Chap8 The Transportation and Assignment Problems

❑ Example: Three canneries and four warehouse

		Shipping Cost per Truckload				Output
		Warehouse				
		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	

x_{ij} = the number of truckloads to be shipped from cannery i to warehouse j.

❑ The Transportation Problem

- ✓ Distribute goods from sources to destinations with minimum cost.
- ✓ s_i : number of units being supplied by source i.
- ✓ d_j : number of units being received by destination j.
- ✓ c_{ij} : cost per unit distributed from source i to destination j.
- ✓ x_{ij} : amount distributed from source i to destination j.
- ✓ Parameter table for the transportation problem:

		Cost per Unit Distributed				Supply
		Destination				
		1	2	…	n	
Source	1	c_{11}	c_{12}	…	c_{1n}	s_1
	2	c_{21}	c_{22}	…	c_{2n}	s_2
	…	…	…	…	…	…
	m	c_{m1}	c_{m2}	…	c_{mn}	s_m
Demand		d_1	d_2	…	d_n	

- ✓ Any problem (whether involving transportation or not) fits the model for a transportation problem if it can be described completely in terms of a parameter table.

- ✓ Formulation

  Min $Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$

  S.T. $\sum_{j=1}^{n} x_{ij} = s_i$, for i = 1, 2, …, m

  $\sum_{i=1}^{m} x_{ij} = d_j$, for j = 1, 2, …, n

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

- ❑ **The feasible solutions property: A transportation problem will have feasible solution if and only if $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$.**

- ❑ **Integer solution property: For transportation problems where every s_i and d_j has an integer value, all the basic variables in every basic feasible solution (including an optimal one) also have integer values.**
  - ✓ We will discuss the reasoning after introducing the solution algorithm.

- ❑ **Another example (production schedule) – with a dummy destination**

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	

  - ✓ Tradition modeling way—x_j be the number to be produced in month j.

✓ Transportation problem modeling way

- x_{ij} = number of engines produced in month i for installation in month j.
- c_{ij} = cost associated with each unit of x_{ij}.
- d_j = number of scheduled installations in month j.
- s_i = production of engines in month i.

		Cost per Unit Distributed				**Supply**
		Destination				
		1	**2**	**3**	**4**	
Source	**1**	1.080	1.095	1.110	1.125	?
	2	M	1.110	1.125	1.140	?
	3	M	M	1.100	1.115	?
	4	M	M	M	1.130	?
Demand		10	15	25	20	

✓ How can we make it fit into the transportation model?

- Adopt the supply upper bounds as the supply.
- This results in extra supply =
- In order to guarantee $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$, we need a **dummy destination** to consume the extra supply (unused production capacity) in the respective month.

		Cost per Unit Distributed					**Supply**
		Destination					
		1	**2**	**3**	**4**	**5(D)**	
Source	**1**	1.080	1.095	1.110	1.125		
	2	M	1.110	1.125	1.140		
	3	M	M	1.100	1.115		
	4	M	M	M	1.130		
Demand		10	15	25	20		

❑ One more example (distribution of water resources) — with dummy source.

	Cost per Acre Foot				Supply
	City1	City2	City3	City4	
River 1	16	13	22	17	50
River 2	14	13	19	15	60
River 3	19	20	23	*M*	50
Minimum Needed	30	70	0	10	
Requested	50	70	30	Infinity	

- ✓ Demand (requested amount) is greater than supply→Use requested amount as demand→For city4, the maximum amount could be requested is limited by (50+60+50)-(30+70+0)=60→Use dummy supply to provide extra supply→The amount supplied by this dummy source is (50+70+30+60)−(50+60+50)=50.
- ✓ Amount gotten from dummy source means the amount not satisfy the demand.

	Cost per Acre Foot				Supply
	City1	City2	City3	City4	
River 1	16	13	22	17	50
River 2	14	13	19	15	60
River 3	19	20	23		50
(Dummy) River 4					
Demand	50	70	30	60	

- ✓ **For City 3**, minimum request is OK.
- ✓ **For City 4**, demand is 60 and the maximum supply from dummy source is 50. So, there must be at least 10 units from the real river, which satisfy the minimum requirement.
- ✓ **For City 2**, minimum demand and requested demand is the same. So, it is not allowed to get any water from the dummy source. Use extreme large cost (*M*) to prevent getting any water from dummy source.
- ✓ **For City 1**, we have to make sure that the minimum demand is from real rivers. The exceed amount (50-30) can be from real river or dummy.

	Cost per Acre Foot					Supply
			City2	City3	City4	
River 1			13	22	17	50
River 2			13	19	15	60
River 3			20	23	M	50
Dummy 4				0	0	50
Demand			70	30	60	

❑ Transportation simplex method

- ✓ The transportation problem is a special type of LP problem.
- ✓ Of course, we can use the simplex method to solve this problem.
- ✓ Objective: Min $Z=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij}$ → Max $-Z = -\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij}$
- ✓ Row 0: $-Z+\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij} = 0$
- ✓ After any iteration, the coefficient of x_{ij} in row 0 is $c_{ij} - u_i - v_j$.
  - ➢ u_i = multiple of original row i that has been subtracted from original row 0 (u_i is the dual variables associated with the supply constraint).
  - ➢ v_j = multiple of original row $m+j$ that has been subtracted from original row 0 (v_i is the dual variables associated with the demand constraint).

- ✓ An efficient form of simplex is available by taking advantage of this special structure.
  - ➢ No need to use Big-M method to obtain the initial BF solution.
  - ➢ Row 0 can be obtained simply by calculating the current values of u_i and v_j .
  - ➢ The leaving basic variable can be identified in a simple way without using the coefficients of the entering basic variable.
- ✓ We use the following table to represent a transportation problem.

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="4">Destination</td><td rowspan="2">Supply</td><td rowspan="2">u_i</td></tr>
<tr><td>1</td><td>2</td><td>…</td><td>n</td></tr>
<tr><td rowspan="4">Source</td><td>1</td><td>c_{11}</td><td>c_{12}</td><td>…</td><td>c_{1n}</td><td>s_1</td><td></td></tr>
<tr><td>2</td><td>c_{21}</td><td>c_{22}</td><td>…</td><td>c_{2n}</td><td>s_2</td><td></td></tr>
<tr><td>:</td><td>…</td><td>…</td><td>…</td><td>…</td><td>:</td><td></td></tr>
<tr><td>m</td><td>c_{m1}</td><td>c_{m2}</td><td>…</td><td>c_{mn}</td><td>s_m</td><td></td></tr>
<tr><td colspan="2">Demand</td><td>d_1</td><td>d_2</td><td>…</td><td>d_n</td><td colspan="2" rowspan="2">Z =</td></tr>
<tr><td colspan="2">v_j</td><td></td><td></td><td></td><td></td></tr>
</table>

- ✓ If x_{ij} is a basic variable,

c_{ij}	
	x_{ij}

- ✓ If x_{ij} is a nonbasic variable

c_{ij}	
	$c_{ij}-u_i-v_j$

- ✓ The example of water resources distribution:

		Destination					Supply	u_i
		1	**2**	**3**	**4**	**5**		
source	**1**	16	16	13	22	17	50	
	2	14	14	13	19	15	60	
	3	19	19	20	23	M	50	
	4(D)	M	0	M	0	0	50	
Demand		30	20	70	30	60		
v_j								

❑ Number of basic variables of a transportation problem

- ✓ Given m sources and n destinations, number of functional constraints is $m+n$.
- ✓ However, number of basic variables = $m+n-1$, because of equality constraints.

- ✓ Therefore, any basic feasible solution appears on a transportation simplex with exactly $m+n-1$ nonnegative allocations, where the sum of the allocations for each row or column equals its supply or demand.
- ✓ How can we determine these $m+n-1$ basic variables in an easy way?

❑ General procedure for constructing an initial BF solution

- ✓ From the rows and columns still under consideration, select the next basic variable (allocation) according to some criterion.
- ✓ Make that allocation (value of basic variable) large enough to exactly use up the remaining supply in its row or the remaining demand in its column (whichever is smaller).
- ✓ Eliminate that row or column (which has the smaller remaining supply or demand) from further consideration.

- ✓ If only one row or only one column remains under consideration, the procedure is completed by selecting every remaining variable associated with that row or column to be basic with the only feasible allocation. Otherwise, go back to the first step.

❑ Criteria one: Northwest corner rule

- ✓ Begin with x_{11} (starting from the northwest corner).
- ✓ If $x_{i,j}$ was the last basic variable selected, then next select $x_{i,j+1}$ (move one column to the right) if source i has any supply remaining.
- ✓ Otherwise, next select $x_{i+1,j}$ (move one row down).
- ✓ An Example (the distribution of water resources example)

		Destination					**Supply**	u_i
		1	**2**	**3**	**4**	**5**		
Source	**1**	16	16	13	22	17	50	
	2	14	14	13	19	15	60	
	3	19	19	20	23	M	50	
	4(D)	M	0	M	0	0	50	
Demand		30	20	70	30	60	$Z=$	
v_j								

❑ Criteria two: Vogel's approximation methods

- ✓ For each row and column remaining under consideration, calculate its **difference**, which is defined as the arithmetic difference between the smallest and next-to-the-smallest unit cost c_{ij} still remaining in that row or column.
- ✓ In that row or column having the largest difference, select the variable having he smallest remaining unit cost.
- ✓ Use up the remaining supply in its row or the remaining demand in its column (whichever is smaller).
- ✓ Eliminate that row or column. Repeat the procedure.

✓ An Example (the distribution of water resources example)

		Destination 1	2	3	4	5	Supply	Row diff.
source	1	16	16	13	22	17	50	
	2	14	14	13	19	15	60	
	3	19	19	20	23	M	50	
	4(D)	M	0	M	0	0	50	
Demand		30	20	70	30	60		
Column diff.								

		Destination 1	2	3	4	5	Supply	Row diff.
source	1	16	16	13	22	17	50	
	2	14	14	13	19	15	60	
	3	19	19	20	23	M	50	
	4(D)	M	0	M	0	0	20	
Demand		30	20	70	0	60		
Column diff.								

		Destination 1	2	3	4	5	Supply	Row diff.
source	1	16	16	13	22	17	50	
	2	14	14	13	19	15	60	
	3	19	19	20	23	M	50	
	4(D)	M	0	M	0	0	0	
Demand		30	20	70	0	40		
Column diff.								

		Destination 1	2	3	4	5	Supply	Row diff.
source	1	16	16	13	22	17	0	
	2	14	14	13	19	15	60	
	3	19	19	20	23	M	50	
	4(D)	M	0	M	0	0	0	
Demand		30	20	20	0	40		
Column diff.								

		Destination 1	2	3	4	5	Supply	Row diff.
source	1	16	16	13	22	17	0	
	2	14	14	13	19	15	20	
	3	19	19	20	23	M	50	
	4(D)	M	0	M	0	0	0	
Demand		30	20	20	0	0		
Column diff.								

		Destination 1	2	3	4	5	Supply	Row diff.
source	1	16	16	13	22	17	0	
	2	14	14	13	19	15	0	
	3	19	19	20	23	M	50	
	4(D)	M	0	M	0	0	0	
Demand		30	20	0	0	0		
Column diff.								

❑ Criteria three: Russell's approximation method

- ✓ For each source row i remaining consideration, determine its $\bar{u}_i$, which is the largest unit cost c_{ij} still remaining that row.
- ✓ For each destination column j remaining under consideration, determine its $\bar{v}_j$, which is the largest unit cost c_{ij} still remaining in that column.
- ✓ For each variable x_{ij} not previously selected in these rows and columns, calculate $\Delta_{ij} = c_{ij} - \bar{u}_i - \bar{v}_j$.
- ✓ Select the variable having the largest (in absolute terms) negative value of Δ_{ij}.

		Destination					Supply	$\bar{u}_i$
		1	**2**	**3**	**4**	**5**		
source	**1**	16	16	13	22	17	50	
	2	14	14	13	19	15	60	
	3	19	19	20	23	M	50	
	4(D)	M	0	M	0	0	50	
Demand		30	20	70	30	60		
$\bar{v}_j$								

	$\bar{u}_1$	$\bar{u}_2$	$\bar{u}_3$	$\bar{u}_4$	$\bar{v}_1$	$\bar{v}_2$	$\bar{v}_3$	$\bar{v}_4$	$\bar{v}_5$	**Largest negative** Δ_{ij}	**Allocation**
1	22	19	M	M	M	19	M	23	M	$\Delta_{45} = -2M$	$x_{45}=50$
2	22	19	M		19	19	29	23	M	$\Delta_{15} = -5-M$	$x_{15}=10$
3	22	19	M		19	19	29	23		$\Delta_{13} = -29$	$x_{13}=40$
4		19	23		19	19	29	23		$\Delta_{23} = -26$	$x_{23}=30$
5		19	23		19	19		23		$\Delta_{21} = -24$	$x_{21}=30$
6										irrelevant	$x_{31}=0$ $x_{32}=20$ $x_{34}=30$ $Z=2570$

❑ Comparison of these three methods

- ✓ Northwest corner rule is easy. However, the initial solution obtained is far from optimal because it pays no attention to unit costs c_{ij}.
- ✓ Expending a little more effort to find a good initial BF solution might greatly reduce the number of computation iterations.

❑ Optimality test—A basic feasible solution is optimal if and only if c_{ij}-u_i-$v_j \geq 0$ for every (i, j) such that x_{ij} is nonbasic.

- ✓ We need to derive the values of u_i and v_j for the current basic feasible solution and then the calculation of these c_{ij}-u_i-v_j.
- ✓ Since c_{ij}-u_i-v_j is required to be zero if x_{ij} is a basic variable, u_i and v_j satisfy the set of equations.

  $c_{ij}=u_i+ v_j$ for each (i, j) such that x_{ij} is basic.
- ✓ There are $m+n-1$ basic variables, and so there are $m+n-1$ of these equations.
- ✓ Since the number of unknown is $m+n$, one of these variables can be assigned a value arbitrarily without violating the equations.
  - ➢The choice of this one variable and its value does not affect the value of any c_{ij}-u_i-v_j, even when x_{ij} is nonbasic.
- ✓ A convenient choice for this purpose is to select the u_i that has the largest number of allocations in its row and to assign to it the value zero.
- ✓ Example (the initial BF solution is obtained by adopting the Russel method):

		Destination					**Supply**	u_i
		1	**2**	**3**	**4**	**5**		
Source	**1**	16	16	13 40	22	17 10	50	
	2	14 30	14	13 30	19	15	60	
	3	19 0	19 20	20	23 30	M	50	
	4(D)	M	0	M	0	0 50	50	
Demand		30	20	70	30	60	Z=	
v_j								

- ✓ After obtaining u_i and v_j, calculate the value of $c_{ij}-u_i-v_j$ for each nonbasic variable.

❑ Step 1 for an iteration transportation simplex (find the entering variable)

- ✓ Since $c_{ij}-u_i-v_j$ represents coefficients of nonbasic variables in row 0, the entering basic variable must have a negative $c_{ij}-u_i-v_j$.
- ✓ To choose between the candidates, select the one having the larger (in absolute terms) negative value of $c_{ij}-u_i-v_j$.
- ✓ Pick _____ as the entering variable in this example.

		Destination					Supply	u_i
		1	**2**	**3**	**4**	**5**		
Source	**1**	16 / +2	16 / +2	13 / 40	22 / +4	17 / 10	50	–5
	2	14 / 30	14 / 0	13 / 30	19 / +1	15 / –2	60	–5
	3	19 / 0	19 / 20	20 / +2	23 / 30	M / M–22	50	0
	4(D)	M / M+3	0 / +3	M / M+4	0 / –1	0 / 50	50	–22
Demand		30	20	70	30	60	Z=2,570	
v_j		19	19	18	23	22		

❑ Step 2 for an iteration for transportation simplex (find the leaving variable)

- ✓ Increasing the entering basic variable from zero sets off a chain reaction of compensating changes in other basic variables, in order to continue satisfying the supply and demand constraints.
- ✓ The first basic variable to be decreased to zero then becomes the leaving basic variable.

			Destination			Supply
			3	**4**	**5**	
Source	1	…	13 / 40 +	22 / +4	17 / 10 --	50
	2	…	13 / 30 --	19 / +1	15 / + –2	60
		…	…	…	…	
Demand			70	30	60	

- ✓ Cell (2, 5) and (1, 3) are the recipient cell (receiving allocation).
- ✓ Cell (1, 5) and (2, 3) are the donor cell (giving allocation).

- ✓ The donor cell that starts with the smallest allocation must reach zero first as the entering basic variable is increased. So, ____is the leaving variable.

❑ **Step 3 for an iteration for transportation simplex (Find the new BF solution)**

- ✓ The new basic feasible solution is identified simply by adding the value of the leaving basic variable to the allocation for each recipient cell and subtracting this same amount from the allocation for each donor cell.

		Destination				Supply
			3	4	5	
Source	**1**	…	13 50	22	17	50
	2	…	13 20	19	15 10	60
		…	…	…	…	
Demand			70	30	60	

- ✓ Continue the process until the optimal solution is found.
- ✓ Iteration 1:

		Destination					Supply	u_i
		1	**2**	**3**	**4**	**5**		
Source	**1**	16	16	13 50	22	17	50	
	2	14 30	14	13 20	19	15 10	60	
	3	19 0	19 20	20	23 30	M	50	
	4(D)	M	0	M	0	0 50	50	
Demand		30	20	70	30	60	Z=	
v_j								

✓ Iteration 2:

		Destination					Supply	u_i
		1	**2**	**3**	**4**	**5**		
source	**1**	16	16	13 / 50	22	17	50	
	2	14	14	13 / 20	19	15 / 40	60	
	3	19 / 30	19 / 20	20	23 / 0	*M*	50	
	4(D)	*M*	0	*M*	0 / 30	0 / 20	50	
Demand		30	20	70	30	60	*Z*=	
v_j								

✓ Iteration 3:

		Destination					Supply	u_i
		1	**2**	**3**	**4**	**5**		
source	**1**	16	16	13 / 50	22	17	50	
	2	14	14	13 / 20	19	15 / 40	60	
	3	19 / 30	19 / 20	20 / 0	23	*M*	50	
	4(D)	*M*	0	*M*	0 / 30	0 / 20	50	
Demand		30	20	70	30	60	*Z*=	
v_j								

❑ **The assignment problem**

- ✓ Assignees are being assigned to perform tasks.

❑ Basic assumptions of assignment problem

- ✓ The number of assignees and the number of tasks are the same (denoted by n).
- ✓ Each assignee is to be assigned to exactly one task.
- ✓ Each task is to be performed by exactly one assignee.
- ✓ There is a cost c_{ij} associated with assignee i performing task j.
- ✓ The objective is to determine how all n assignments should be made to minimize the total cost.

❑ A prototype example of assignment problem

- ✓ Assign three machines to four locations.

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

- ✓ Introduce a dummy machine for the extra location.

		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

❑ Mathematical model of assignment problem

- ✓ $x_{ij} = \begin{cases} 1 & \text{if assignee } i \text{ performs task } j \\ 0 & \text{if not} \end{cases}$
- ✓ Min $Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$

S.T. $\sum_{j=1}^{n} x_{ij} = 1$ for $i = 1, 2, \ldots, n$

$\sum_{i=1}^{n} x_{ij} = 1$ for $j = 1, 2, \ldots, n$

$x_{ij} \geq 0$ (and binary) for all i and j

❑ Assignment problem v.s. transportation problem

- ✓ The assignment problem is a special type of transportation problem
- ✓ Sources are equivalent to assignees and destinations are tasks.
- ✓ Number of sources m = number of destinations n.
- ✓ Every supply $s_i = 1$ and every demand $d_j = 1$.

❑ Solution procedures for assignment problem

- ✓ We can apply the transportation simplex method to solve an assignment problem.
- ✓ Our prototype example: just add demand row and supply columns to previous table.

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

- ✓ It is no coincidence that the solution has many degenerate basic variables.
  - ➢ There are n sources and n destination ($m = n$).
  - ➢ There are $(2n-1)$ basic variables. Exactly n of these basic variables equal 1. That is, there are $(n-1)$ degenerate basic variables.
  - ➢ These degenerate basic variables do cause **wasted iterations** when apply the transportation simplex.
- ✓ It is better to adopt specialized algorithms to take advantage of this special structure ($m = n$, every $s_i = 1$, and every $d_j = 1$).

❑ Another example – Assigning products to plants

- ✓ There are 4 products and 3 plants.

		Unit Cost ($) for Product				Capacity Available
		1	**2**	**3**	**4**	
Plant	**1**	41	27	28	24	75
	2	40	29	--	23	75
	3	37	30	27	21	45
Production rate		20	30	30	40	

- ✓ Option 1: Permit product splitting, where the same product can be produced in more than one plant.
- ✓ Option 2: Prohibit product splitting. Every plant should be assigned at lease one of the products.
- ✓ Formulation of option 1
  - ➢Dummy demand = (75+75+45) – (20+30+30+40) = 75.

		Cost per Unit Distribution					**Supply**
		Destination (Product)					
		1	**2**	**3**	**4**		
Source	**1**	41	27	28	24		75
(Plant)	**2**	40	29	*M*	23		75
	3	37	30	27	21		45
Demand		20	30	30	40		

  - ➢The optimal solution is: x_{12}=30, x_{13}=30, x_{15}=15, x_{24}=15, x_{25}=60, x_{31}=20, and x_{34}=25. Total cost is \$3,260.

- ✓ Formulation of option 2
  - ➢Each product must be assigned to just one plant.
  - ➢One of the plants will need to be assigned two products.
  - ➢By observation, plant 3 only has enough capacity to produce one product. So, either plant 1 or plant 2 will take the extra product.
  - ➢Plant 1 and 2 each are split into two assignees. Thus, there are five assignees.
  - ➢A dummy task (product 5) needs to be introduced.
  - ➢Notice that the cost entries are **not** the unit cost anymore. The cost c_{ij} is the **total cost** associated with assignee *i* performing task *j*.
  - ➢Example: total cost of assigning plant 1 to product 1

		Task (Product)				
		1	**2**	**3**	**4**	**5(D)**
Assignee (Plant)	**1a**					
	1b					
	2a					
	2b					
	3					

- **The role of equivalent cost tables**
  - ✓ We can add or subtract any constant from every element of a row or column of the cost table without really changing the problem (i.e. the optimal solution won't change).
- **The rationale of the Hungarian algorithm for the assignment problem**
  - ✓ It converts the original cost table into a series of equivalent cost tables until it reaches one where an optimal solution is obvious.
  - ✓ This final equivalent cost table is one consisting of only positive or zero elements where all the assignments can be made to the zero element position.
  - ✓ Since the total cost cannot be negative, this set of assignments with a zero total cost is clearly optimal.
- **Procedures of the Hungarian algorithm (part 1)**
  - ✓ The algorithm begins by subtracting the smallest number in each row from every number in the row.
  - ✓ This **row reduction** process will create an equivalent cost table that has a zero element in every row.
  - ✓ If this cost table has any columns without a zero element, the next step is to perform a **column reduction** process by subtracting the smallest number in each such column from every number in the column.
  - ✓ Note that the individual rows and columns can be reduced in any order.
  - ✓ Our prototype 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				
3				
4(D)				

- ✓ Another example (not that lucky anymore).

	1	2	3	4	5(D)
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*

	1	2	3	4	5(D)
1a					
1b					
2a					
2b					
3					

  - ➢Suppose we begin the process by column reduction.
  - ➢Now every row and column has at least one zero element, but a complete set of assignments with zero elements is **not** possible this time (the maximum number of assignments is 3).
  - ➢Thus, **we need more zero elements**.

❑ **Procedures of the Hungarian algorithm (part 2) – creation of additional zero elements**

- ✓ We now add or subtract a constant from a combination of rows and columns.
- ✓ This procedure begins by drawing a set of lines through some of the rows and columns in such a way as to **cover all the zeros**. This is preferably done with a **minimum** number of lines.

	1	2	3	4	5(D)
1a	80	0	30	120	0
1b	80	0	30	120	0
2a	60	60	*M*	80	0
2b	60	60	*M*	80	0
3	0	90	0	0	*M*

- ✓ The minimum element not crossed out is 30. Thus, subtracting 30 from every element in the entire table (from every row or column) will create new zeros.
- ✓ In order to restore the previous zero element and eliminate negative elements, we add 30 to each row or column with a line covering it.
- ✓ A shortcut is to subtract 30 from just the element without a line through them and then add 30 to every element that lies at the intersection of two lines.

	1	2	3	4	5(D)
1a					
1b					
2a					
2b					
3					

- ✓ It now is possible to make four assignments to zero elements, but still not five (Note that column 1 and 4 have only a single zero element and they both are in the same row).
- ✓ Repeat the same procedure to create more zeros.
- ✓ Four lines (the same number as the maximum number of assignments) now are the minimum needed to cover all zeros.

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

- ✓ The minimum element not covered by a line is again 30.
- ✓ Subtract 30 from every **uncovered** element and add 30 to every **doubly covered** element.

	1	2	3	4	5(D)
1a					
1b					
2a					
2b					
3					

- ✓ This table actually has several ways of making a complete set of assignments to zero element positions (several optimal solutions).

❑ Summary of the Hungarian algorithm

- ✓ Please refer to p. 362 of the text book.