Operations Research (OR)

Unit -II

Transportation, Assignment, and Transshipment Problems

Transportation, Assignment, and Transshipment Problems

- A network model is one which can be represented by a set of nodes, a set of arcs, and functions (e.g. costs, supplies, demands, etc.) associated with the arcs and/or nodes.
- Transportation, assignment, and transshipment problems of this chapter, as well as the shortest route, minimal spanning tree, and maximal flow problems (Chapter 9) and PERT/CPM problems (Chapter 10) are all examples of network problems.

Transportation, Assignment, and Transshipment Problems

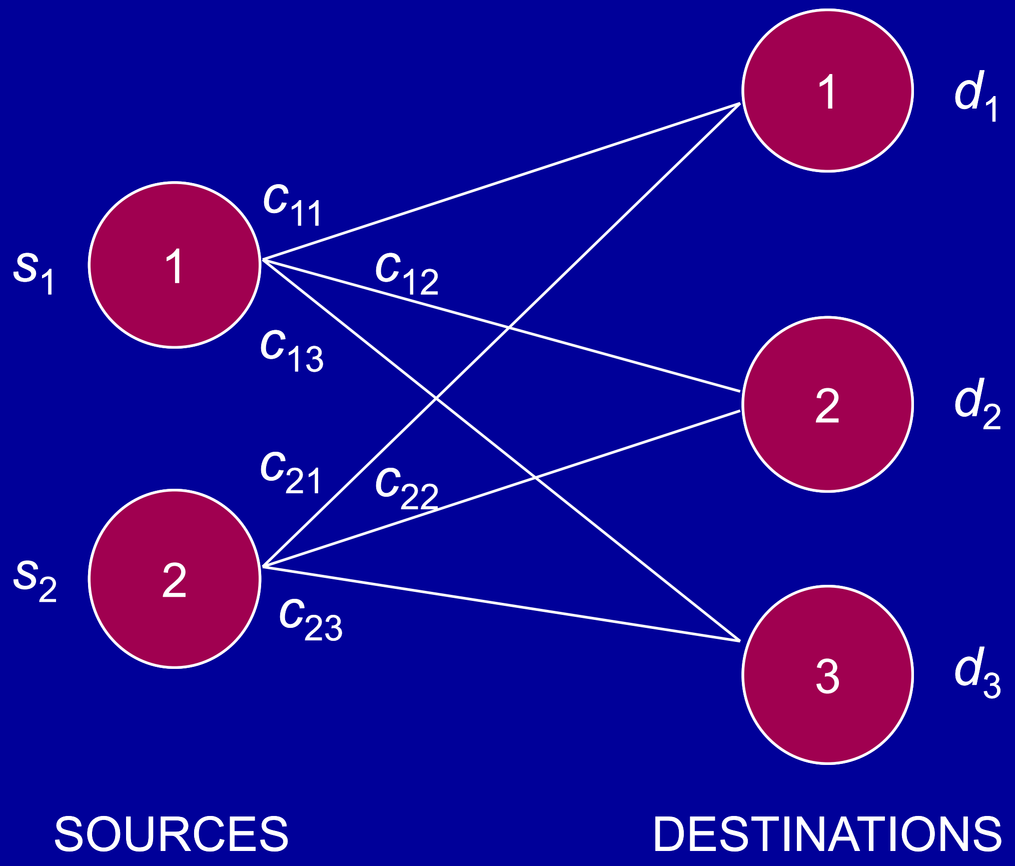
- Each of the three models of this chapter (transportation, assignment, and transshipment models) can be formulated as linear programs and solved by general purpose linear programming Algorithms (simplex method).
- For each of the three models, if the right-hand side of the linear programming formulations are all integers, the optimal solution will be in terms of integer values for the decision variables.
- However, there are many computer packages (including *The Management Scientist, DS, QSB*) which contain separate computer codes for these models which take advantage of their network structure.

Transportation Problem

- The transportation problem seeks to minimize the total shipping costs of transporting goods from *m* origins or sources (each with a supply s_i) to *n* destinations (each with a demand d_j), when the unit shipping cost from source, *i*, to a destination, *j*, is c_{ij}.
- The network representation for a transportation problem with two sources and three destinations is given on the next slide.

Transportation Problem

- Network Representation



Transportation Problem

- LP Formulation

The linear programming formulation in terms of the amounts shipped from the sources to the destinations, x_{ij} , can be written as:

Min $\sum_{i}\sum_{j} c_{ij}x_{ij}$ *(total transportation cost)*

s.t. $\sum_{j} x_{ij} \le s_i$ for each source i *(supply constraints)*

$\sum_{i} x_{ij} = d_j$ for each destination j *(demand constraints)*

$x_{ij} \ge 0$ for all i and j *(nonnegativity constraints)*

Transportation Problem

- To solve the transportation problem by its special purpose algorithm, it is required that the sum of the supplies at the sources equal the sum of the demands at the destinations. If the total supply is greater than the total demand, a dummy destination is added with demand equal to the excess supply, and shipping costs from all sources are zero. Similarly, if total supply is less than total demand, a dummy source is added.
- When solving a transportation problem by its special purpose algorithm, unacceptable shipping routes are given a cost of +*M* (a large number).

Transportation Problem

- A transportation tableau is given below. Each cell represents a shipping route (which is an arc on the network and a decision variable in the LP formulation), and the unit shipping costs are given in an upper right hand box in the cell.

	D1	D2	D3	Supply
S1	15	30	20	50
S2	30	40	35	30
Demand	25	45	10	

Problem formulation

- The LP model for this problem is as follows:

Min $Z = 15\ X_{11} + 30\ X_{12} + 20\ X_{13} + 30\ X_{21} + 40X_{22} + 35X_{23}$

S.t.

$X_{11} + X_{12} + X_{13} \leq 50$

Supply constraints

$X_{21} + X_{22} + X_{23} \leq 30$

$X_{11} + X_{21} = 25$

$X_{12} + X_{22} = 45$

$X_{13} + X_{23} = 10$

demand constraints

$X_{11}, \dots, X_{23} \geq 0$

Transportation Problem

- The transportation problem is solved in two phases:
  - Phase I -- Obtaining an initial feasible solution
  - Phase II -- Moving toward optimality
- In Phase I, the Minimum-Cost Procedure can be used to establish an initial basic feasible solution without doing numerous iterations of the simplex method.
- In Phase II, the Stepping Stone, by using the MODI method for evaluating the reduced costs may be used to move from the initial feasible solution to the optimal one.

Initial Tableau

- There are many method for finding the initial tableau for the transportation problem which are:

1. Northwest corner
2. Minimum cost of the row
3. Minimum cost of the column
4. Least cost
5. Vogle's approximation method
6. Russell's approximation method

Northwest corner

- Northwest corner: Begin by selecting X_{11} (that is, start in the northwest corner of the transportation tableau). Therefore, if X_{ij} was the last basic variable (occupied cell) selected, then select X_{ij+1} (that is, move one column to the right) if source I has any supply remaining. Otherwise, next select $X_{i+1\,j}$ (that is, move one row down).

	D1	D2	D3	Supply
S1	15 *25*	30 *25*	20	50
S2	30	40 *20*	35 *10*	30
Demand	25	45	10	

Total cost is $2275

Transportation Algorithm

- Phase I - Minimum-Cost Method
  - Step 1: Select the cell with the least cost. Assign to this cell the minimum of its remaining row supply or remaining column demand.
  - Step 2: Decrease the row and column availabilities by this amount and remove from consideration all other cells in the row or column with zero availability/demand. (If both are simultaneously reduced to 0, assign an allocation of 0 to any other unoccupied cell in the row or column before deleting both.) GO TO STEP 1.

	D1	D2	D3	Supply
S1	15 25	30 15	20 10	50
S2	30	40 30	35	30
Demand	25	45	10	

Total cost is $2225

Transportation Algorithm

- Phase II - Stepping Stone Method
  - Step 1: For each unoccupied cell, calculate the reduced cost by the MODI method described below. Select the unoccupied cell with the most negative reduced cost. (For maximization problems select the unoccupied cell with the largest reduced cost.) If none, STOP.
  - Step 2: For this unoccupied cell generate a stepping stone path by forming a closed loop with this cell and occupied cells by drawing connecting alternating horizontal and vertical lines between them.

    Determine the minimum allocation where a subtraction is to be made along this path.

Transportation Algorithm

- Phase II - Stepping Stone Method (continued)
  - Step 3: Add this allocation to all cells where additions are to be made, and subtract this allocation to all cells where subtractions are to be made along the stepping stone path.

    (Note: An occupied cell on the stepping stone path now becomes 0 (unoccupied). If more than one cell becomes 0, make only one unoccupied; make the others occupied with 0's.)

    GO TO STEP 1.

Transportation Algorithm

- MODI Method (for obtaining reduced costs)

  Associate a number, u_i, with each row and v_j with each column.

  - Step 1: Set $u_1 = 0$.
  - Step 2: Calculate the remaining u_i's and v_j's by solving the relationship $c_{ij} = u_i + v_j$ for occupied cells.
  - Step 3: For unoccupied cells (i,j), the reduced cost = $c_{ij} - u_i - v_j$.

Example: BBC

Building Brick Company (BBC) has orders for 80 tons of bricks at three suburban locations as follows: Northwood -- 25 tons, Westwood -- 45 tons, and Eastwood -- 10 tons. BBC has two plants, each of which can produce 50 tons per week.

How should end of week shipments be made to fill the above orders given the following delivery cost per ton:

	Northwood	Westwood	Eastwood
Plant 1	24	30	40
Plant 2	30	40	42

Example: BBC

- Initial Transportation Tableau

  Since total supply = 100 and total demand = 80, a dummy destination is created with demand of 20 and 0 unit costs.

	Northwood	Westwood	Eastwood	Dummy	Supply
Plant 1	24	30	40	0	50
Plant 2	30	40	42	0	50
Demand	25	45	10	20	

Example: BBC

- Least Cost Starting Procedure
  - Iteration 1: Tie for least cost (0), arbitrarily select x_{14}. Allocate 20. Reduce s_1 by 20 to 30 and delete the Dummy column.
  - Iteration 2: Of the remaining cells the least cost is 24 for x_{11}. Allocate 25. Reduce s_1 by 25 to 5 and eliminate the Northwood column.
  - Iteration 3: Of the remaining cells the least cost is 30 for x_{12}. Allocate 5. Reduce the Westwood column to 40 and eliminate the Plant 1 row.
  - Iteration 4: Since there is only one row with two cells left, make the final allocations of 40 and 10 to x_{22} and x_{23}, respectively.

Example: BBC

- Initial tableau

	Northwood	Westwood	Eastwood	Dummy	Supply
Plant 1	24 *25*	30 *5*	40	0 *20*	50
Plant 2	30	40 *40*	42 *10*	0	50
Demand	25	45	10	20	

Total transportation cost is $2770

Example: BBC

- Iteration 1
  - MODI Method
    1. Set $u_1 = 0$
    2. Since $u_1 + v_j = c_{1j}$ for occupied cells in row 1, then $v_1 = 24$, $v_2 = 30$, $v_4 = 0$.
    3. Since $u_i + v_2 = c_{i2}$ for occupied cells in column 2, then $u_2 + 30 = 40$, hence $u_2 = 10$.
    4. Since $u_2 + v_j = c_{2j}$ for occupied cells in row 2, then $10 + v_3 = 42$, hence $v_3 = 32$.

Example: BBC

- Iteration 1
  - MODI Method (continued)

    Calculate the reduced costs (circled numbers on the next slide) by c_{ij} - u_i + v_j.

Unoccupied Cell	Reduced Cost
(1,3)	40 - 0 - 32 = 8
(2,1)	30 - 24 -10 = -4
(2,4)	0 - 10 - 0 = -10

•**Since some of the reduced cost are negative, the current solution is not optimal.**

• **Cell (2,4) has the most negative; therefore, it will be the basic variable that must be occupied in the next iteration.**

Example: BBC

- Iteration 1 Tableau

	Northwood	Westwood	Eastwood	Dummy	u_i
Plant 1	24 25	30 5	40 +8	0 20	0
Plant 2	30 -4	40 40	42 10	0 -10	10
v_j	24	30	32	0	

Example: BBC

- Iteration 1
  - Stepping Stone Method

The stepping stone path for cell (2,4) is (2,4), (1,4), (1,2), (2,2). The allocations in the subtraction cells are 20 and 40, respectively. The minimum is 20, and hence reallocate 20 along this path. Thus for the next tableau:

x_{24} = 0 + 20 = 20 (0 is its current allocation)

x_{14} = 20 - 20 = 0 (blank for the next tableau)

x_{12} = 5 + 20 = 25

x_{22} = 40 - 20 = 20

The other occupied cells remain the same.

Example: BBC

	Northwood	Westwood	Eastwood	Dummy	Supply
Plant 1	24 25	30 25	40	0	50
Plant 2	30	40 20	42 10	0 20	50
Demand	25	45	10	20	

New quantity

Total transportation cost is $2570 = 2770 – 10 (20)

Reduced cost of cell (2,4)

Example: BBC

- Iteration 2
  - MODI Method

The reduced costs are found by calculating the u_i's and v_j's for this tableau.

1. Set $u_1 = 0$.
2. Since $u_1 + v_j = c_{ij}$ for occupied cells in row 1, then $v_1 = 24$, $v_2 = 30$.
3. Since $u_i + v_2 = c_{i2}$ for occupied cells in column 2, then $u_2 + 30 = 40$, or $u_2 = 10$.
4. Since $u_2 + v_j = c_{2j}$ for occupied cells in row 2, then $10 + v_3 = 42$ or $v_3 = 32$; and, $10 + v_4 = 0$ or $v_4 = -10$.

Example: BBC

- Iteration 2
  - MODI Method (continued)

    Calculate the reduced costs (circled numbers on the next slide) by c_{ij} - u_i + v_j.

Unoccupied Cell	Reduced Cost
(1,3)	40 - 0 - 32 = 8
(1,4)	0 - 0 - (-10) = 10
(2,1)	30 - 10 - 24 = -4

Since there is still negative reduced cost for cell (2,1), the solution is not optimal.

Cell (2,1) must be occupied

Example: BBC

- Iteration 2 Tableau

	Northwood	Westwood	Eastwood	Dummy	u_i
Plant 1	24 25	30 25	40 +8	0 +10	0
Plant 2	30 -4	40 20	42 10	0 20	10
v_j	24	30	36	-6	

Example: BBC

- Iteration 2
  - Stepping Stone Method

The most negative reduced cost is = -4 determined by x_{21}. The stepping stone path for this cell is (2,1),(1,1),(1,2),(2,2). The allocations in the subtraction cells are 25 and 20 respectively. Thus the new solution is obtained by reallocating 20 on the stepping stone path. Thus for the next tableau:

x_{21} = 0 + 20 = 20 (0 is its current allocation)

x_{11} = 25 - 20 = 5

x_{12} = 25 + 20 = 45

x_{22} = 20 - 20 = 0 (blank for the next tableau)

The other occupied cells remain the same.

Example: BBC

	Northwood	Westwood	Eastwood	Dummy	Supply
Plant 1	24 *5*	30 *45*	40	0	50
Plant 2	30 *20*	40	42 *10*	0 *20*	50
Demand	25	45	10	20	

Total cost is $2490 = 2570 - 4(20)

Example: BBC

- Iteration 3
  - MODI Method

The reduced costs are found by calculating the u_i's and v_j's for this tableau.

1. Set $u_1 = 0$
2. Since $u_1 + v_j = c_{1j}$ for occupied cells in row 1, then $v_1 = 24$ and $v_2 = 30$.
3. Since $u_i + v_1 = c_{i1}$ for occupied cells in column 2, then $u_2 + 24 = 30$ or $u_2 = 6$.
4. Since $u_2 + v_j = c_{2j}$ for occupied cells in row 2, then $6 + v_3 = 42$ or $v_3 = 36$, and $6 + v_4 = 0$ or $v_4 = -6$.

Example: BBC

- Iteration 3
  - MODI Method (continued)

    Calculate the reduced costs (circled numbers on the next slide) by c_{ij} - u_i + v_j.

Unoccupied Cell	Reduced Cost
(1,3)	40 - 0 - 36 = 4
(1,4)	0 - 0 - (-6) = 6
(2,2)	40 - 6 - 30 = 4

Since all the reduced cost are nonnegative, the current solution is optimal

Example: BBC

- Iteration 3 Tableau

  Since all the reduced costs are non-negative, this is the optimal tableau.

	Northwood	Westwood	Eastwood	Dummy	u_i
Plant 1	24 5	30 45	40 +4	0 +6	0
Plant 2	30 20	40 +4	42 10	0 20	6
v_j	24	30	36	-6	

Example: BBC

- Optimal Solution

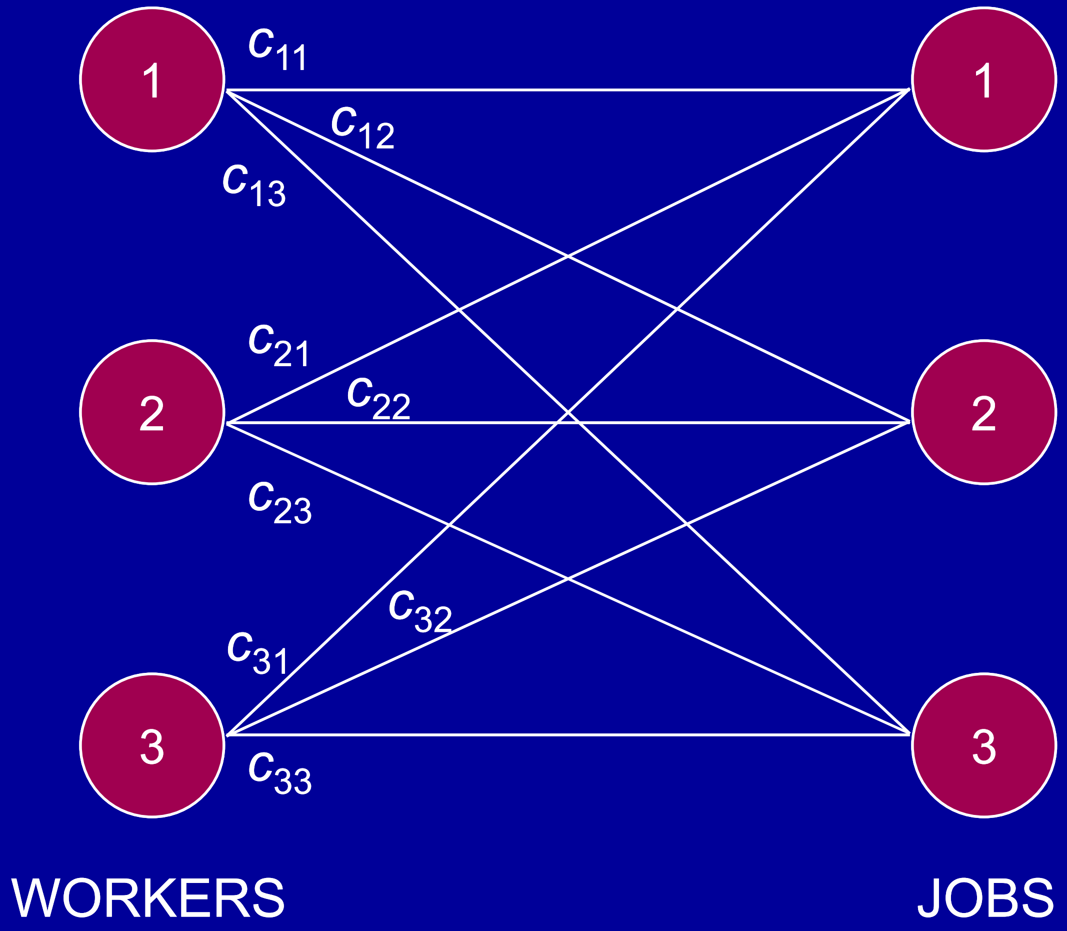
From	To	Amount	Cost
Plant 1	Northwood	5	120
Plant 1	Westwood	45	1,350
Plant 2	Northwood	20	600
Plant 2	Eastwood	10	420
		Total Cost =	$2,490

Assignment Problem

- An assignment problem seeks to minimize the total cost assignment of m workers to m jobs, given that the cost of worker i performing job j is c_{ij}.
- It assumes all workers are assigned and each job is performed.
- An assignment problem is a special case of a transportation problem in which all supplies and all demands are equal to 1; hence assignment problems may be solved as linear programs.
- The network representation of an assignment problem with three workers and three jobs is shown on the next slide.

Assignment Problem

- Network Representation



Assignment Problem

- Linear Programming Formulation

$$\text{Min} \quad \sum_{i}\sum_{j} c_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j} x_{ij} = 1 \qquad \text{for each worker } i$$

$$\sum_{i} x_{ij} = 1 \qquad \text{for each job } j$$

$$x_{ij} = 0 \text{ or } 1 \qquad \text{for all } i \text{ and } j.$$

- Note: A modification to the right-hand side of the first constraint set can be made if a worker is permitted to work more than 1 job.

Example: Hungry Owner

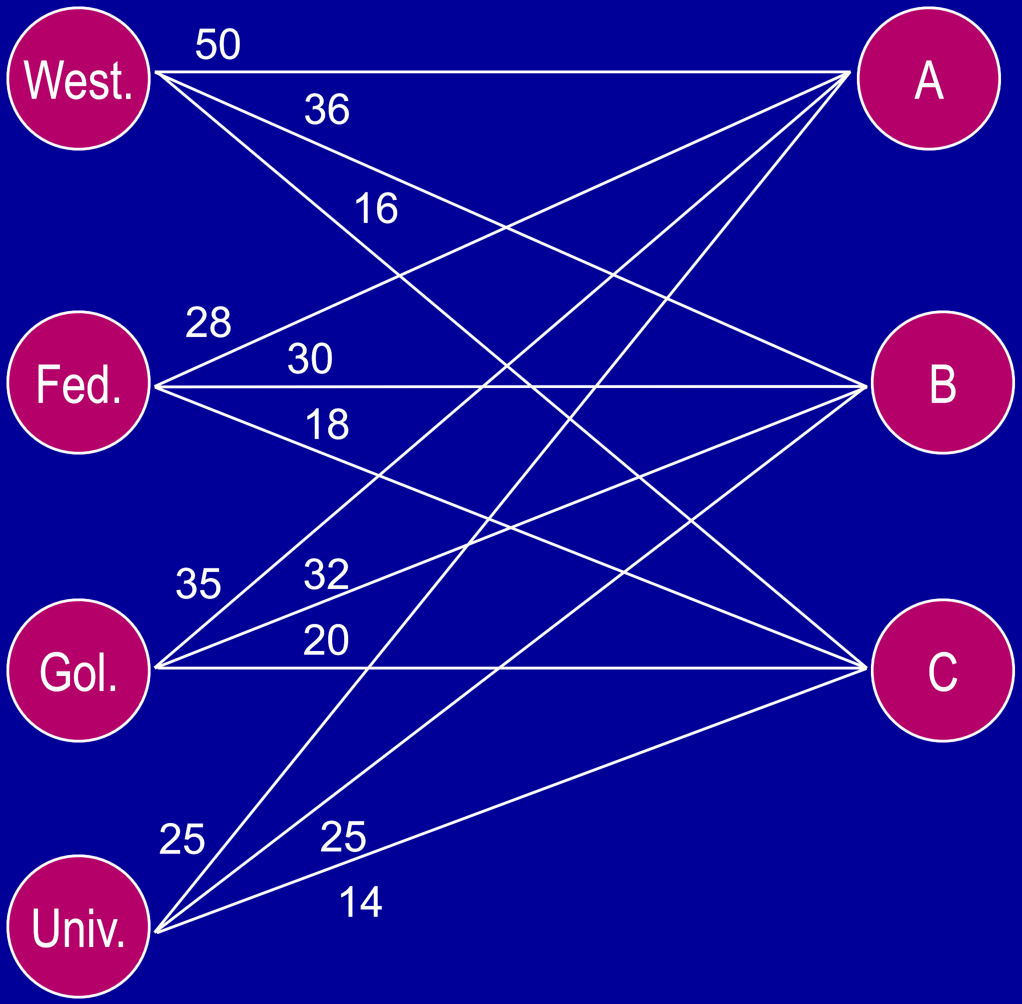
A contractor pays his subcontractors a fixed fee plus mileage for work performed. On a given day the contractor is faced with three electrical jobs associated with various projects. Given below are the distances between the subcontractors and the projects.

		Project		
		A	B	C
Subcontractors	Westside	50	36	16
	Federated	28	30	18
	Goliath	35	32	20
	Universal	25	25	14

How should the contractors be assigned to minimize total distance (and total cost)?

Example: Hungry Owner

- Network Representation



Example: Hungry Owner

- LP Formulation
  - Decision Variables Defined

x_{ij} = 1 if subcontractor i is assigned to project j

= 0 otherwise

where: i = 1 (Westside), 2 (Federated),
3 (Goliath), and 4 (Universal)

j = 1 (A), 2 (B), and 3 (C)

Example: Hungry Owner

- LP Formulation
  - Objective Function

    Minimize total distance:

    $$\text{Min } 50x_{11} + 36x_{12} + 16x_{13} + 28x_{21} + 30x_{22} + 18x_{23} + 35x_{31} + 32x_{32} + 20x_{33} + 25x_{41} + 25x_{42} + 14x_{43}$$

Example: Hungry Owner

- LP Formulation
  - Constraints

$$
\begin{aligned}
x_{11} + x_{12} + x_{13} &\le 1 && \text{(no more than one}\\
x_{21} + x_{22} + x_{23} &\le 1 && \text{project assigned}\\
x_{31} + x_{32} + x_{33} &\le 1 && \text{to any one}\\
x_{41} + x_{42} + x_{43} &\le 1 && \text{subcontractor)}\\
x_{11} + x_{21} + x_{31} + x_{41} &= 1 && \text{(each project must}\\
x_{12} + x_{22} + x_{32} + x_{42} &= 1 && \text{be assigned to just}\\
x_{13} + x_{23} + x_{33} + x_{43} &= 1 && \text{one subcontractor)}\\
\text{all } x_{ij} &\ge 0 && \text{(non-negativity)}
\end{aligned}
$$

Example: Hungry Owner

- Optimal Assignment

Subcontractor	Project	Distance
Westside	C	16
Federated	A	28
Universal	B	25
Goliath	(unassigned)	

Total Distance = 69 miles

Variations of Assignment Problem

- Total number of agents not equal to total number of tasks
- Maximization objective function
- Unacceptable assignments

Hungarian Method

- The Hungarian method solves minimization assignment problems with m workers and m jobs.
- Special considerations can include:
  - number of workers does not equal the number of jobs -- add dummy workers or jobs with 0 assignment costs as needed
  - worker i cannot do job j -- assign $c_{ij} = +M$
  - maximization objective -- create an opportunity loss matrix subtracting all profits for each job from the maximum profit for that job before beginning the Hungarian method

Hungarian Method

- Step 1: For each row, subtract the minimum number in that row from all numbers in that row.
- Step 2: For each column, subtract the minimum number in that column from all numbers in that column.
- Step 3: Draw the minimum number of lines to cover all zeroes. If this number = *m*, STOP -- an assignment can be made.
- Step 4: Determine the minimum uncovered number (call it *d*).
    - Subtract *d* from uncovered numbers.
    - Add *d* to numbers covered by two lines.
    - Numbers covered by one line remain the same.
    - Then, GO TO STEP 3.

Hungarian Method

- Finding the Minimum Number of Lines and Determining the Optimal Solution
  - Step 1: Find a row or column with only one unlined zero and circle it. (If all rows/columns have two or more unlined zeroes choose an arbitrary zero.)
  - Step 2: If the circle is in a row with one zero, draw a line through its column. If the circle is in a column with one zero, draw a line through its row. One approach, when all rows and columns have two or more zeroes, is to draw a line through one with the most zeroes, breaking ties arbitrarily.
  - Step 3: Repeat step 2 until all circles are lined. If this minimum number of lines equals *m*, the circles provide the optimal assignment.

Example: Hungry Owner

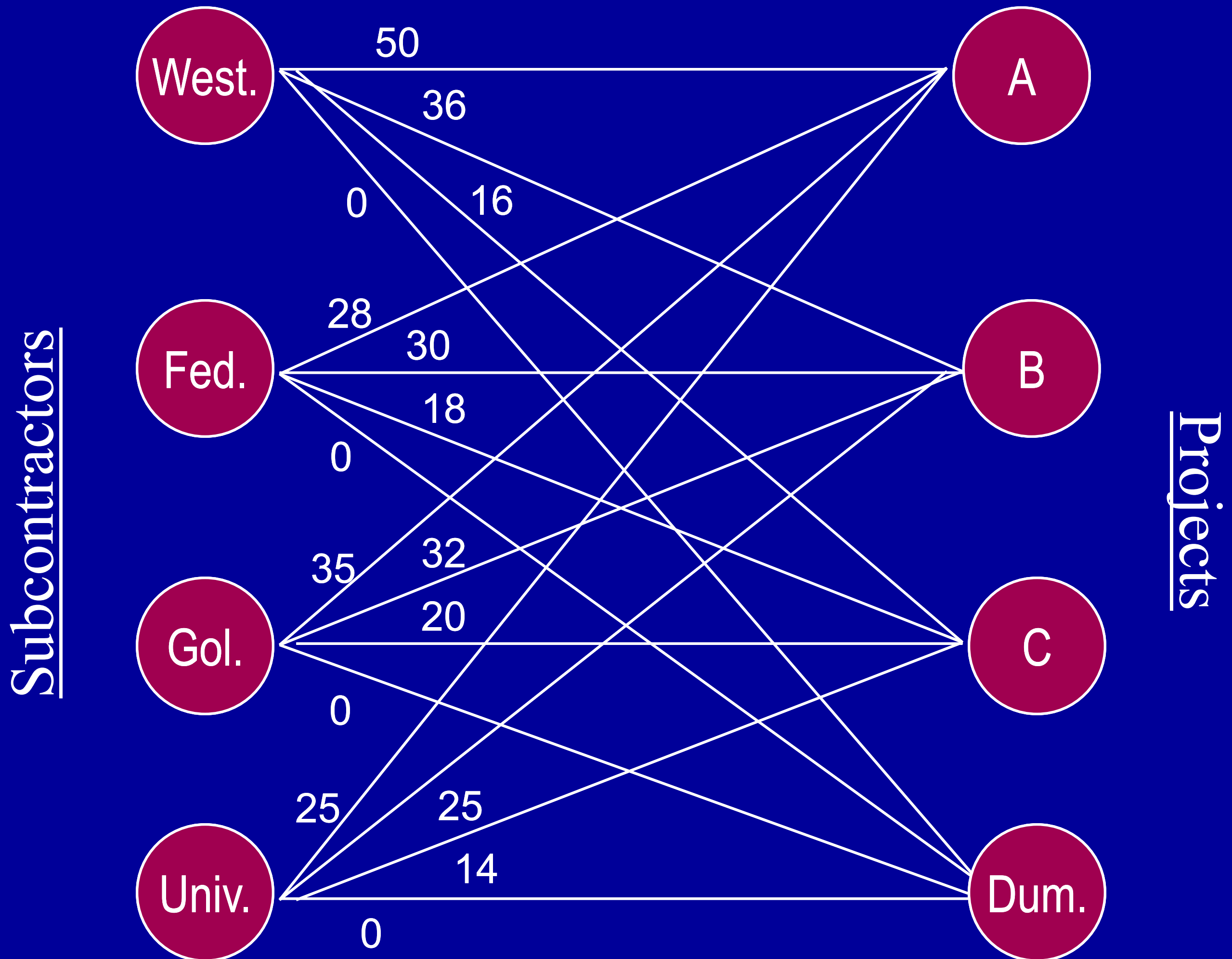
A contractor pays his subcontractors a fixed fee plus mileage for work performed. On a given day the contractor is faced with three electrical jobs associated with various projects. Given below are the distances between the subcontractors and the projects.

		Projects		
		A	B	C
Subcontractors	Westside	50	36	16
	Federated	28	30	18
	Goliath	35	32	20
	Universal	25	25	14

How should the contractors be assigned to minimize total costs?

Example: Hungry Owner

- Network Representation (note the dummy project)



Example: Hungry Owner

- Initial Tableau Setup

  Since the Hungarian algorithm requires that there be the same number of rows as columns, add a Dummy column so that the first tableau is:

	A	B	C	Dummy
Westside	50	36	16	0
Federated	28	30	18	0
Goliath	35	32	20	0
Universal	25	25	14	0

Example: Hungry Owner

- Step 1: Subtract minimum number in each row from all numbers in that row. Since each row has a zero, we would simply generate the same matrix above.
- Step 2: Subtract the minimum number in each column from all numbers in the column. For A it is 25, for B it is 25, for C it is 14, for Dummy it is 0. This yields:

	A	B	C	Dummy
Westside	25	11	2	0
Federated	3	5	4	0
Goliath	10	7	6	0
Universal	0	0	0	0

Example: Hungry Owner

- Step 3: Draw the minimum number of lines to cover all zeroes. Although one can "eyeball" this minimum, use the following algorithm. If a "remaining" row has only one zero, draw a line through the column. If a remaining column has only one zero in it, draw a line through the row.

	A	B	C	Dummy
Westside	25	11	2	0
Federated	3	5	4	0
Goliath	10	7	6	0
Universal	0	0	0	0

- Step 4: The minimum uncovered number is 2 (circled).

Example: Hungry Owner

- Step 5: Subtract 2 from uncovered numbers; add 2 to all numbers covered by two lines. This gives:

	A	B	C	Dummy
Westside	23	9	0	0
Federated	1	3	2	0
Goliath	8	5	4	0
Universal	0	0	0	2

Example: Hungry Owner

- Step 3: Draw the minimum number of lines to cover all zeroes.

	A	B	C	Dummy
Westside	~~23~~	~~9~~	~~0~~	~~0~~
Federated	①	3	2	~~0~~
Goliath	8	5	4	~~0~~
Universal	~~0~~	~~0~~	~~0~~	~~2~~

- Step 4: The minimum uncovered number is 1 (circled).

Example: Hungry Owner

- Step 5: Subtract 1 from uncovered numbers. Add 1 to numbers covered by two lines. This gives:

	A	B	C	Dummy
Westside	23	9	0	1
Federated	0	2	1	0
Goliath	7	4	3	0
Universal	0	0	0	3

Example: Hungry Owner

- Step 4: The minimum number of lines to cover all 0's is four. Thus, there is a minimum-cost assignment of 0's with this tableau. The optimal assignment is:

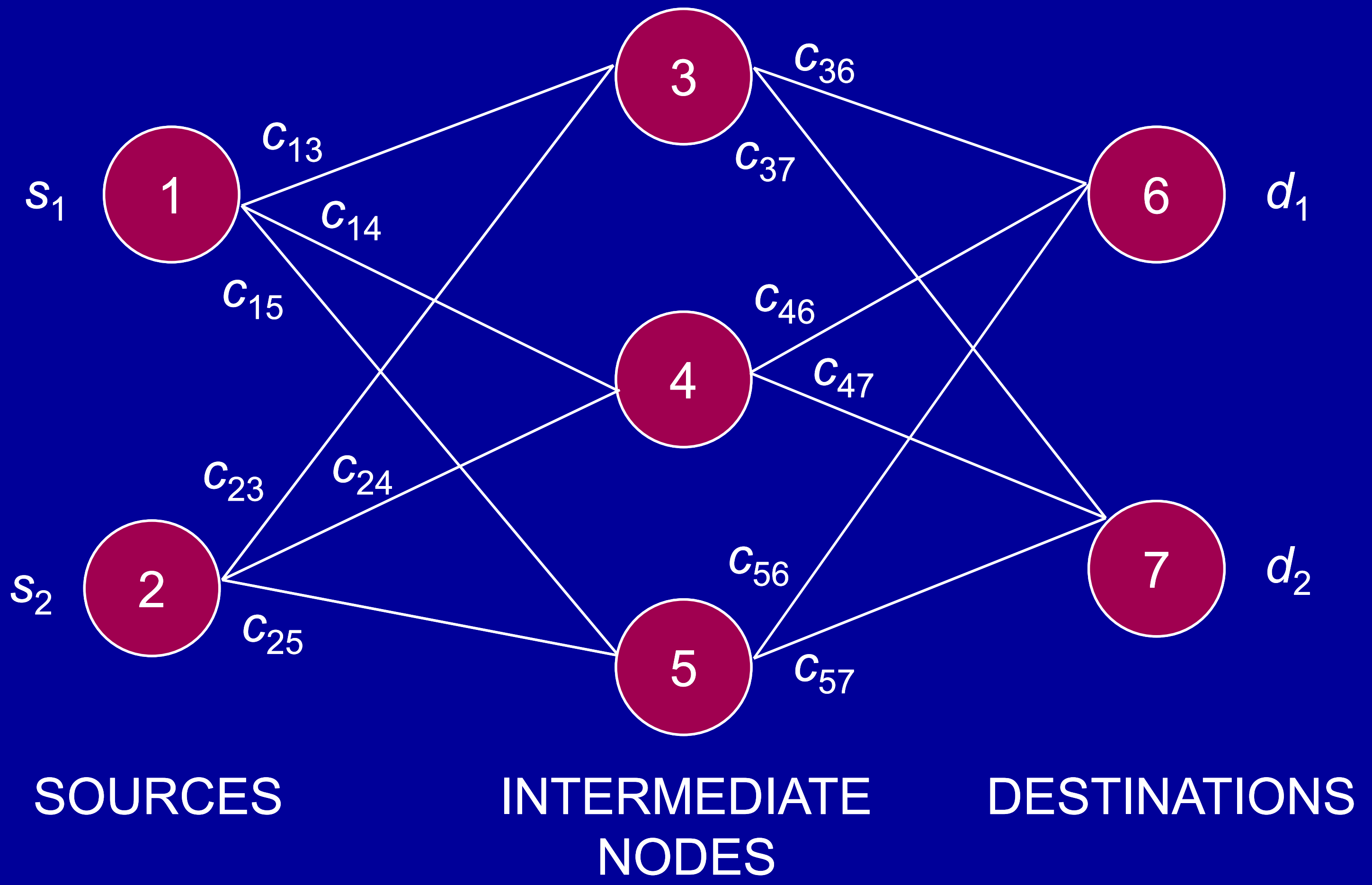
Subcontractor	Project	Distance
Westside	C	16
Federated	A	28
Goliath	(unassigned)	
Universal	B	25
	Total Distance =	69 miles

Transshipment Problem

- Transshipment problems are transportation problems in which a shipment may move through intermediate nodes (transshipment nodes)before reaching a particular destination node.
- Transshipment problems can be converted to larger transportation problems and solved by a special transportation program.
- Transshipment problems can also be solved by general purpose linear programming codes.
- The network representation for a transshipment problem with two sources, three intermediate nodes, and two destinations is shown on the next slide.

Transshipment Problem

- Network Representation



"Transshipment points"

Transshipment Problem

- Linear Programming Formulation

x_{ij} represents the shipment from node *i* to node *j*

$$\text{Min} \quad \sum_i \sum_j c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_j x_{ij} \leq s_i \qquad \text{for each source } i$$

$$\sum_i x_{ik} - \sum_j x_{kj} = 0 \qquad \text{for each intermediate node } k$$

$$\sum_i x_{ij} = d_j \qquad \text{for each destination } j$$

$$x_{\text{ij}} \geq 0 \qquad \text{for all } i \text{ and } j$$

Example: Transshipping

Thomas Industries and Washburn Corporation supply three firms (Zrox, Hewes, Rockwright) with customized shelving for its offices. They both order shelving from the same two manufacturers, Arnold Manufacturers and Supershelf, Inc.

Currently weekly demands by the users are 50 for Zrox, 60 for Hewes, and 40 for Rockwright. Both Arnold and Supershelf can supply at most 75 units to its customers.

Additional data is shown on the next slide.

Example: Transshipping

Because of long standing contracts based on past orders, unit costs from the manufacturers to the suppliers are:

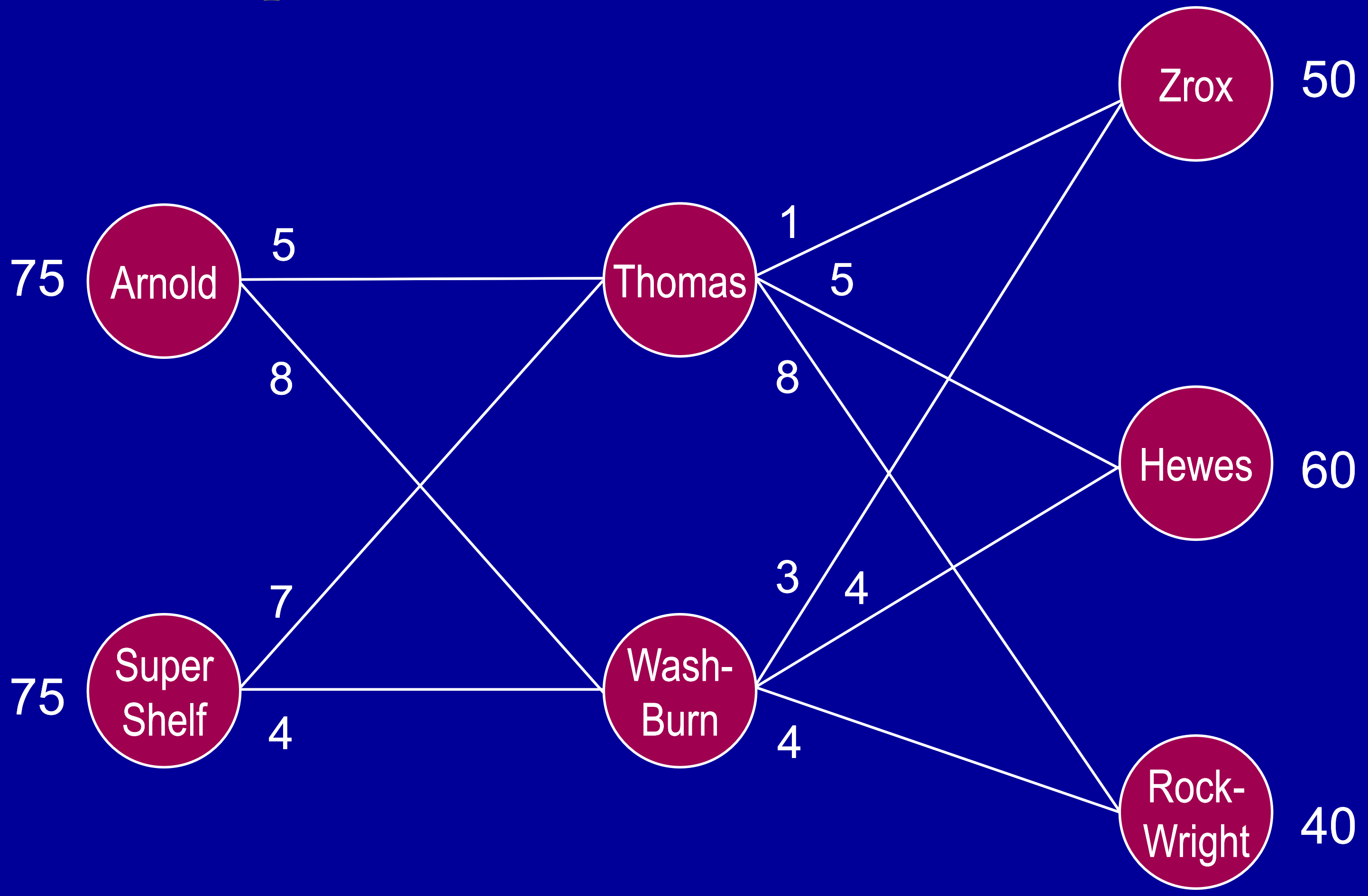
	Thomas	Washburn
Arnold	5	8
Supershelf	7	4

The cost to install the shelving at the various locations are:

	Zrox	Hewes	Rockwright
Thomas	1	5	8
Washburn	3	4	4

Example: Transshipping

- Network Representation



Example: Transshipping

- Linear Programming Formulation
  - Decision Variables Defined

x_{ij} = amount shipped from manufacturer i to supplier j

x_{jk} = amount shipped from supplier j to customer k

where i = 1 (Arnold), 2 (Supershelf)

j = 3 (Thomas), 4 (Washburn)

k = 5 (Zrox), 6 (Hewes), 7 (Rockwright)

  - Objective Function Defined

Minimize Overall Shipping Costs:

$$\text{Min} \quad 5x_{13} + 8x_{14} + 7x_{23} + 4x_{24} + 1x_{35} + 5x_{36} + 8x_{37} + 3x_{45} + 4x_{46} + 4x_{47}$$

Example: Transshipping

- Constraints Defined

Amount Out of Arnold:	$x_{13} + x_{14} \leq 75$
Amount Out of Supershelf:	$x_{23} + x_{24} \leq 75$
Amount Through Thomas:	$x_{13} + x_{23} - x_{35} - x_{36} - x_{37} = 0$
Amount Through Washburn:	$x_{14} + x_{24} - x_{45} - x_{46} - x_{47} = 0$
Amount Into Zrox:	$x_{35} + x_{45} = 50$
Amount Into Hewes:	$x_{36} + x_{46} = 60$
Amount Into Rockwright:	$x_{37} + x_{47} = 40$

Non-negativity of Variables: $x_{ij} \geq 0$, for all *i* and *j*.

Solving the transshipment problem using transportation algorithm

- Supply point: is a point that can send goods to another point but can not receive goods from any other point
- Demand point: is a point that can receive goods form other points but cannot send goods to any other point.
- Transshipment point: is a point that can both receive goods from other points and send goods to other points

Solving the transshipment problem using transportation algorithm

- a transshipment problem can be transformed to a balanced transportation problem by using the following procedure:
  - Step 1: if necessary, add a dummy demand point or a dummy supply point as needed
  - Step 2: construct a transportation tableau as follows:
    - A row in the tableau will be needed for each supply point and transshipment point.
    - A column will be needed for each demand point and transshipment point
    - Each supply point will have a supply equal to its original supply, and each demand point will have a demand equals to its original demand.
    - each transshipment point will have

      supply = its original supply + total available supply, and

      demand = its original demand + total available supply
- Then the problem can be solved as a transportation problem

Example: transshipping

- The balanced transportation tableau for the transshipment problem will be as follows:

	Zrox	Hewes	Rockwright	Thomas	washburn	**Supply**
Arnold	M	M	M	5	8	**75**
Supper shell	M	M	M	7	4	**75**
Thomas	1	5	8	0	M	**150**
Washburn	3	4	4	M	0	**150**
Demand	**50**	**60**	**40**	**150**	**150**	**Total = 450**

- ***Where M is a very large number and used when the route is not valid***

- ***Zero cost is used from a destination to itself.***

- ***There is no need for dummy row or column, since the problem is balanced***