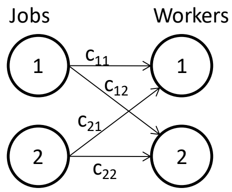
Assignment Problem:

History:
Solution method published by Harold Kuhn (classmate/friend/colleague of John Nash; the "A Beautiful Mind" guy) in 1955. Kuhn called his solution method the *Hungarian Method* because Kuhn had to learn Hungarian to read papers by Dénes Kőnig (theorem) and Jenő Egerváry (proof) in order to prove the method was correct. It should be noted that problems of 10 or more assignments were practically impossible in 1955. Kuhn's method made it so those problems could be solved, by-hand, in a matter of minutes.

Linear Assignment Problem:
Find the least cost 1-to-1 assignment of *m* "jobs" to *m* "workers." This has the same structure as the transportation problem (where the nodes are split between suppliers and customers); with one exception - all supply nodes have a supply of 1 unit, and all customer nodes have a demand of 1 unit.



Indexed Sets:
i: job index (1, …, m)
j: worker index (1, …, m)

Data:
s_i = 1 for all i (assignment problem; only 1 job per i)
d_j = 1 for all j (assignment problem; only 1 worker per j)
c_{ij} = cost of assigning job i to worker j

Variables:
x_{ij} = 1 if job i assigned to worker j; 0 otherwise; for all i,j pairs
Although; due to the structure of the problem (totally unimodular) the variable values will automatically be 0 or 1; thus, $x_{ij} \geq 0$; for all i,j pairs

Primal Problem (for 2 by 2)
Minimize Cost: $c_{11}x_{11} + c_{12}x_{12} + c_{21}x_{21} + c_{22}x_{22}$
Subject To:
Job 1 constraint: $x_{11} + x_{12} = 1$; dual variable λ_1 ~ unrestricted
Job 2 constraint: $x_{21} + x_{22} = 1$; dual variable λ_2 ~ unrestricted
Worker 1 constraint: $x_{11} + x_{21} = 1$; dual variable μ_1 ~ unrestricted
Worker 2 constraint: $x_{12} + x_{22} = 1$; dual variable μ_2 ~ unrestricted
$x_{ij} \geq 0$; for all i,j pairs

In general:

Minimize $\sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij} x_{ij}$

Subject To:

$$\sum_{j=1}^{m} x_{ij} = 1 \; \forall \; i$$

$$\sum_{i=1}^{m} x_{ij} = 1 \; \forall \; j$$

$$x_{ij} \geq 0 \; \forall \; i, j$$

Number of constraints: 2m [not counting non-negativity]
Number of variables: m^2
# of Basic Variables: 2m - 1

- # of Basic Variables with value 1: m
- # of Basic Variables with value 0: m-1

Every Basic Feasible Solution is degenerate with m-1 degenerate variables. This can cause problems with cycling if you solve using simplex algorithm. If you solve this problem with simplex algorithm or transportation simplex algorithm, then (generally) > 90% of your pivots are degenerate.

Dual Problem (for 2 by 2)
Maximize: $\lambda_1 + \lambda_2 + \mu_1 + \mu_2$
Subject To:
$\lambda_1 + \mu_1 \leq c_{11}$ *[Note that since $x_{ij} \geq 0$, the constraints are of the form ≤]*
$\lambda_1 + \mu_2 \leq c_{12}$
$\lambda_2 + \mu_1 \leq c_{21}$
$\lambda_2 + \mu_2 \leq c_{22}$
λ_1 , λ_2 , μ_1 , μ_2 ~ unrestricted

In general, the dual constraints have the form: $\lambda_i + \mu_j \leq c_{ij}$

Complementary Slackness Conditions:
If $\lambda_i + \mu_j < c_{ij} \rightarrow x_{ij} = 0$
If $x_{ij} = 1 \rightarrow \lambda_i + \mu_j = c_{ij}$

Basic Solution Strategy:

1. Choose a dual feasible solution.
2. Use complementary slackness to identify the set of x_{ij} that are allowed to be 1.
3. Using only the x_{ij} found in Step 2, try to find a primal feasible solution (i.e., a 1-to-1 assignment).
4. If Steps 1 to 3 are satisfied, then you are optimal. If not, modify the dual solution and repeat.

Hungarian Algorithm (A Primal-Dual Algorithm):

1. Choose a dual feasible solution, so that at least *m* dual constraints are binding. $\lambda_i + \mu_j \leq c_{ij}$

Ignore μ_j and set $\lambda_i \leq c_{ij}$

Thus: $\lambda_1 \leq c_{11}$, $\lambda_1 \leq c_{12}$, …, $\lambda_1 \leq c_{1m}$

$\rightarrow \lambda_1 \leq \min\{c_{1j}\} \rightarrow \lambda_i = \min_j\{c_{ij}\}$

Using the fixed λ_i find μ_j : $\mu_j \leq c_{ij} - \lambda_i \rightarrow \mu_j = \min_j\{c_{ij} - \lambda_i\}$

2. Use complementary slackness to identify the set of x_{ij} that are allowed to be 1.

Dual binding constraints are allowed to be 1.

Let $Q = \{(i, j) \mid c_{ij} - \lambda_i - \mu_j = 0\}$ = index set of binding dual constraints.

Thus, if (i,j) is not in Q, then $x_{ij} = 0$.

Thus,

$(i,j) \in Q \rightarrow x_{ij} = 0$ or 1

$(i,j) \notin Q \rightarrow x_{ij} = 0$

3. Using only x_{ij} from set Q try to find a 1-to-1 assignment (to get primal feasibility).

If we find a 1-to-1 assignment, then we are optimal (because we satisfy KKT Conditions).

4. If Steps 1 to 3 are satisfied, then you are optimal. If not, modify the dual solution and repeat.

Example:

		Job 1	Job 2	Job 3
Worker	1	2	5	7
	2	4	2	1
	3	2	3	5

Step 1:

2	5	7	$\lambda_1=2$
4	2	1	$\lambda_2=1$
2	3	5	$\lambda_3=2$

Note: $\lambda_1=\min\{2,5,7\} = 2$

Subtracting λ_i from c's. Creates m zeros.

0	3	5
3	1	0
0	1	3
$\mu_1=0$	$\mu_2=1$	$\mu_3=0$

Note: $\mu_1=\min\{2-2,4-1,2-2\}=0$

Subtracting u_j from c-λ_i's. Creates Reduced Matrix (dual slack)

0	2	5
3	0	0
0	0	3

Step 2:

Set Q = {(1,1), (2,2), (2,3), (3,1), (3,2)}. Also, $x_{12} = x_{13} = x_{21} = x_{33} = 0$.

Step 3:
$x_{11} = 1$ (must, because it is the only 0 entry in reduced matrix for worker 1)
$x_{32} = 1$ (must, because it is the only 0 entry in reduced matrix for worker 3 since $x_{11} = 1$)
$x_{23} = 1$ (must, because it is the only 0 entry in reduced matrix for job 3)
Total cost of assignment = 2 + 1 + 3 = 6

Example 2:

		Job 1	Job 2	Job 3
Worker	**1**	2	5	7
	2	4	2	1
	3	2	6	5

Step 1:

2	5	7	λ_1=2
4	2	1	λ_2=1
2	6	5	λ_3=2

Subtracting λ_i from c's. Creates m zeros.

0	3	5
3	1	0
0	4	3
μ_1=0	μ_2=1	μ_3=0

Subtracting u_j from c-λ_i's. Creates Reduced Matrix (dual slack)

0	2	5
3	0	0
0	3	3

Step 2:
Set Q = {(1,1), (2,2), (2,3), (3,1)}.

Step 3:
Only two possible assignments can be made! Must modify the dual solution and try again.

Since only two assignments were possible all zeros in the reduced matrix can be covered by 2 horizontal or vertical lines. Create at least 1 new zero in an uncovered cell. Let c_o = minimum of uncovered cells in reduced matrix (for our example c_o = 2).

Dual Variable Modification: for uncovered rows increase λ_i by c_o for covered columns decrease μ_j by c_o

Cell Type:	Net Change in $c_{ij} - \lambda_i - \mu_j$	Result
Uncovered Row/Uncovered Column:	$c_{ij} - (\lambda_i + c_o) - \mu_j = (c_{ij} - \lambda_i - \mu_j) - c_o$	$- c_o$
Uncovered Row/Covered Column:	$c_{ij} - (\lambda_i + c_o) - (\mu_j - c_o) = (c_{ij} - \lambda_i - \mu_j)$	No change
Covered Row/Uncovered Column:	$(c_{ij} - \lambda_i - \mu_j)$	No change
Covered Row/Covered Column:	$c_{ij} - \lambda_i - (\mu_j - c_o) = (c_{ij} - \lambda_i - \mu_j) + c_o$	$+ c_o$

What this means …
c_o = min(uncovered elements of reduced matrix)
Increase λ_i for each uncovered row; decrease μ_j for each covered row

Uncovered cells decrease by c_o
Single covered cells do not change
Double covered cells increase by c_o

$c_o = 2$

0	0	3	$\lambda_1=4$	λ_1 increases by c_o
5	0	0	$\lambda_2=1$	no change
0	1	1	$\lambda_3=4$	λ_3 increases by c_o
$\mu_1=-2$	$\mu_2=1$	$\mu_3=0$		

μ_1 decreases by c_o

Note that the dual solution is feasible $(c_{ij} - \lambda_i - \mu_j) \geq 0$.
Note that we can update the reduced matrix without actually recomputing λ_i and μ_j
Optimal since 3 assignments were made.
Total cost of assignment = 5 + 1 + 2 = 8

Note: You may have to do the c_o change multiple times (especially for larger problems).

Example:
Use the Hungarian Algorithm to find the minimal cost assignment, assuming that each job must be assigned to exactly one worker. Please state the optimal assignment and minimum cost. The cost matrix is shown below.

		Job 1	Job 2	Job 3
Worker	1	7	8	7
	2	5	5	2
	3	3	7	5
	4	2	4	7

Since there is an extra worker, we have to create a "Dummy Job" in order to have a 1-to-1 assignment problem.

7	8	7	0	$\lambda_1=0$
5	5	2	0	$\lambda_2=0$
3	7	5	0	$\lambda_3=0$
2	4	7	0	$\lambda_4=0$
$\mu_1=2$	$\mu_2=4$	$\mu_3=2$	$\mu_4=0$	

The "Reduced Matrix" can only make 3 assignments; therefore, use 3 vertical or horizontal lines to cover up zeros.

5	4	5	0
3	1	0	0
1	3	3	0
0	0	5	0

c_0 = 1 (i.e., job 1 - worker 3)

4	3	4	0
3	1	0	1
0	2	2	0
0	0	5	1

Optimal Assignment:
Job 1 – Worker 3
Job 2 – Worker 4
Job 3 – Worker 2
Worker 1 – Dummy Job

Optimal Cost: 3 + 4 + 2 + 0 = 9

Excerpt from ***Linear Programming and Extensions*** *(George B. Dantzig, 1963; pages 321-322):*
- *The Marriage Game (Assigning men to women based on preference).*

- *Story (copied from page 322):*
In 1955, at the summer meeting of the Operations Research Society in Los Angeles, I (Dantzig) was interviewed by the press. The reporter turned out to be the brother of my small daughter's piano teacher, and so we became quite friendly. I explained to him that linear programming models originated in the Air Force, and I described their growing application to industrial problems. It became obvious that this veteran Hollywood reporter was having a hard time seeing how to make the material into an exciting news story. In desperation I suggested, "How about something with sex appeal?" "Now you're talking," he said. "Well," I continued, "an interesting by-product of our work with linear programming models is a mathematical proof that of all the possible forms of marriage (monogamy, bigamy, polygamy, etc.), monogamy is the best." "You say monogamy is the best of all possible relations?" he queried. "Yes," I replied. "Man," he said, shaking his head in the negative, "you've been working with the wrong kind of models."

Practice Assignment Problem

Use the Hungarian Algorithm to find the minimal cost assignment, assuming that each job must be assigned to exactly one worker. Please state the optimal assignment and minimum cost. The cost matrix is shown below.

		Job 1	Job 2	Job 3	Job 4
Worker	1	14	5	8	7
	2	2	12	6	5
	3	7	8	3	9
	4	2	4	6	10

Solution: Assignment Problem

Use the Hungarian Algorithm to find the minimal cost assignment, assuming that each job must be assigned to exactly one worker. Please state the optimal assignment and minimum cost. The cost matrix is shown below.

		Job 1	Job 2	Job 3	Job 4
Worker	1	14	5	8	7
	2	2	12	6	5
	3	7	8	3	9
	4	2	4	6	10

14	5	8	7	$\lambda_1=5$
2	12	6	5	$\lambda_2=2$
7	8	3	9	$\lambda_3=3$
2	4	6	10	$\lambda_4=2$

9	0	3	2
0	10	4	3
4	5	0	6
0	2	4	8
$\mu_1=0$	$\mu_2=0$	$\mu_3=0$	$\mu_4=2$

9	0	3	0
0	10	4	1
4	5	0	4
0	2	4	6

c_0 = 1

10	0	4	0
0	9	4	0
4	4	0	3
0	1	4	5

Job 2 – Worker 1
Job 4 – Worker 2
Job 3 – Worker 3
Job 1 – Worker 4
Cost = 15