## Randomly-Generated 9×9 Assignment Problem

| \ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 13 | 8 | 10 | 11 | 6 | 14 | 10 | 7 | 13 |
| **2** | 12 | 13 | 11 | 1 | 2 | 11 | 2 | 12 | 10 |
| **3** | 6 | 8 | 4 | 12 | 12 | 5 | 3 | 13 | 13 |
| **4** | 12 | 2 | 6 | 4 | 11 | 5 | 10 | 4 | 6 |
| **5** | 1 | 2 | 4 | 10 | 7 | 8 | 14 | 9 | 15 |
| **6** | 11 | 2 | 7 | 10 | 3 | 8 | 5 | 11 | 11 |
| **7** | 10 | 2 | 2 | 4 | 8 | 10 | 2 | 1 | 4 |
| **8** | 11 | 3 | 2 | 12 | 15 | 3 | 3 | 14 | 14 |
| **9** | 6 | 8 | 15 | 4 | 12 | 7 | 4 | 15 | 12 |



### LINGO model:

```
MODEL:   !  Linear assignment problem;

SETS:
    JOB/1..9/: ;
    MACHINE/1..9/: ;
    ASSIGN(MACHINE,JOB): COST, X;
ENDSETS

DATA:
    COST= 13  8 10 11  6 14 10  7 13
          12 13 11  1  2 11  2 12 10
           6  8  4 12 12  5  3 13 13
          12  2  6  4 11  5 10  4  6
           1  2  4 10  7  8 14  9 15
          11  2  7 10  3  8  5 11 11
          10  2  2  4  8 10  2  1  4
          11  3  2 12 15  3  3 14 14
           6  8 15  4 12  7  4 15 12;
ENDDATA
```



```
MIN = @SUM(ASSIGN(I,J):  COST(I,J)*X(I,J) );

@FOR(MACHINE(I):
    @SUM(JOB(J):  X(I,J) ) = 1;
 );

@FOR(JOB(J):
    @SUM(MACHINE(I):  X(I,J) ) = 1;
 );

END
```

LINGO solution:

```
 Global optimal solution found at step:      25
 Objective value:                      28.00000

Variable           Value        Reduced Cost
X( 1, 5)        1.000000           0.0000000
X( 2, 4)        1.000000           0.0000000
X( 3, 6)        1.000000           0.0000000
X( 4, 9)        1.000000           0.0000000
X( 5, 1)        1.000000           0.0000000
X( 6, 2)        1.000000           0.0000000
X( 7, 8)        1.000000           0.0000000
X( 8, 3)        1.000000           0.0000000
X( 9, 7)        1.000000           0.0000000
```



```
Row    Slack or Surplus      Dual Price
 1         28.00000          1.000000
 2        0.0000000        -10.000000
 3        0.0000000         -6.000000
 4        0.0000000         -6.000000
 5        0.0000000         -6.000000
 6        0.0000000         -6.000000
 7        0.0000000         -7.000000
 8        0.0000000         -4.000000
 9        0.0000000         -4.000000
10        0.0000000         -8.000000

11        0.0000000          5.000000
12        0.0000000          5.000000
13        0.0000000          2.000000
14        0.0000000          5.000000
15        0.0000000          4.000000
16        0.0000000          1.000000
17        0.0000000          4.000000
18        0.0000000          3.000000
19        0.0000000          0.000000
```

Note that the dual variables are

u= [10, 6, 6, 6, 6, 7, 4, 4, 8], v=[–5, –5, –2, –5, –4, –1, –4, –3, 0]

The reduced cost matrix is $\underline{C} = C - (u \oplus v)$

where ⊕ indicates *outer product:*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 8 | 3 | 2 | 6 | 0 | 5 | 4 | 0 | 3 |
| 11 | 12 | 7 | 0 | 0 | 6 | 0 | 9 | 4 |
| 5 | 7 | 0 | 11 | 10 | 0 | 1 | 10 | 7 |
| 11 | 1 | 2 | 3 | 9 | 0 | 8 | 1 | 0 |
| 0 | 1 | 0 | 9 | 5 | 3 | 12 | 6 | 9 |
| 9 | 0 | 2 | 8 | 0 | 2 | 2 | 7 | 4 |
| 11 | 3 | 0 | 5 | 8 | 7 | 2 | 0 | 0 |
| 12 | 4 | 0 | 13 | 15 | 0 | 3 | 13 | 10 |
| 3 | 5 | 9 | 1 | 8 | 0 | 0 | 10 | 4 |

*That is, if we reduce rows of C by u, and the columns by v, we obtain a reduced cost matrix (as in Hungarian method) with a zero-cost assignment!*



## Hungarian Method

**Row reduction:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 7 | 2 | 4 | 5 | 0 | 8 | 4 | 1 | 7 |
| 11 | 12 | 10 | 0 | 1 | 10 | 1 | 11 | 9 |
| 3 | 5 | 1 | 9 | 9 | 2 | 0 | 10 | 10 |
| 10 | 0 | 4 | 2 | 9 | 3 | 8 | 2 | 4 |
| 0 | 1 | 3 | 9 | 6 | 7 | 13 | 8 | 14 |
| 9 | 0 | 5 | 8 | 1 | 6 | 3 | 9 | 9 |
| 9 | 1 | 1 | 3 | 7 | 9 | 1 | 0 | 3 |
| 9 | 1 | 0 | 10 | 13 | 1 | 1 | 12 | 12 |
| 2 | 4 | 11 | 0 | 8 | 3 | 0 | 11 | 8 |

**Column reduction:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 7 | 2 | 4 | 5 | 0 | 7 | 4 | 1 | 4 |
| 11 | 12 | 10 | 0 | 1 | 9 | 1 | 11 | 6 |
| 3 | 5 | 1 | 9 | 9 | 1 | 0 | 10 | 7 |
| 10 | 0 | 4 | 2 | 9 | 2 | 8 | 2 | 1 |
| 0 | 1 | 3 | 9 | 6 | 6 | 13 | 8 | 11 |
| 9 | 0 | 5 | 8 | 1 | 5 | 3 | 9 | 6 |
| 9 | 1 | 1 | 3 | 7 | 8 | 1 | 0 | 0 |
| 9 | 1 | 0 | 10 | 13 | 0 | 1 | 12 | 9 |
| 2 | 4 | 11 | 0 | 8 | 2 | 0 | 11 | 5 |

Covering zeroes (8 lines required):

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **7** | **2** | **4** | **5** | **0** | **7** | **4** | **1** | **4** |
| **11** | **12** | **10** | **0** | **1** | **9** | **1** | **11** | **6** |
| **3** | **5** | **1** | **9** | **9** | **1** | **0** | **10** | **7** |
| **10** | **0** | **4** | **2** | **9** | **2** | **8** | **2** | **1** |
| **0** | **1** | **3** | **9** | **6** | **6** | **13** | **8** | **11** |
| **9** | **0** | **5** | **8** | **1** | **5** | **3** | **9** | **6** |
| **9** | **1** | **1** | **3** | **7** | **8** | **1** | **0** | **0** |
| **9** | **1** | **0** | **10** | **13** | **0** | **1** | **12** | **9** |
| **2** | **4** | **11** | **0** | **8** | **2** | **0** | **11** | **5** |

Result of reduction:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 7 | 2 | 3 | 5 | 0 | 6 | 4 | 0 | 3 |
| 11 | 12 | 9 | 0 | 1 | 8 | 1 | 10 | 5 |
| 3 | 5 | 0 | 9 | 9 | 0 | 0 | 9 | 6 |
| 10 | 0 | 3 | 2 | 9 | 1 | 8 | 1 | 0 |
| 0 | 1 | 2 | 9 | 6 | 5 | 13 | 7 | 10 |
| 9 | 0 | 4 | 8 | 1 | 4 | 3 | 8 | 5 |
| 10 | 2 | 1 | 4 | 8 | 8 | 2 | 0 | 0 |
| 10 | 2 | 0 | 11 | 14 | 0 | 2 | 12 | 9 |
| 3 | 5 | 11 | 1 | 9 | 2 | 1 | 11 | 5 |



Covering zeroes (8 lines required):

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 7 | 2 | 3 | 5 | 0 | 6 | 4 | 0 | 3 |
| 11 | 12 | 9 | 0 | 1 | 8 | 1 | 10 | 5 |
| 3 | 5 | 0 | 9 | 9 | 0 | 0 | 9 | 6 |
| 10 | 0 | 3 | 2 | 9 | 1 | 8 | 1 | 0 |
| 0 | 1 | 2 | 9 | 6 | 5 | 13 | 7 | 10 |
| 9 | 0 | 4 | 8 | 1 | 4 | 3 | 8 | 5 |
| 10 | 2 | 1 | 4 | 8 | 8 | 2 | 0 | 0 |
| 10 | 2 | 0 | 11 | 14 | 0 | 2 | 12 | 9 |
| 3 | 5 | 11 | 1 | 9 | 2 | 1 | 11 | 5 |

Result of reduction:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 8 | 3 | 3 | 6 | 0 | 6 | 4 | 0 | 3 |
| 11 | 12 | 8 | 0 | 0 | 7 | 0 | 9 | 4 |
| 4 | 6 | 0 | 10 | 9 | 0 | 0 | 9 | 6 |
| 11 | 1 | 3 | 3 | 9 | 1 | 8 | 1 | 0 |
| 0 | 1 | 1 | 9 | 5 | 4 | 12 | 6 | 9 |
| 9 | 0 | 3 | 8 | 0 | 3 | 2 | 7 | 4 |
| 11 | 3 | 1 | 5 | 8 | 8 | 2 | 0 | 0 |
| 11 | 3 | 0 | 12 | 14 | 0 | 2 | 12 | 9 |
| 3 | 5 | 10 | 1 | 8 | 1 | 0 | 10 | 4 |

Covering zeroes (9 lines required):

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 8 | 3 | 3 | 6 | 0 | 6 | 4 | 0 | 3 |
| 11 | 12 | 8 | 0 | 0 | 7 | 0 | 9 | 4 |
| 4 | 6 | 0 | 10 | 9 | 0 | 0 | 9 | 6 |
| 11 | 1 | 3 | 3 | 9 | 1 | 8 | 1 | 0 |
| 0 | 1 | 1 | 9 | 5 | 4 | 12 | 6 | 9 |
| 9 | 0 | 3 | 8 | 0 | 3 | 2 | 7 | 4 |
| 11 | 3 | 1 | 5 | 8 | 8 | 2 | 0 | 0 |
| 11 | 3 | 0 | 12 | 14 | 0 | 2 | 12 | 9 |
| 3 | 5 | 10 | 1 | 8 | 1 | 0 | 10 | 4 |

Recovering solution:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 8 | 3 | 3 | 6 | 0 | 6 | 4 | 0 | 3 |
| 11 | 12 | 8 | 0 | 0 | 7 | 0 | 9 | 4 |
| 4 | 6 | 0 | 10 | 9 | 0 | 0 | 9 | 6 |
| 11 | 1 | 3 | 3 | 9 | 1 | 8 | 1 | 0 |
| 0 | 1 | 1 | 9 | 5 | 4 | 12 | 6 | 9 |
| 9 | 0 | 3 | 8 | 0 | 3 | 2 | 7 | 4 |
| 11 | 3 | 1 | 5 | 8 | 8 | 2 | 0 | 0 |
| 11 | 3 | 0 | 12 | 14 | 0 | 2 | 12 | 9 |
| 3 | 5 | 10 | 1 | 8 | 1 | 0 | 10 | 4 |

(or $X_{36}=X_{83}=1$ & $X_{33}=X_{86}=0$ which is also optimal!)

*Only 4 iterations (reductions) were required, compared with 25 for the simplex method!*