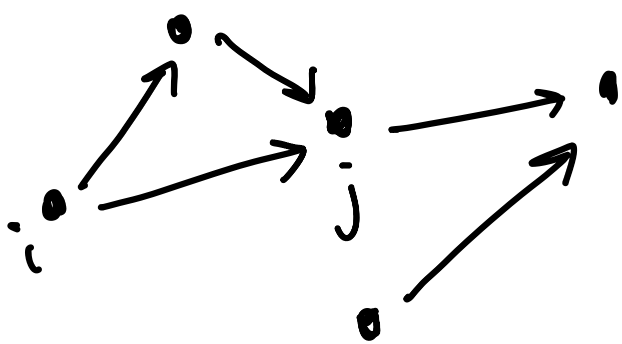
Network Flow ([V] Chapter 14)

$\mathcal{N}$ = set of nodes $\{i\}$ $(\mathcal{V}$, vertices$)$

$\mathcal{A}$ = set of (directed) arcs $(\mathcal{E}$, edges$)$

$\subseteq \{(i,j) : i,j \in \mathcal{N}\}$



Network $= (\mathcal{N}, \mathcal{A})$ $(\mathcal{V}, \mathcal{E})$

graph (or digraph)

Network Flow ([V] Chapter 14)

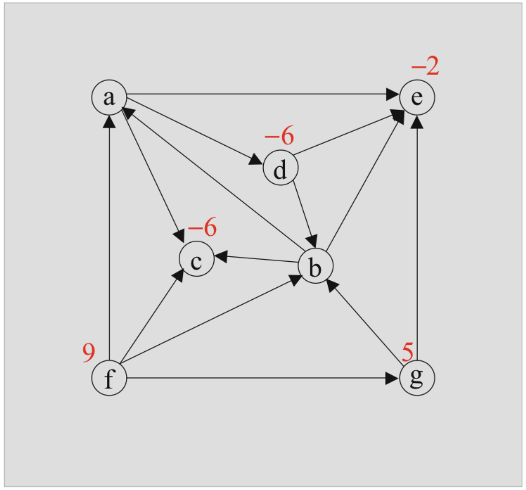


FIGURE 14.1. A network having 7 nodes and 14 arcs. The numbers written next to the nodes denote the supply at the node (negative values indicate demands; missing values indicate no supply or demand).

b_i

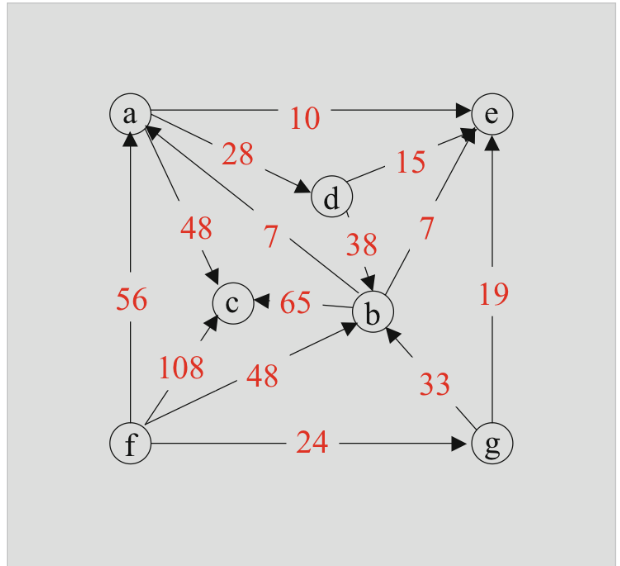


FIGURE 14.2. The costs on the arcs for the network in Figure 14.1.

c_{ij}

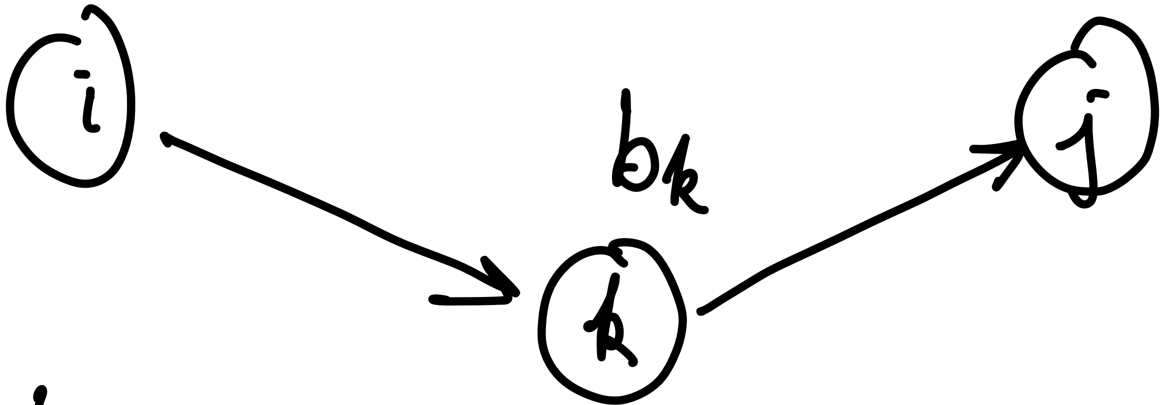
(1) $b_i > 0$ (supply) ; $b_i < 0$ (demand) $\quad \sum_{i \in N} b_i = 0$

(2) x_{ij} = amount transported from i to j

(3) c_{ij} = cost of transportation from i to j

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

Network Flow ([V] Chapter 14)

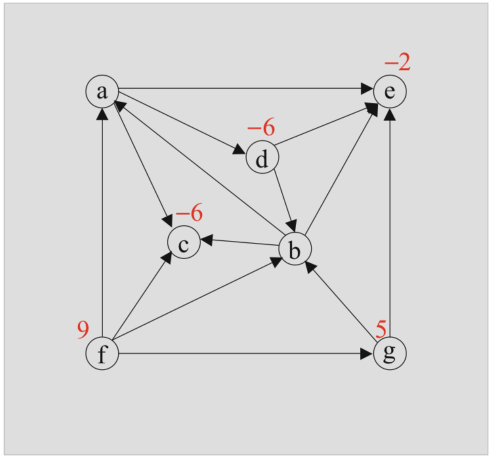


Balanced eqn at node k: $\sum_i x_{ik} + b_k = \sum_j x_{kj}$

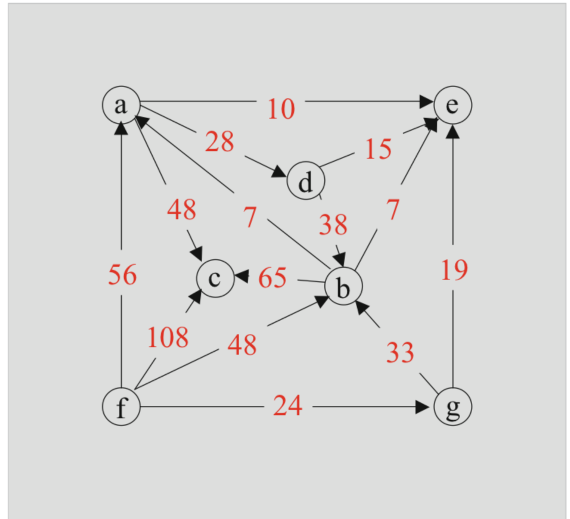
$$\boxed{\sum_i x_{ik} - \sum_j x_{kj} = -b_k}$$

$$\boxed{\begin{array}{l} \min \quad c^T X \\ AX = -b, \quad X \geq 0 \end{array}}$$

Network Flow ([V] Chapter 14)



b_i



c_{ij}

FIGURE 14.1. A network having 7 nodes and 14 arcs. The numbers written next to the nodes denote the supply at the node (negative values indicate demands; missing values indicate no supply or demand).

FIGURE 14.2. The costs on the arcs for the network in Figure 14.1.

$$x^T = \overbrace{\begin{bmatrix} x_{ac} & x_{ad} & x_{ae} & x_{ba} & x_{bc} & x_{be} & x_{db} & x_{de} & x_{fa} & x_{fb} & x_{fc} & x_{fg} & x_{gb} & x_{ge} \end{bmatrix}}^{\text{arcs}},$$

incidence matrix

$$\text{nodes}\left\{\begin{array}{c} (a) \\ (b) \\ (c) \\ (d) \\ (e) \\ (f) \\ (g) \end{array}\right. \quad A = \begin{bmatrix} -1 & -1 & -1 & 1 & & & & & 1 & & & & & \\ & & & -1 & -1 & -1 & 1 & & & 1 & & & 1 & \\ 1 & & & & 1 & & & & & & 1 & & & \\ & 1 & & & & & -1 & -1 & & & & & & \\ & & 1 & & & 1 & & 1 & & & & & & 1 \\ & & & & & & & & -1 & -1 & -1 & -1 & & \\ & & & & & & & & & & & 1 & -1 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ -6 \\ -6 \\ -2 \\ 9 \\ 5 \end{bmatrix}$$

A

$$c^T = \begin{bmatrix} 48 & 28 & 10 & 7 & 65 & 7 & 38 & 15 & 56 & 48 & 108 & 24 & 33 & 19 \end{bmatrix}.$$

Network Flow ([V] Chapter 14)

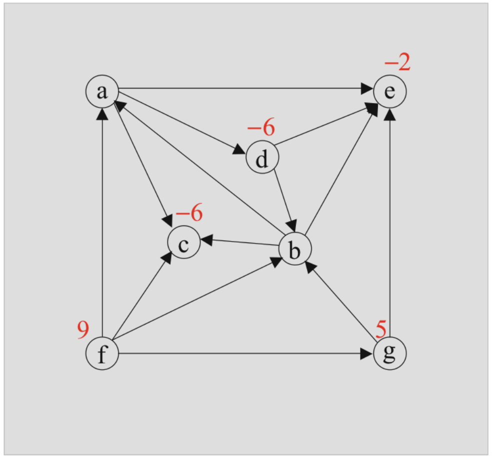


FIGURE 14.1. A network having 7 nodes and 14 arcs. The numbers written next to the nodes denote the supply at the node (negative values indicate demands; missing values indicate no supply or demand).

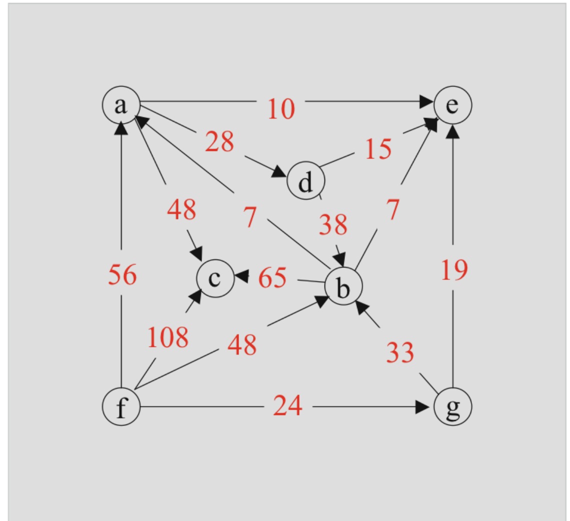


FIGURE 14.2. The costs on the arcs for the network in Figure 14.1.

Properties of A

(1)
$$A_{i,(k,l)} = \begin{cases} -1 & i=k \\ 1 & i=l \\ 0 & i\neq k,l \end{cases}$$

(2) Each column has one +1 and one −1

Network Flow ([V] Chapter 14)

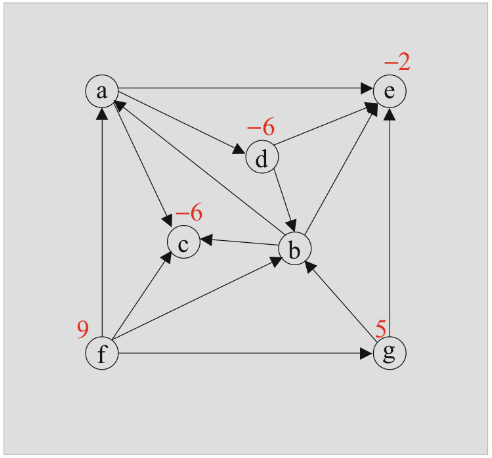


FIGURE 14.1. A network having 7 nodes and 14 arcs. The numbers written next to the nodes denote the supply at the node (negative values indicate demands; missing values indicate no supply or demand).

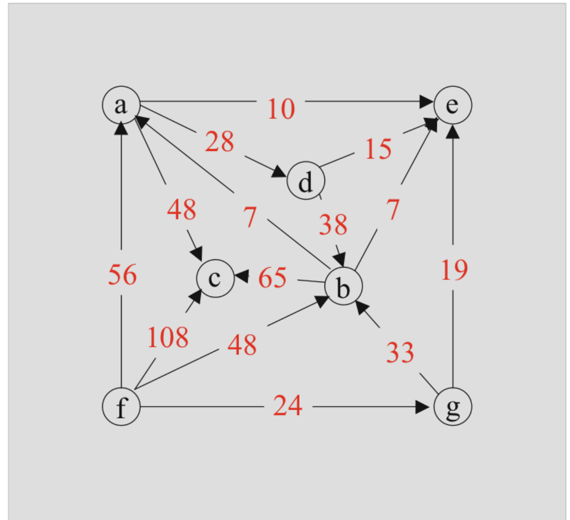


FIGURE 14.2. The costs on the arcs for the network in Figure 14.1.

Properties of A

(3) Rank(A) = m − 1

- Sum of all rows = 0 (Rank(A) ≤ m−1)

(• Can delete one row (Root node))

Network Flow ([V] Chapter 14)

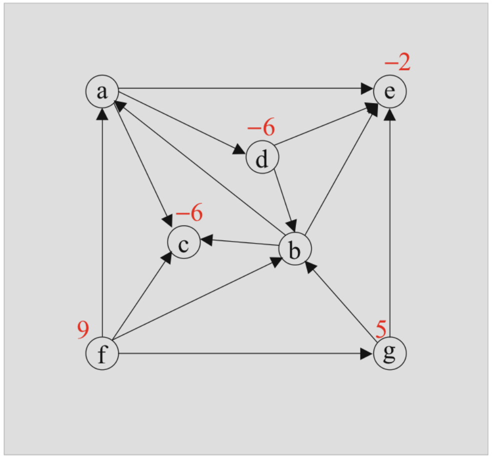


FIGURE 14.1. A network having 7 nodes and 14 arcs. The numbers written next to the nodes denote the supply at the node (negative values indicate demands; missing values indicate no supply or demand).

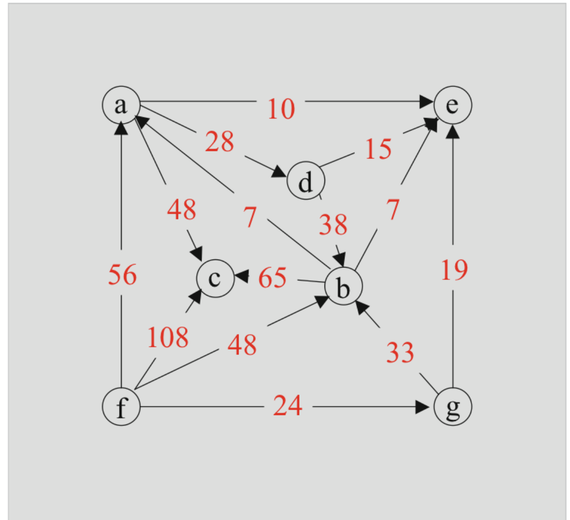


FIGURE 14.2. The costs on the arcs for the network in Figure 14.1.

Properties of A

(4) A is unimodular :

any m-minors has det. +1 or −1

$\Rightarrow$ If b is integral, any basic solution is integral.

6. The Integrality Theorem

[V], p. 247

In this section, we consider network flow problems for which all the supplies and demands are integers. Such problems are called *network flow problems with integer data*. As we explained in Section 14.2, for network flow problems, basic primal solutions are computed without any multiplication or division. The following important theorem follows immediately from this property:

THEOREM 14.2. *Integrality Theorem. For network flow problems with integer data, every basic feasible solution and, in particular, every basic optimal solution assigns integer flow to every arc.*

This theorem is important because many real-world network flow problems have integral supplies/demands and require their solutions to be integral too. This integrality restriction typically occurs when one is shipping indivisible units through a network. For example, it would not make sense to ship one-third of a car from an automobile assembly plant to one dealership with the other two-thirds going to another dealership.

Problems that are linear programming problems with the additional stipulation that the optimal solution values must be integers are called *integer programming problems*. Generally speaking, these problems are much harder to solve than linear programming problems (see Chapter 23). However, if the problem is a network flow problem with integer data, it can be solved efficiently using the simplex method to compute a basic optimal solution, which the integrality theorem tells us will be integer valued.

Network Flow ([V] Chapter 14)

(P)
$$\boxed{\begin{array}{l} \min \quad c^T x \\ Ax = -b, \quad x \geq 0 \end{array}}$$

(D)
$$\boxed{\begin{array}{ll} \max & -b^T y \\ \text{s.t.} & A^T y \leq c \end{array}}$$

$\Updownarrow$

$A^T y + z = c, \quad z \geq 0$

Network Flow ([V] Chapter 14)

$$(P)\quad \boxed{\begin{array}{l} \min \quad c^T x \\ Ax = -b, \quad x \geq 0 \end{array}}$$

$$(D)\quad \boxed{\begin{array}{l} \max \quad -b^T y \\ \text{s.t.} \quad A^T y + z = c, \quad z \geq 0 \end{array}}$$

$$\max \; -\sum_{i \in N} b_i y_i$$

$$\text{s.t.} \quad y_j - y_i + z_{ij} = c_{ij}, \quad (i,j) \in A$$

$$z_{ij} \geq 0$$

Network Flow ([V] Chapter 14)

$$(P) \quad \boxed{\begin{array}{l} \min \quad c^T x \\ Ax = -b, \quad x \geq 0 \end{array}}$$

$$(D) \quad \boxed{\begin{array}{ll} \max & -b^T y \\ \text{s.t.} & A^T y + z = c, \quad z \geq 0 \end{array}}$$

$$\max \; -\sum_{i \in N} b_i y_i$$

$$\text{s.t.} \quad \underline{y_j - y_i} + z_{ij} = c_{ij}, \quad (i,j) \in A$$

$$z_{ij} \geq 0$$

$y_j \to y_j + \text{const.}$

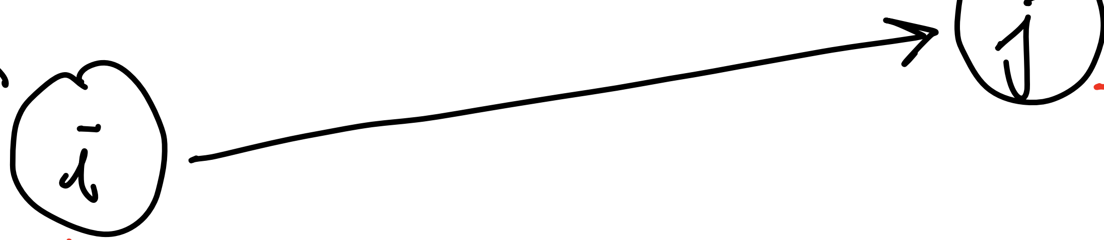
Economic Interpretation of Network Dual

$$
\boxed{
\begin{aligned}
(D)\quad & \max\ -\sum_{i\in N} b_i y_i \\
\text{s.t.}\quad & y_j - y_i + z_{ij} = c_{ij}, \quad (i,j)\in A \\
& z_{ij} \geq 0
\end{aligned}
}
$$

$\boxed{y_j - y_i \leq c_{ij}}$

you get y_i

you pay y_j

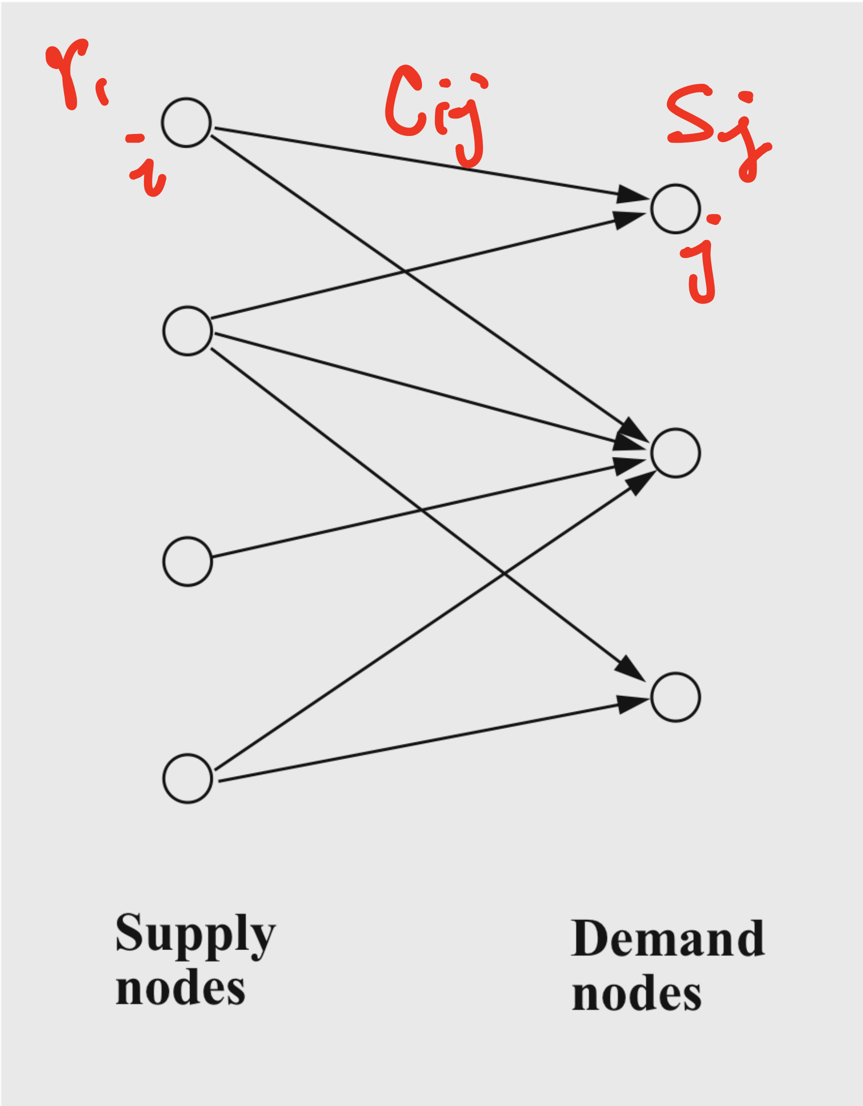


new company pays at y_i

new company buys at y_j

min cost $\sum_i b_i y_i$

Transportation Problem [V] p. 258

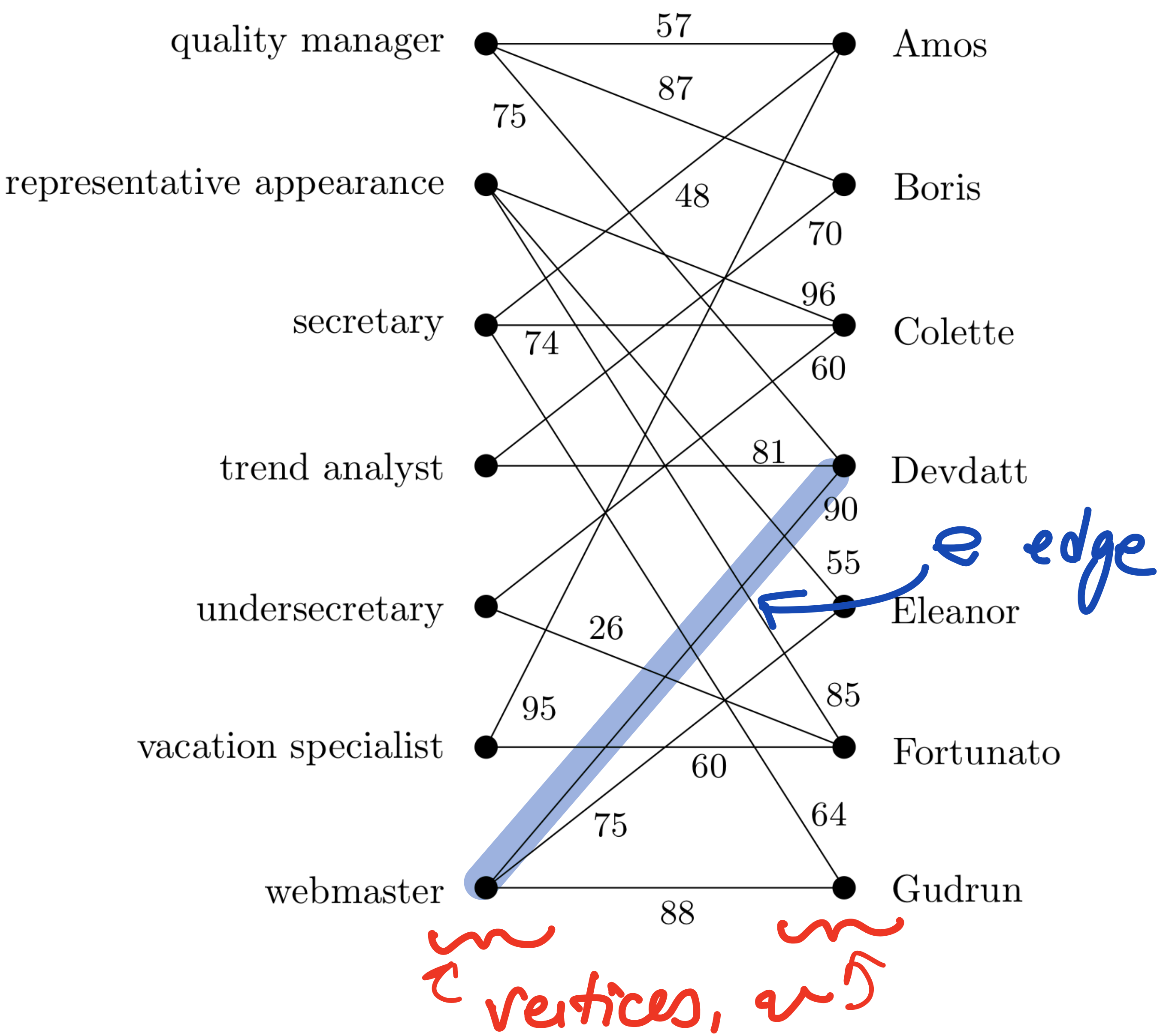


$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{D}} c_{ij} x_{ij} \\
\text{subject to} \quad & \sum_{j \in \mathcal{D}} x_{ij} = r_i \qquad i \in \mathcal{S} \\
& \sum_{i \in \mathcal{S}} x_{ij} = s_j \qquad j \in \mathcal{D} \\
& x_{ij} \geq 0 \qquad i \in \mathcal{S}, j \in \mathcal{D}.
\end{aligned}
$$

(bipartite graph)

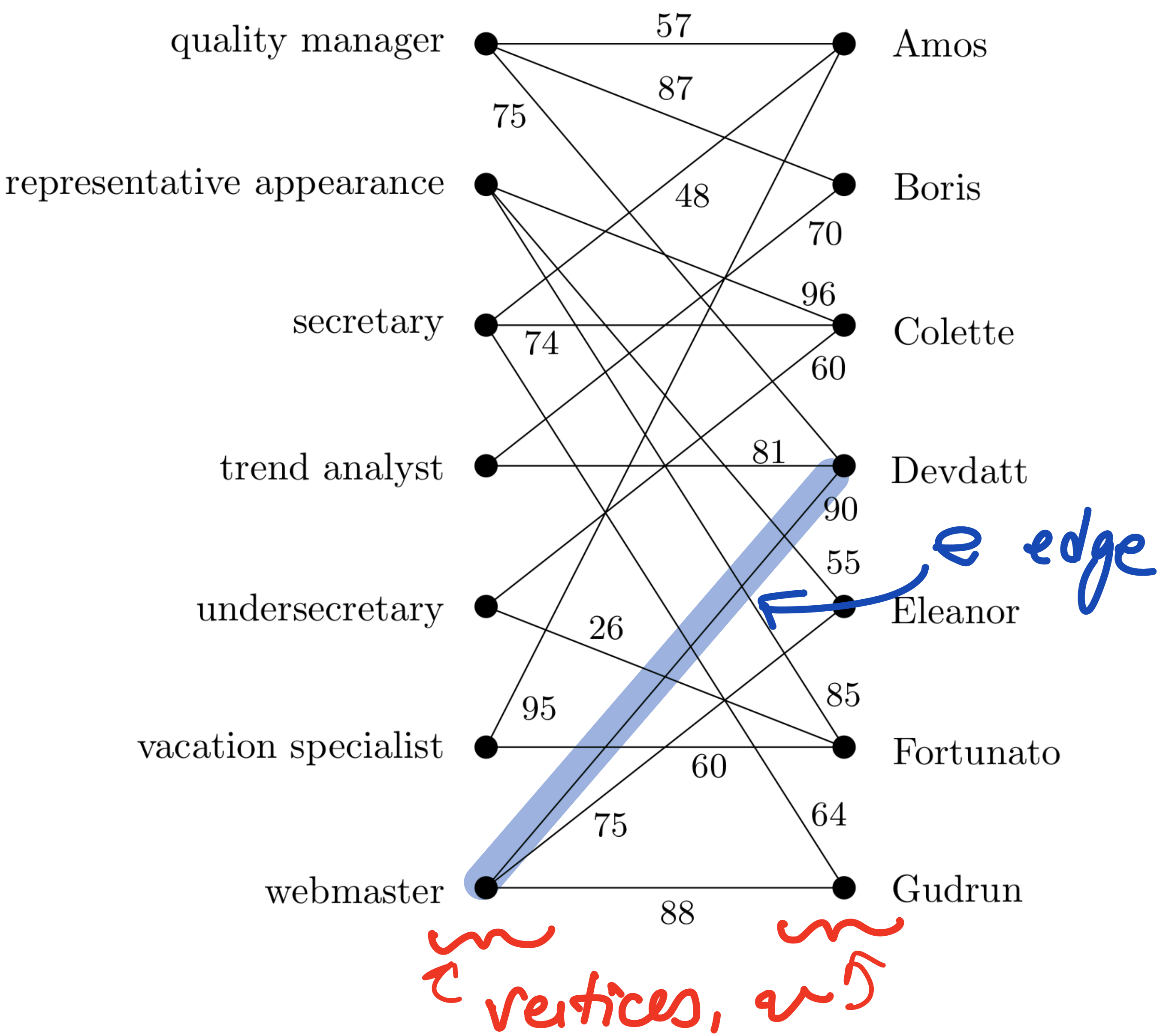
Integer Programming

[MG] p. 31 Maximum Weight Matching



Integer Programming

[MG] p. 31 Maximum Weight Matching



$$\max \sum_e w_e x_e$$

s.t. for any v,

$$\sum_{e,\, v \in e} x_e = 1$$

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

[V] p.259

2. The Assignment Problem

Given a set $\mathcal{S}$ of m people, a set $\mathcal{D}$ of m tasks, and for each $i \in \mathcal{S}$, $j \in \mathcal{D}$ a cost c_{ij} associated with assigning person i to task j, the *assignment problem* is to assign each person to one and only one task in such a manner that each task gets covered by someone and the total cost of the assignments is minimized. If we let

$$x_{ij} = \begin{cases} 1 & \text{if person } i \text{ is assigned task } j, \\ 0 & \text{otherwise,} \end{cases}$$

then the objective function can be written as

$$\text{minimize} \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{D}} c_{ij} x_{ij}.$$

The constraint that each person is assigned exactly one task can be expressed simply as

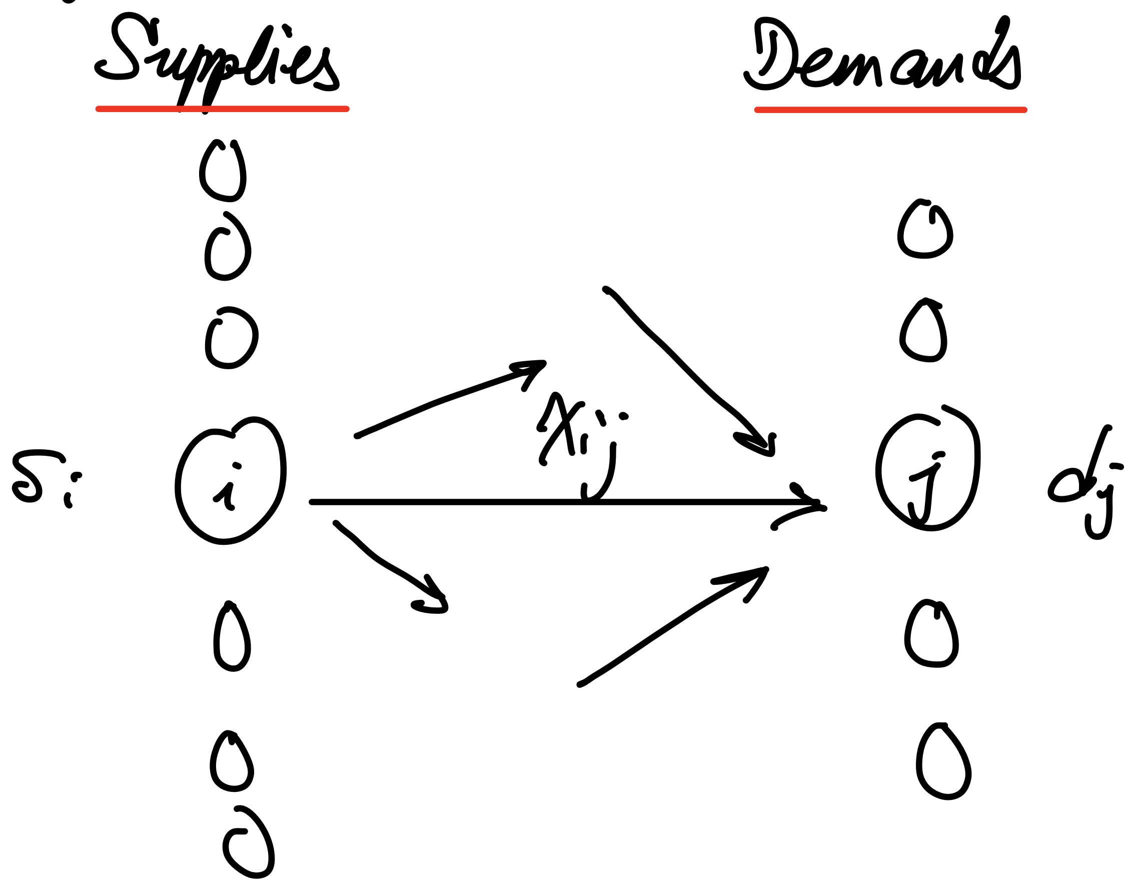
$$\sum_{j \in \mathcal{D}} x_{ij} = 1, \qquad \text{for all } i \in \mathcal{S}.$$

Also, the constraint that every task gets covered by someone is just

$$\sum_{i \in \mathcal{S}} x_{ij} = 1, \qquad \text{for all } j \in \mathcal{D}.$$

Variants of Network Flows

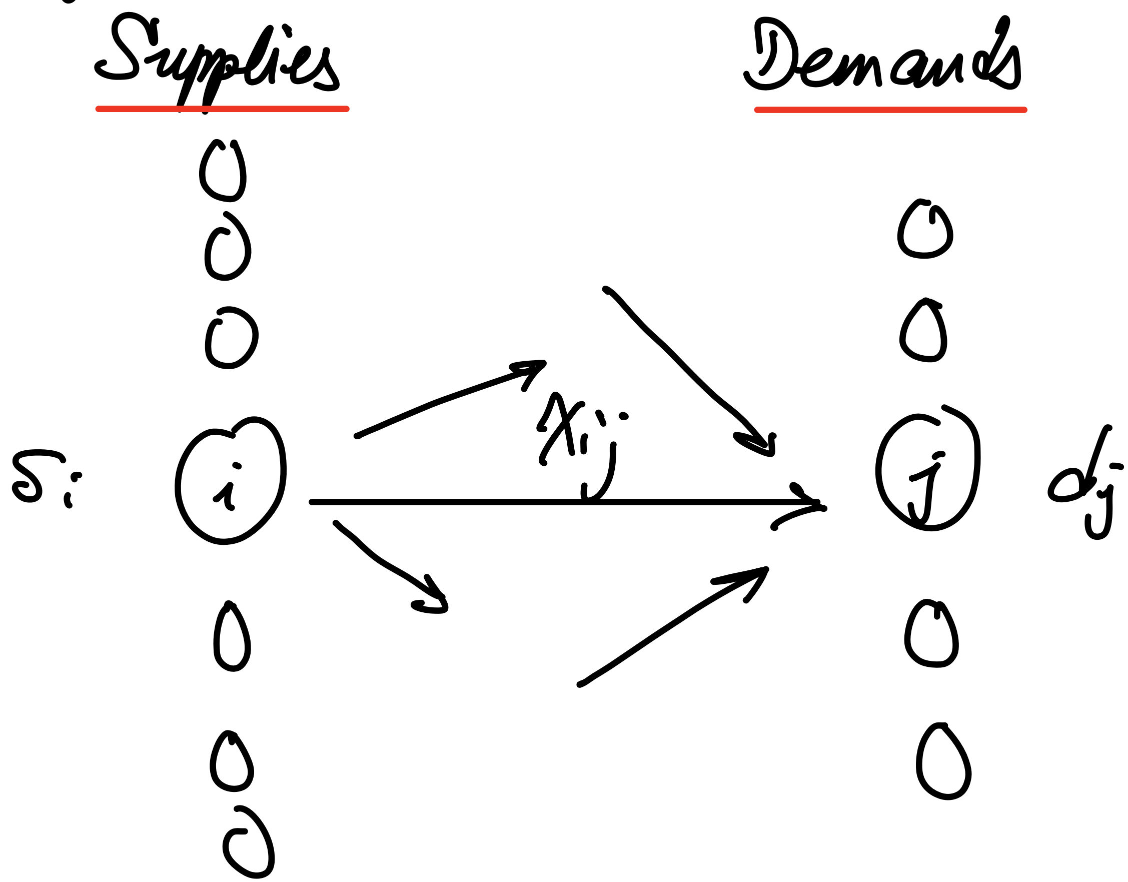
Inequality constraints (Transportation Problem) [C] p.320



$$\sum_j x_{ij} = s_i, \qquad \sum_i x_{ij} = d_j$$

Variants of Network Flows

Inequality constraints (Transportation Problem)

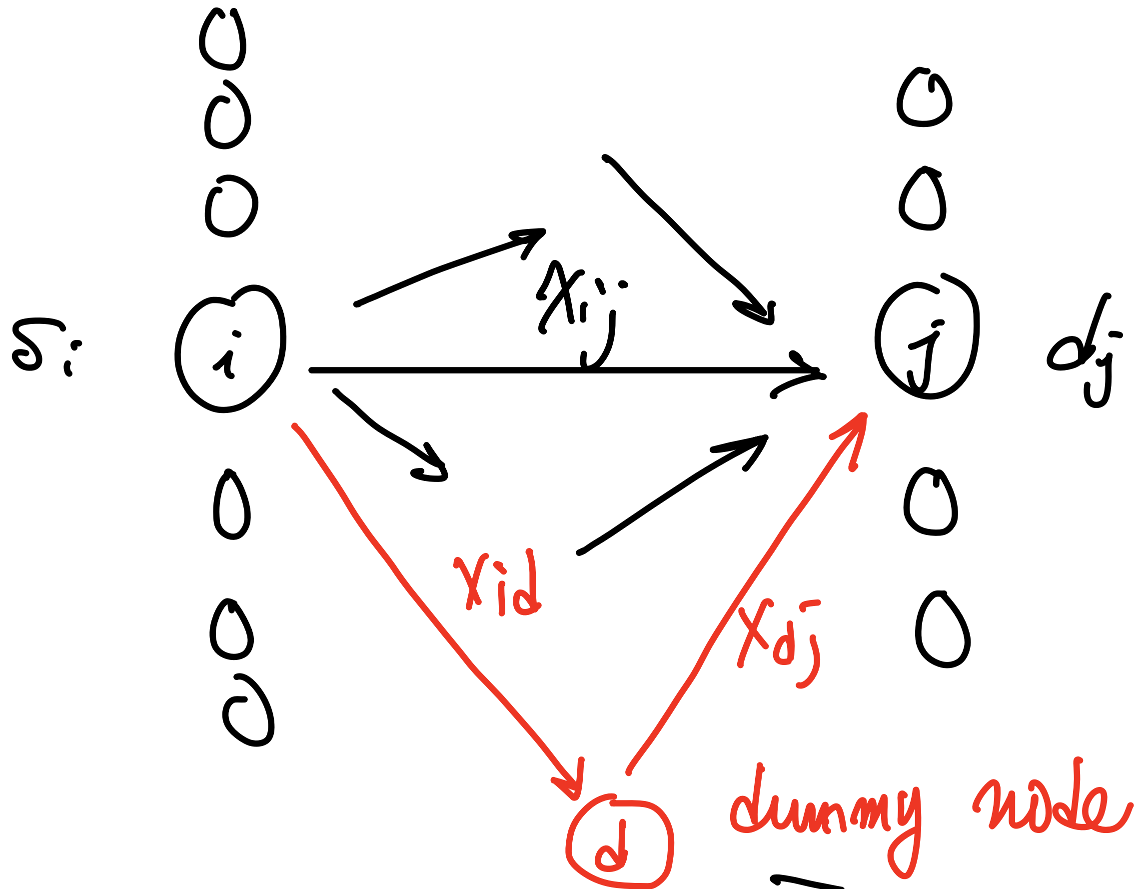


$$\sum_j x_{ij} \leq s_i \,, \qquad \sum_i x_{ij} \geq d_j$$

Variants of Network Flows

Inequality constraints (Transportation Problem)

Supplies | Demands



$x_{id}, x_{dj} \geq 0$

$$\sum_j x_{ij} + \underline{x_{id}} = s_i , \qquad \sum_i x_{ij} = d_j + \underline{x_{dj}}$$

Variants of Network Flows

Inequality constraints (Transportation Problem)

$$\sum_j x_{ij} + \underline{x_{id}} = s_i, \qquad \sum_i x_{ij} = d_j + \underline{x_{dj}}$$

$$\underline{\sum_{i,j} x_{ij}} + \sum_i x_{id} = \sum_i s_i \qquad \underline{\sum_{j,i} x_{ij}} = \sum_j d_j + \sum_j x_{dj}$$

$=$

Hence, the new additional constraint:

$$\boxed{\sum_i s_i - \sum_i x_{id} = \sum_j d_j + \sum_j x_{dj}}$$

Inequality constraints (Transportation Problem) [C] p.321

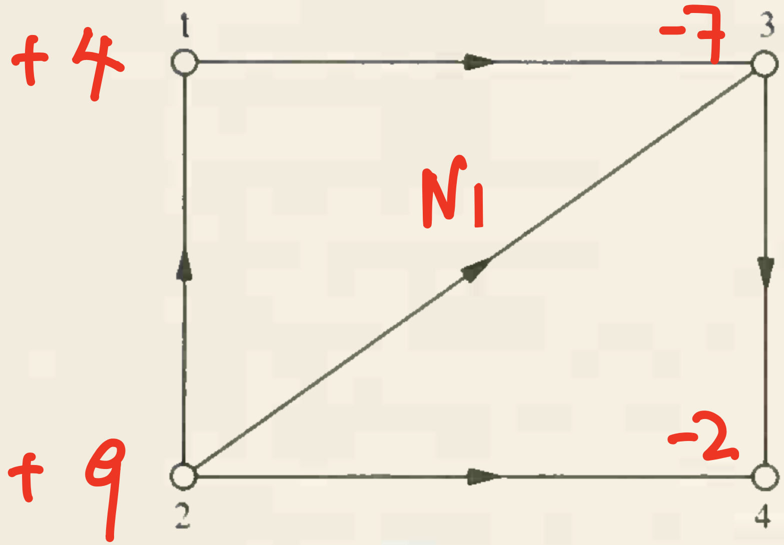


Figure 20.1

4 units at the source 1

9 units at the source 2

and demands of

7 units at the sink 3

2 units at the sink 4.

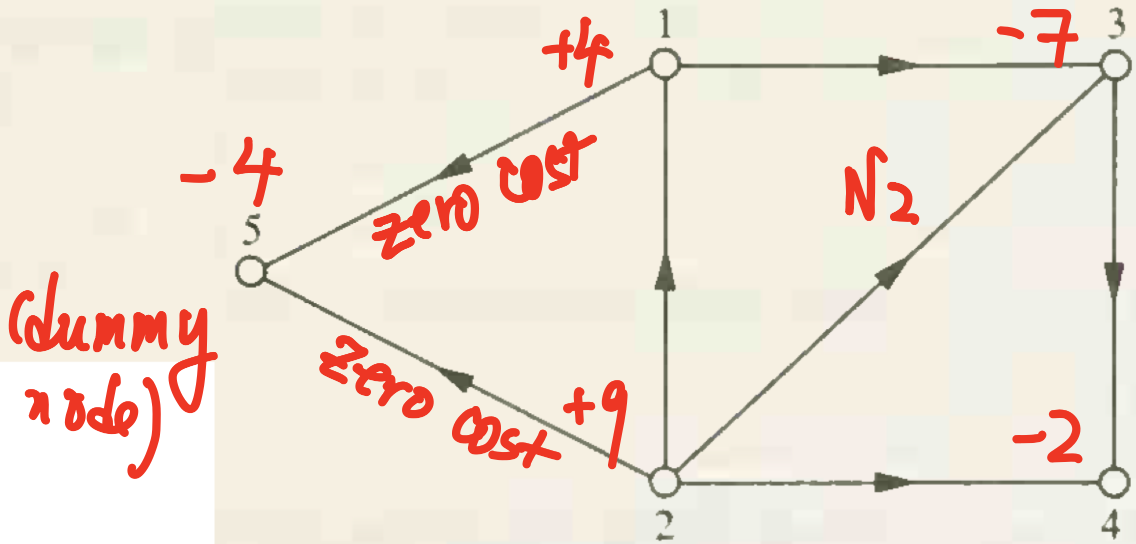


Figure 20.2

Inequality constraints (Transportation Problem) [c] p.321

$$
\begin{array}{llr}
\text{minimize} & z = c_{13}x_{13} + c_{21}x_{21} + c_{23}x_{23} + c_{24}x_{24} + c_{34}x_{34} & \\
\text{subject to} & -x_{13} + x_{21} & \geq -4 \\
 & -x_{21} - x_{23} - x_{24} & \geq -9 \\
 & x_{13} + x_{23} - x_{34} & = 7 \\
 & x_{24} + x_{34} & = 2 \\
 & x_{13}, x_{21}, x_{23}, x_{24}, x_{34} & \geq 0
\end{array}
$$

N_1

$$
\begin{array}{llr}
\text{minimize} & z & \\
\text{subject to} & -x_{13} + x_{21} - x_{15} & = -4 \\
 & -x_{21} - x_{23} - x_{24} - x_{25} & = -9 \\
 & x_{13} + x_{23} - x_{34} & = 7 \\
 & x_{24} + x_{34} & = 2 \\
 & x_{15} + x_{25} & = 4 \\
 & x_{13}, x_{21}, x_{23}, x_{24}, x_{34}, x_{15}, x_{25} & \geq 0.
\end{array}
$$

N_2

Inequality constraints (More generally, [c] p.322)

$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ - incident matrix

$$A_1 \vec{X} \geq -\vec{b}_1$$

$$A_2 \vec{X} \leq -\vec{b}_2$$

$$A_1 \vec{X} = -\vec{b}_1 + \vec{W}_1, \quad \vec{W}_1 \geq 0$$

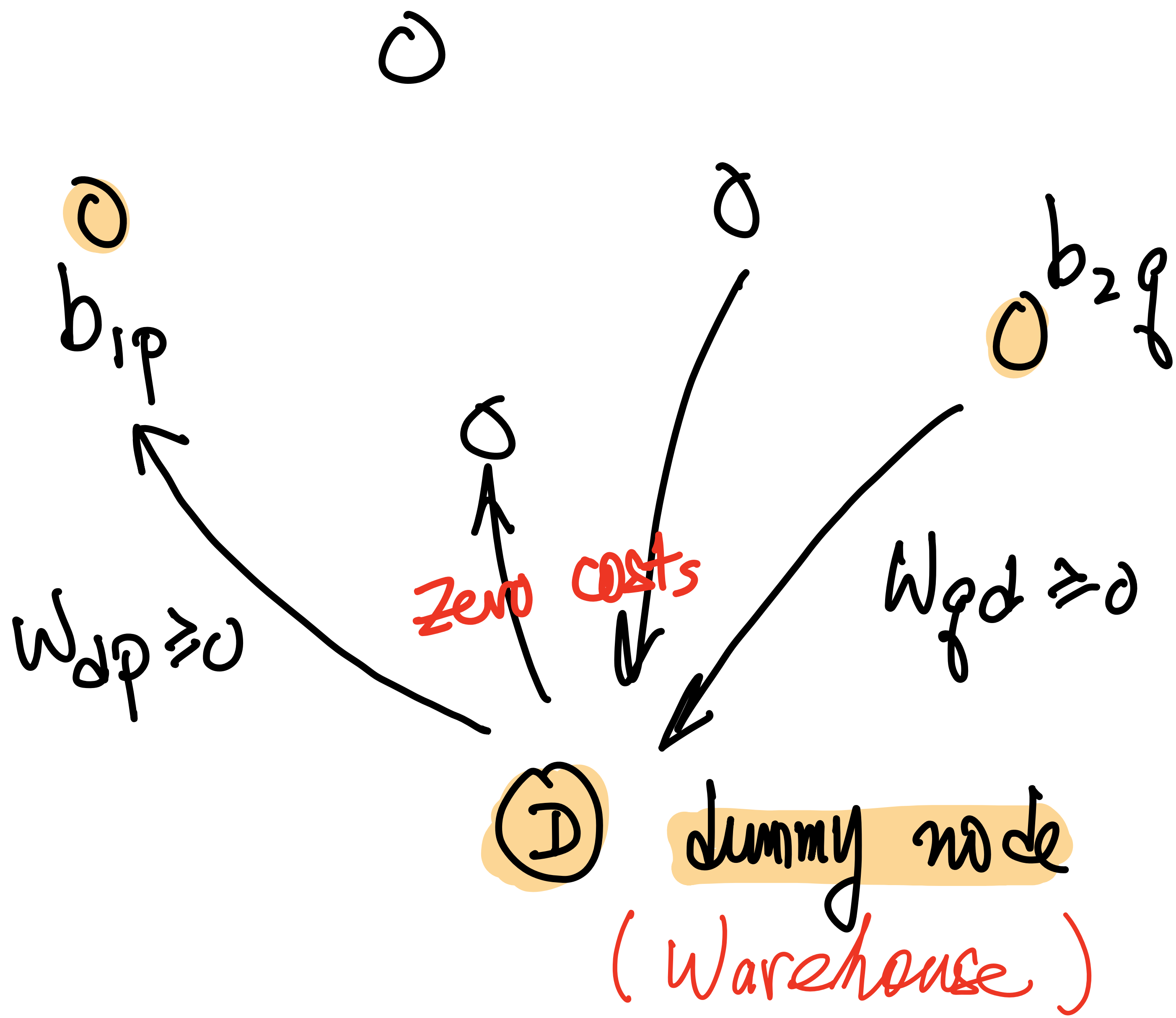
$$A_2 \vec{X} = -\vec{b}_2 - \vec{W}_2 \quad \vec{W}_2 \geq 0$$

+

new constraint

$$\boxed{\sum_p W_{1p} - \sum_q W_{2q} = \sum_p b_{1p} + \sum_q b_{1q}}$$

Inequality constraints (More generally, [c] p.322)



Upper Bounded Transshipment Problem [V]

p. 263

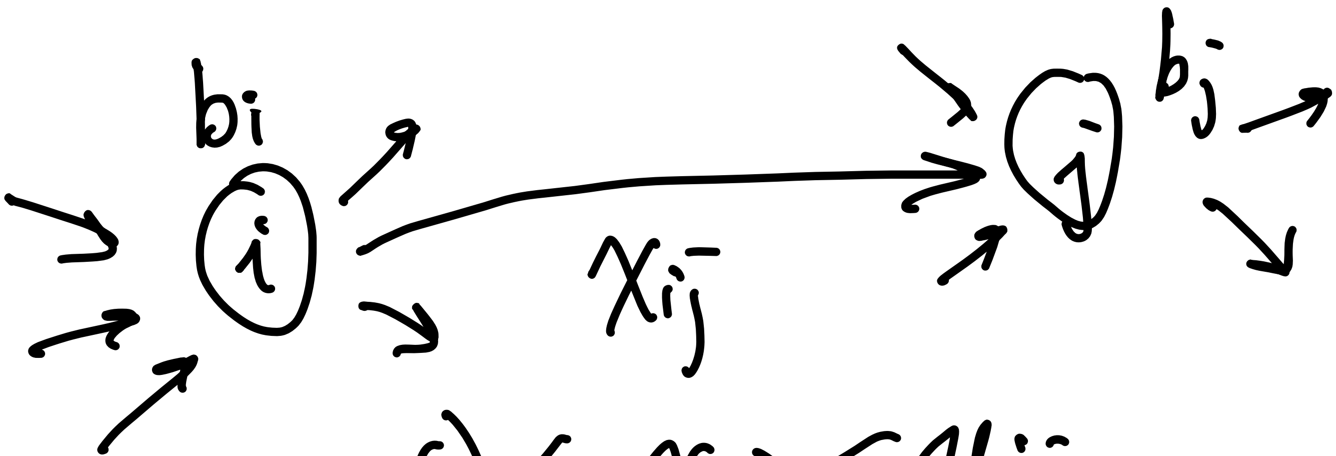
$$AX = b, \qquad 0 \leq X \leq U$$

$$i \xrightarrow{\quad x_{ij} \quad} j$$

$$0 \leq x_{ij} \leq u_{ij}$$

Upper Bounded Transshipment Problem

$$AX = -b, \qquad 0 \le X \le u$$



$$0 \le x_{ij} \le u_{ij}$$

1 at i $\qquad \cdots\ - x_{ij}\ \cdots = -b_i$

2 at j $\qquad \cdots\ + x_{ij}\ \cdots = -b_j$

3 $\qquad x_{ij} + t_{ij} = u_{ij} \quad (t_{ij} \ge 0)$

Upper Bounded Transhipment Problem

(1) $\quad \cdots\cdots \; -x_{ij} \; \cdots\cdots = -b_i \quad$ at (i)

(2) $\quad \cdots\cdots \; +x_{ij} \; \cdots\cdots = -b_j \quad$ at (j)

(3) $\quad x_{ij} + t_{ij} = u_{ij}$

(1) $\quad \cdots\cdots \; -x_{ij} \; \cdots = -b_i$

(2) - (3) $\quad \cdots\cdots \; \cdots -t_{ij} = -b_j - u_{ij}$

(3) $\quad +x_{ij} + t_{ij} = u_{ij}$

Upper Bounded Transhipment Problem

$$
\begin{cases}
(1) & \cdots\cdots - x_{ij} \cdots = -b_i \\
(2)' & \cdots\cdots - t_{ij} = -b_j - u_{ij} \\
(3) & + x_{ij} + t_{ij} = u_{ij}
\end{cases}
$$