

# Transportation & Assignment Problems

---

Taek-Ho Lee

Department of Industrial Engineering, SeoulTech

Mail: [taekho.lee@seoultech.ac.kr](mailto:taekho.lee@seoultech.ac.kr)

# Introduction

- Transportation Problem
  - A problem concerning “how to transport goods in an optimal way”
  - The main application areas of transportation problems are not necessarily limited to transportation itself.
- Assignment Problem
  - A problem concerning “how to assign people (assignee) to one of several tasks”
  - This is a special case of the transportation problem.
- Characteristics of transportation and assignment problems
  - Both problems belong to LP.
  - They typically involve a very large number of constraints and variables, which makes the computational burden of the Simplex Method increase significantly.
  - However, due to their structural nature, most of the constraint coefficients  $a_{ij}$  are 0, and the relatively small number of nonzero coefficients appear with a special pattern.
  - By exploiting these structural characteristics, efficient solution algorithms have been developed.

# Transportation Problems

# Transportation Problem

- Prototype Example

- We want to transport cans produced at three canneries to four warehouses.
  - ✓ Canneries: (1) Bellingham, (2) Eugene, and (3) Albert Lea
  - ✓ Warehouses: (1) Sacramento, (2) Salt Lake City, (3) Rapid City, and (4) Albuquerque
- The primary expense involved is the shipping cost.
- For decision-making, we have forecasted production at each cannery and forecasted demand at each warehouse for the upcoming season.

Shipping cost		Warehouse				Output
		1	2	3	4	
Cannery	1	464	513	654	867	75
	2	352	416	690	791	125
	3	995	682	388	685	100
Allocation		80	65	70	85	300

- ✓ The objective is to determine a shipping plan for each (cannery-warehouse) pair that minimizes total transportation cost.

# Transportation Problem

- LP Formulation

- Minimize Total Shipping Cost  $Z$

- $$Z = 464 x_{11} + 513 x_{12} + 654 x_{13} + 867 x_{14} + 352 x_{21} + 416 x_{22} + 690 x_{23} + 791 x_{24} + 995 x_{31} + 682 x_{32} + 388 x_{33} + 685 x_{34}$$

- Cannery constraints

- $$\begin{aligned} - x_{11} + x_{12} + x_{13} + x_{14} &= 75 \\ - x_{21} + x_{22} + x_{23} + x_{24} &= 125 \\ - x_{31} + x_{32} + x_{33} + x_{34} &= 100 \end{aligned}$$

- Warehouse constraints

- $$\begin{aligned} - x_{11} + x_{21} + x_{31} &= 80 \\ - x_{12} + x_{22} + x_{32} &= 65 \\ - x_{13} + x_{23} + x_{33} &= 70 \\ - x_{14} + x_{24} + x_{34} &= 85 \end{aligned}$$

- $x_{ij}$ : quantity shipped from cannery  $i$  to warehouse  $j$ , where  $i \in \{1, 2, 3\}$ ,  $j \in \{1, 2, 3, 4\}$ .

# Transportation Problem

- Assumptions about Supply and Demand in Transportation Problem
  - Each source has a fixed supply  $s_i$ , which must be distributed among several destinations.
  - Each destination has a fixed demand  $d_j$ , which must be satisfied by shipments from multiple sources.
  - There should be no surplus or shortage in the overall system, i.e.,
    - ✓  $\sum_{i=1}^m s_i = \sum_{j=1}^n d_j$
  - Assumption about unit shipping cost
    - ✓ The shipping cost from a source to a destination is proportional to the amount shipped (the proportionality assumption in LP).
    - ✓ Therefore, the shipping cost from source  $i$  to destination  $j$  is expressed as:  $c_{ij} \times x_{ij}$
    - ✓ Here,  $c_{ij}$  is unit cost of shipping one unit from source  $i$  to destination  $j$  and  $x_{ij}$  is a quantity shipped along the route  $i \rightarrow j$ .

# Transportation Problem

- Information for the Transportation Model
  - Objective: Minimize the total shipping cost  $Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$ 
    - ✓ Constraints
      - $\sum_{j=1}^n x_{ij} = s_i \ (\forall i = 1, \dots, m)$
      - $\sum_{i=1}^m x_{ij} = d_j \ (\forall j = 1, \dots, n)$
      - $x_{ij} \geq 0 \ \forall i, j$
  - Parameter table
    - ✓ (1) supply at each source, (2) demand at each destination, (3) unit shipping cost for all pairs

		Destination				Supply
		1	2	...	$n$	
Sources	1	$c_{11}$	$c_{12}$	...	$c_{1n}$	$s_1$
	2	$c_{21}$	$c_{22}$	...	$c_{2n}$	$s_2$
	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
	$m$	$c_{m1}$	$c_{m2}$	...	$c_{mn}$	$s_m$
Demand		$d_1$	$d_2$	...	$d_n$	$\sum_{i=1}^m s_i = \sum_{j=1}^n d_j$

# | Transportation Problem

- Integer Solution Property of the Transportation Problem
  - If all supply values ( $s_i$ ) and demand values ( $d_j$ ) are integers, then every BFS, including the optimal solution, will also have all BVs as integers.

# Transportation Problem

- Example with a Dummy Destination
  - Machine Production Scheduling (over four months)
    - ✓  $x_j$ : The number of machines to be produced in month  $j$

Month	Scheduled Installations	Maximum Production	Unit Cost of Production	Unit Cost of Storage
1	10	25	1.08	0.015
2	15	35	1.11	0.015
3	25	30	1.10	0.015
4	20	10	1.13	

# Transportation Problem

## ▪ Example with a Dummy Destination

### • Machine production scheduling (4 months)

- ✓ Source  $i$ : Machine production in month  $i$
- ✓ Destination  $j$ : Machine installation in month  $j$
- ✓  $x_{ij}$ : The number of machines produced in month  $i$  to be installed in month  $j$ 
  - Constraint:  $x_{ij} = 0$  if  $i > j$  (cannot produce later for an earlier installation)
- ✓  $c_{ij}$ :  $x_{ij}$ 
  - If  $i = j$ , unit production cost in month  $j$ . If  $i < j$ , unit production + storage cost until month  $j$
- ✓  $s_i$ : Given as an upper bound, not a fixed supply → introduce a dummy destination to absorb unused production ( $c_{ij} = 0$  for dummy)
  - $x_{11} + x_{12} + x_{13} + x_{14} \leq 25$
  - $x_{21} + x_{22} + x_{23} + x_{24} \leq 35$
  - $x_{31} + x_{32} + x_{33} + x_{34} \leq 30$
  - $x_{41} + x_{42} + x_{43} + x_{44} \leq 10$
- ✓  $d_j$ : scheduled number of machine installation in month  $j$

		Destination					Supply
		1	2	3	4	5(D)	
Source	1	1.080	1.095	1.110	1.125	0	25
	2	M	1.110	1.125	1.140	0	35
	3	M	M	1.100	1.115	0	30
	4	M	M	M	1.130	0	10
Demand		10	15	25	20	30	100

# Transportation Problem

- Example with a Dummy Source

- There are three plants (A, B, C) and four cities (1, 2, 3, 4)
- The supply capacities of the plants are 50, 60, 50, and the demands of the cities are 50, 70, 30, 60.
  - ✓ Thus, total supply is 160 total demand is 210, so supply is insufficient.
- Assume the penalty (loss) cost for unsatisfied demand is the same regardless of the city.
- The objective is to minimize total cost in distributing the available supply to the demands.
- As indicated in the parameter table, shipping from plant C to city 4 is assumed to be impossible.

		Destination				Supply
		1	2	3	4	
Source	A	16	13	22	17	50
	B	14	13	19	15	60
	C	19	20	23	M	50
Demand		50	70	30	60	

# Transportation Problem

- Example with a Dummy Source

- We assume that the penalty cost for unsatisfied demand is the same across all cities.
  - ✓ Therefore, we do not need to consider which city's demand is left unsatisfied, and only the shipping cost matters in this example.
- To balance total supply and demand, we introduce a dummy source.
  - ✓ Shipments from the dummy source to a city do not represent actual deliveries but instead indicate the shortfall of demand at that city.
  - ✓ If necessary, different penalty costs can be assigned to the dummy source-city pairs to reflect differences in the loss of unsatisfied demand across cities.

		Destination				Supply
		1	2	3	4	
Source	A	16	13	22	17	50
	B	14	13	19	15	60
	C	19	20	23	M	50
	D	M	M	M	M	50
Demand		50	70	30	60	210

# Transportation Simplex Method

- (General) Simplex Method
  - Simplex tableau (with artificial variables introduced)
    - ✓  $z_i$  and  $z_{m+j}$  are artificial variables.

BV	Z	...	$x_{ij}$	...	$z_i$	...	$z_{m+j}$	...	RHS
Z	-1		$c_{ij}$		M		M		0
⋮	⋮								
$z_i$	0		1		1				$s_i$
⋮	⋮								
$z_{m+j}$	0		1				1		$d_j$
⋮	⋮								

(at any iteration)    Z    -1     $c_{ij} - u_i - v_j$      $M - u_i$      $M - v_j$      $-\sum_{i=1}^m s_i u_i - \sum_{j=1}^n d_j v_j$

# Transportation Simplex Method

- (General) Simplex Method
  - Simplex tableau (with artificial variables introduced)
    - ✓ For a NBV  $x_{ij}$ 
      - $u_i$ : the multiple of row  $i$  that has been subtracted from row 0 to form the current simplex tableau.
      - $v_j$ : the multiple of row  $m + j$  that has been subtracted from row 0 to form the current simplex tableau.
    - ✓ For a BV  $x_{ij}$ :  $c_{ij} - u_i - v_j = 0$

BV	Z	...	$x_{ij}$	...	$z_i$	...	$z_{m+j}$	...	RHS
Z	-1		$c_{ij}$		M		M		0
⋮	⋮								
$z_i$	0		1		1				$s_i$
⋮	⋮								
$z_{m+j}$	0		1				1		$d_j$
⋮	⋮								

(at any iteration)

Z	-1	$c_{ij} - u_i - v_j$	$M - u_i$	$M - v_j$	$-\sum_{i=1}^m s_i u_i - \sum_{j=1}^n d_j v_j$
---	----	----------------------	-----------	-----------	--

# Transportation Simplex Method

- (General) Simplex Method
  - Initialization
    - ✓ Objective: To obtain an initial BFS that serves as the starting point for the Simplex method
    - ✓ In a general LP, the number of BVs is  $m+n$ , but in the transportation problem, the number of BVs is only  $m+n-1$ .
      - This is because, under the balance condition (total supply = total demand), the final equation is automatically satisfied.
    - ✓ To construct an initial BFS, select  $m+n-1$  variables and assign values so that each chosen variable satisfies one of the constraints.

# Transportation Simplex Method

- General Initialization
  - 1. From the current row/column under consideration, select an entering BV according to a chosen rule.
  - 2. Assign as much as possible so that either the remaining supply of that row or the remaining demand of that column is exhausted.
    - ✓ Consume the smaller of the two (remaining supply or demand).
  - 3. Remove the row/column that has been exhausted from further consideration (if both are exhausted simultaneously, remove the row first.)
  - 4. When only one row or column remains, allocate the remaining feasible assignments, select all associated variables, and terminate the initialization process.

# Transportation Simplex Method

- Transportation Simplex Method
  - Northwest Corner Method
    - ✓ Start with  $x_{11}$ .
    - ✓ If the current source  $i$  still has remaining supply, then proceed to select  $x_{i,j+1}$ .
    - ✓ Otherwise, move down and select  $x_{i+1,j}$ .

# | Transportation Simplex Method

- Transportation Simplex Method
  - Vogel's Approximation Method
    - ✓ For each row and each column, calculate the difference between the smallest unit shipping cost  $c_{ij}$  and the next smallest  $c_{ij}$ .
      - If they are the same, the difference is taken as 0.
    - ✓ Select the row or column with the largest difference.
      - Within that row or column, choose the variable  $x_{ij}$  corresponding to the smallest  $c_{ij}$ .
      - If there is a tie (multiple choices with the same value), select the one randomly.

# Transportation Simplex Method

- Transportation Simplex Method
  - 러셀의 근사법(Russel's Approximation Method)
    - ✓ 남은 source  $i$ 에 대해  $\bar{u}_i$  (남아있는 가장 큰 단위비용  $c_{ij}$ )
    - ✓ 남은 destination  $j$ 에 대해  $\bar{v}_j$  (남아있는 가장 큰 단위비용  $c_{ij}$ )
    - ✓ 아직 선택되지 않은  $x_{ij}$ 에 대해  $\Delta_{ij} = c_{ij} - \bar{u}_i - \bar{v}_j$  계산
    - ✓ 가장 작은 음수  $\Delta_{ij}$  선택(같은 값은 임의 선택)

# Transportation Simplex Method

- Transportation Simplex Method
  - Comparison of three initialization alternatives
    - ✓ Northwest Corner Method
      - Fast and simple
      - Does not consider the shipping cost  $c_{ij}$ .
      - Likely to produce an initial BFS that is far from optimal, which may increase the number of iterations needed later.
    - ✓ Vogel's Approximation Method
      - Generally requires fewer iterations than the Northwest Corner Method.
      - Still relatively easy to apply by hand.
    - ✓ Russel's Approximation Method
      - Tends to find even better initial BFS compared to Vogel's method.
      - More complex to compute manually, but well-suited for computer-based implementation.

# Transportation Simplex Method

- Transportation Simplex Method
  - Optimality Test
    - ✓ A BFS is optimal if and only if for every NBV  $x_{ij}$ ,  $c_{ij} - u_i - v_j \geq 0$  holds.
      - To perform this test, derive the values of  $u_i$  and  $v_j$  for the current BFS and compute  $c_{ij} - u_i - v_j$ .
      - To determine  $u_i, v_j$ , use the equality  $c_{ij} - u_i - v_j = 0$  that must hold for every BV  $x_{ij}$ .
    - ✓ Since there are  $m + n - 1$  BVs and  $m + n$  unknowns ( $m$  values of  $u_i$  and  $n$  values of  $v_j$ ), one of them can be assigned arbitrarily.
      - A convenient choice is to set  $u_i = 0$  for the row with the most allocations.
  - Example: Performing optimality test starting from Russel's initial solution:

# Transportation Simplex Method

- Transportation Simplex Method

- Iterations

- ✓ 1. Determining the entering BV: For an NBV  $x_{ij}$ ,  $c_{ij} - u_i - v_j$  indicates the change in the objective function  $Z$  if  $x_{ij}$  increases.
      - To reduce the total cost  $Z$ , choose the  $x_{ij}$  with the most negative value  $c_{ij} - u_i - v_j$ .
    - ✓ 2. Determining the leaving BV: Increasing the chosen NBV triggers a chain reaction due to the constraints.
      - Since this chain reaction occurs by the same amount throughout, the leaving BV is the donor cell  $x_{ij}$  with the smallest allocation.
    - ✓ 3. Find the new BFS
      - Update allocations by adding and subtracting the leaving BV's value along the chain (recipient/donor cells).
      - The total cost change is  $\Delta Z = (\text{value of the leaving BV}) \times (c_{ij} - u_i - v_j \text{ of the entering BV})$

# Transportation Simplex Method

- Transportation Simplex Method
  - Initialization: Use one of the initialization methods such as the Northwest Corner Method, Vogel's Approximation method, Russel's Approximation method.
  - Optimality Test:
    - ✓ Set  $u_i = 0$  for the row with the largest number of allocations.
    - ✓ For each BV  $x_{ij}$ , use the condition  $c_{ij} - u_i - v_j = 0$  to recursively determine the values of  $u_i, v_j$ .
    - ✓ For each NBV  $x_{ij}$ , if  $c_{ij} - u_i - v_j \geq 0$ , the solution is optimal. If any negative values exist, proceed to iterations.
  - Iterations:
    - ✓ Determine the entering BV: Choose the NBV  $x_{ij}$  with the most negative value of  $c_{ij} - u_i - v_j$ .
    - ✓ Determine the leaving BV: Increasing the entering BV triggers a chain reaction. Identify the donor cells in this cycle, and select the BV with the smallest allocation as the leaving BV.
    - ✓ Compute the new BFS: Adjust allocations by adding/subtracting the leaving BV's value across recipient/donor cells.

# Transportation Simplex Method

- Transportation Simplex Method
  - Special characteristics of the example
    - ✓ 1. Since the BV  $x_{31} = 0$ , the initial BFS is degeneracy. In this example,  $x_{31}$  increases in a later iterations.
    - ✓ 2. Both  $x_{21}, x_{34}$  simultaneously reach 0; therefore, one of them is arbitrarily chosen as the leaving BV, while the other becomes degeneracy.
    - ✓ 3. If  $c_{ij} - u_i - v_j \geq 0$ , the solution is optimal. However, due to (2), two alternative optimal solutions occur.

# | Transportation Simplex Method

- Transportation Simplex Method
  - Solve the example to the optimal point:

# Assignment Problems

# Assignment Problem

- Overview of the Assignment Problem
  - An LP problem where assignees are assigned to tasks.
  - Characteristics
    - ✓ 1. The number of assignees equals the number of tasks. ( $n$ )
    - ✓ 2. Each assignee must be assigned to exactly one task.
    - ✓ 3. Each task must be performed by exactly one assignee.
    - ✓ 4. The cost  $c_{ij}$  is given for assignee  $i$  performing task  $j$ .
    - ✓ 5. The objective is to minimize the total cost.

# Assignment Problem

- Prototype Example

- There are three machines (1, 2, 3) and four locations (1, 2, 3, 4) available for installation.
  - ✓ Constraint: Machine 2 cannot be installed at Location 2.

		Location			
		1	2	3	4
Machine	1	13	16	12	11
	2	15	-	13	20
	3	5	7	10	6

# Assignment Problem

- Prototype Example

- There are three machines (1, 2, 3) and four possible locations (1, 2, 3, 4).
  - ✓ However, machine 2 cannot be installed at location 2. This restriction is modeled using the Big-M method (assigning a very large penalty cost).
- Just as in the transportation problem, the balance condition requires total supply = total demand, in the assignment problem (a special case of the transportation problem) the number of assignees and the number of tasks must be equal.

		Location			
		1	2	3	4
Machine	1	13	16	12	11
	2	15	M	13	20
	3	5	7	10	6
	4(D)	0	0	0	0

# Assignment Problem

- Mathematical Model

- $x_{ij} = 1$  if assignee  $i$  performs task  $j$ ; otherwise,  $x_{ij} = 0$ .
  - ✓ With the balance assumption (same number of assignees and tasks),  $i \in \{1, 2, \dots, n\}$ ,  $j \in \{1, 2, \dots, n\}$
- Objective: Minimize  $Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$ 
  - ✓ Constraints:
    - $\sum_{j=1}^n x_{ij} = 1, \quad \forall i \in \{1, 2, \dots, n\}$
    - $\sum_{i=1}^n x_{ij} = 1, \quad \forall j \in \{1, 2, \dots, n\}$
    - $x_{ij} \geq 0$  (Binary variables)  $\forall i, j \in \{1, 2, \dots, n\}$
- Relation to the transportation model (special case)
  - ✓ Interpretable as a transportation problem with # sources = # destinations =  $n$ , where  $s_i = 1$  for all  $i = 1, \dots, n$  &  $d_j = 1$  for all  $j = 1, \dots, n$ .

# Assignment Problem

- Solution Methods for the Assignment Problem
  - General Simplex Method (without explicit integer constraints)
  - Transportation Simplex Method
- Applying the Transportation Simplex Method
  - The number of BVs is  $m + n - 1 \rightarrow$  in the assignment, this becomes  $2n - 1$ .
    - ✓ However, in the assignment problem, exactly  $n$  variables take the value 1, and the remaining  $n - 1$  BVs are degenerate (value 0).
  - Therefore, the transportation simplex method does not fully exploit the special structure of the assignment problem.

# Assignment Problem

- Approach via Transportation/Assignment Models
  - We need to produce four products (1, 2, 3, 4) using three plants (1, 2, 3).
    - ✓ Constraint: plant 2 cannot produce product 3.

Unit production cost		Product				Supply
		1	2	3	4	
Source	1	41	27	28	24	75
	2	40	29	-	23	75
	3	37	30	27	21	45
Demand		20	30	30	40	

# Assignment Problem

- Approach via Transportation/Assignment Models
  - We need to produce four products (1, 2, 3, 4) using three plants (1, 2, 3).
    - ✓ Constraint: plant 2 cannot produce product 3.

Unit production cost		Product				Supply
		1	2	3	4	
Source	1	41	27	28	24	75
	2	40	29	-	23	75
	3	37	30	27	21	45
Demand		20	30	30	40	

- ✓ If the production of a single product can be split across multiple plants, model it as a transportation problem.
  - Optimal solution in transportation formulation:  
 $x_{12} = 30, x_{13} = 30, x_{15} = 15, x_{24} = 15, x_{25} = 60, x_{31} = 20, x_{34} = 25 \rightarrow Z = 3260$

# Assignment Problem

- Approach via Transportation/Assignment Models
  - We need to produce four products (1, 2, 3, 4) using three plants (1, 2, 3).
    - ✓ Constraint: plant 2 cannot produce product 3.

단위 생산 비용		Product				
		1	2	3	4	5(D)
Assignee	1a	820	810	840	960	0
	1b	820	810	840	960	0
	2a	800	870	M	920	0
	2b	800	870	M	920	0
	3	740	900	810	840	M

- ✓ If a product's production cannot be split across multiple plants → Assignment problem
  - Optimal solution:  $x_{12} = 1, x_{13} = 1, x_{21} = 1, x_{34} = 1 \rightarrow Z = 3290$

# Algorithm for Assignment Problem

- Role of Equivalent Cost Tables


- An **equivalent cost table** is a transformed version of the original cost table where all final entries are either 0 or positive.
  - ✓ every assignment can be made only at positions where the cost value is 0.
- Key points
  - ✓ How it works
    - By adding or subtracting constants to entire rows or columns, we can transform the table.
    - These transformations do not change the actual optimal assignment or the total cost. They only simplify the problem.
  - ✓ Interpretation
    - The 0's in the equivalent cost table represent feasible positions for assignments.
    - As long as the assignment follows the one-to-one rule (each person → one task, each task → one person), the solution obtained is equivalent to the original problem.

# Algorithm for Assignment Problem


- Equivalent Cost Table

- Example.

	1	2	3	4
1	13	16	12	11
2	15	M	13	20
3	5	7	10	6
4(D)	0	0	0	0



	1	2	3	4
1	2	5	1	0
2	15	M	13	20
3	5	7	10	6
4(D)	0	0	0	0



	1	2	3	4
1	2	5	1	0
2	2	M	0	7
3	0	2	5	1
4(D)	0	0	0	0

- ✓ In the original table, if you sum the costs at the assigned positions, you get  $Z = 11 + 13 + 5 + 0 = 29$

# Algorithm for Assignment Problem

- Equivalent Cost Table

- Example.

- ✓ Since every row already contains a zero (except the last row), subtract the minimum of each column from that column.

	1	2	3	4	5(D)		1	2	3	4	5(D)
1a	820	810	840	960	0	1a	80	0	30	120	0
1b	820	810	840	960	0	1b	80	0	30	120	0
2a	800	870	M	920	0	2a	60	60	M	80	0
2b	800	870	M	920	0	2b	60	60	M	80	0
3	740	900	810	840	M	3	0	90	0	0	M

- ✓ Even after transformation, a complete assignment using only the zero positions is still not possible. 변형 후에도 여전히 0의 위치만을 이용한 온전한 할당해 도출 불가능

# Algorithm for Assignment Problem

- Equivalent Cost Table: Generating Additional Zeros

- Example.

- ✓ Draw lines across rows/columns to cover all zeros with the minimum number of lines.

	1	2	3	4	5(D)		1	2	3	4	5(D)		1	2	3	4	5(D)	
1a	820	810	840	960	0		1a	80	0	30	120	0	1a	50	0	0	90	0
1b	820	810	840	960	0		1b	80	0	30	120	0	1b	50	0	0	90	0
2a	800	870	M	920	0	➡	2a	60	60	M	80	0	2a	30	60	M	50	0
2b	800	870	M	920	0		2b	60	60	M	80	0	2b	30	60	M	50	0
3	740	900	810	840	M		3	0	90	0	0	M	3	0	120	0	0	M

- ✓ Smallest remaining value =30 (choose one from the two belows)

- Method 1. Subtract 30 from all entries, then restore the positions of zeros, and add 30 to each line (adding 60 at intersections).
- Method 2. Subtract 30 from the entries not covered by any line, add 30 to the entries at the intersections of lines, and leave the other covered entries unchanged.

# Algorithm for Assignment Problem

- Equivalent Cost Table: Generating Additional Zeros
  - Example.
    - ✓ Once an assignment using only zero positions becomes possible, derive the optimal assignment solution:
      - $Z = 810 + 840 + 800 + 0 + 840 = 3290$

	1	2	3	4	5(D)		1	2	3	4	5(D)
1a	50	0	0	90	0	1a	50	0	0	90	30
1b	50	0	0	90	0	1b	50	0	0	90	30
2a	30	60	M	50	0	2a	0	30	M	20	0
2b	30	60	M	50	0	2b	0	30	M	20	0
3	0	120	0	0	M	3	0	120	0	0	M

# Algorithm for Assignment Problem

- Hungarian Algorithm
  - 1. Subtract the minimum value in each row (row reduction)
  - 2. Subtract the minimum value in each column (column reduction)
  - 3. Check for optimality: if the minimum number of lines needed to cover all zeros equals the number of rows (or columns)  $n$ , then an optimal solution can be found.
  - 4. If the number of lines is less than  $n$ : (a) Subtract the smallest uncovered value from all uncovered elements, (b) Add this value to the elements at the intersections of the lines, (c) Leave all other elements unchanged.
  - 5. Repeat steps 3 and 4 until an optimal solution found.
  - 6. Make the assignment using only zero positions (start with rows/columns that have exactly one zero).

**Q&A**