# Allocation Problems

## 1 Cost Matrices

- Require the same number of tasks as workers

- Only relative costs are important

To find a reduced cost matrix:

1. Subtract the least value in each row from each element of that row

2. Using the new matrix, subtract the least value in each column from each element in that column

## 2 Hungarian algorithm

1. Find the reduced cost matrix

2. Determine the minimum number of horizontal or vertical straight lines which will cover all the zeros in the matrix

    - If lines $\geqslant$ n, where the matrix is n×n then STOP
    - If lines <n, the matrix can be improved and continue

3. Draw in the lines and look for the smallest uncovered element, call it $e$

4. Add $e$ to all covered elements, add it twice if covered twice

5. Subtract $e$ from **all** elements of the matrix (even those that just had $e$ added)

6. Repeat steps 2-5 until an optimal solution is found

### 2.1 Example

|          | Dig | Weed | Cut |
|----------|-----|------|-----|
| Boris    | 250 | 80   | 160 |
| Percival | 230 | 90   | 150 |
| Spike    | 230 | 110  | 140 |

**Find the reduced cost matrix - First process rows**

|          | Dig | Weed | Cut |
|----------|-----|------|-----|
| Boris    | 170 | 0    | 80  |
| Percival | 140 | 0    | 60  |
| Spike    | 120 | 0    | 30  |

**Find the reduced cost matrix - process columns**

|          | Dig | Weed | Cut |
|----------|-----|------|-----|
| Boris    | 50  | 0    | 50  |
| Percival | 20  | 0    | 30  |
| Spike    | 0   | 0    | 0   |

**Draw lines to cover zeros**

|          | Dig | Weed | Cut |
|----------|-----|------|-----|
| Boris    | 50  | 0    | 50  |
| Percival | 20  | 0    | 30  |
| Spike    | 0   | 0    | 0   |

**Compare to dimension of matrix**
2 Lines$\leqslant$3×3 matrix

**Find the smallest uncovered element**
20 (Percival digging)

**Add e to all covered elements, and add twice for double covering**

|          | Dig | Weed | Cut |
|----------|-----|------|-----|
| Boris    | 50  | 20   | 50  |
| Percival | 20  | 20   | 30  |
| Spike    | 20  | 40   | 20  |

**Subtract e from all elements**

|          | Dig | Weed | Cut |
|----------|-----|------|-----|
| Boris    | 30  | 0    | 30  |
| Percival | 0   | 0    | 10  |
| Spike    | 0   | 20   | 0   |

**Draw lines to cover zeros**

|          | *Dig* | *Weed* | *Cut* |
|----------|-----|------|-----|
| *Boris*    | 30  | 0    | 30  |
| *Percival* | 0   | 0    | 10  |
| *Spike*    | 0   | 20   | 0   |

**Compare to dimension of matrix**

3 lines=$3 \times 3$ matrix - Optimal solution

# 3 Dummy Locations

When the problem is not n×n a dummy location is used, this has zeroes in all the elements.

## 3.1 Example

|       | Task A | Task B | Task C |
|-------|--------|--------|--------|
| Mark  | 12     | 23     | 15     |
| Nicky | 14     | 21     | 17     |
| Nigel | 13     | 22     | 20     |
| Susie | 14     | 24     | 13     |

**Add another task with zero cost**

|       | Task A | Task B | Task C | Task D |
|-------|--------|--------|--------|--------|
| Mark  | 12     | 23     | 15     | 0      |
| Nicky | 14     | 21     | 17     | 0      |
| Nigel | 13     | 22     | 20     | 0      |
| Susie | 14     | 24     | 13     | 0      |

**This is then processed as usual**

# 4 Incomplete data

The algorithm can also work on incomplete data (where a person cannot to a task). To do this assign values to the tasks that cannot be done that are at least **twice as large** as the largest value.

## 4.1 Example

|        | Chinese | French | Indian | Italian |
|--------|---------|--------|--------|---------|
| Denis  | -       | 27     | 15     | 40      |
| Hilary | 14      | 21     | 17     | 13      |
| Robert | 20      | -      | 13     | -       |
| Trudy  | 14      | 24     | 10     | 30      |

**The cells with dashes in are then replaced with 100**

|        | Chinese | French | Indian | Italian |
|--------|---------|--------|--------|---------|
| Denis  | 100     | 27     | 15     | 40      |
| Hilary | 14      | 21     | 17     | 13      |
| Robert | 20      | 100    | 13     | 100     |
| Trudy  | 14      | 24     | 10     | 30      |

**This can then be processed as normal**