

# NETWORK FLOWS: THE MINIMUM COST NETWORK FLOW PROBLEM

# elements of the MCNFP

Defined on a network:  $G = (N, A)$

where  $N$  is a set of  $n$  nodes:  $\{1, 2, \dots, n\}$

and  $A$  is a set of  $m$  arcs as a subset of  $N \times N$

Each node  $i$  has an associated value  $b(i)$

- $b(i) > 0 \rightarrow$  node  $i$  is a *supply* node with a supply of  $b(i)$  units
- $b(i) < 0 \rightarrow$  node  $i$  is a *demand* node with a demand for  $b(i)$  units of some commodity
- $b(i) = 0 \rightarrow$  node  $i$  is a *transshipment* node

# elements of the MCNFP

The set  $A$  is a set of *arcs*, e.g.  $(i, j)$  for  $i \in N, j \in N$   
each of which may carry *flow of a commodity*

Decision variable:  $x_{ij}$  determines the units of flow on arc  $(i, j)$

Arc  $(i, j)$  has certain characteristics:

- cost  $c_{ij}$  per unit of flow on arc  $(i, j)$
- upper bound on flow of  $u_{ij}$  (capacity)
- lower bound on flow of  $\ell_{ij}$  (usually 0)

# example MCNFP

$$N = \{1, 2, 3, 4\}$$

$$b(1) = 5, b(2) = -2, b(3) = 0, b(4) = -3$$

Supply since positive demand since negative Transshipment since 0

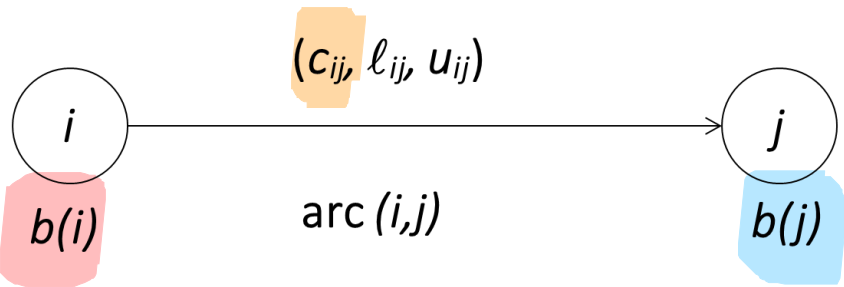
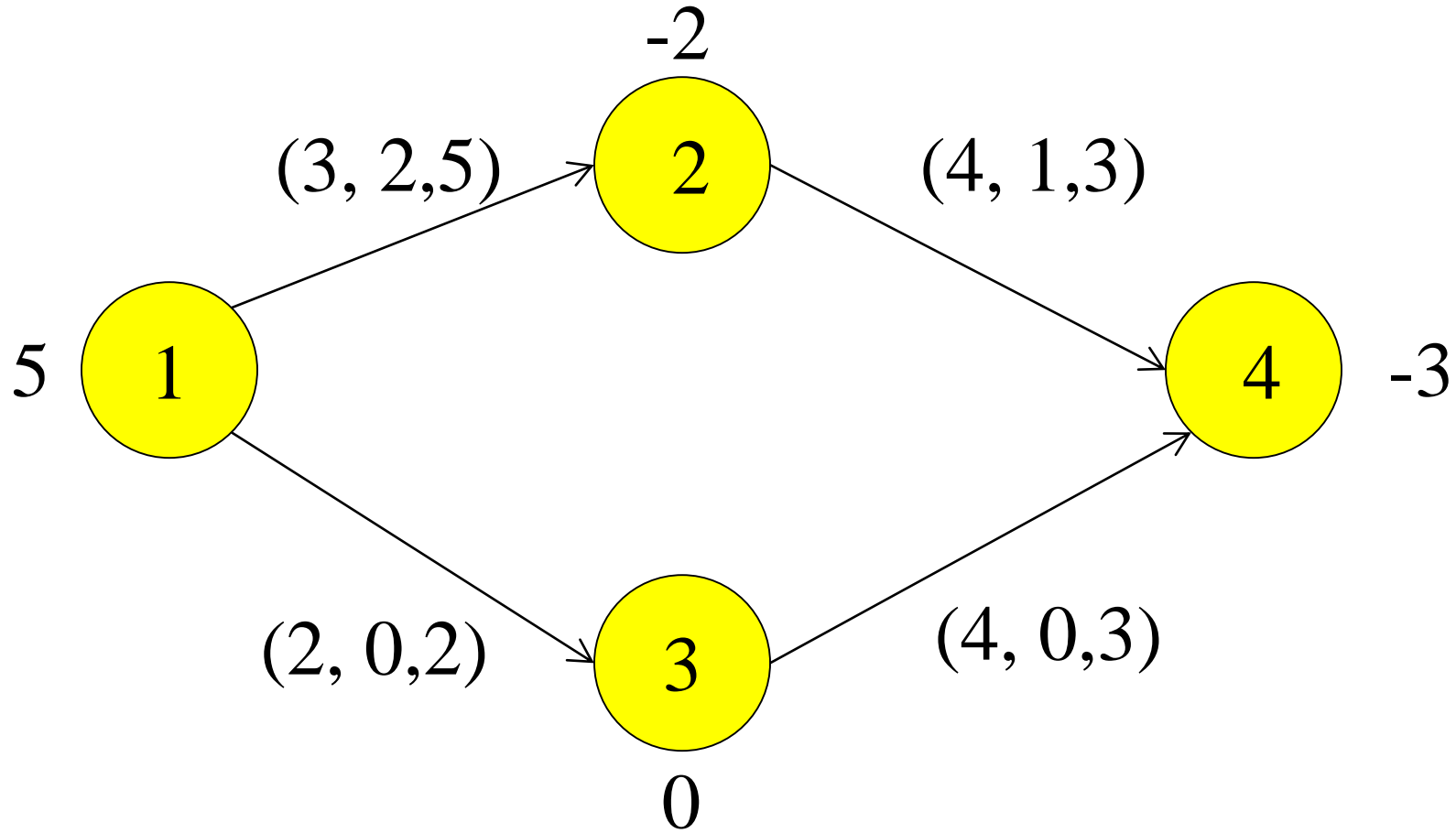
$$A = \{(1,2), (1,3), (2,4), (3,4)\}$$

$$c_{12} = 3, c_{13} = 2, c_{24} = 4, c_{34} = 4$$

$$\ell_{12} = 2, \ell_{13} = 0, \ell_{24} = 1, \ell_{34} = 0$$

$$u_{12} = 5, u_{13} = 2, u_{24} = 3, u_{34} = 3$$

# example MCNFP



# requirements for a feasible flow

- Flow on all arcs is within the allowable bounds:  
 $\ell_{ij} \leq x_{ij} \leq u_{ij}$  for all arcs  $(i, j)$
- Flow is balanced at all nodes:  
flow out of node  $i$  – flow into node  $i = b(i)$
- MCNFP: find a minimum-cost feasible flow

# MCNFP formulation

$$\text{minimize } \sum_{(i,j) \in A} c_{ij} x_{ij}$$

$$\text{subject to } \sum_{j: (i,j) \in A} x_{ij} - \sum_{j: (j,i) \in A} x_{ji} = b_i \quad \forall i \in N$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in A$$

# MCNFP example formulation (expanded)

$$\text{minimize } 3x_{12} + 2x_{13} + 4x_{24} + 4x_{34}$$

$$\text{s.t. } x_{12} + x_{13} = 5 \quad (\text{node 1})$$

$$x_{24} - x_{12} = -2 \quad (\text{node 2})$$

$$x_{34} - x_{13} = 0 \quad (\text{node 3})$$

$$-x_{24} - x_{34} = -3 \quad (\text{node 4})$$

$$2 \leq x_{12} \leq 5$$

$$0 \leq x_{13} \leq 2$$

$$1 \leq x_{24} \leq 3$$

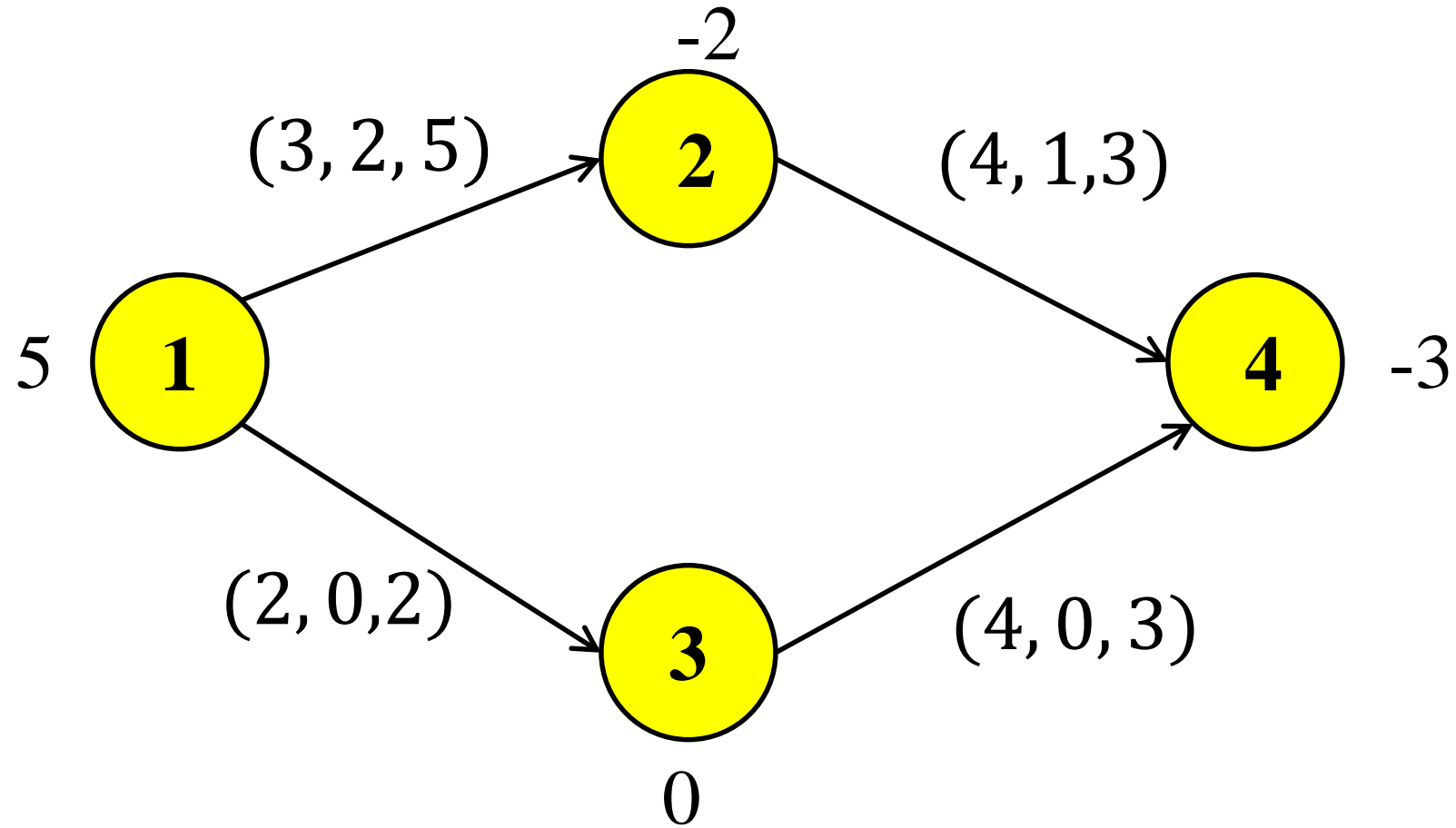
$$0 \leq x_{34} \leq 3$$



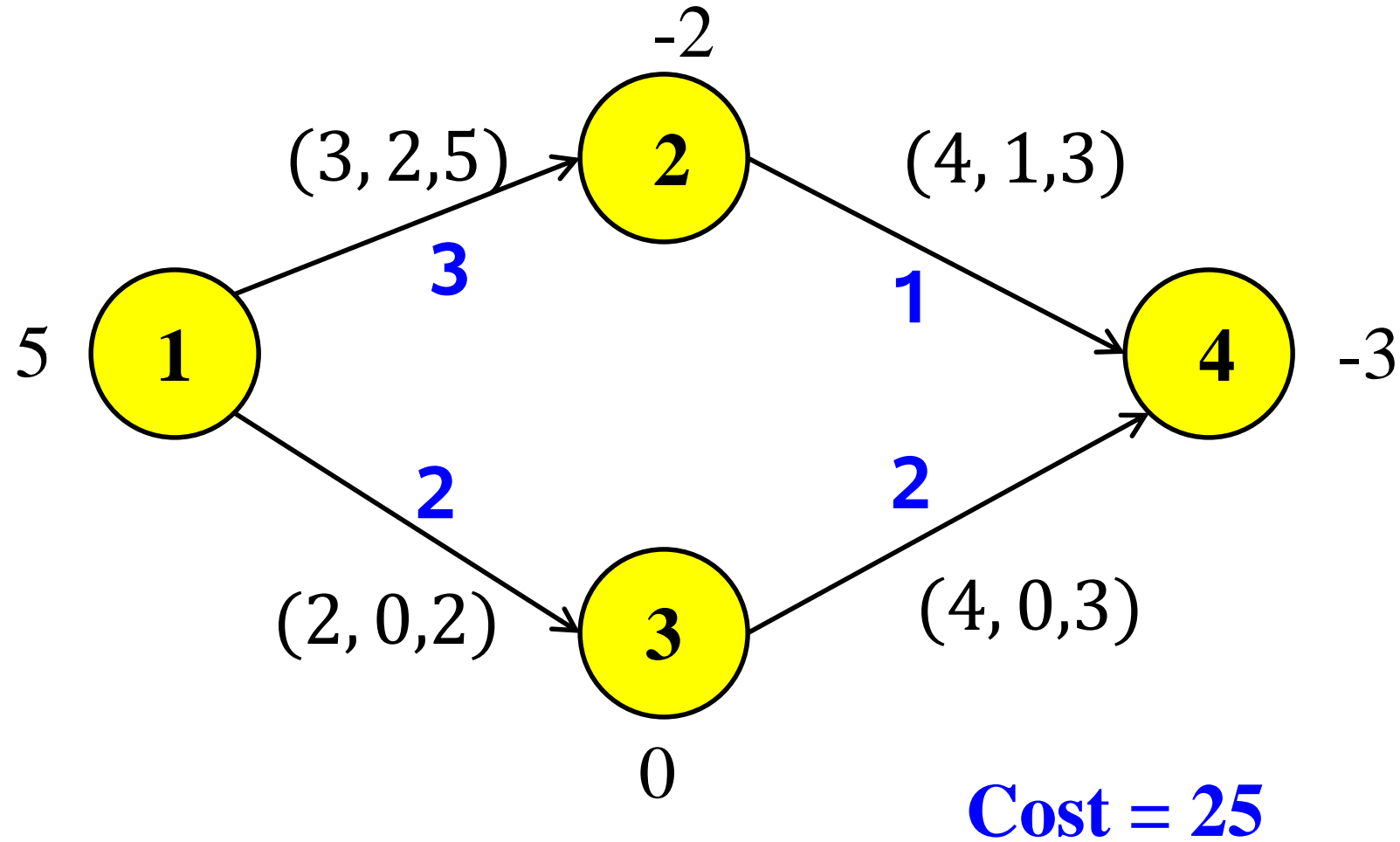
# node arc incidence matrix

		$x_{12}$	$x_{13}$	$x_{24}$	$x_{34}$
Represents arc relationships Positive = starting node Negative = ending node	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>
	<b>2</b>	<b>-1</b>	<b>0</b>	<b>1</b>	<b>0</b>
	<b>3</b>	<b>0</b>	<b>-1</b>	<b>0</b>	<b>1</b>
	<b>4</b>	<b>0</b>	<b>0</b>	<b>-1</b>	<b>-1</b>

# example feasible solution



# optimal solution for example



Since,

$$\mathbf{Ax} = \mathbf{b}$$

QUESTION: WHY  
ARE NETWORK  
MODELS  
NATURALLY  
INTEGER???

it must hold that,

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

# Cramer's rule

Use determinants to solve  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$

$$x_j = \frac{|B_j|}{|A|}$$

solution of  $j$ th variable  $x$  is from  $\det(B_j) / \det(A)$

where,

$$B_j = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1,j-1} & b_1 & a_{1,j+1} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2,j-1} & b_2 & a_{2,j+1} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{n,j-1} & b_n & a_{n,j+1} & \dots & a_{nn} \end{bmatrix}$$

i.e., take the matrix  $A$  and replace column  $j$  with the vector  $b$  to form matrix  $B_j$ .

# Cramer's rule to solve for $x_{12}$

Is  $|B_{(1,2)}|$  an integer?

$$x_{12} = \frac{\begin{vmatrix} 5 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ -3 & 0 & -1 & -1 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{vmatrix}} = \frac{|B_{(1,2)}|}{|A|}$$

Does  $|A| = \pm 1$ ?

# total unimodularity

So 1 or -1 means integer solution naturally

- A square, integer matrix is *unimodular* if its determinant is 1 or -1.
- An integer matrix  $A$  is called *totally unimodular* (TU) if every square, nonsingular submatrix of  $A$  is unimodular.

# total unimodularity

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Not TU

Not TU because the determinant value must be 1 or -1.

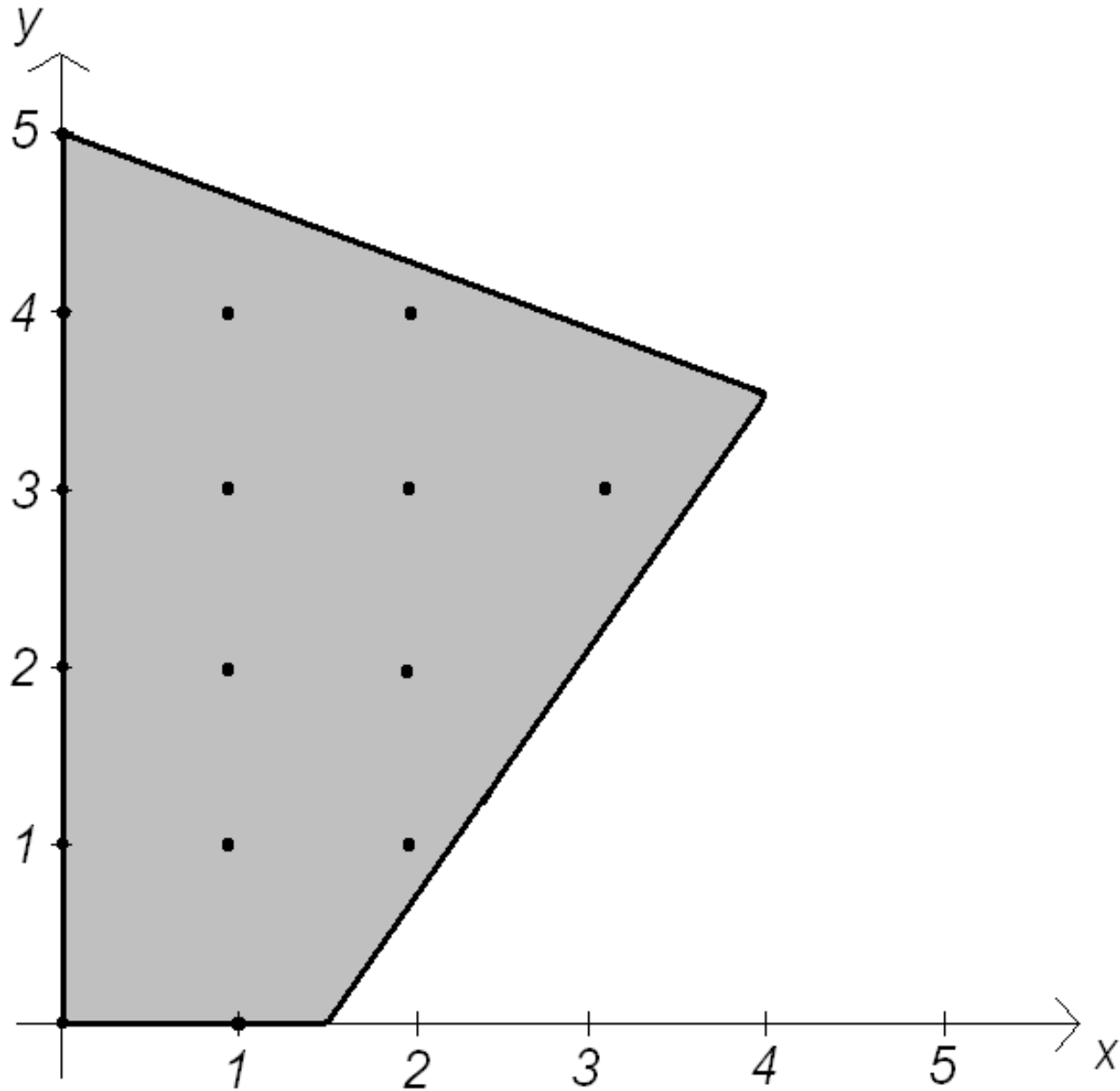
$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix}$$

TU

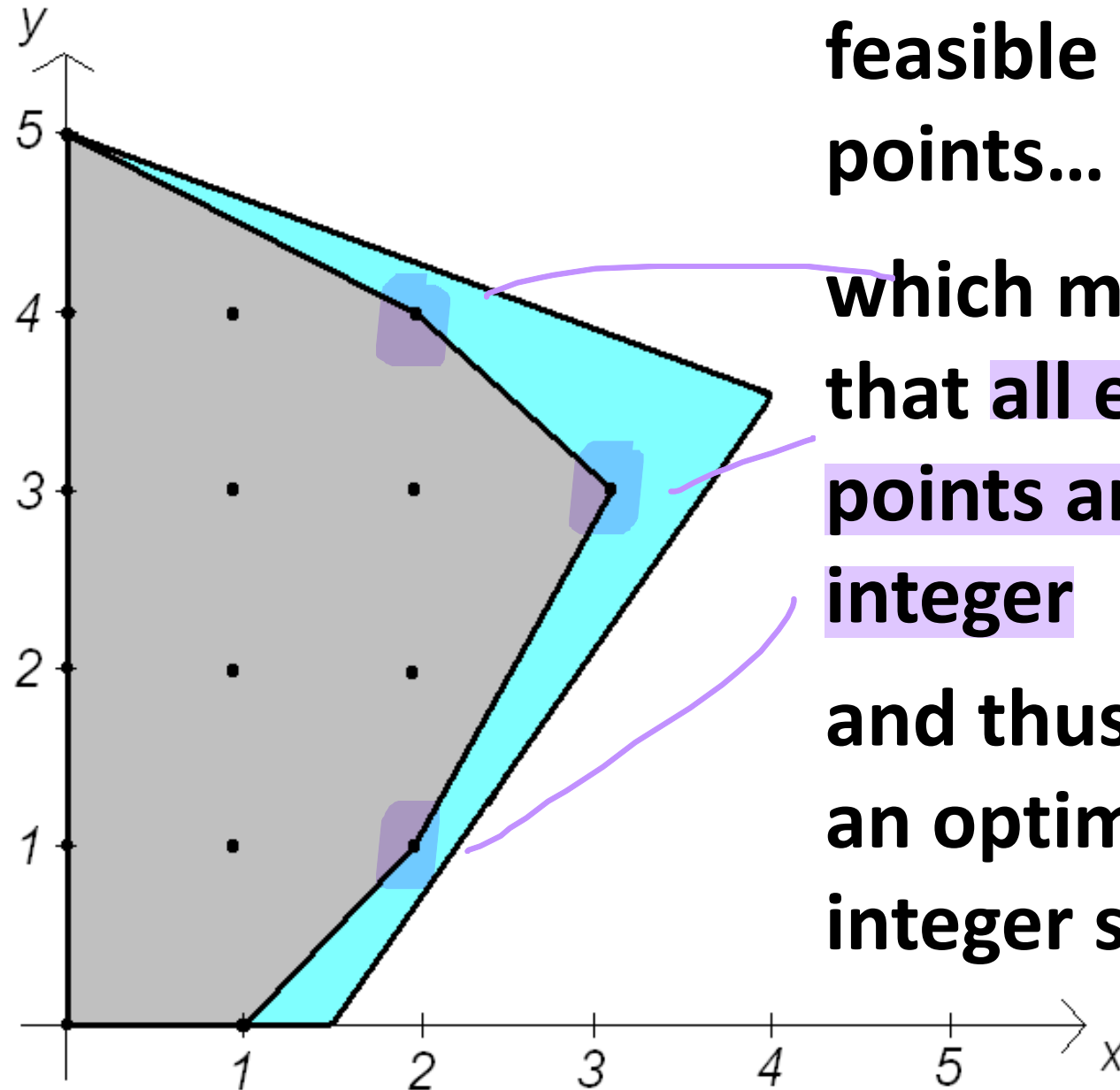


$$\begin{bmatrix}
 1 & 1 & 0 & 0 \\
 -1 & 0 & 1 & 0 \\
 0 & -1 & 0 & 1 \\
 0 & 0 & -1 & -1
 \end{bmatrix}
 \begin{vmatrix}
 1 & 0 \\
 -1 & 1
 \end{vmatrix}
 = 1$$

# total unimodularity



# total unimodularity



Convex hull of  
feasible integer  
points...

which means  
that all extreme  
points are  
integer

and thus, there is  
an optimal  
integer solution

# total unimodularity theorem

An integer matrix  $A$  is TU if:

1. all entries are -1, 0 or 1
2. at most two non-zero entries appear in any column
3. the rows of  $A$  can be partitioned into two disjoint sets such that
  - if a column has two entries of the same sign, their rows are in different sets
  - if a column has two entries of different signs, their rows are in the same set

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix}$$

**Properties (1) and (2)  
hold by definition of an  
node arc incidence matrix**

**Let M1 and M2 be a  
partition of rows.**

**In particular, let M1 = {A}  
and M2 = {}.**

**As such, property (3) holds.**

**Digression: what about the Transportation problem constraint matrix?**

**ANSWER: INT**

1	1	1	1	1	1														
						1	1	1	1	1	1								
												1	1	1	1	1	1		
																		1	1
1						1						1					1		
	1						1						1					1	
		1						1						1					1
			1						1						1				
				1						1						1			
					1						1						1		
						1						1						1	
							1						1						1

Properties (1) and (2) hold.

Let M1 = supply constraint rows.

and M2 = demand constraint rows.

As such, property (3) holds.

# using Cramer's rule to solve for $x_{12}$

Is  $|B_{(1,2)}|$  an integer?

**Yes!**

Does  $|A| = \pm 1$ ?

**Yes!**

$$x_{12} = \frac{\begin{vmatrix} 5 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ -3 & 0 & -1 & -1 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{vmatrix}} = \frac{|B_{(1,2)}|}{|A|}$$

All MCNFP have TU constraint matrices. All MCNFP feasible solutions will be integer!



# MIN-COST NETWORK FLOW PROBLEM (MCNFP) CONTINUED

# Network Flow Problems

- There are a variety of “network flow” problems
- The MCNFP is one, common, and useful type
- Many problems (in general) can be represented as network flow problems and the MCNFP in particular
- The “flow” might be actual “things”
  - e.g. people, commodities, water, electricity in a physical system (transportation, pipelines, powergrid, etc.)
- Or it might represent other more abstract concepts
  - e.g., information, financial transactions, or logical decisions

# MCNFP Examples

(Minimum Cost Flow Applications) (Minimum Cost Flow Applications:)

- Transportation problem (again)
- Multi-period planning (again)
- Shortest path problem
- Maximum flow problem

# MCNFP formulation

(Minimum Cost Flow Applications):

USES THE SAME FORMULATION FOR EACH

$$\text{minimize} \quad \sum_{(i,j) \in A} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{j: (i,j) \in A} x_{ij} - \sum_{j: (j,i) \in A} x_{ji} = b_i \quad \forall i \in N$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in A$$

# File: MCNFP.txt

# AMPL model for the Minimum Cost Network Flow Problem

# SETS AND PARAMETERS-----

set **NODES**; # nodes in the network

set **ARCS** within {NODES, NODES}; # arcs in the network

param **b** {NODES} default 0; # supply/demand for node i

param **c** {ARCS} default 0; # cost of one unit flow on (i,j)

param **l** {ARCS} default 0; # lower bound on flow on (i,j)

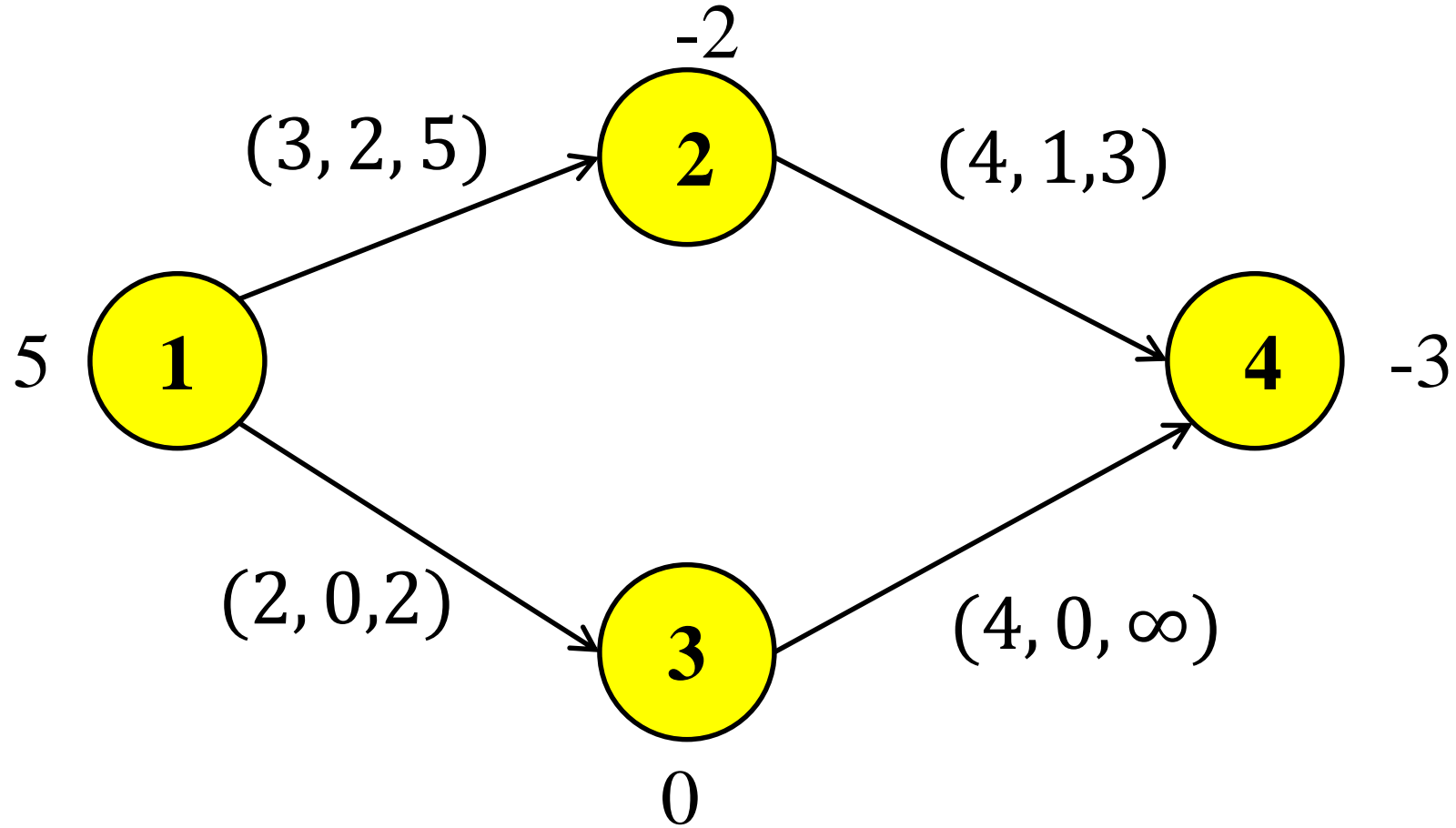
param **u** {ARCS} default Infinity; # upper bound on (i,j)

```
#DECISION VARIABLES -----  
var x {ARCS};           # flow on arc (i,j)  
  
#OBJECTIVE -----  
#minimize arc flow cost  
minimize cost: sum{(i,j) in ARCS} c[i,j] * x[i,j];  
  
#CONSTRAINTS -----  
subject to flow_balance {i in NODES}:  
    sum{j in NODES: (i,j) in ARCS} x[i,j]  
    - sum{j in NODES: (j,i) in ARCS} x[j,i] = b[i];  
  
subject to capacity {(i,j) in ARCS}:  
    l[i,j] <= x[i,j] <= u[i,j];
```

Constraints can vary, but it is  
mainly visualizing and  
getting the data

# example network

Visualize, THEN convert to a  
data file



```
#MCNFP Problem - data file for MCNFP.txt
#note: default arc costs and lower bounds are 0
#      default arc upper bounds are infinity
#      default node requirements are 0
```

```
set NODES := 1 2 3 4 ;
```

```
set ARCS := (1, 2) (1,3) (2,4) (3,4) ;
```

```
param b :=
      1  5
      2 -2
      3  0
      4 -3;
```

```
param:
      c  l  u :=
      [1,2] 3  2  5
      [1,3] 2  .  2
      [2,4] 4  1  3
      [3,4] 4  .  .      ;
```



# MIN-COST NETWORK FLOW PROBLEM EXAMPLES: Transportation Problem (again!)

A mobile-home manufacturer channels its units through distribution centers located in Elkhart, Ind., Albany, N.Y., Camden, N.J., and Petersburg, Va.

The distribution centers have in inventory 30, 75, 60, and 35 mobile homes, respectively.

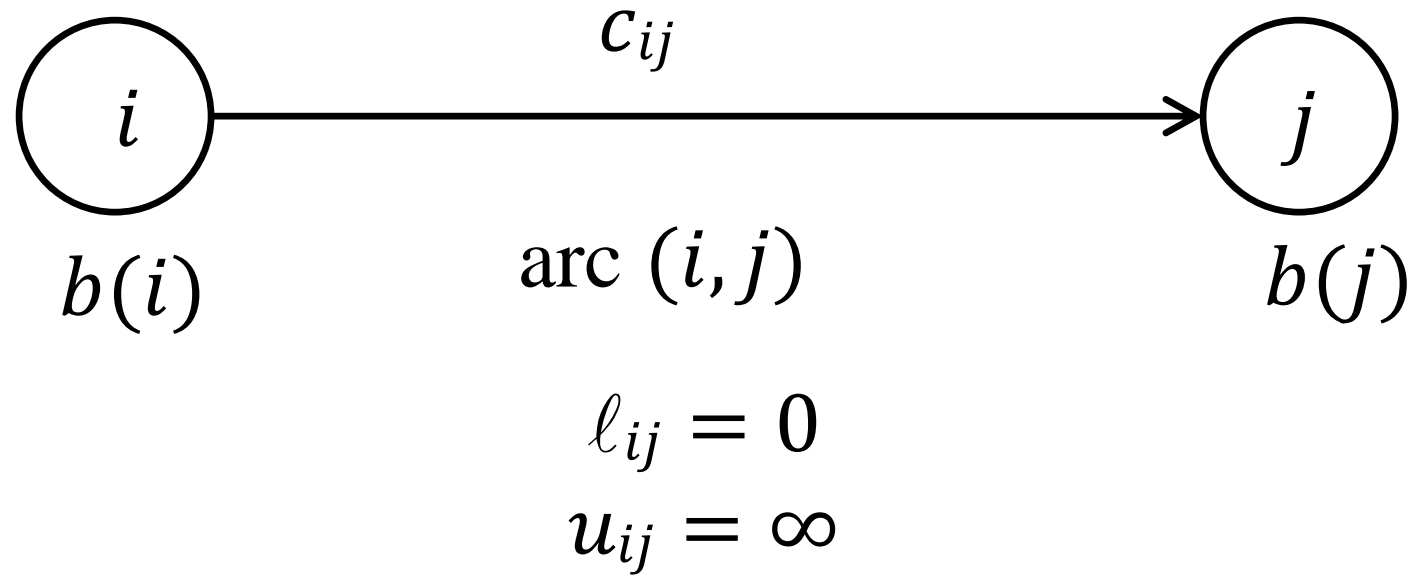
Demand at the dealerships:

Dealer	A	B	C	D	E	F
Units	25	40	15	25	50	45

Transportation costs (dollars per unit) between each distribution center and the dealerships are:

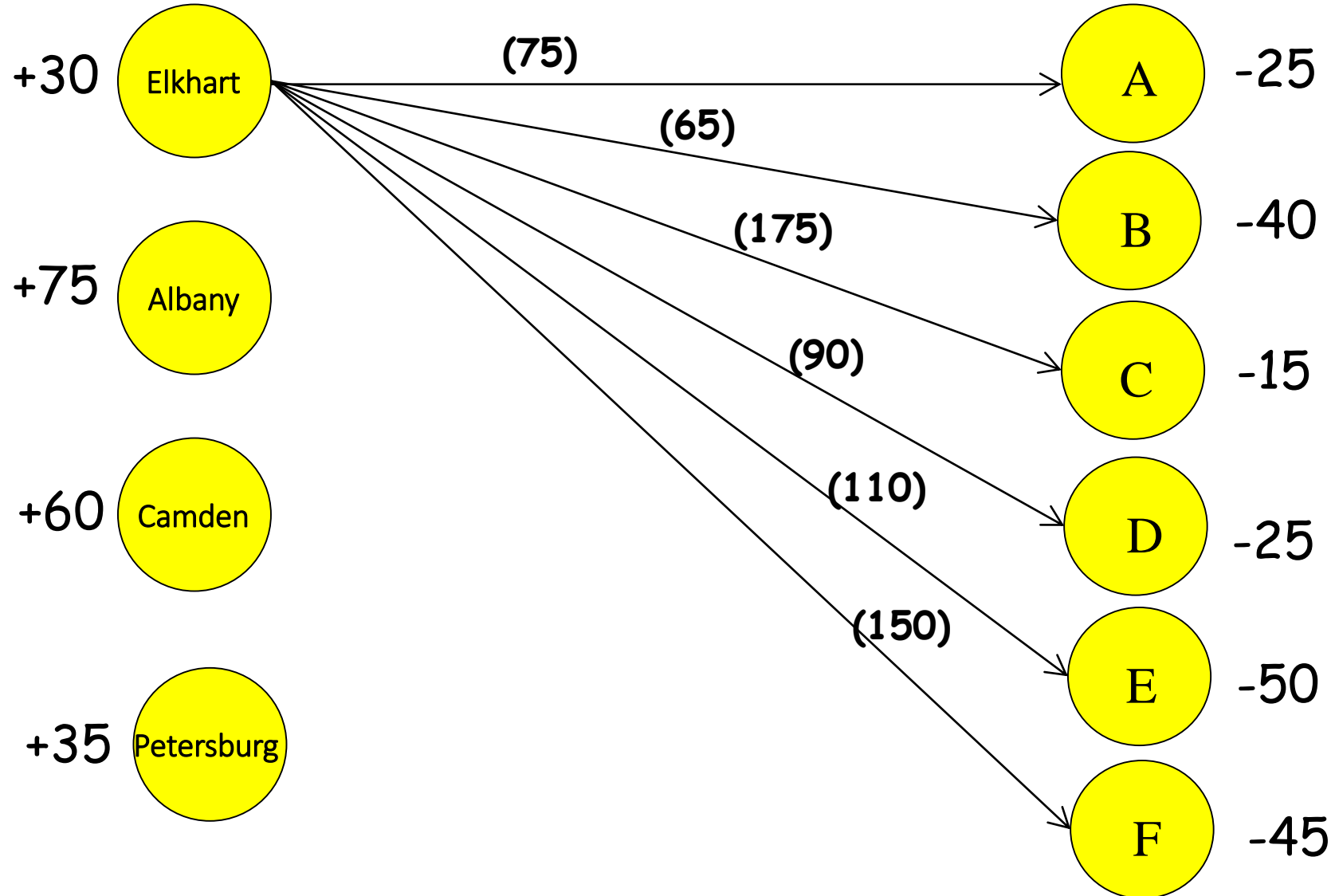
	A	B	C	D	E	F
Elkhart	75	65	175	90	110	150
Albany	90	30	45	50	105	130
Camden	40	55	35	80	70	75
Petersburg	95	150	100	115	55	55

# Transportation Problems (MCNFP Formulation)



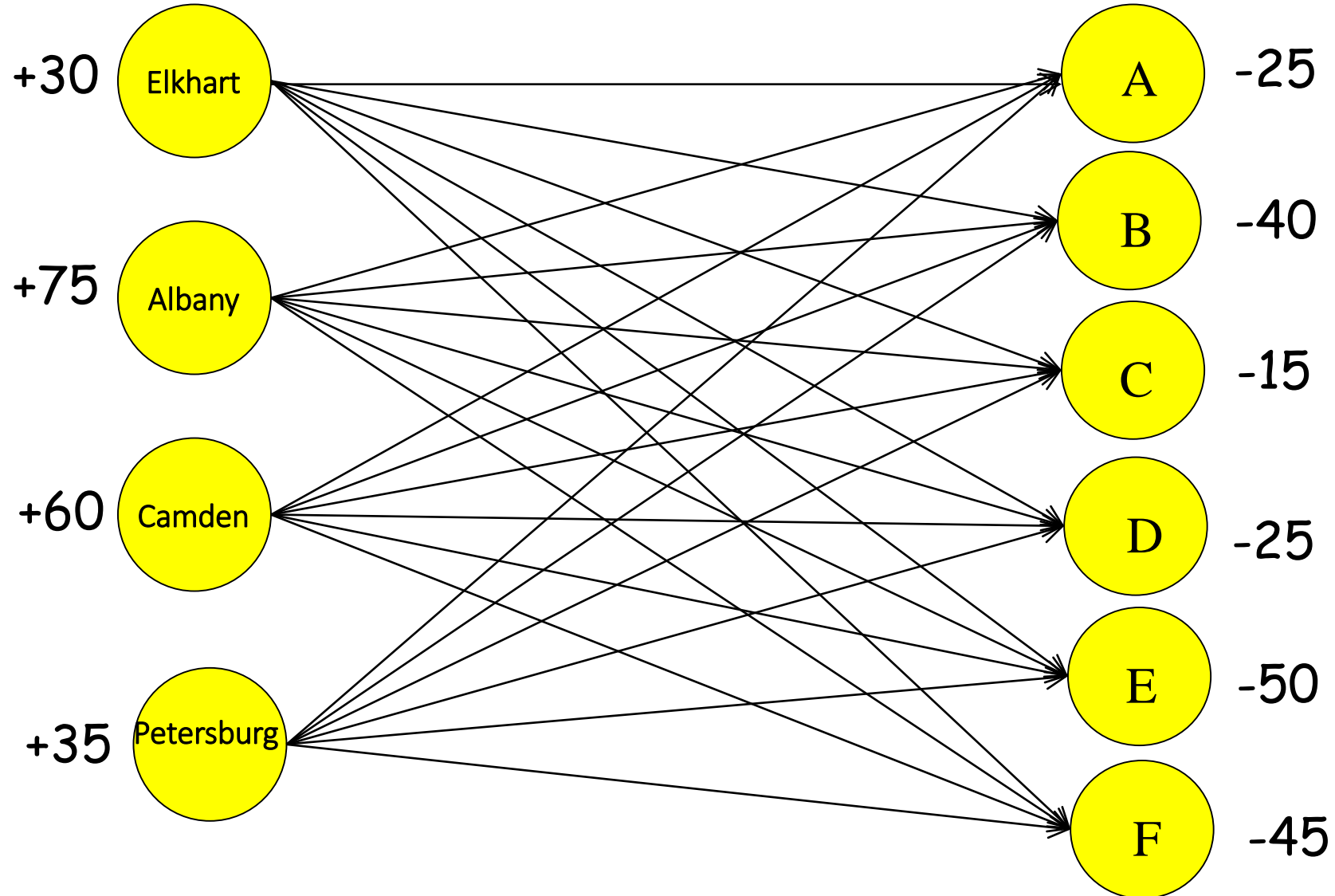
## Distribution Center (Supply Nodes)

## Dealers (Demand Nodes)



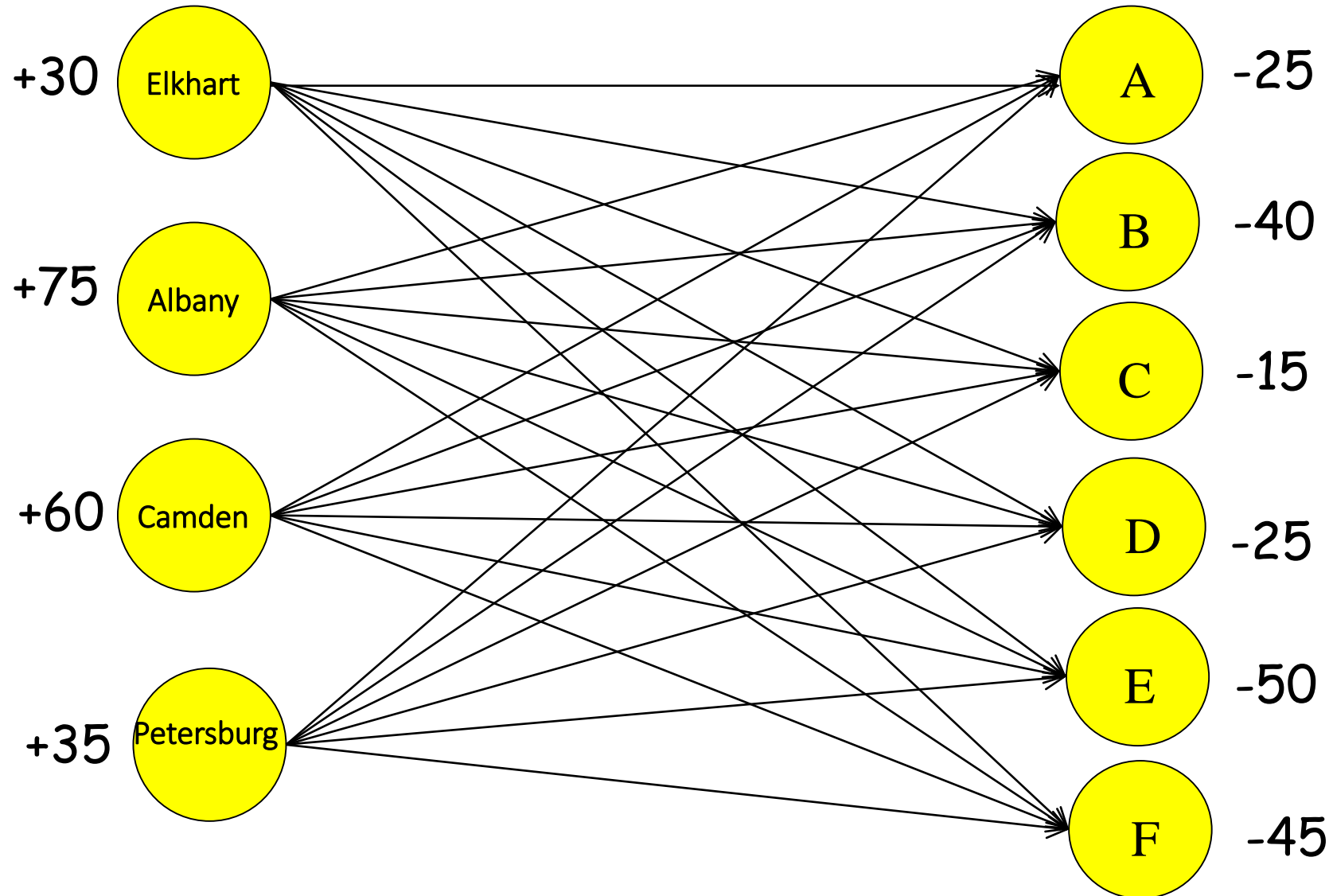
## Distribution Center (Supply Nodes)

## Dealers (Demand Nodes)



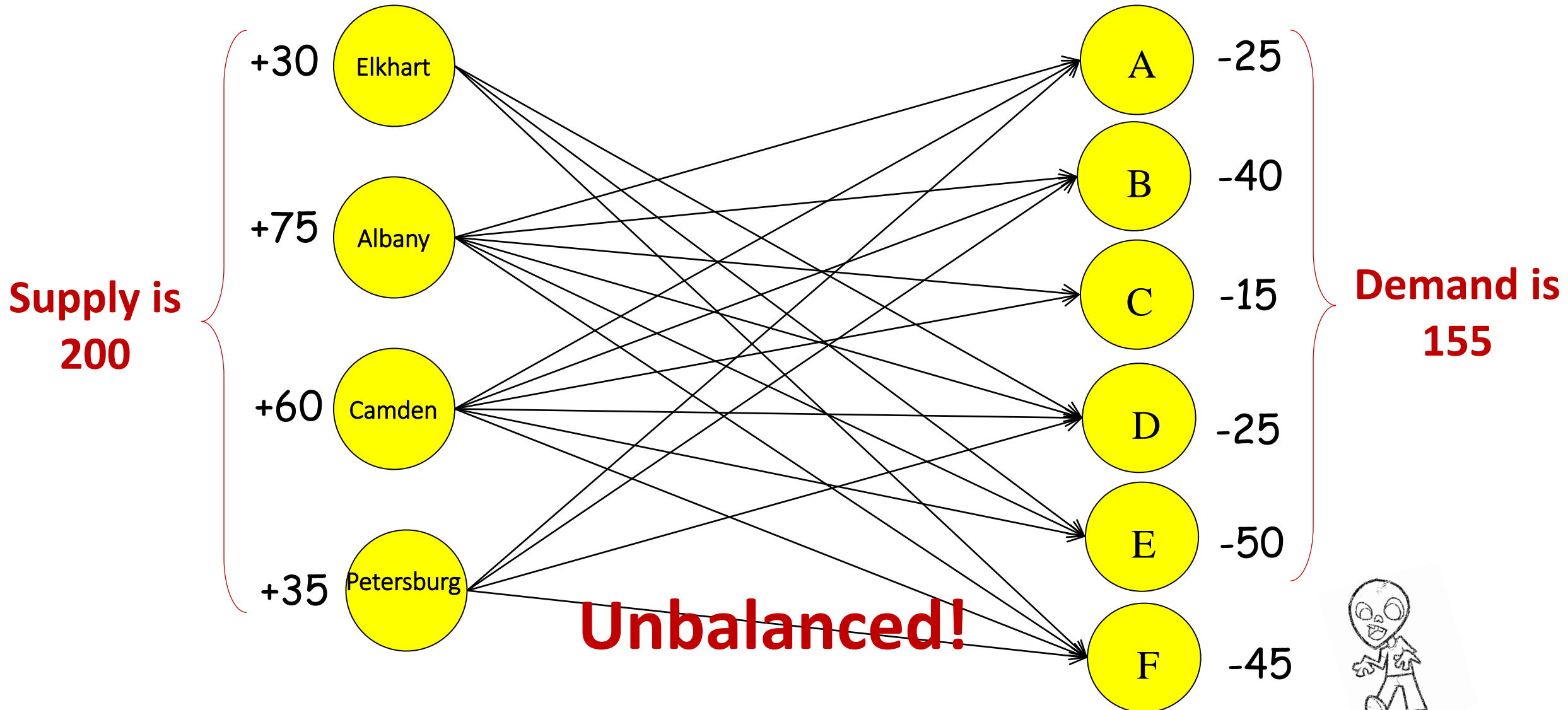
## Distribution Center (Supply Nodes)

## Dealers (Demand Nodes)



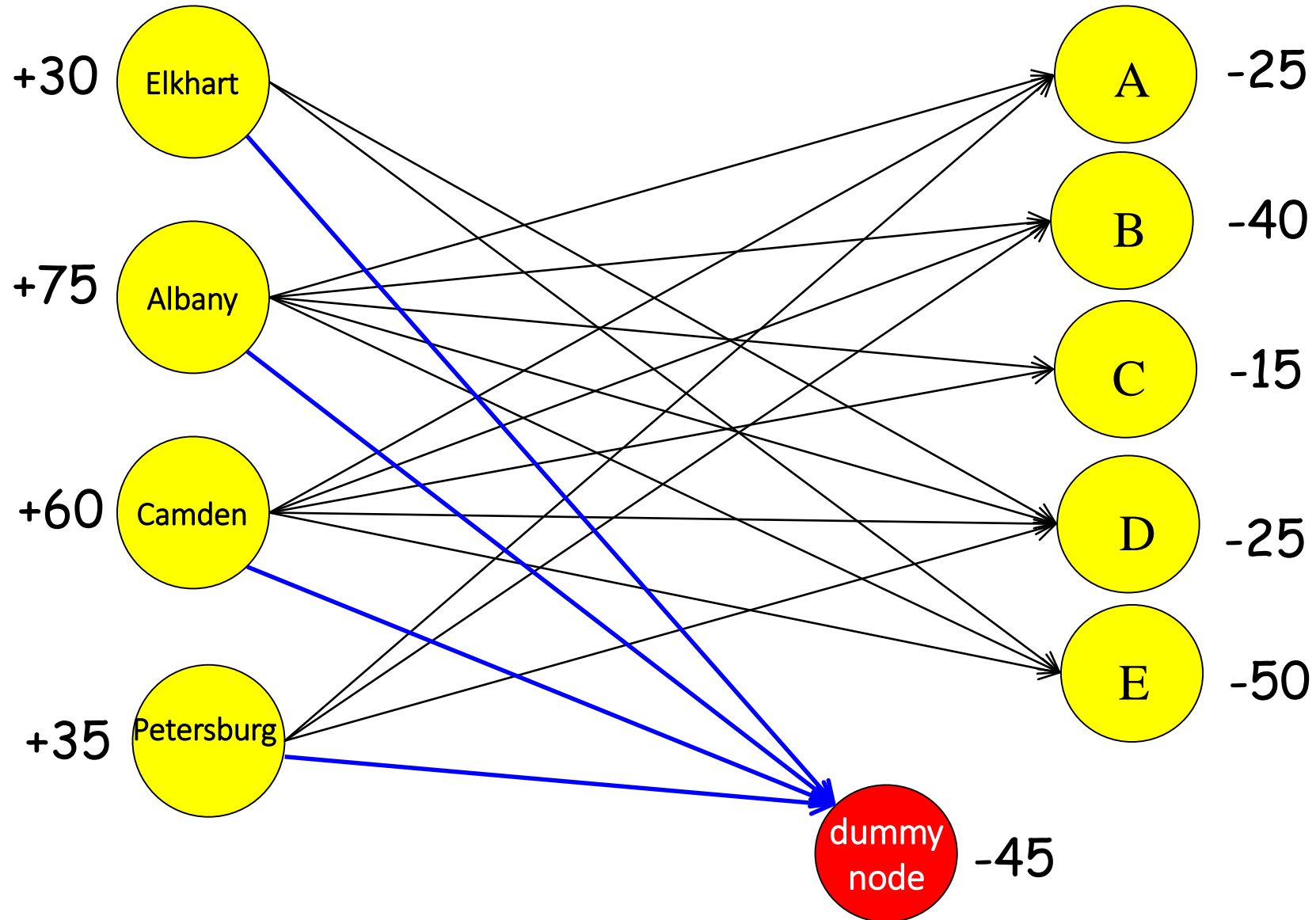
## Distribution Center (Supply Nodes)

## Dealers (Demand Nodes)



## Distribution Center (Supply Nodes)

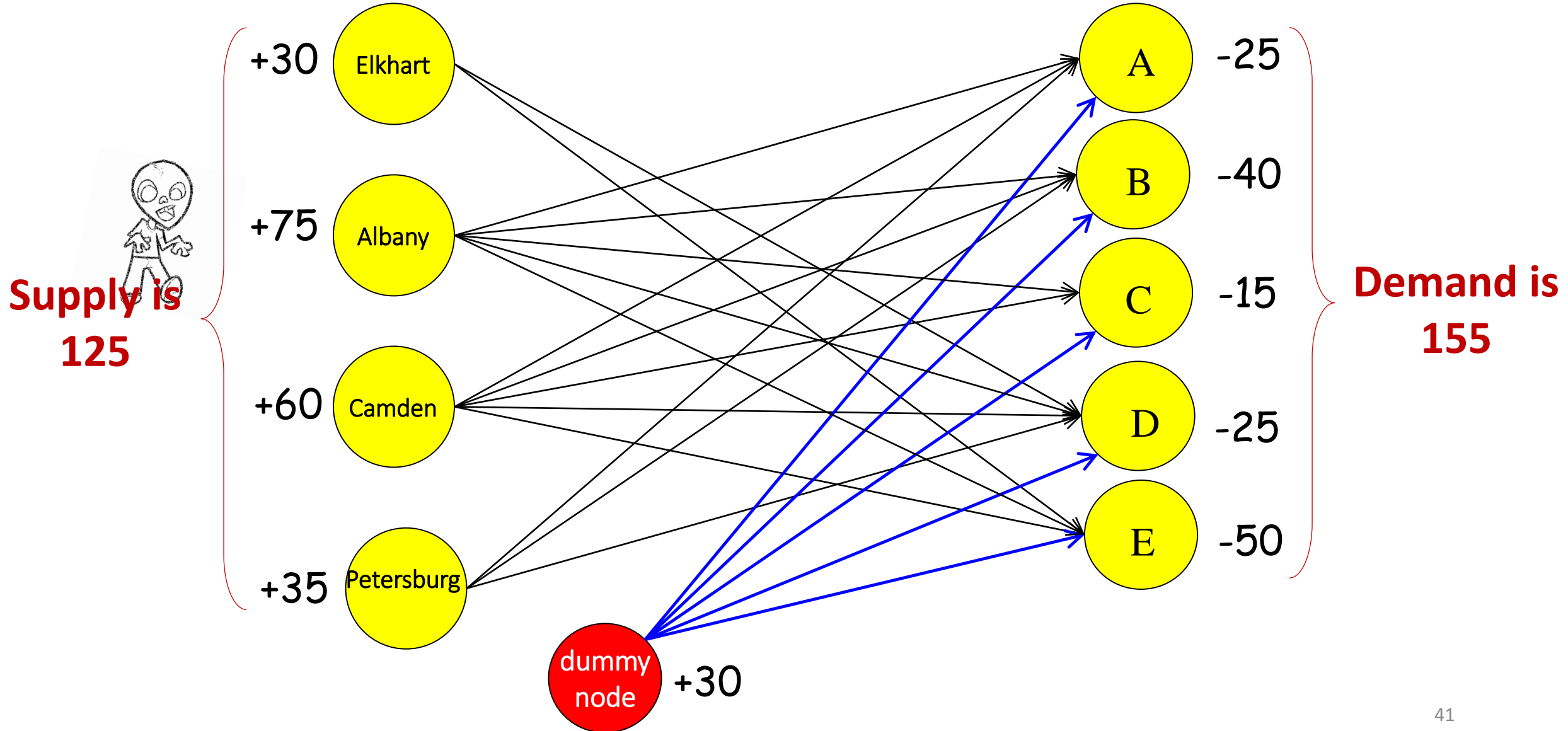
## Dealers (Demand Nodes)





## Distribution Center (Supply Nodes)

## Dealers (Demand Nodes)



# MIN-COST NETWORK FLOW

## PROBLEM EXAMPLES:

### Multi-period Planning (again!)

NSC produces steel.  
*Demand* for the next four months is given:

Jan	Feb	Mar	Apr
2400	2200	2700	2500

NSC can meet demand by producing steel, by drawing from its inventory or by a combination. January inventory is 1000 tons.

*Production costs* vary from month to month as shown:

Jan	Feb	Mar	Apr
7400	7500	7600	7800

Inventory holding costs: \$120

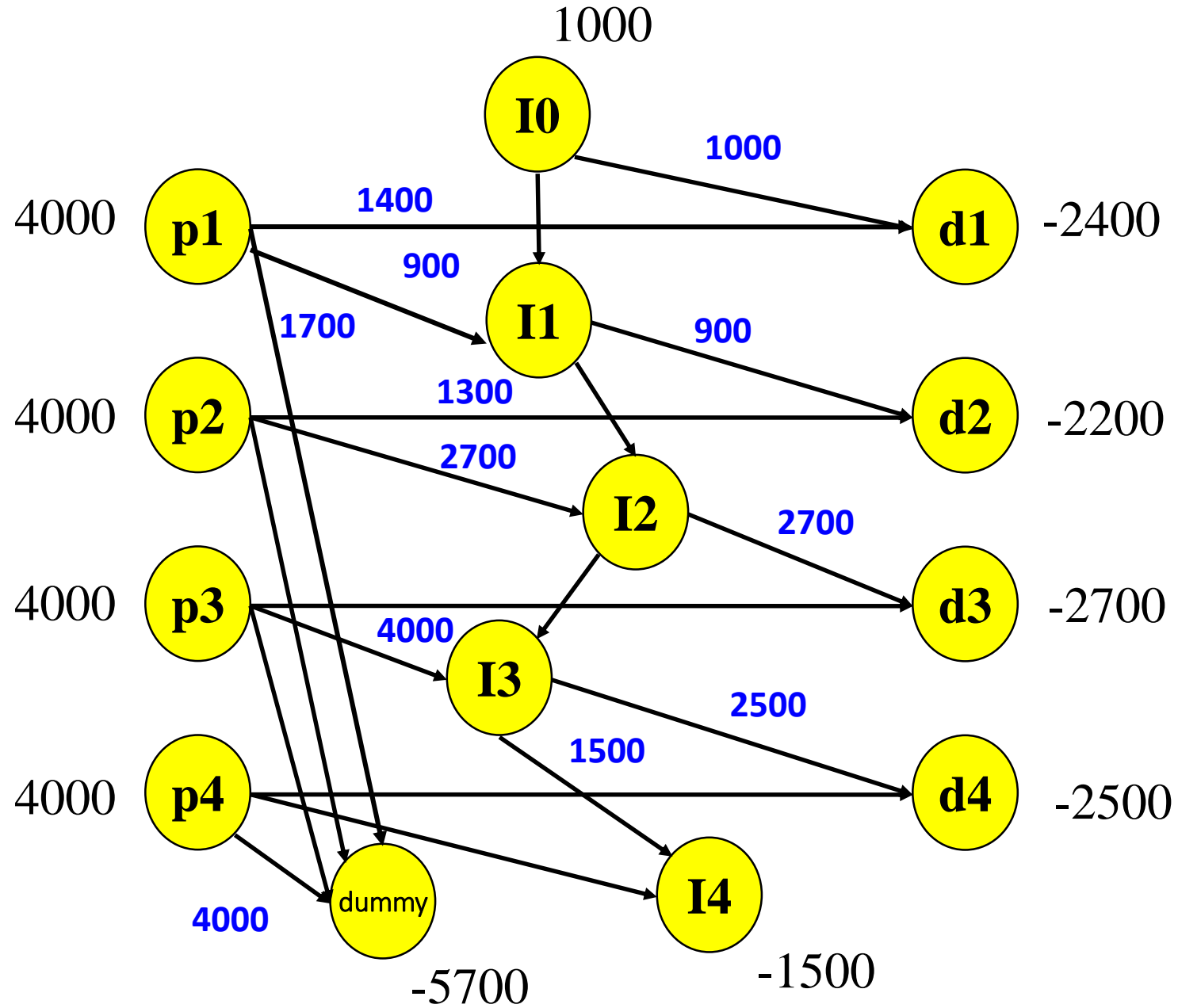
Monthly production capacity is 4000 tons.

Operations requires ending inventory for April: 1500 tons.

# **Formulate the NSC multi-period production planning problem as a network flow problem.**

1. Draw the network.
2. Denote the node requirement values.
3. Clarify the arc flow upper and lower bounds and costs on arcs as needed.

# MCNFP formulation of NSC problem



# arc parameters

- All arcs have  $\ell_{ij} = 0$  and  $u_{ij} = \infty$
- Arcs  $(p_i, \text{dummy})$  have cost 0.
- Arcs  $(I_i, d_{i+1})$  and  $(I_i, I_{i+1})$  have cost 120.

Production to meet demand

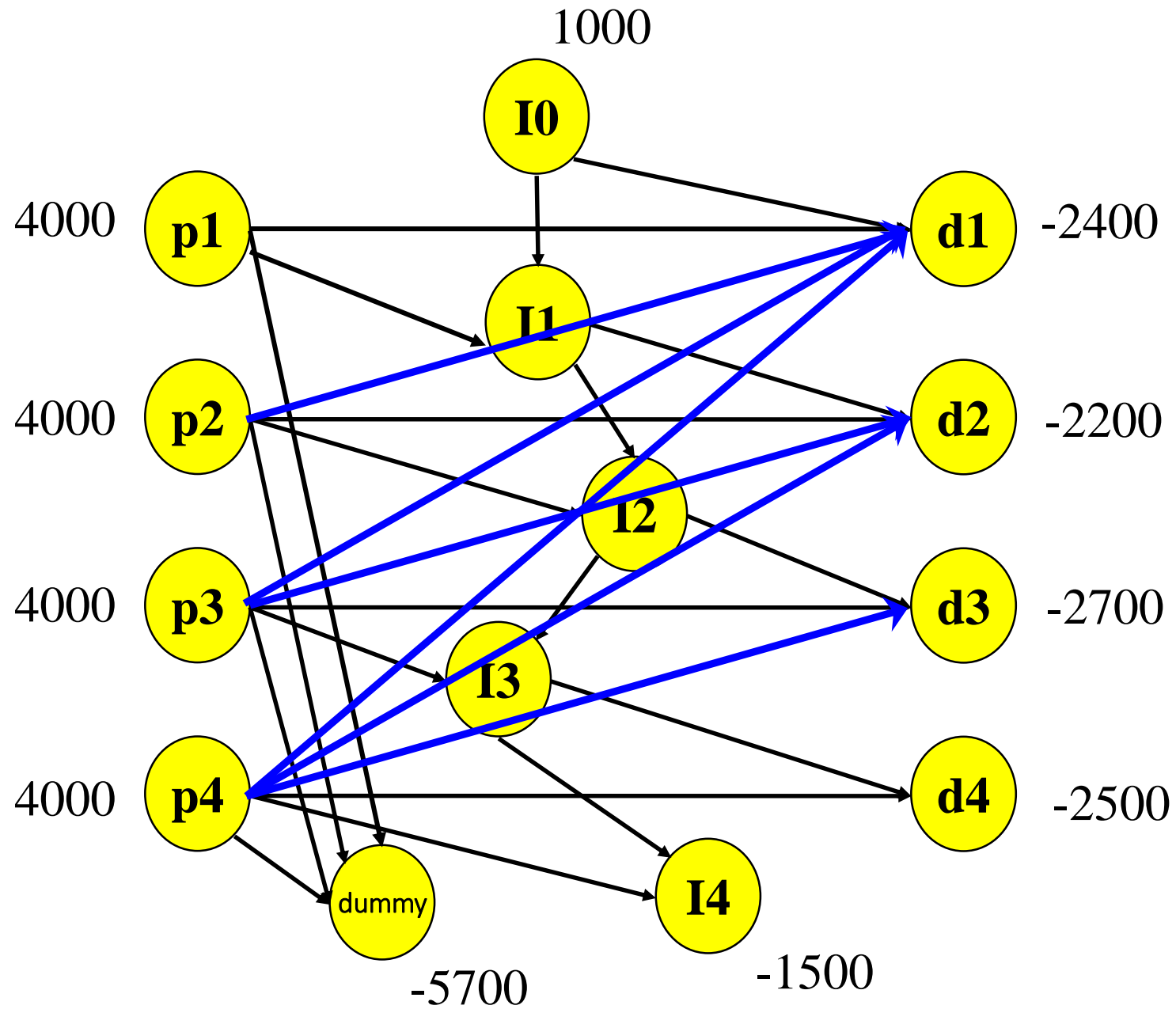
Arc	Cost
(p1,d1)	7400
(p2,d2)	7500
(p3,d3)	7600
(p4,d4)	7500

Production for extra-inventory

Arc	Cost
(p1,i1)	7400
(p2,i2)	7500
(p3,i3)	7600
(p4,i4)	7500

What if you want to  
allow backorders?

# backorders





# additional arc parameters with back orders

- All arcs have  $\ell_{ij} = 0$  and  $u_{ij} = \infty$

Production to meet demand (backorder of \$200/ton/month cost included)

Arc	Cost
(p2,d1)	7700
(p3,d1)	8000
(p3,d2)	7800
(p4,d1)	8400
(p4,d2)	8200
(p4,d3)	8000

# MIN-COST NETWORK FLOW

## PROBLEM EXAMPLES:

### Shortest Path Problem

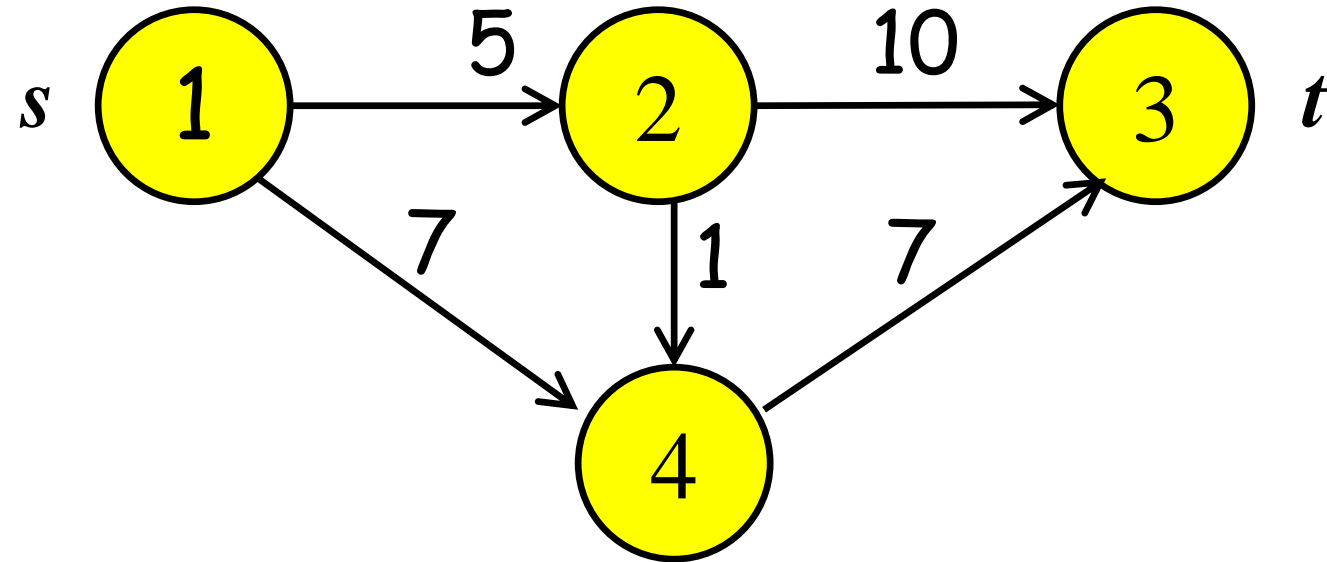
# Shortest Path Problem (MCNFP Formulation)

- Defined on a network with two special nodes:  $s$  and  $t$
- A *path* from  $s$  to  $t$  is an alternating sequence of distinct nodes and distinct arcs starting at  $s$  and ending at  $t$ :  
 $s, (s, n_1), n_1, (n_1, n_2), \dots, (n_i, n_j), n_j, (n_j, t), t$ 
  - There are *directed* and *undirected* paths
  - A network is *connected* if an undirected path exists between any pairs of nodes
  - If  $s = t$ , the path is called a *cycle*
  - A connected network without cycles is called a *tree*

# Shortest Path Problem (MCNFP Formulation)

- Defined on a network with two special nodes:  $s$  and  $t$
- A *path* from  $s$  to  $t$  is an alternating sequence of distinct nodes and distinct arcs starting at  $s$  and ending at  $t$ :  
 $s, (s, n_1), n_1, (n_1, n_2), \dots, (n_i, n_j), n_j, (n_j, t), t$
- Find a minimum-cost directed path from  $s$  to  $t$

# shortest path example



1,(1,2),2,(2,3),3

Length = 15

1,(1,2),2,(2,4),4,(4,3)

*Length = 13*

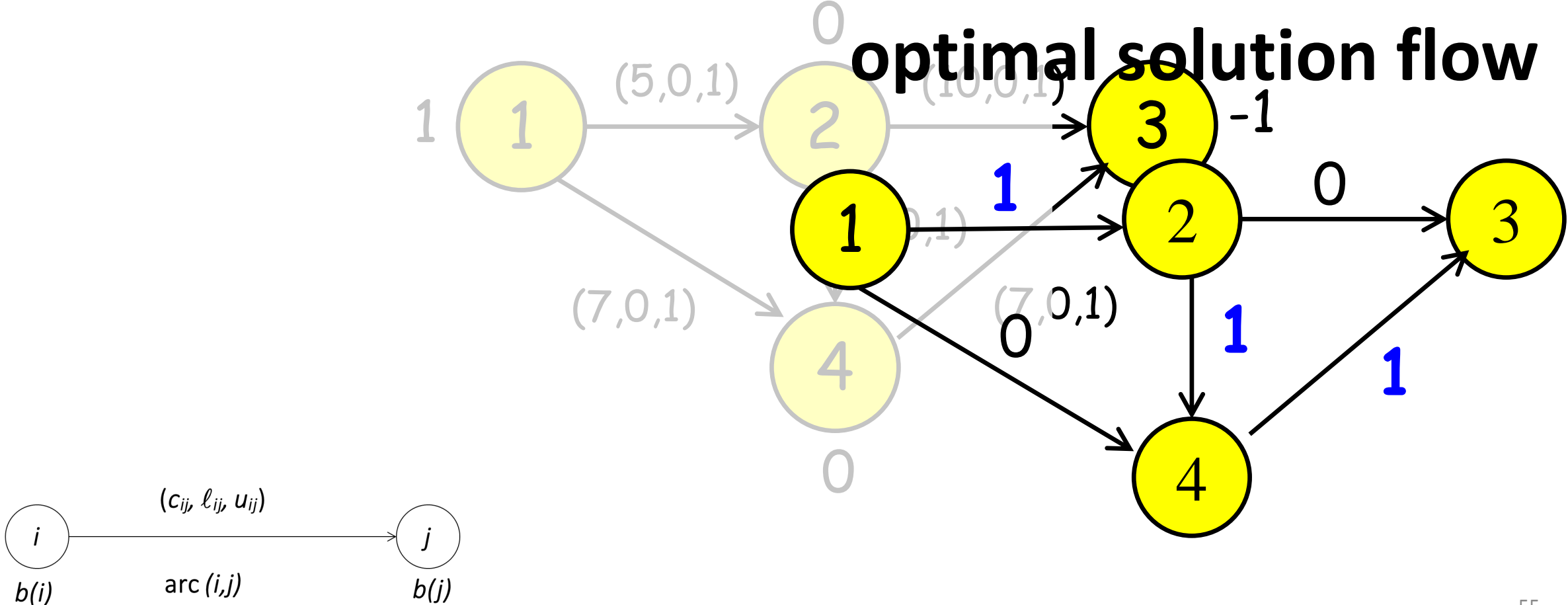
1,(1,4),4,(4,3),3

Length = 14

# MCNFP formulation of shortest path problem

- *Source* node  $s$  has a supply of 1
- *Terminal* node  $t$  has a demand of 1
- All other nodes are transshipment nodes
- Each arc has capacity 1
- Tracing the unit of flow from  $s$  to  $t$  gives a path from  $s$  to  $t$

# shortest path as MCNFP



# shortest path example

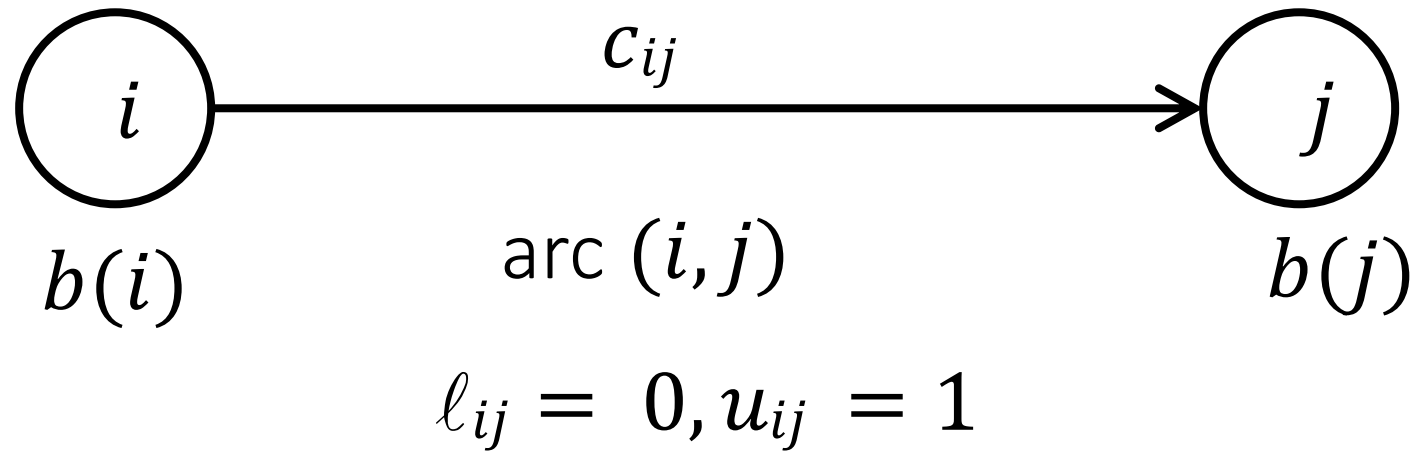
In a rural area of Texas, there are six farms connected by small roads. The distances in miles between the farms are given in the table.

What is the minimum distance to get from Farm 1 to Farm 6?

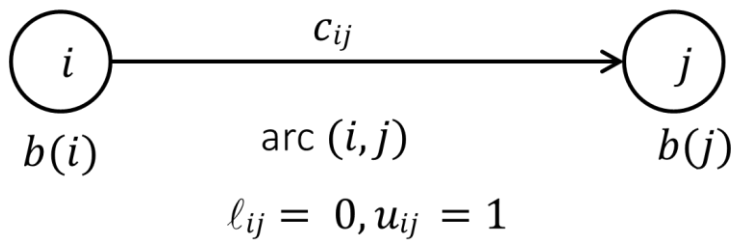
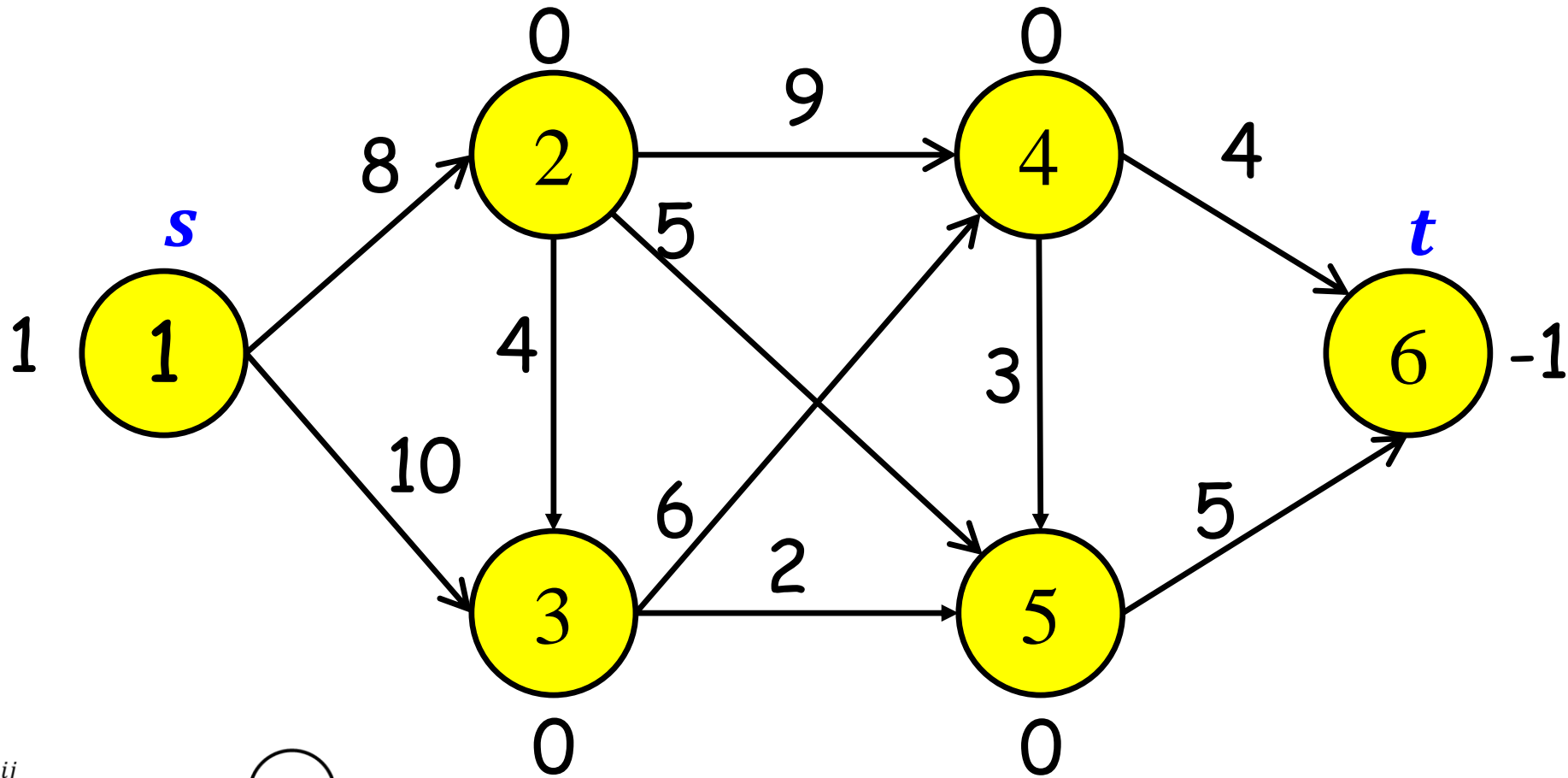
From Farm	To Farm	Distance
1	2	8
1	3	10
2	3	4
2	4	9
2	5	5
3	4	6
3	5	2
4	5	3
4	6	6
5	6	5



# graphical network flow formulation



# formulation as shortest path





# matrix representation

node requirements:

+1 for node s;

-1 for node t

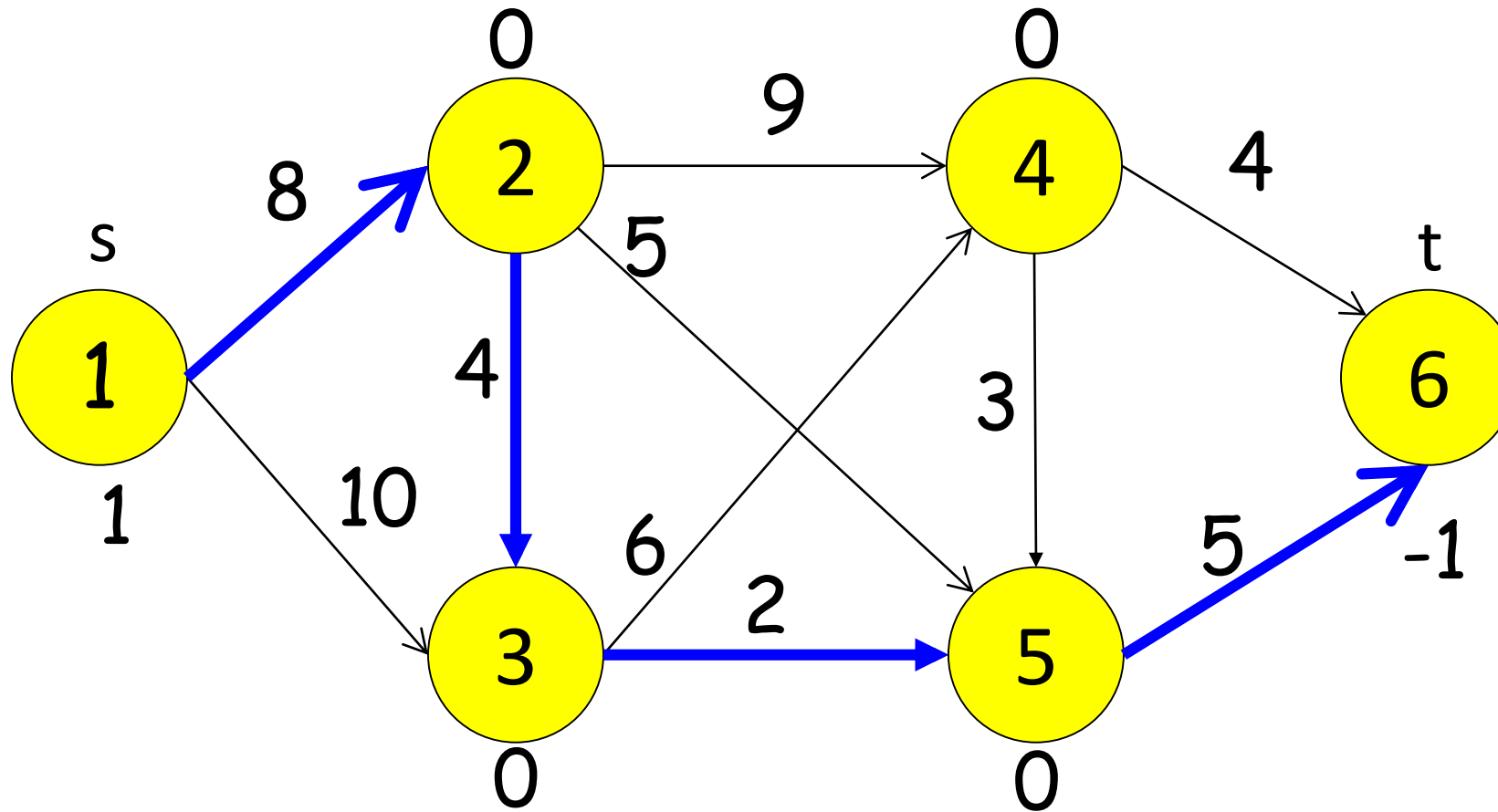
one variable per arc: denotes the “flow” on an edge


$$A = \begin{bmatrix} x_{12} & x_{13} & x_{23} & x_{24} & x_{25} & x_{34} & x_{35} & x_{45} & x_{46} & x_{56} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$


node-arc incidence matrix stipulates the “flow-balance constraints”

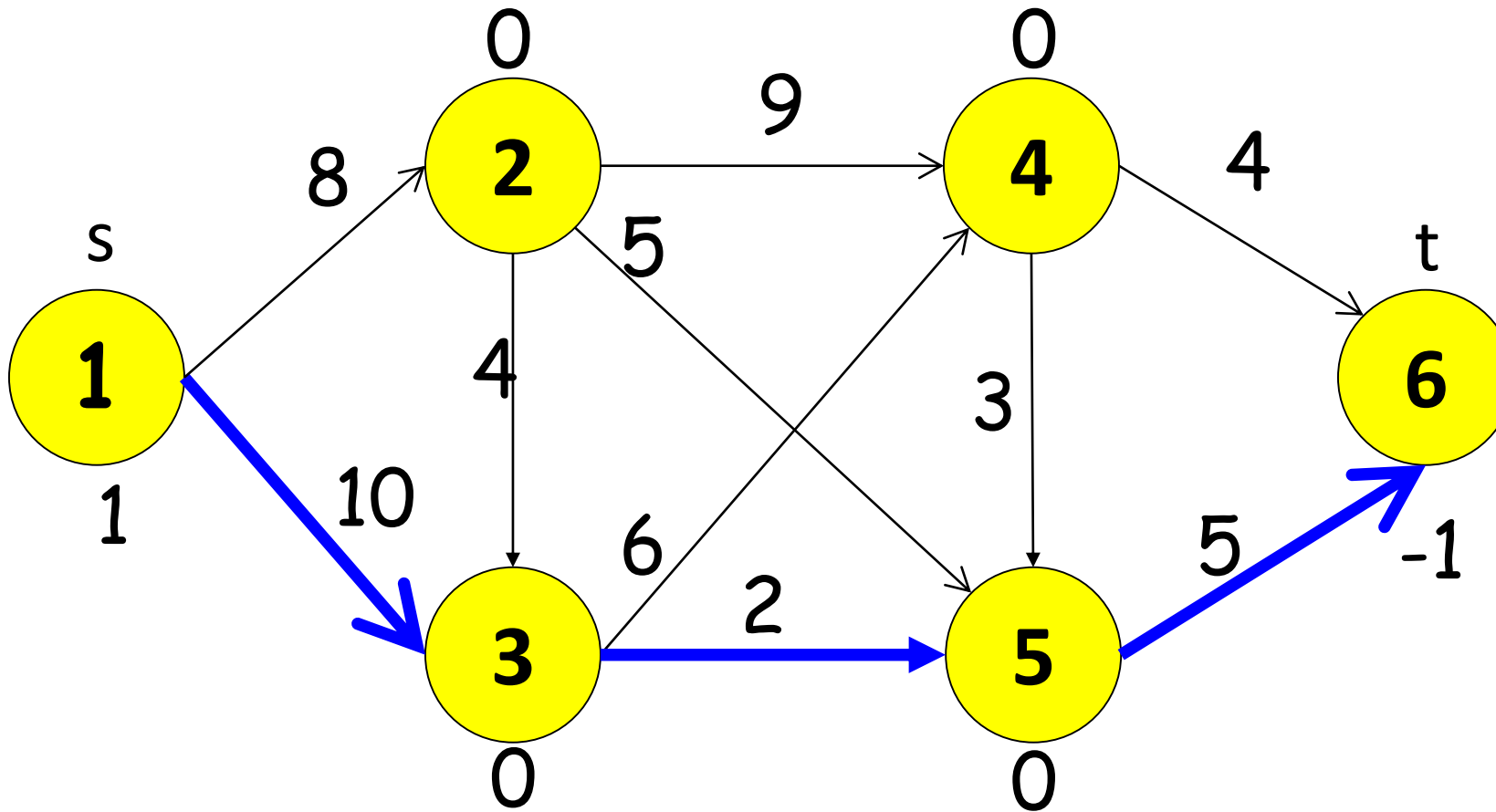
- every column relates to an arc
- every row relates to a node

# “greedy” solution



$x_{13} = x_{23} = x_{35} = x_{56} = 1$ ,  $x_{ij} = 0$  for all other arcs.  
Objective function value = 19.

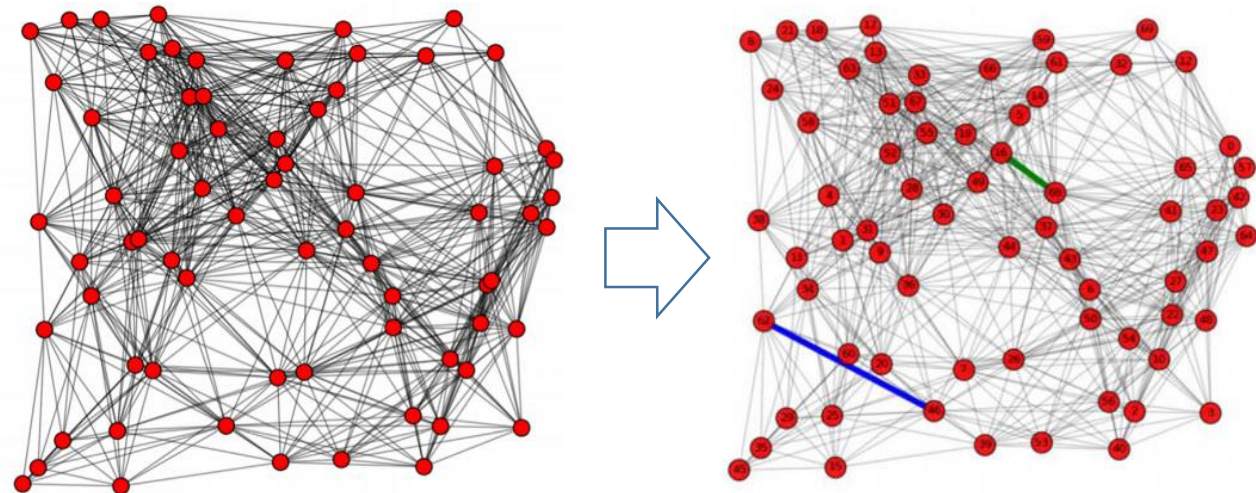
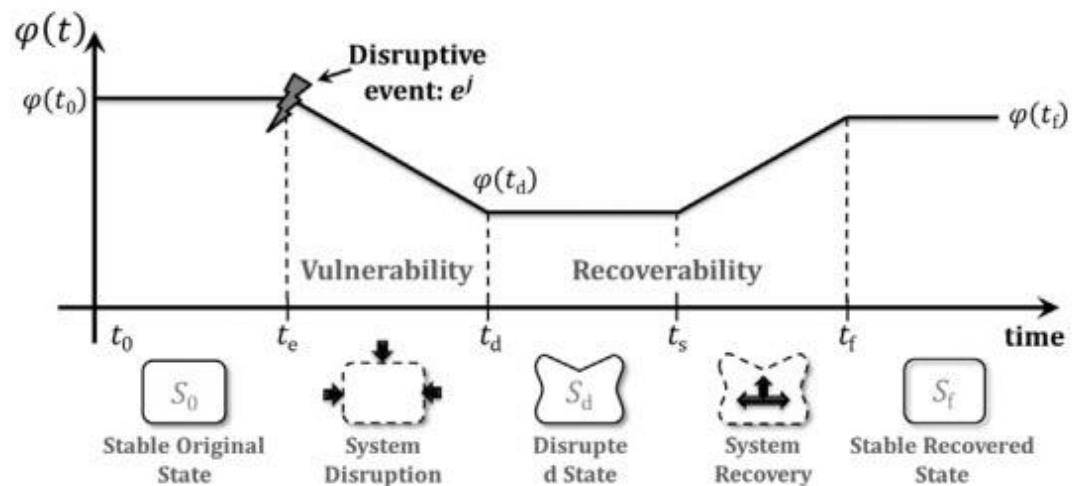
# shortest path: optimal solution



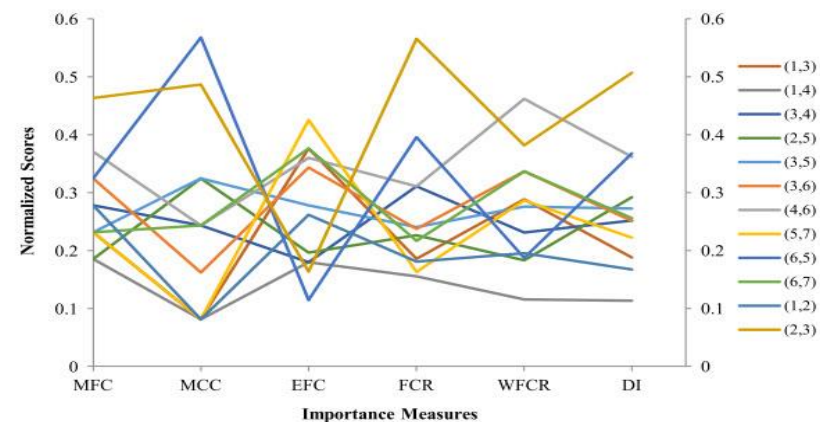
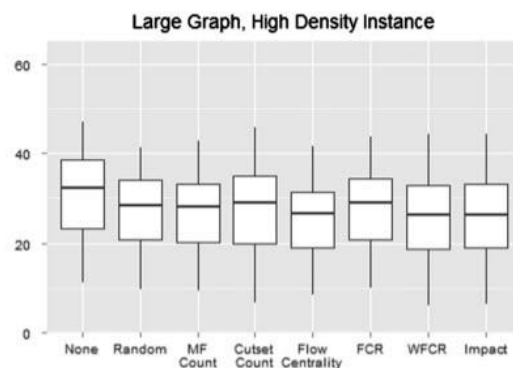
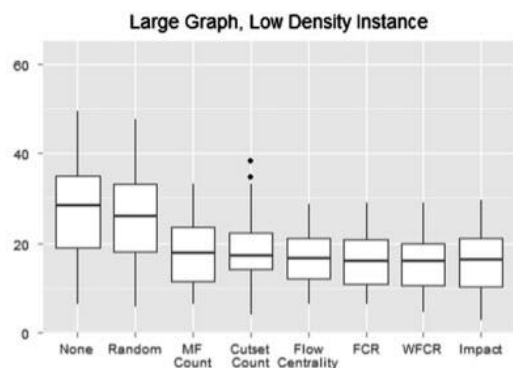
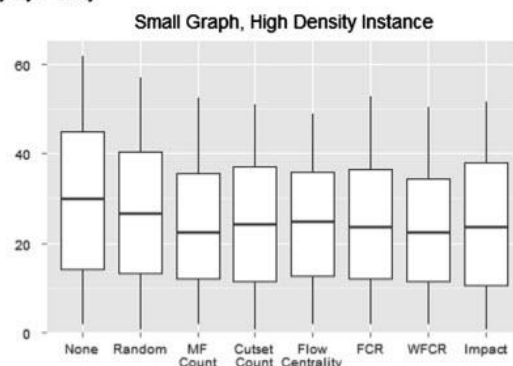
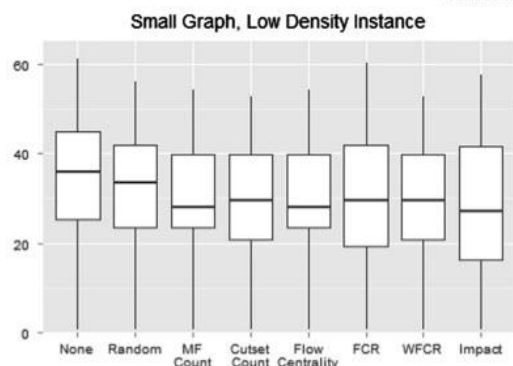
$x_{13} = x_{35} = x_{56} = 1$ ,  $x_{ij} = 0$  for all other arcs.  
Objective function value = 17.

# MIN-COST NETWORK FLOW PROBLEM EXAMPLES:

## Maximum Flow Problem



Vulnerability by Policy



Policy	SGLD		SGHD		LGLD		LGHD	
	Mean	Max	Mean	Max	Mean	Max	Mean	Max
None	34.6%	61.2%	30.0%	61.7%	27.6%	49.5%	31.1%	46.9%
Random	32.4%	55.9%	27.2%	56.8%	26.0%	47.6%	27.7%	<b>41.4%</b>
MF count	29.6%	54.3%	24.3%	52.4%	17.7%	33.2%	27.5%	42.7%
Cutset count	<b>28.9%</b>	<b>52.7%</b>	24.3%	51.1%	18.1%	38.1%	27.9%	45.8%
Flow centrality	29.6%	54.3%	24.8%	<b>48.7%</b>	16.8%	<b>28.6%</b>	<b>26.0%</b>	<b>41.4%</b>
FCR	31.3%	60.2%	24.8%	52.7%	15.9%	28.8%	28.4%	43.8%
WFCR	<b>28.9%</b>	<b>52.7%</b>	<b>23.6%</b>	50.4%	<b>15.6%</b>	28.9%	<b>26.0%</b>	44.4%
Impact	29.6%	57.6%	24.2%	51.4%	<b>15.6%</b>	29.6%	26.2%	44.4%

# rerouting airline passengers example

Due to a mechanical problem, Fly-By-Night Airlines had to cancel flight 162 - its only non-stop flight from San Francisco to New York.

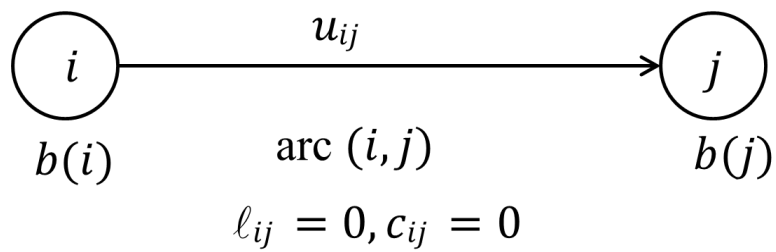
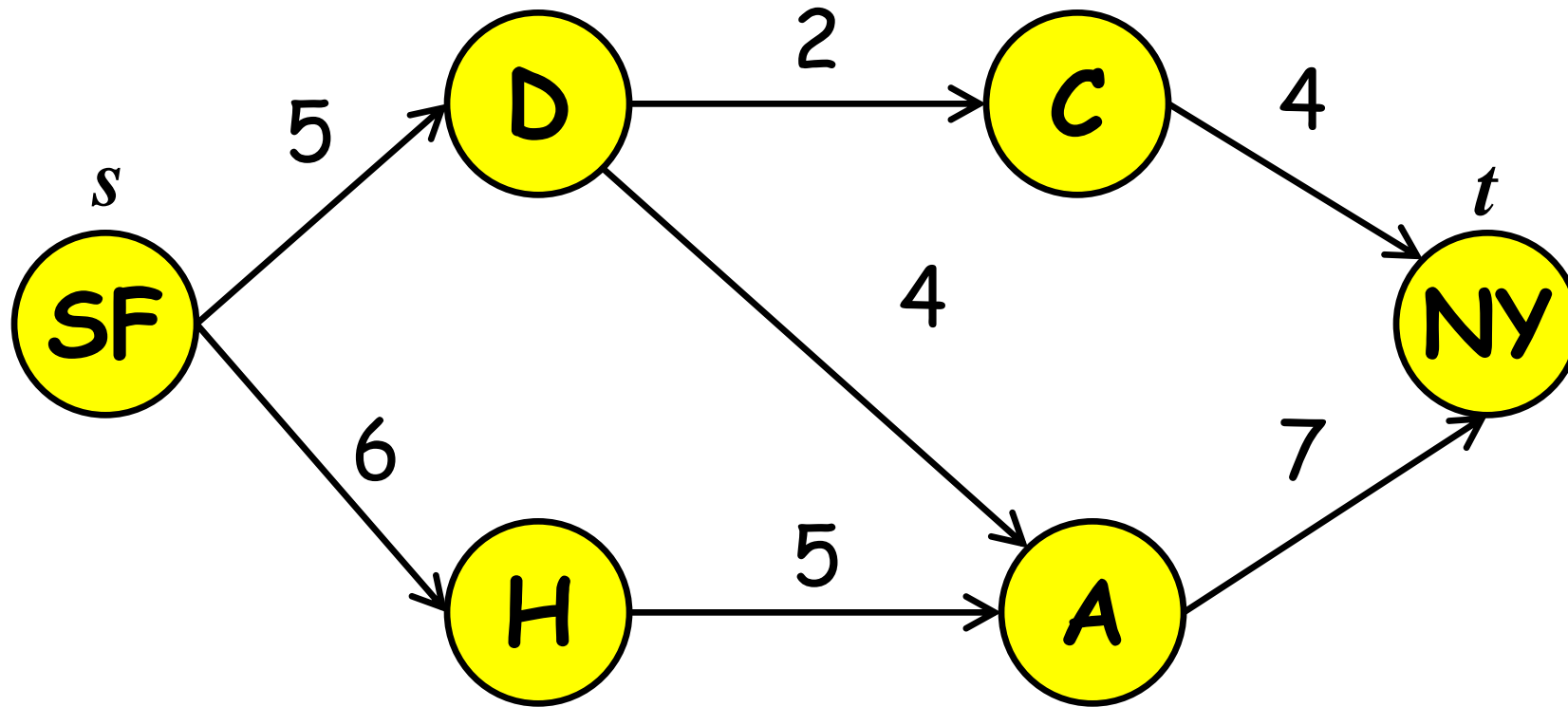
Formulate a maximum flow problem to reroute as many passengers as possible from San Francisco to New York.





<b>Flight</b>	<b>From</b>	<b>To</b>	<b># of seats</b>
<b>160</b>	<b>San Francisco</b>	<b>Denver</b>	<b>5</b>
<b>115</b>	<b>San Francisco</b>	<b>Houston</b>	<b>6</b>
<b>153</b>	<b>Denver</b>	<b>Atlanta</b>	<b>4</b>
<b>102</b>	<b>Denver</b>	<b>Chicago</b>	<b>2</b>
<b>170</b>	<b>Houston</b>	<b>Atlanta</b>	<b>5</b>
<b>150</b>	<b>Atlanta</b>	<b>New York</b>	<b>7</b>
<b>180</b>	<b>Chicago</b>	<b>New York</b>	<b>4</b>

# network representation

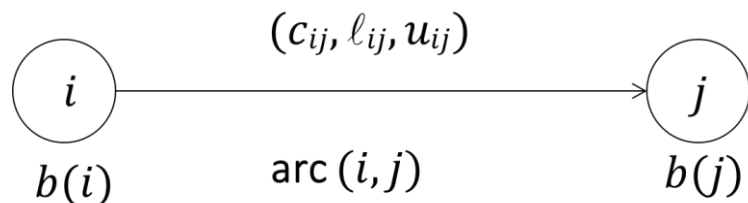
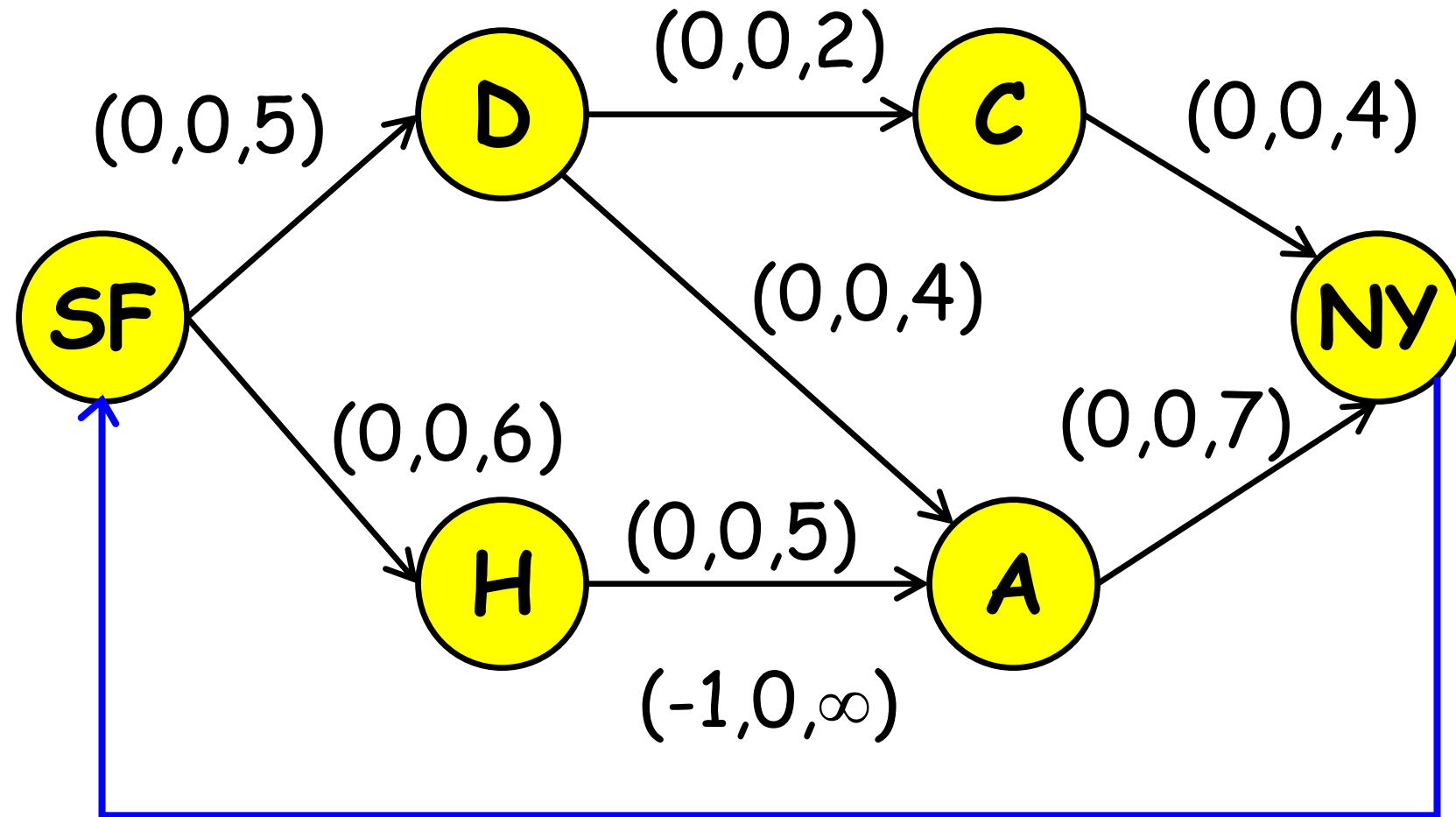


# MCNFP formulation of maximum flow problem

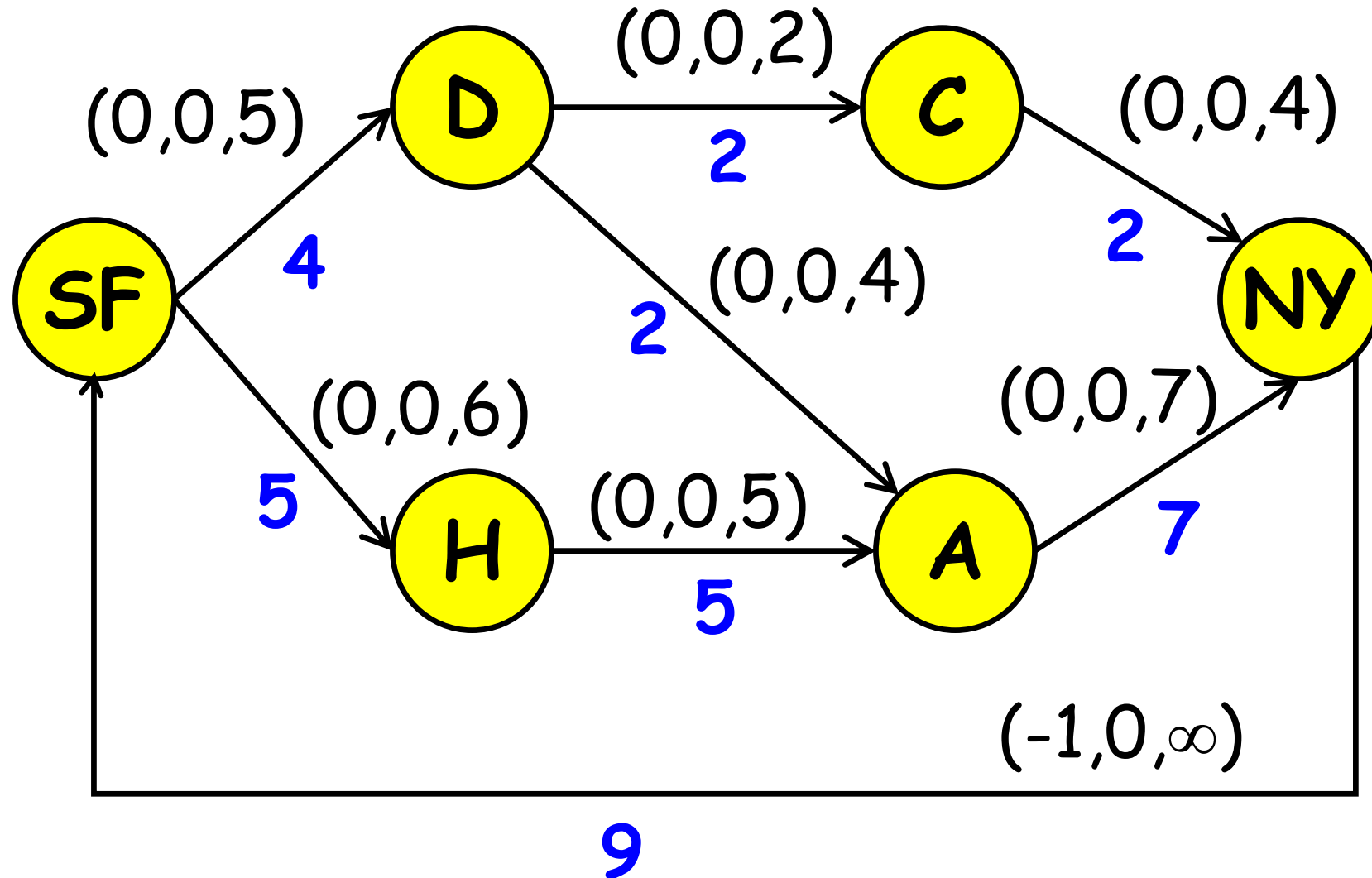
1. Let arc cost = 0 for all arcs
2. Add an arc from  $t$  to  $s$   
Give this arc a cost of  $-1$  and infinite capacity
3. All nodes are transshipment nodes

*“Circulation Problem” formulation*

# formulation of maximum flow problem as MCNFP



# optimal solution



# Widgets International Inc.

**Widgets International Inc. (WII)** manufactures widgets in various locations around the country.

The cost of manufacturing (excluding raw material costs), for each widget along with the plant's minimum and maximum monthly production levels is given in the table.

Plant	Cost per widget	Production max	Production min
1	\$5	500	0
2	\$7	750	100
3	\$3	1000	500
4	\$4	250	250

Each widget requires 20 lbs of raw material.

There are two suppliers of the necessary raw material.

- The suppliers can provide as much raw material as **WII** wants.
- Long standing contracts stipulate that **WII** will purchase 14,000 lbs of raw material from each (at minimum).
- The suppliers charge different amounts per pound of raw material:

<b>Supplier #1</b>	\$0.11/lbs
<b>Supplier #2</b>	\$0.09/lbs



Shipping costs from suppliers to plants (\$ per lbs):

	Plant 1	Plant 2	Plant 3	Plant 4
Supplier #1	\$0.01	\$0.02	\$0.035	\$0.04
Supplier #2	\$0.04	\$0.03	\$0.02	\$0.02

The widgets are sold in 4 cities: Norman; Ft. Collins; Austin; and San Jose. Transportation costs (\$ per widget) between plants and cities are:

Plant	Norman	Ft. Collins	Austin	San Jose
1	\$1	\$1	\$2	\$0
2	\$3	\$6	\$7	\$3
3	\$3	\$1	\$5	\$3
4	\$8	\$2	\$1	\$4

**WII** knows it's market really well.

In particular, the selling price per widget, the maximum demand, and minimum demand at each city is known.

City	Selling price	Max Demand	Min Demand
Norman	\$19	2,000	500
Ft. Collins	\$15	400	100
Austin	\$22	1,500	500
San Jose	\$24	1,550	400

Note: The max demand implies that the max number of widgets that will be bought in the city (if delivered); the min demand is the min amount of demand that must be sent to the city (or the population will riot).

***Can you formulate this as a MCNFP problem? If so, how? If not, why?***