interest?

- Supply chain networks
- Transportation
- Energy
- Water supply and management
- Communications



(Norouzi et al, 2018)

Challenges
- Network modeling
- Handling uncertainty
- Large scale problems

Transportation Problem

Sets:

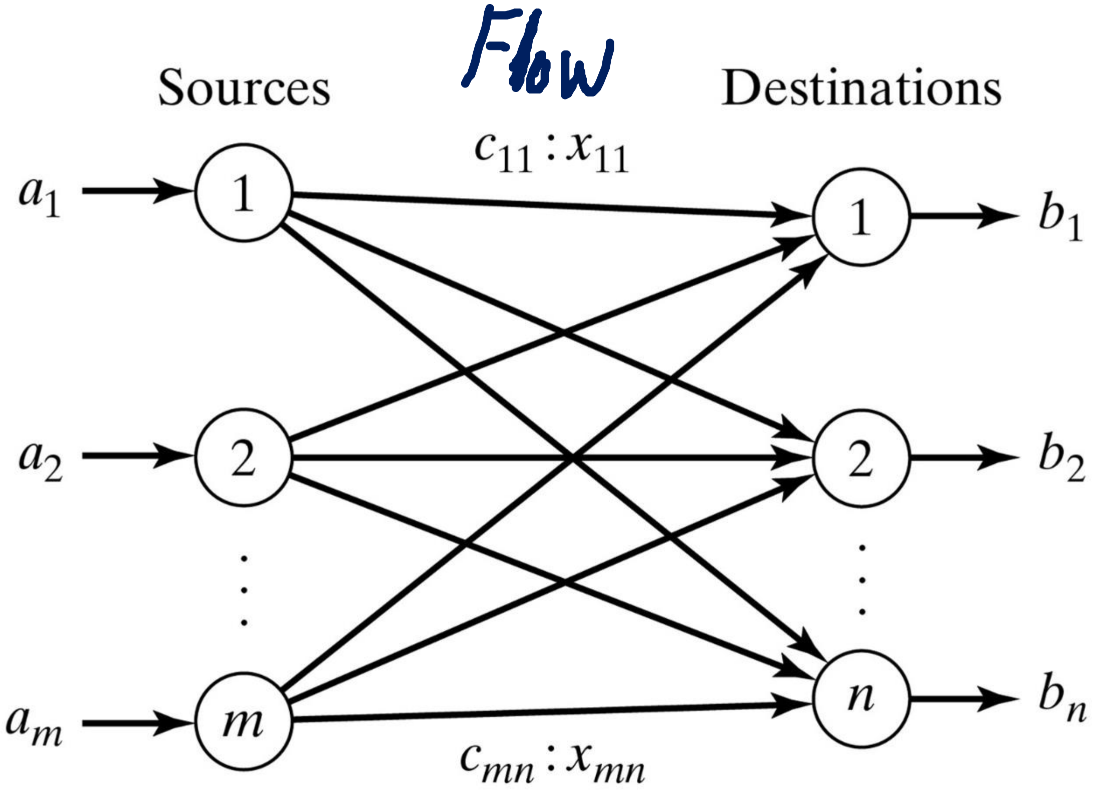
- S = set of *sources* $\{1,2,\dots,m\}$
- D = set of *destinations* $\{1,2,\dots,n\}$

Parameters:

- a_i = supply of source node $i \in S$
- b_j = demand of destination node $j \in D$
- c_{ij} = unit cost of flow through arc (i,j), where $i \in S$ and $j \in D$

Objective:

- Minimize the transportation cost while satisfying demand and supply constraints



Transportation Problem

Sets:

- S = set of *sources* $\{1,2,\ldots,m\}$
- D = set of *destinations* $\{1,2,\ldots,n\}$

Parameters:

- a_i = supply of source node $i \in S$
- b_j = demand of destination node $j \in D$
- c_{ij} = unit cost of flow through arc (i,j), where $i \in S$ and $j \in D$

Decision variables:

x_{ij}= flow through arc (i,j), where $i \in S$ and $j \in D$

Objective function:

$$\min \underbrace{\sum_{i\in S}\sum_{j\in D}}\, c_{ij}x_{ij}$$

Assumes ALL Source nodes are connected to a Destination node

Constraints:

$$\sum_{i\in S} x_{ij} = b_j \qquad ,\forall j \in D$$

$$\sum_{j\in D} x_{ij} = a_i \qquad ,\forall i \in S$$

Assumes that it is a balanced network

$$x_{ij} \geq 0 \qquad ,\forall i \in S, \forall j \in D$$

Example – Shipment of pens and pencils

Demand

City	Demand (pencils)	Demand (pens)
Boston	50	40
New York	50	30
Seattle	10	30

City	Production capacity (pencils)	Production capacity (pens)
Detroit	50	60
Denver	60	40

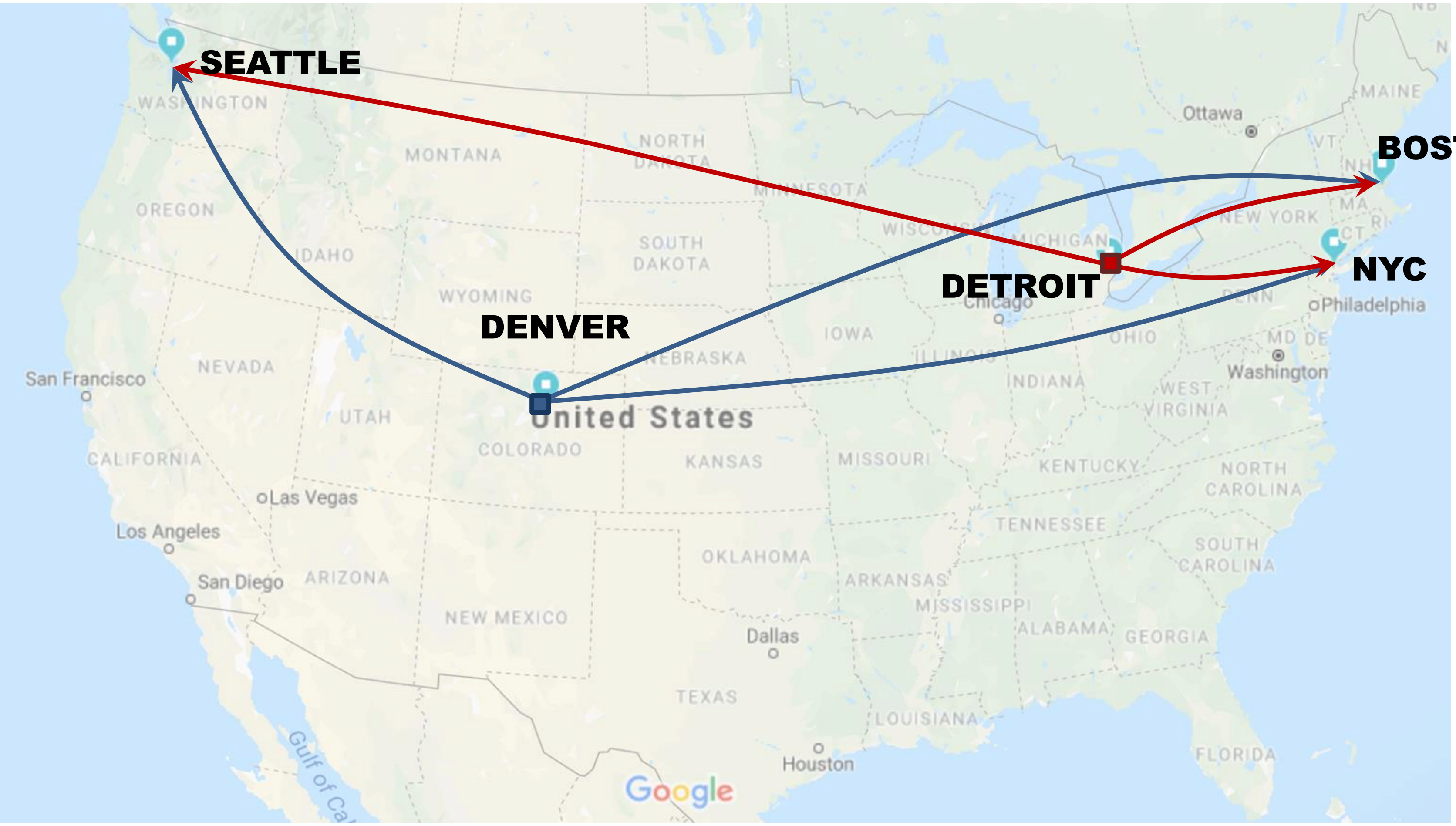
Supply

Actual amount shipped

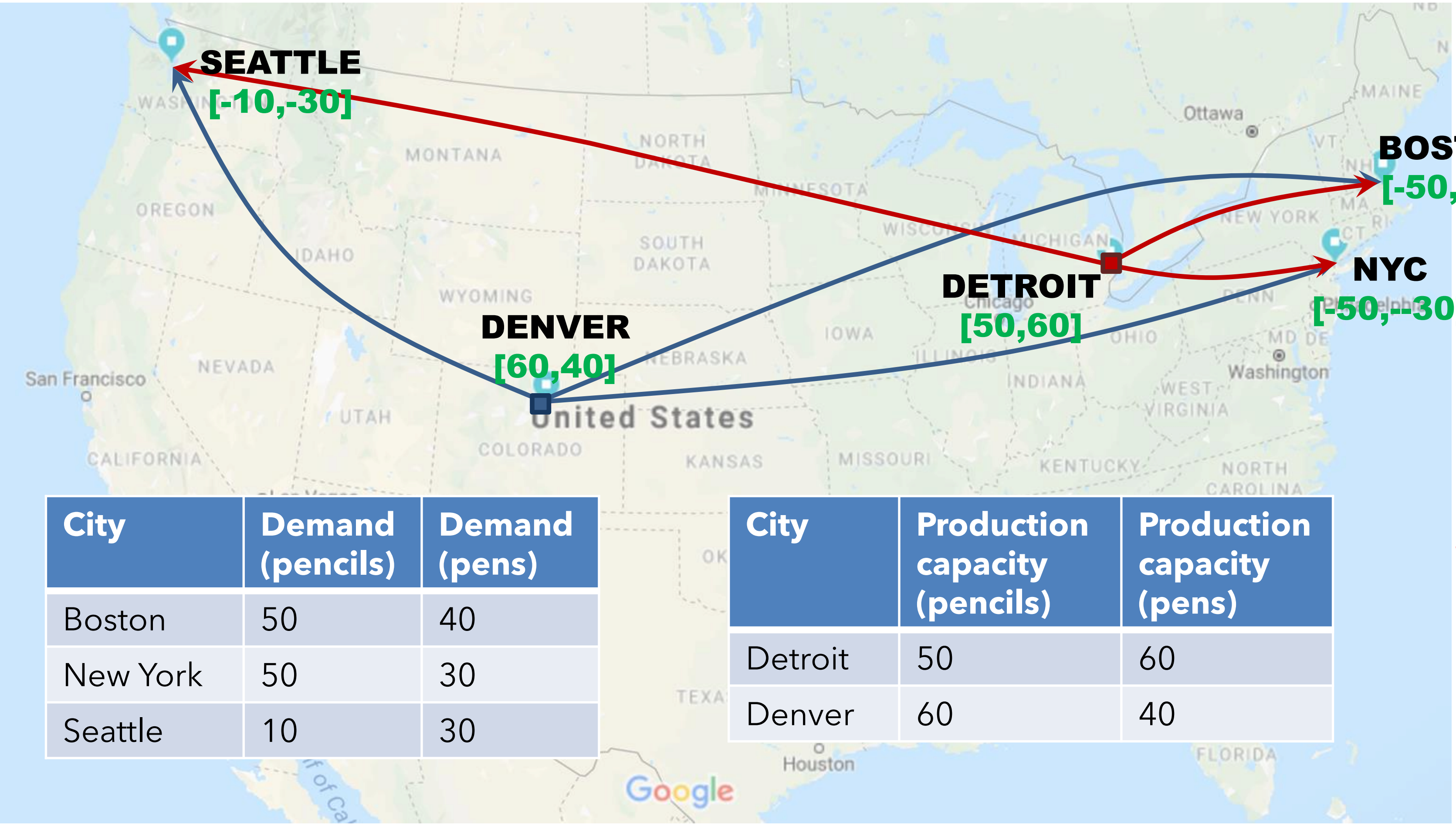
Arc	Cost (pencils)	Cost (pens)	Arc capacity
Detroit → Boston	10	20	100
Detroit → New York	20	20	80
Detroit → Seattle	60	80	120
Denver → Boston	40	60	120
Denver → New York	40	70	120
Denver → Seattle	30	30	120

Maximum capacity of the flow from the arc

Example – Shipment of pens and pencils



Example – Shipment of pens and pencils



Demand = Negative Value

City	Demand (pencils)	Demand (pens)
Boston	50	40
New York	50	30
Seattle	10	30

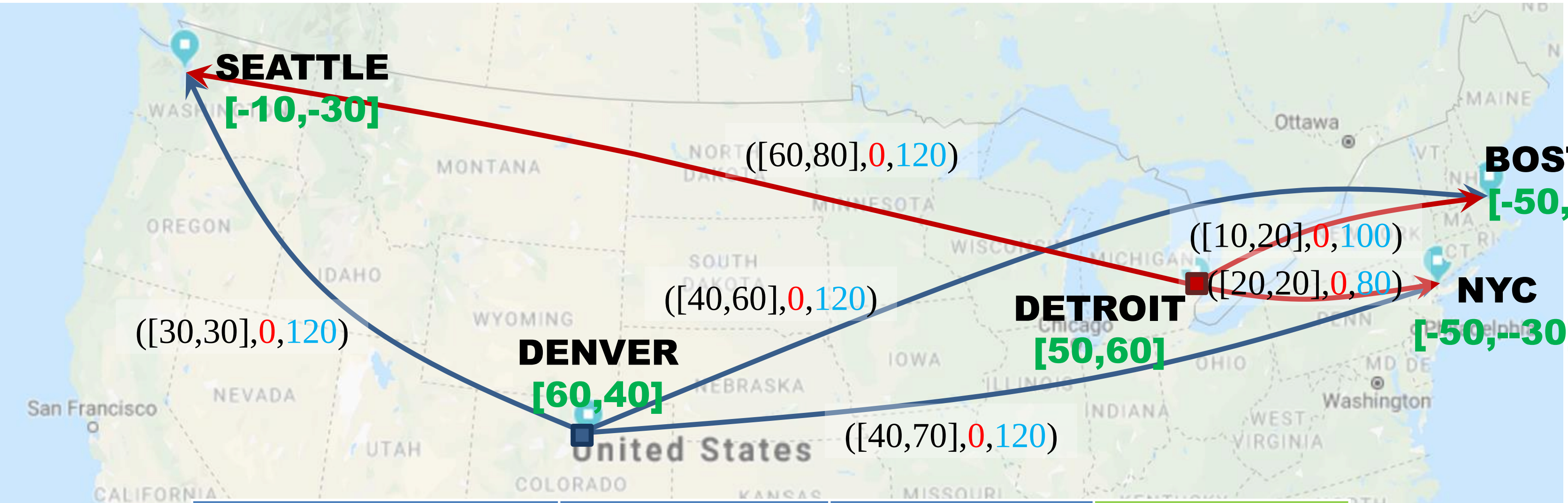
City	Production capacity (pencils)	Production capacity (pens)
Detroit	50	60
Denver	60	40

Supply = Positive Value

Example – Shipment of pens and pencils

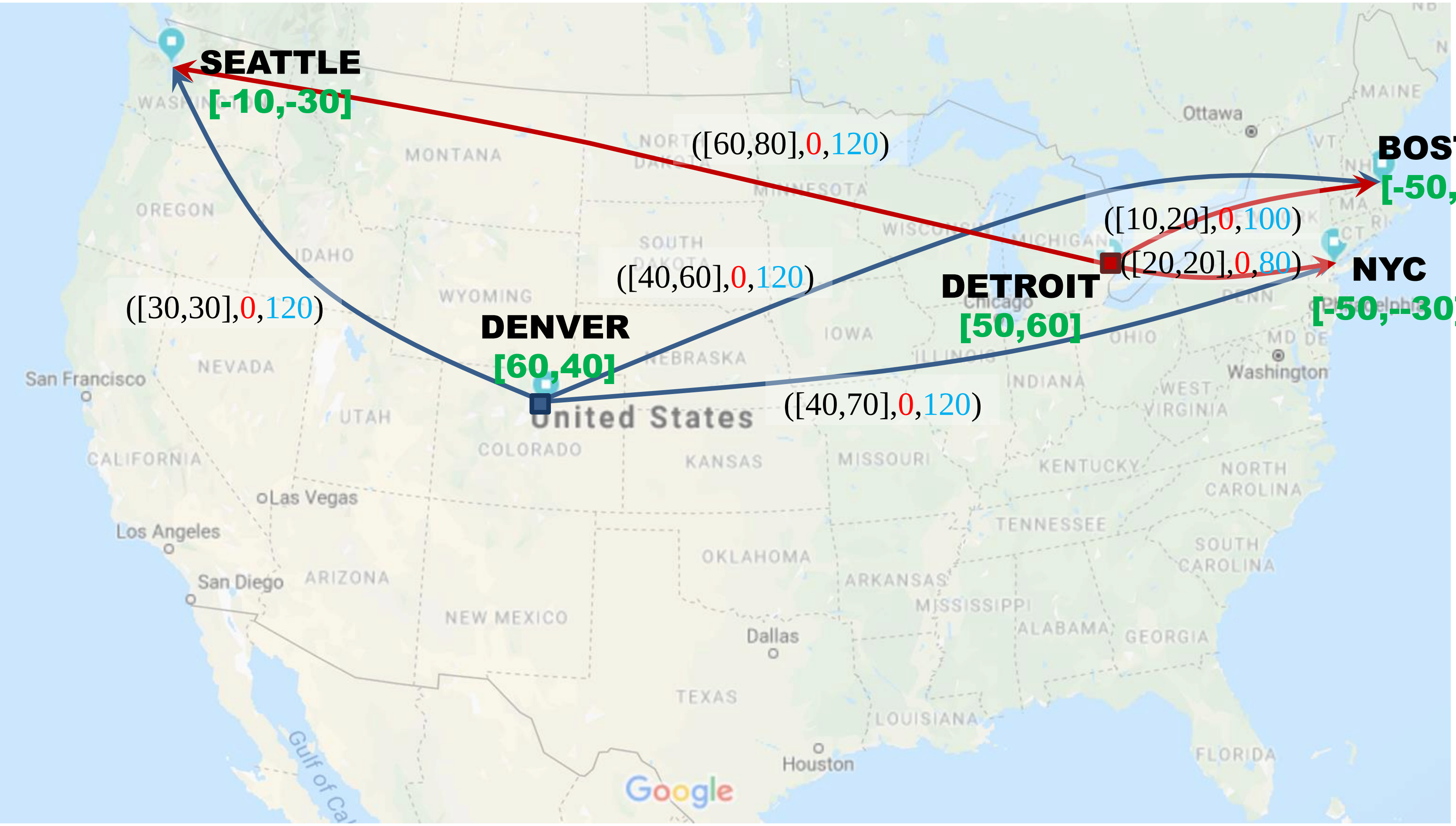
Blue = Upper bound

Red = Lower Bound



Arc	Cost (pencils)	Cost (pens)	Arc capacity
Detroit → Boston	10	20	100
Detroit → New York	20	20	80
Detroit → Seattle	60	80	120
Denver → Boston	40	60	120
Denver → New York	40	70	120
Denver → Seattle	30	30	120

Example – Shipment of pens and pencils



Assignment Problem

Sets:

- W = set of *workers* $\{1, 2, \ldots, n\}$
- J = set of *jobs* $\{1, 2, \ldots, n\}$

Cost doesn't mean the cost of the employee. It means the relationship between the worker and the job. So you want to maximize the "cost", which would represent the worker's preference

Parameters:

- c_{ij} = cost of assigning worker $i \in W$ to job $j \in J$

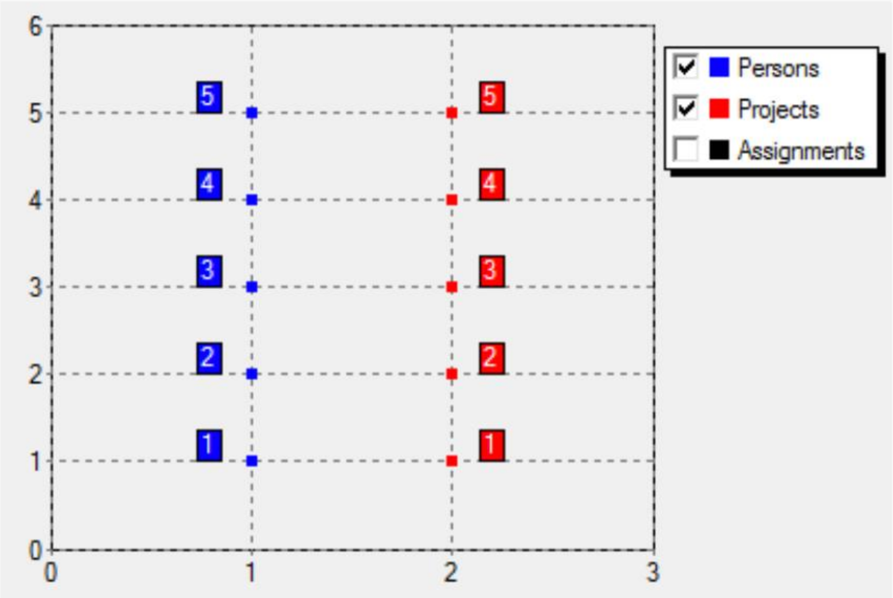
Objective:

- Minimize the total assignment cost while guaranteeing that (i) each worker has exactly one job assigned and (ii) each job has exactly one worker assigned

Assignment Problem - Example

```
TYPE:          Assignment problem
DIFFICULTY:    1
FEATURES:      simple LP problem, graphical representation of results
DESCRIPTION:   A set of projects is assigned to persons with the
               objective to maximize the overall satisfaction.
               A preference rating per person and project is given.
               In this model formulation the solution to the LP problem
               is integer, there is no need to define the decision
               variables explicitly as binaries.
```

```
PREF:: [1, 2, 3, 5, 4,
        3, 2, 5, 4, 1,
        3, 4, 1, 5, 2,
        4, 3, 2, 5, 1,
        2, 3, 5, 4, 1]
```

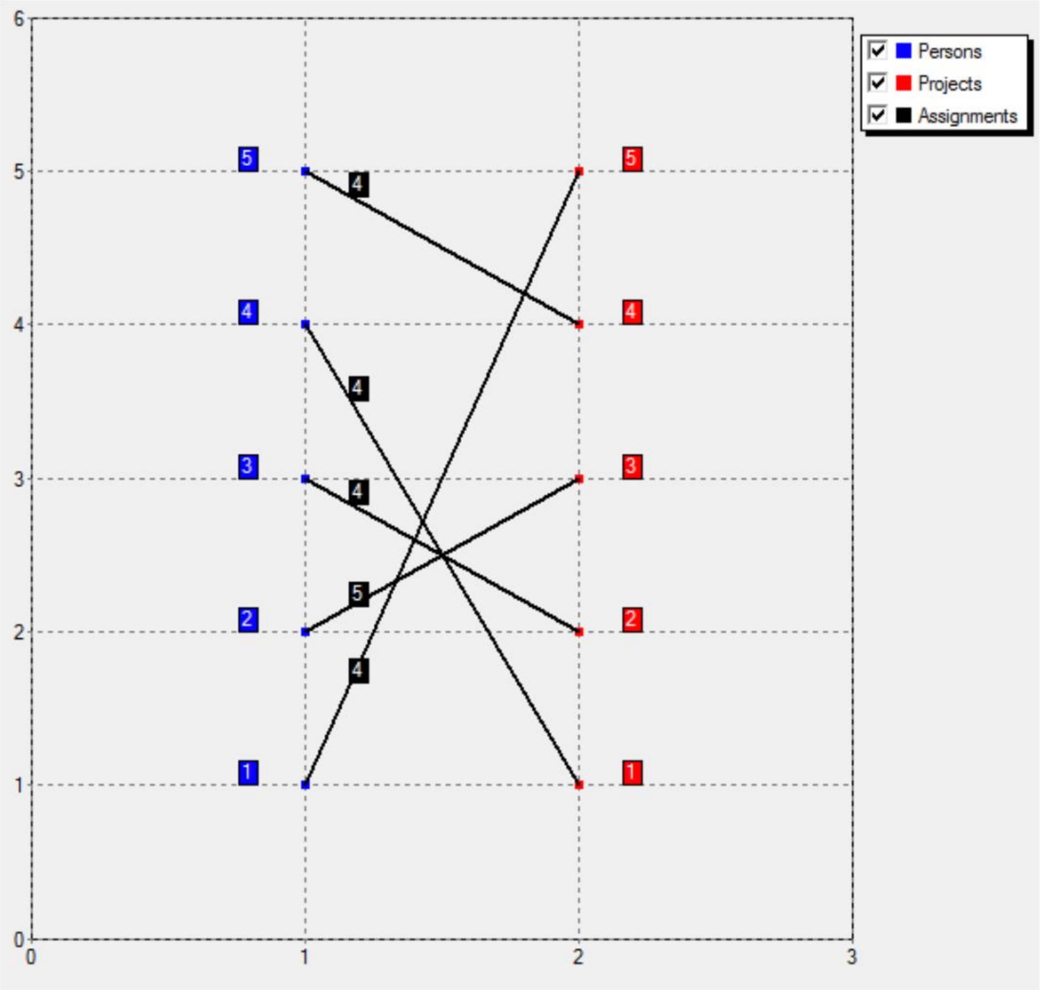


{i} RP	{i} RP	[ ] PREF
1	1	1
1	2	2
1	3	3
1	4	5
1	5	4
2	1	3
2	2	2
2	3	5
2	4	4
2	5	1
3	1	3
3	2	4
3	3	1
3	4	5
3	5	2
4	1	4
4	2	3
4	3	2
4	4	5
4	5	1
5	1	2
5	2	3
5	3	5
5	4	4
5	5	1

Assignment Problem - Example

```
PREF:: [1, 2, 3, 5, 4,
        3, 2, 5, 4, 1,
        3, 4, 1, 5, 2,
        4, 3, 2, 5, 1,
        2, 3, 5, 4, 1]
```

```
Total satisfaction score: 21
Person 1: project 5
Person 2: project 3
Person 3: project 2
Person 4: project 1
Person 5: project 4
```



Assignment Problem

Sets:

- W = set of *workers* $\{1,2,\dots,n\}$
- J = set of *jobs* $\{1,2,\dots,n\}$

Parameters:

- c_{ij} = cost of assigning worker $i \in W$ to job $j \in J$

Variables:

- x_{ij}: binary variable that is 1 if worker $i \in W$ is assigned to job $j \in J$; 0 otherwise.

Objective function:

$$\min z = \sum_{i \in W} \sum_{j \in J} c_{ij} x_{ij}$$

Constraints:

$$\sum_{i \in W} x_{ij} = 1, \qquad \forall j \in J$$

$$\sum_{j \in J} x_{ij} = 1, \qquad \forall i \in W$$

Has to be 1 or 0

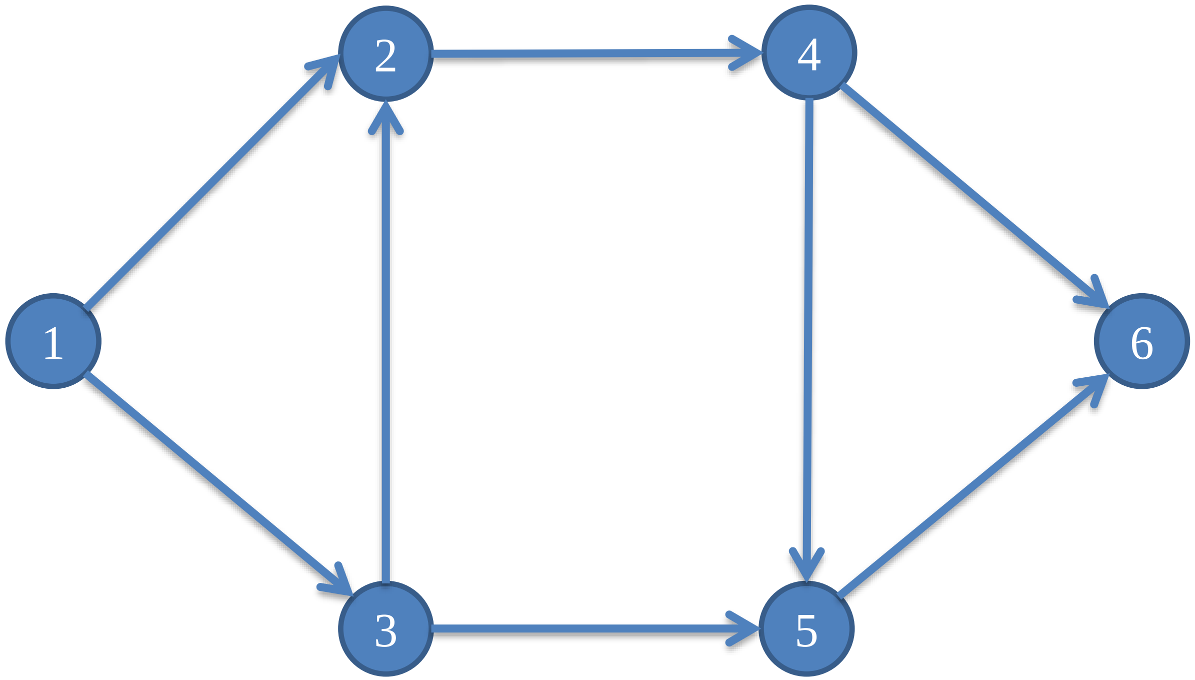
$$x_{ij} \in \{0,1\}, \quad \forall i \in W, \ \forall j \in J$$

Binary set

Minimum cost flow problem

Given:

- G = *directed network* (N, A)
- N = set of *n nodes*
- A = set of *m directed arcs*.

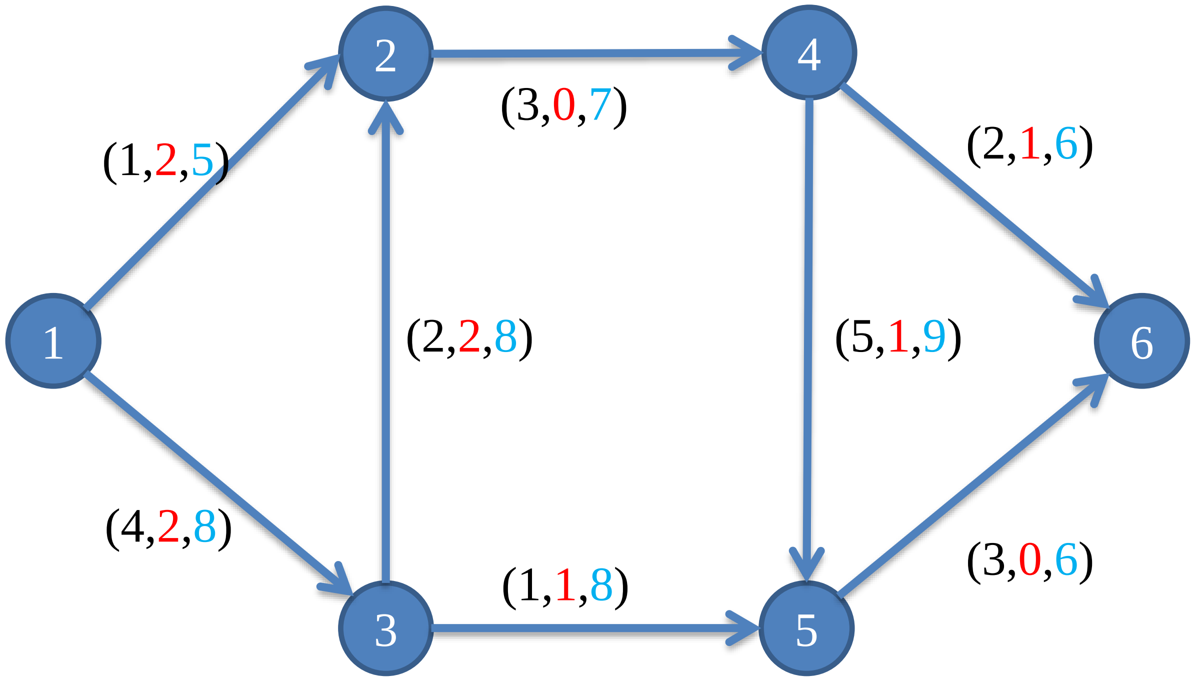


(Ahuja, Magnanti & Orlin, 1993; Kennington, 2006; Vaidyanathan, 2010)

Minimum cost flow problem

Each arc $(i, j) \in A$ has an associated:

- *Cost* $\boldsymbol{c_{ij}}$ (per unit flow on that arc),
- Lower bound $\boldsymbol{l_{ij}}$ *and a capacity* $\boldsymbol{u_{ij}}$ (minimum and maximum flow on the arc).

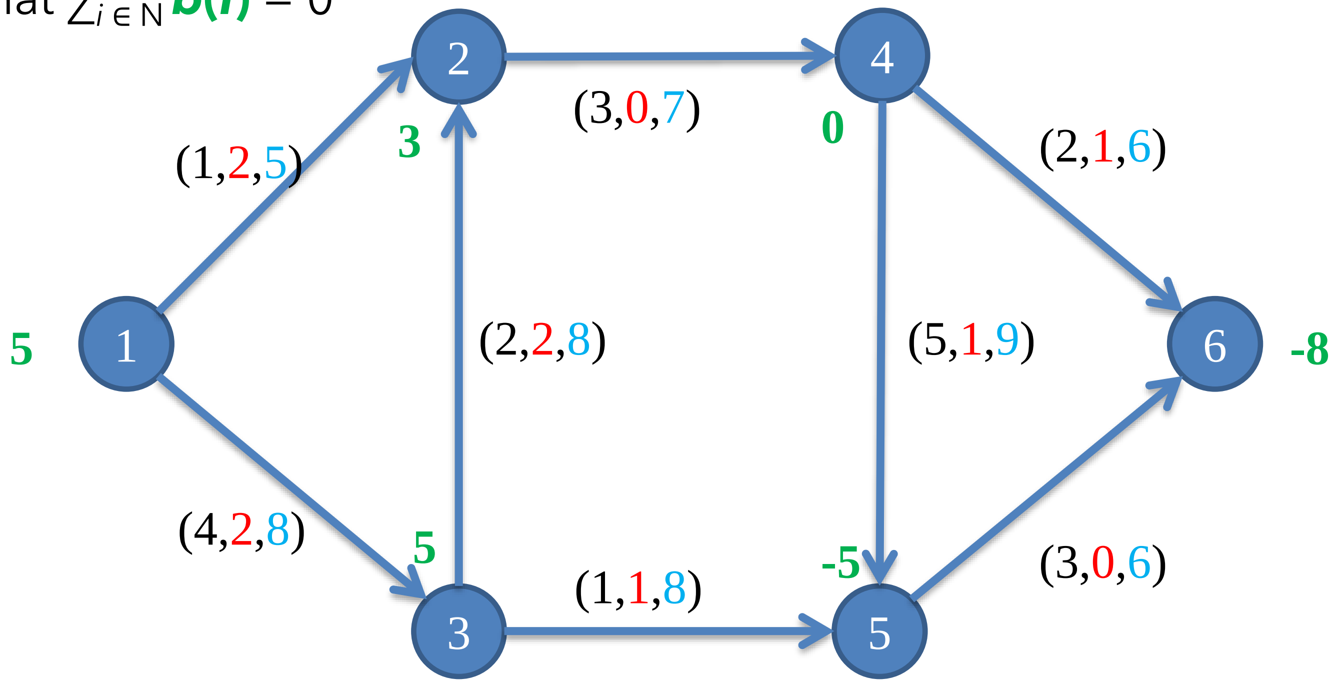


Blue = Upper bound

Red = Lower Bound

Minimum cost flow problem

- We associate with each node $i \in N$ an integer $\boldsymbol{b(i)}$ representing its supply/demand.
  - If $\boldsymbol{b(i)} > 0$, node i is a *supply node*
  - If $\boldsymbol{b(i)} < 0$, then node i is a *demand node* with a demand of $-\boldsymbol{b(i)}$
  - If $\boldsymbol{b(i)} = 0$, then node i is a *transshipment node*
  - We assume that $\sum_{i \in N} \boldsymbol{b(i)} = 0$



Blue = Upper bound

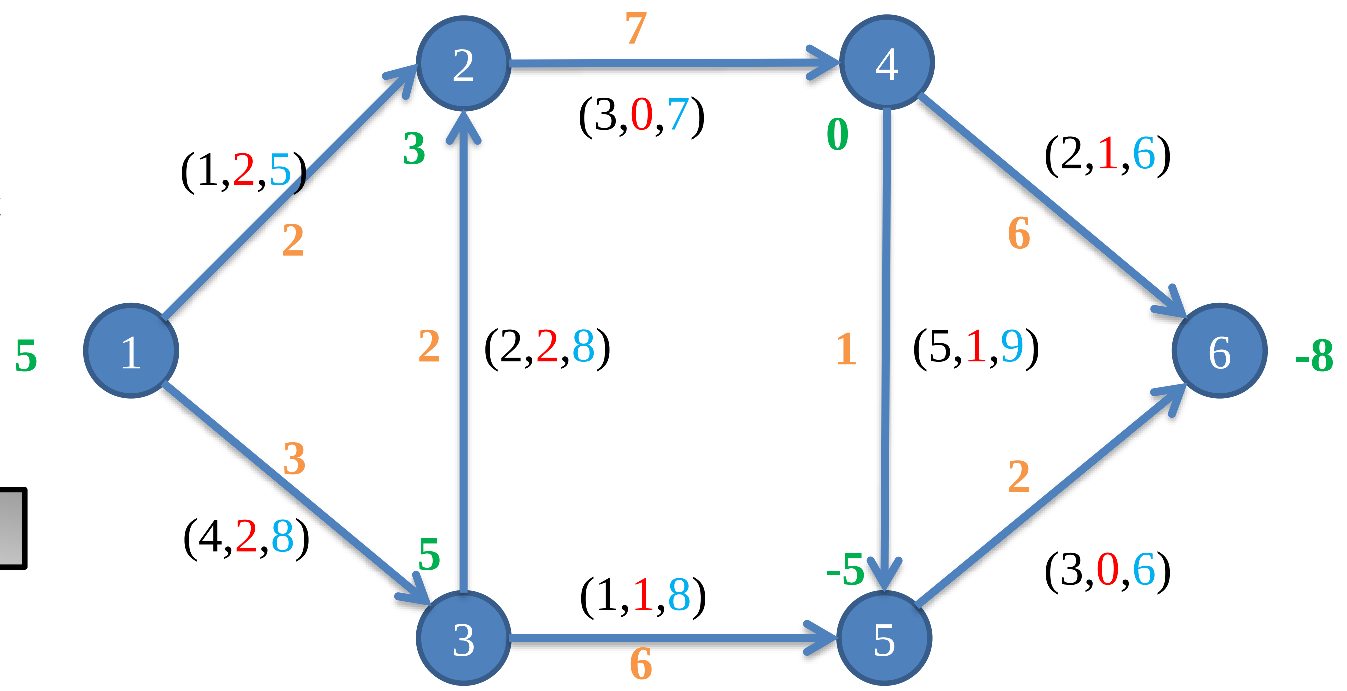
Red = Lower Bound

Minimum cost flow problem

Objective
The *minimum cost flow problem* seeks a least cost shipment of a commodity through a network to satisfy demands at certain nodes by available supplies at other nodes.

The decision variables x_{ij} are arc flows defined for each arc $(i, j) \in A$.

Total cost = 68



Blue = Upper bound

Red = Lower Bound

Minimum Cost Flow Problem (MCFP)

- Sets:

N: Set of nodes $\{1,2,3,\ldots,n\}$

A: Set of arcs

- Parameters:

c_{ij}: unit cost of sending a commodity through arc $(i,j) \in A$

b_i: demand/supply of commodities in node $i \in N$

u_{ij}: maximum flow through arc $(i,j) \in A$

l_{ij}: minimum flow through arc $(i,j) \in A$

- Variables:

x_{ij}: flow through arc $(i,j) \in A$

- Objective function:

$$\min z = \sum_{(i,j)\in A} c_{ij}x_{ij}$$

- Constraints:

$$\sum_{j:(i,j)\epsilon\, A} x_{ij} - \sum_{j:(j,i)\epsilon\, A} x_{ji} = b_i, \qquad \forall i \in N$$

$$x_{ij} \le u_{ij}, \qquad \forall (i,j) \in A$$

$$x_{ij} \ge l_{ij}, \qquad \forall (i,j) \in A$$

Minimum cost flow problem

```
from gurobipy import *

#create model
m=Model('MinimumCostFlowProblem')

#SETS AND #PARAMETERS

#Set N, parameter b
N, b= multidict({
    ('node1'): 5,
    ('node2'): 3,
    ('node3'):5,
    ('node4'):0,
    ('node5'):-5,
    ('node6'):-8
})

#Set A, Parameters l, u, c

A, l, u, c = multidict({
    ('node1','node2'): [2,5,1],
    ('node1','node3'): [2,8,4],
    ('node3','node2'): [2,8,2],
    ('node2','node4'): [0,7,3],
    ('node3','node5'): [1,8,1],
    ('node4','node5'): [1,9,5],
    ('node4','node6'): [1,6,2],
    ('node5','node6'): [0,6,3]
})
```

Minimum cost flow problem

```
#VARIABLES
x=m.addVars(A, obj=c, name="x")

#OBJ FUNCTION
#Already done (when writing obj=c).
#You could also write:
#z=sum(c[i,j]*x[i,j] for i,j in A)

#CONSTRAINTS
#upper bound
m.addConstrs((x[i,j]<=u[i,j] for i,j in A), "maxFlow")

#lower bound
m.addConstrs((x[i,j]>=l[i,j] for i,j in A), "minFlow")

#flow-balance constraint
m.addConstrs((x.sum(i,'*')-x.sum('*',i)==b[i] for i in N), "flowBalance")

m.update()

m.params.outputflag=0
m.optimize()

if m.status==GRB.OPTIMAL:
    print('**Optimal Solution Found**\n--Objective Function--\n %g' % m.objVal)
    print('--Decision Variables--')
    for v in m.getVars(): print('%s: %g' % (v.varName, v.x))
    print('--Dual Variables--')
    for constraint in m.getConstrs(): print ('Dual variable of constraint %s: %g'%(constraint.constrName,constraint.Pi))
```

Minimum cost flow problem

```
**Optimal Solution Found**
--Objective Function--
 68
--Decision Variables--
x[node1,node2]: 2
x[node1,node3]: 3
x[node3,node2]: 2
x[node2,node4]: 7
x[node3,node5]: 6
x[node4,node5]: 1
x[node4,node6]: 6
x[node5,node6]: 2
--Dual Variables--
Dual variable of constraint maxFlow[node1,node2]: 0
Dual variable of constraint maxFlow[node1,node3]: 0
Dual variable of constraint maxFlow[node3,node2]: 0
Dual variable of constraint maxFlow[node2,node4]: 0
Dual variable of constraint maxFlow[node3,node5]: 0
Dual variable of constraint maxFlow[node4,node5]: 0
Dual variable of constraint maxFlow[node4,node6]: -6
Dual variable of constraint maxFlow[node5,node6]: 0
Dual variable of constraint minFlow[node1,node2]: 4
Dual variable of constraint minFlow[node1,node3]: 0
Dual variable of constraint minFlow[node3,node2]: 9
Dual variable of constraint minFlow[node2,node4]: 0
Dual variable of constraint minFlow[node3,node5]: 0
Dual variable of constraint minFlow[node4,node5]: 0
Dual variable of constraint minFlow[node4,node6]: 0
Dual variable of constraint minFlow[node5,node6]: 0
Dual variable of constraint flowBalance[node1]: 8
Dual variable of constraint flowBalance[node2]: 11
Dual variable of constraint flowBalance[node3]: 4
Dual variable of constraint flowBalance[node4]: 8
Dual variable of constraint flowBalance[node5]: 3
Dual variable of constraint flowBalance[node6]: 0
```

Multicommodity Minimum Cost Flow Problem (MMCFP)

- Sets:

N: Set of nodes $\{1,2,3,\dots,n\}$

A: Set of arcs

L: Set of commodities

- Parameters:

c_{ijl}: unit cost of sending a commodity $l \in L$ through arc $(i,j) \in A$

b_{il}: demand/supply of commodity $l \in L$ in node $i \in N$

u_{ij}: maximum total flow through arc $(i,j) \in A$

l_{ij}: minimum total flow through arc $(i,j) \in A$

- Variables:

x_{ijl}: flow of commodity $l \in L$ through arc $(i,j) \in A$

- Objective function:

$$\min z = \sum_{l \in L} \sum_{(i,j) \in A} c_{ijl} x_{ijl}$$

- Constraints:

$$\sum_{j:(i,j) \epsilon A} x_{ijl} - \sum_{j:(j,i) \epsilon A} x_{jil} = b_{il}, \quad \forall i \in N, \forall l \in L$$

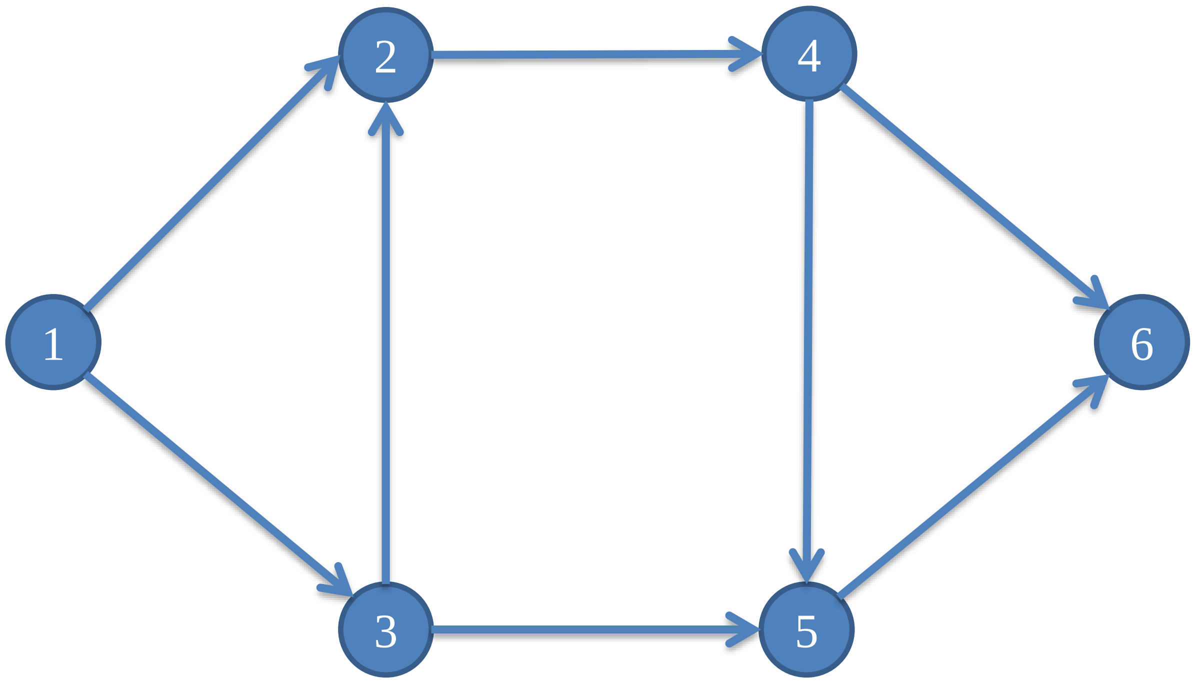
$$\sum_{l \in L} x_{ijl} \le u_{ij}, \qquad \forall (i,j) \in A$$

$$\sum_{l \in L} x_{ijl} \ge l_{ij}, \qquad \forall (i,j) \in A$$

Minimum cost flow problem: related problems

The minimum cost flow problem can be applied to solve several other network flow problems such as:

- Shortest path problem
- Maximum flow problem
- Transportation problem
- Assignment problem
- Airplane hopping problem



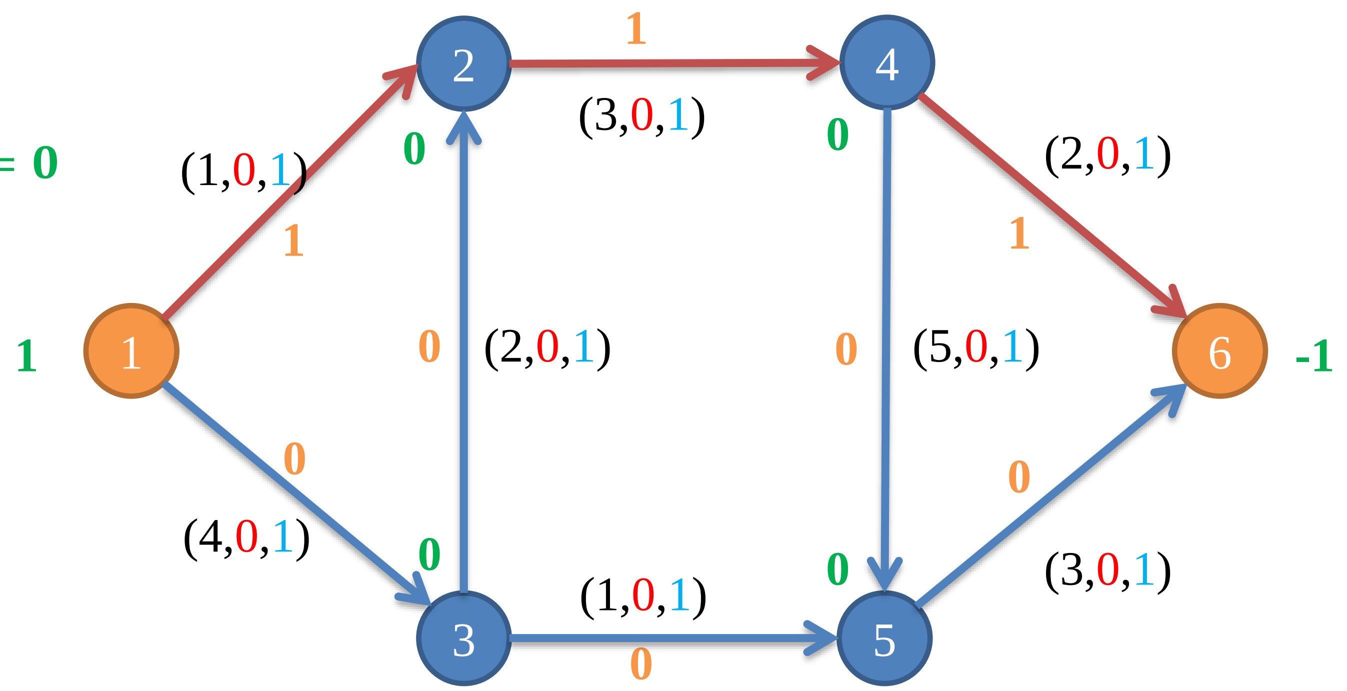
(Ahuja, Magnanti & Orlin, 1993; Kennington, 2006; Vaidyanathan, 2010)

Shortest path problem

Objective
The *shortest path problem* seeks a least cost shipment of a commodity between a two nodes.

$l_{ij} = 0$
$u_{ij} = 1$
$b(i \neq start\ or\ end) = 0$
$b(start) = 1$
$b(end) = -1$

Total cost = 6



Blue = Upper bound
Red = Lower Bound

Maximum flow problem

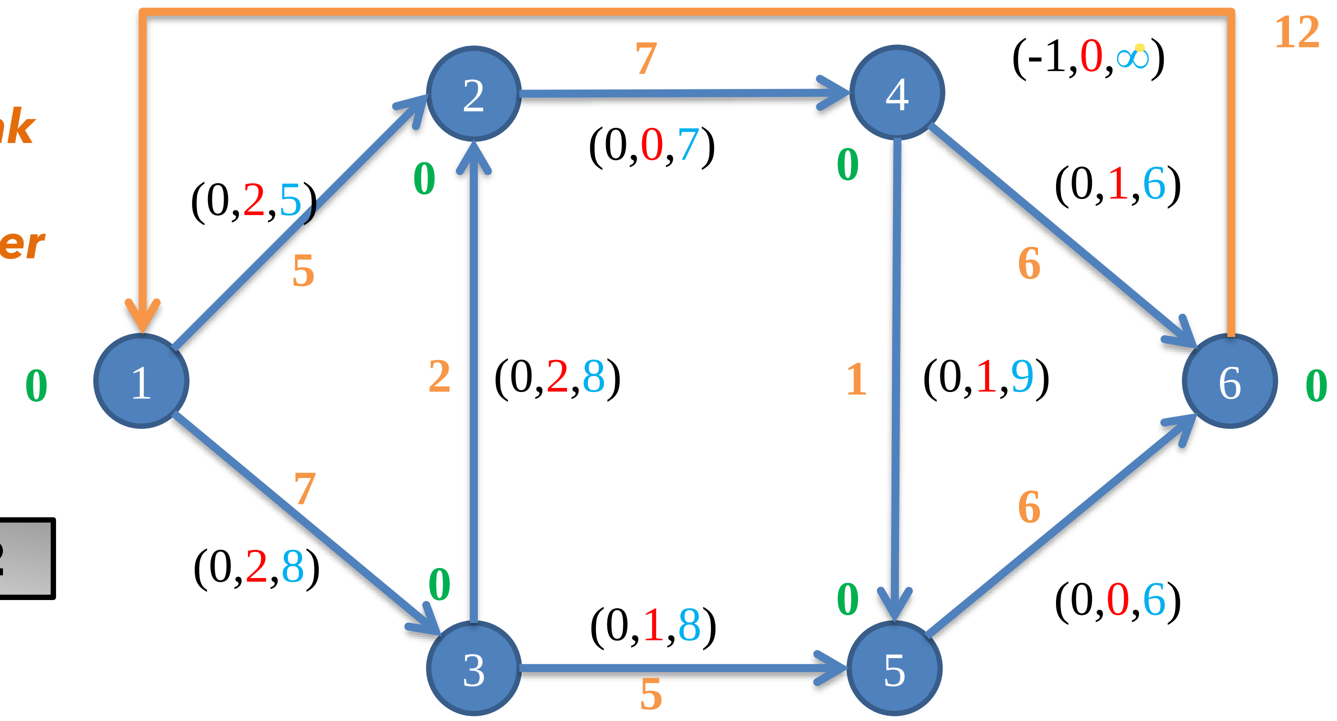
Objective

The *maximum flow problem* is to find the maximum feasible flow between a source node and a sink node

Artificial arc between the sink and the source, with cost 0, lower bound 0.

$b(i) = 0$

Total flow= 12



Blue = Upper bound

Red = Lower Bound

THANK YOU

QUESTIONS?

Andrés D. González | andres.gonzalez@ou.edu