DegreeLive

Bachelor of Business Administration (BBA) | Fifth Semester | UK5DSCMGT303

# Management Science

Course Code: UK5DSCMGT303 • Module 3 Notes


degreelive.in

# Module 3: Assignment Problems

Course Code: UK5DSCMGT303 • Module 3 Notes

## Chapter 1: Nature, Assumptions, and Mathematical Formulation of Assignment Problems

### 1.1 Meaning, Operational Context, and Distinctive Character

The Assignment Problem is a specialized, foundational sub-class of linear programming and transportation problems concerned with the optimal allocation of a finite set of indivisible resources (such as operating personnel, specialized machines, sales representatives, delivery vehicles, or software development teams) to an equal number of discrete tasks, jobs, territories, or workstations on a **strict one-to-one basis**. The overriding operational objective is to either minimize the aggregate cost, time, or distance associated with completing all jobs, or to maximize total commercial productivity, revenue, or customer satisfaction.

In classical industrial operations, the assignment problem arises whenever management must pair n facilities with n destinations, where every facility has the productive capacity to perform exactly one job, and every job requires the complete operational commitment of exactly one facility. The unit cost or efficiency of performing job j by resource i varies based on operator skill, machine speed, geographic proximity, or technological compatibility, creating a cost matrix C = [cij].

### 1.2 Mathematical Formulation of the Assignment Model

Mathematically, the classical assignment problem is formulated as an integer linear programming problem with binary decision variables:

> Minimize $Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$
>
> Subject to structural constraints:
>
> 1. Resource Allocation Constraints: $\sum_{j=1}^{n} x_{ij} = 1$ (for each resource $i = 1, 2, \ldots, n$)
>
> 2. Job Requirement Constraints: $\sum_{i=1}^{n} x_{ij} = 1$ (for each job $j = 1, 2, \ldots, n$)
>
> 3. Binary Integrality Restrictions: $x_{ij} \in \{0, 1\}$ (for all $i, j$)

Legend: Z = Aggregate assignment expenditure; c_{ij} = Monetary cost or time required when resource i is assigned to job j; x_{ij} = Binary indicator variable equal to 1 if resource i is assigned to job j, and 0 otherwise.

The structural constraints guarantee two indispensable physical conditions: (1) Each worker i is assigned to exactly one job (row sum equals 1); and (2) Each job j receives exactly one worker (column sum equals 1). Consequently, exactly n variables will equal 1 in the final optimal solution, while all remaining n2 – n variables will equal 0.

### 1.3 Theoretical Reasons for Specialized Solution Algorithms

Although the assignment problem can theoretically be cast as a standard transportation problem with $m = n$ sources (each having capacity = 1) and $n$ destinations (each having requirement = 1), attempting to solve it using standard transportation algorithms (such as the MODI method or Stepping Stone method) is computationally catastrophic. In any transportation model, a basic feasible solution requires exactly $m + n - 1 = 2n - 1$ non-negative allocations. However, an assignment schedule possesses exactly $n$ non-zero allocations! This creates an extreme degree of **mathematical degeneracy**, with $(2n - 1) - n = n - 1$ zero allocations requiring artificial basic status ($\epsilon$), causing standard transportation simplex loops to stall in cycling loops.

Furthermore, evaluating all possible permutation pairings by brute-force total enumeration requires examining $n!$ factorial alternatives. For an assignment of 20 machines to 20 jobs, $20! \approx 2.43 \times 1018$ possibilities exist—requiring centuries of high-speed supercomputer time. This computational barrier led to the formulation of an extraordinarily efficient combinatorial algorithm: **The Hungarian Method**.

## Chapter 2: The Hungarian Method (Kuhn-Munkres Algorithm)

### 2.1 Mathematical Foundations and König's Theorem

The Hungarian Method was developed by American mathematician Harold Kuhn in 1955, building upon the theoretical theorems of Hungarian mathematicians Dénes Kőnig and Jenő Egerváry. The method operates on the fundamental **Reduction Principle**:

> **Theorem (Kőnig):** If a real constant is added to or subtracted from any row or column of the cost matrix in an assignment problem, the optimal assignment schedule that minimizes total cost remains completely unchanged; only the aggregate objective function value changes by that constant amount.

By systematically subtracting row and column minimums, the Hungarian Method reduces the original cost matrix to an equivalent non-negative **Reduced Cost Matrix** containing at least one zero in every row and column. Since costs cannot be negative ($c_{ij}* \geq 0$), any assignment schedule that successfully allocates workers to jobs exclusively along cells with **zero reduced cost ($c_{ij}* = 0$)** automatically achieves the global mathematical minimum cost of zero in the reduced matrix, thereby guaranteeing absolute optimality in the original cost matrix.

### 2.2 The Systematic Six-Step Hungarian Algorithm

To solve an $n \times n$ assignment problem, apply the following rigorous procedure:

| Stage | Algorithmic Operational Procedure | Mathematical Mechanism & Objective |
|---|---|---|
| **Step 1: Row Reduction** | Identify the minimum element in each row of the cost matrix. Subtract this row minimum from | Creates at least one zero in every row. Reduces total matrix values while preserving relative cost differentials across activities. |

| Stage | Algorithmic Operational Procedure | Mathematical Mechanism & Objective |
|---|---|---|
| | every element in that respective row. | |
| **Step 2: Column Reduction** | Identify the minimum element in each column of the row-reduced matrix. Subtract this column minimum from every element in that column. | Creates at least one zero in every column that previously lacked a zero. Completes the preliminary reduced cost matrix. |
| **Step 3: Line Covering Test** | Draw the **minimum number of horizontal and vertical straight lines (k)** necessary to cover all zeros in the reduced matrix. | Tests matrix rank. If the minimum number of covering lines equals the order of the matrix (k = n), an optimal assignment exists. Proceed to Step 5. If k < n, proceed to Step 4. |
| **Step 4: Matrix Augmentation** | Find the smallest uncovered element (u > 0). Subtract u from all uncovered elements. Add u to all elements at line intersections. Leave elements covered by a single line unchanged. | Shifts zero coordinates without altering row/column equivalence. Eliminates unproductive zero clustering, creating new viable zero pairings. Return to Step 3. |
| **Step 5: Zero Assignment Protocol** | Examine rows with exactly one zero; box that zero ([0]) and cross out (×) any other zeros in its column. Next, examine columns with one unassigned zero; box it and cross out column zeros. Repeat until all rows/columns are assigned. | Enforces the strict one-to-one assignment constraint. Prevents resource double-booking and ensures complete job coverage. |
| **Step 6: Optimal Cost Calculation** | Map each assigned pair back to the **original unreduced cost matrix**. Sum the corresponding original costs to determine the total minimum operational cost. | Provides corporate management with the definitive optimal scheduling plan and audited expenditure baseline. |

## Chapter 3: Special Operational Variations in Assignment Problems

### 3.1 Unbalanced Assignment Problems

In commercial reality, the number of available workers or machines rarely equals the exact number of jobs to be completed (m ☐= n). If m > n, surplus workers exist; if m < n, surplus jobs remain

unstaffed. Because the Hungarian Method strictly requires a square matrix ($n \times n$), an unbalanced problem must be balanced prior to algorithmic execution:

- If the number of rows exceeds the number of columns ($m > n$), add $m - n$ **Dummy Columns** with all zero cost coefficients ($c_{ij} = 0$). The worker assigned to a dummy column represents an idle employee who receives no physical job assignment.
- If the number of columns exceeds the number of rows ($m < n$), add $n - m$ **Dummy Rows** with all zero cost coefficients ($c_{ij} = 0$). The job assigned to a dummy row represents an unfulfilled operational task that must be outsourced or deferred.

### 3.2 Maximization Assignment Problems

When the matrix entries represent revenues, gross margins, efficiency scores, or sales units rather than operational costs, management seeks to **maximize total payoff**. Because the Hungarian algorithm is fundamentally a minimization engine, the maximization matrix must be converted into an equivalent relative opportunity loss (regret) matrix:

1. Identify the single **maximum element (Mmax)** across the entire profit matrix.
2. Subtract every matrix element from Mmax:

```
rij = Mmax - pij  (where pij = profit coefficient)
```

Legend: r_{ij} = Relative opportunity loss / regret; M_{\max} = Highest individual profit entry in original matrix.

3. Apply the standard Hungarian minimization algorithm to the resulting regret matrix. The optimal assignment schedule derived minimizes total regret, which mathematically maximizes total corporate profit!

### 3.3 Restricted and Infeasible Assignments

Due to technological incompatibilities, physical safety regulations, trade union contracts, or licensing limitations, certain facilities cannot be assigned to specific tasks (e.g., an apprentice technician is legally prohibited from servicing high-voltage electrical transformers). To prevent the Hungarian algorithm from selecting forbidden pairings, assign a **prohibitively large mathematical penalty cost $M \to \infty$** to the restricted cells. When row and column reductions are performed, $M$ remains sufficiently large that the cell will never contain a zero, preventing assignment.

### 3.4 Multiple Optimal Assignments and The Traveling Salesman Problem (TSP)

If during the final zero assignment stage, multiple rows and columns contain two or more unassigned zeros without single-zero constraints, **alternative optimal schedules** exist. Corporate planners can generate alternative pairings by arbitrarily selecting one zero and tracing corresponding dependent assignments, providing operational flexibility.

The **Traveling Salesman Problem (TSP)** represents a specialized routing variant of the assignment problem where a salesperson must visit n cities exactly once and return to the starting city at minimum total distance. In addition to standard one-to-one assignment constraints, TSP incorporates **Sub-tour Elimination Constraints** to prevent the formation of isolated closed loops that fail to visit all cities in a single continuous circuit.

## Chapter 4: Comprehensive Worked Numerical Problems

### 4.1 Comprehensive Problem 1: Cost Minimization Assignment

A precision engineering firm has four skilled toolmakers—**W1, W2, W3, W4**—available to fabricate four customized components—**J1, J2, J3, J4**. The estimated production times (in hours) are given in the cost matrix below:

| Toolmaker | Job 1 (J1) | Job 2 (J2) | Job 3 (J3) | Job 4 (J4) |
|---|---|---|---|---|
| **W1** | 12 hrs | 30 hrs | 21 hrs | 15 hrs |
| **W2** | 18 hrs | 33 hrs | 9 hrs | 31 hrs |
| **W3** | 44 hrs | 25 hrs | 24 hrs | 21 hrs |
| **W4** | 23 hrs | 30 hrs | 28 hrs | 14 hrs |

#### Step 1: Row Reduction

- Row 1 minimum = 12. Subtract 12: [12-12, 30-12, 21-12, 15-12] = **[0, 18, 9, 3]**
- Row 2 minimum = 9. Subtract 9: [18-9, 33-9, 9-9, 31-9] = **[9, 24, 0, 22]**
- Row 3 minimum = 21. Subtract 21: [44-21, 25-21, 24-21, 21-21] = **[23, 4, 3, 0]**
- Row 4 minimum = 14. Subtract 14: [23-14, 30-14, 28-14, 14-14] = **[9, 16, 14, 0]**

#### Step 2: Column Reduction

Column minimums of row-reduced matrix: Col 1 min = 0; Col 2 min = 4; Col 3 min = 0; Col 4 min = 0. Subtract 4 from Column 2:

| Toolmaker | Col 1 (J1) | Col 2 (J2) | Col 3 (J3) | Col 4 (J4) |
|---|---|---|---|---|
| **W1** | 0 | 14 | 9 | 3 |
| **W2** | 9 | 20 | 0 | 22 |
| **W3** | 23 | 0 | 3 | 0 |
| **W4** | 9 | 12 | 14 | 0 |

#### Step 3: Minimum Line Covering and Optimality Test

DegreeLive

Let us cover all zeros using the minimum number of straight lines:

- Line 1: Vertical line covering Column 1 (covers zero at W1-J1).
- Line 2: Vertical line covering Column 3 (covers zero at W2-J3).
- Line 3: Horizontal line covering Row 3 (covers zeros at W3-J2 and W3-J4).
- Line 4: Horizontal line covering Row 4 (covers zero at W4-J4).
- Total covering lines = **4 lines**. Since the order of the matrix n = 4, and k = 4 = n, an optimal assignment is immediately achievable!

### Step 4: Zero Assignment Execution

1. Examine Row 1: Has exactly one zero at (W1, J1). Assign **W1 → J1**. Cross out any other zeros in Column 1 (none).
2. Examine Row 2: Has exactly one zero at (W2, J3). Assign **W2 → J3**. Cross out any other zeros in Column 3 (none).
3. Examine Row 4: Has exactly one zero at (W4, J4). Assign **W4 → J4**. Cross out any other zeros in Column 4 (cross out zero at W3, J4).
4. Examine Row 3: Remaining zero is at (W3, J2). Assign **W3 → J2**.

### Step 5: Final Optimal Schedule and Total Minimum Hours

| Toolmaker | Assigned Component / Job | Original Time Required (Hours) |
|---|---|---|
| **W1** | Job 1 (J1) | 12 hours |
| **W2** | Job 3 (J3) | 9 hours |
| **W3** | Job 2 (J2) | 25 hours |
| **W4** | Job 4 (J4) | 14 hours |
| **Total** | **Optimal Enterprise Schedule** | **60 Hours (Minimum)** |

Managerial Conclusion: The optimal assignment pairs W1 to J1, W2 to J3, W3 to J2, and W4 to J4, completing all manufacturing requirements in a minimum total time of **60 hours**.

## 4.2 Comprehensive Problem 2: Unbalanced Maximization Assignment

A regional sales director must assign four sales managers—**M1, M2, M3, M4**—to five regional sales territories—**T1, T2, T3, T4, T5**. The estimated monthly sales revenues (in Thousands of Rupees) are given below:

| Sales Manager | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|
| **M1** | 40 | 42 | 39 | 45 | 41 |

DegreeLive

| Sales Manager | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|
| **M2** | 38 | 44 | 43 | 40 | 46 |
| **M3** | 45 | 41 | 47 | 43 | 44 |
| **M4** | 42 | 46 | 40 | 44 | 45 |

### Step 1: Balancing the Problem (Adding Dummy Row)

There are 4 managers and 5 territories (m = 4 < n = 5). Add **Dummy Manager M5** with zero sales revenue across all territories:

Dummy Row M5: [0, 0, 0, 0, 0]. The matrix is now balanced at 5 × 5.

### Step 2: Conversion to Minimization (Opportunity Loss Matrix)

The maximum revenue across the entire matrix is Mmax = 47 (Territory T3 under M3). Subtract all matrix elements from 47:

- Row M1: [47-40, 47-42, 47-39, 47-45, 47-41] = **[7, 5, 8, 2, 6]**
- Row M2: [47-38, 47-44, 47-43, 47-40, 47-46] = **[9, 3, 4, 7, 1]**
- Row M3: [47-45, 47-41, 47-47, 47-43, 47-44] = **[2, 6, 0, 4, 3]**
- Row M4: [47-42, 47-46, 47-40, 47-44, 47-45] = **[5, 1, 7, 3, 2]**
- Row M5 (Dummy): [47-0, 47-0, 47-0, 47-0, 47-0] = **[47, 47, 47, 47, 47]**

### Step 3: Row and Column Reduction

- Row minimums: M1 min = 2; M2 min = 1; M3 min = 0; M4 min = 1; M5 min = 47.
- Row-reduced matrix:
  - Row M1: [5, 3, 6, 0, 4]
  - Row M2: [8, 2, 3, 6, 0]
  - Row M3: [2, 6, 0, 4, 3]
  - Row M4: [4, 0, 6, 2, 1]
  - Row M5: [0, 0, 0, 0, 0]
- Column minimums: All columns already contain a zero from Row M5! Column reduction leaves the matrix unchanged.

### Step 4: Line Covering and Final Assignment

Cover all zeros with straight lines:

- Line 1: Col 4 (covers zero in M1).
- Line 2: Col 5 (covers zero in M2).
- Line 3: Col 3 (covers zero in M3).
- Line 4: Col 2 (covers zero in M4).

DegreeLive

- Line 5: Row M5 (covers zeros across all columns in dummy row).
- Total covering lines = **5 lines**. Since k = 5 = n, an optimal assignment schedule is achieved!

### Step 5: Optimal Pairing and Total Maximum Revenue

| Sales Manager | Assigned Territory | Original Monthly Revenue (Rs Thousands) |
|---|---|---|
| **M1** | Territory T4 | Rs 45,000 |
| **M2** | Territory T5 | Rs 46,000 |
| **M3** | Territory T3 | Rs 47,000 |
| **M4** | Territory T2 | Rs 46,000 |
| **M5 (Dummy)** | Territory T1 (Unassigned / Outsource) | Rs 0 |
| **Total** | **Optimal Territory Deployment** | **Rs 1,84,000 (Maximum Monthly Sales)** |

Managerial Recommendation: Deploy Manager 1 to Territory 4, Manager 2 to Territory 5, Manager 3 to Territory 3, and Manager 4 to Territory 2. Territory 1 is left unstaffed for temporary local dealer representation, achieving the maximum possible sales revenue of **Rs 1,84,000** per month.

## Chapter 5: Detailed Matrix Augmentation under Suboptimal Line Covering

### 5.1 The Mathematical Mechanics of Augmentation (k < n)

When the minimum number of straight lines (k) required to cover all zeros in the reduced matrix is strictly less than the order of the matrix (k < n), the current distribution of zeros does not admit a complete, mutually independent one-to-one assignment schedule. The matrix is considered **sub-optimal**, necessitating a rigorous algebraic transformation known as **Matrix Augmentation**:

| Matrix Cell Status | Algebraic Augmentation Rule | Economic & Combinatorial Justification |
|---|---|---|
| **Uncovered Elements** | Subtract the minimum uncovered element (u): $c_{ij}^{new} = c_{ij}^{old} - u$. | Reduces the cost of unallocated pairings, bringing the most attractive alternative down to zero without creating negative values. |
| **Intersection Elements** | Add the minimum uncovered element (u): $c_{ij}^{new} = c_{ij}^{old} + u$. | Compensates for the fact that intersection elements are covered by |

| Matrix Cell Status | Algebraic Augmentation Rule | Economic & Combinatorial Justification |
|---|---|---|
| | | two lines, preserving relative column-row cost equilibrium. |
| **Single-Line Covered Elements** | Leave element completely unchanged: $c_{ij}^{new} = c_{ij}^{old}$. | Maintains mathematical indifference across previously balanced linear dimensions. |

## 5.2 Comprehensive Worked Numerical Problem Requiring Matrix Augmentation

A manufacturing enterprise must assign four specialized computer-controlled CNC milling machines—**M1, M2, M3, M4**—to four production runs—**J1, J2, J3, J4**. The setup and calibration costs (in hundreds of Rupees) are given in the initial cost matrix:

| CNC Machine | Run 1 (J1) | Run 2 (J2) | Run 3 (J3) | Run 4 (J4) |
|---|---|---|---|---|
| **M1** | Rs 10 | Rs 12 | Rs 19 | Rs 11 |
| **M2** | Rs 5 | Rs 10 | Rs 7 | Rs 8 |
| **M3** | Rs 12 | Rs 14 | Rs 13 | Rs 11 |
| **M4** | Rs 8 | Rs 15 | Rs 11 | Rs 9 |

### Step 1: Row Reduction

- Row M1 min = 10: [10-10, 12-10, 19-10, 11-10] = **[0, 2, 9, 1]**
- Row M2 min = 5: [5-5, 10-5, 7-5, 8-5] = **[0, 5, 2, 3]**
- Row M3 min = 11: [12-11, 14-11, 13-11, 11-11] = **[1, 3, 2, 0]**
- Row M4 min = 8: [8-8, 15-8, 11-8, 9-8] = **[0, 7, 3, 1]**

### Step 2: Column Reduction

- Col minimums: Col 1 min = 0; Col 2 min = 2; Col 3 min = 2; Col 4 min = 0.
- Subtract Col min from each column:
  - Col 1: [0, 0, 1, 0]
  - Col 2: [2-2, 5-2, 3-2, 7-2] = [0, 3, 1, 5]
  - Col 3: [9-2, 2-2, 2-2, 3-2] = [7, 0, 0, 1]
  - Col 4: [1, 3, 0, 1]

The resulting preliminary reduced matrix is:

DegreeLive

| CNC Machine | Col 1 (J1) | Col 2 (J2) | Col 3 (J3) | Col 4 (J4) |
|---|---|---|---|---|
| **M1** | 0 | 0 | 7 | 1 |
| **M2** | 0 | 3 | 0 | 3 |
| **M3** | 1 | 1 | 0 | 0 |
| **M4** | 0 | 5 | 1 | 1 |

### Step 3: Line Covering Test (Failure: k < n)

Let us cover all zeros using the absolute minimum number of straight lines:

- Line 1: Vertical line covering Column 1 (covers three zeros at M1, M2, M4).
- Line 2: Horizontal line covering Row 1 (covers zero at M1-J2).
- Line 3: Horizontal line covering Row 3 (covers zeros at M3-J3 and M3-J4).
- Notice: Zero at M2-J3 is already covered by Column 1? No, M2-J3 is in Col 3, Row 2. To cover M2-J3, can we cover Col 3 instead of Row 3?
  - Cover Col 1 (covers M1-J1, M2-J1, M4-J1).
  - Cover Col 3 (covers M2-J3, M3-J3).
  - Cover Row 1 (covers M1-J2).
  - Cover Row 3 (covers M3-J4).

  Wait, can all zeros be covered by 3 lines? Let's test: Line 1 = Col 1; Line 2 = Row 1; Line 3 = Row 3. This leaves only zero at (M2, J3) uncovered! If we draw a line on Row 2 or Col 3, total lines = 3 lines? No, 3 lines cannot cover 4 independent zeros because (M4, J1), (M1, J2), (M2, J3), (M3, J4) lie in distinct rows and columns!

Let us verify: in Row 4, the only zero is at (M4, J1). In Row 1, zeros are at (M1, J1) and (M1, J2). In Row 2, zeros are at (M2, J1) and (M2, J3). In Row 3, zeros are at (M3, J3) and (M3, J4).
If we cover: Line 1 = Col 1 (covers M1, M2, M4 in col 1); Line 2 = Row 1 (covers M1-J2); Line 3 = Row 3 (covers M3-J3, M3-J4). All zeros EXCEPT (M2, J3) are covered. To cover (M2, J3), we need a 4th line (either Row 2 or Col 3). Hence, the minimum lines is 4! Since k = 4 = n, an optimal assignment is immediately possible without augmentation!

### Let us trace the optimal zero assignments:

1. Row 4 has a single zero at (M4, J1). Assign **M4 → J1**. Cross out other zeros in Col 1: cross out (M1, J1) and (M2, J1).
2. Now Row 1 has a single surviving zero at (M1, J2). Assign **M1 → J2**. Cross out other zeros in Col 2 (none).
3. Row 2 now has a single surviving zero at (M2, J3). Assign **M2 → J3**. Cross out other zeros in Col 3: cross out (M3, J3).
4. Now Row 3 has a single surviving zero at (M3, J4). Assign **M3 → J4**.

DegreeLive

| CNC Machine | Assigned Production Run | Original Cost (Hundreds of Rs) |
|---|---|---|
| **M1** | Run 2 (J2) | Rs 1,200 (12) |
| **M2** | Run 3 (J3) | Rs 700 (7) |
| **M3** | Run 4 (J4) | Rs 1,100 (11) |
| **M4** | Run 1 (J1) | Rs 800 (8) |
| **Total** | **Optimal Calibration Schedule** | **Rs 3,800 (Minimum Setup Cost)** |

# Chapter 6: Restricted Assignment Models and Industrial Constraints

## 6.1 Forbidden Allocations in Practical Operations

Real-world industrial assignments are frequently bounded by physical safety protocols, trade union jurisdictional rules, or operator certification requirements. For example, an unlicensed operator cannot pilot heavy transport cranes, or a sensitive chemical compound cannot be processed on an unlined steel vat.

To incorporate these operational restrictions without altering the mathematical structure of the Hungarian algorithm, the analyst assigns a **prohibitively large positive cost penalty (M)** to each prohibited cell ($c_{ij} = M$).

## 6.2 Worked Numerical Case: Restricted Scheduling

Four engineering teams—**T1, T2, T3, T4**—are assigned to four structural design projects—**P1, P2, P3, P4**. Due to proprietary non-compete agreements, Team T1 cannot work on Project P3, and Team T3 cannot work on Project P1. The engineering hours required are:

| Design Team | Project P1 | Project P2 | Project P3 | Project P4 |
|---|---|---|---|---|
| **T1** | 50 hrs | 40 hrs | **RESTRICTED (M)** | 60 hrs |
| **T2** | 40 hrs | 30 hrs | 50 hrs | 40 hrs |
| **T3** | **RESTRICTED (M)** | 45 hrs | 40 hrs | 35 hrs |
| **T4** | 60 hrs | 55 hrs | 50 hrs | 45 hrs |

### Step 1: Row Reduction with Penalty Cost M

- Row T1 min = 40: [50-40, 40-40, M-40, 60-40] = **[10, 0, M, 20]**
- Row T2 min = 30: [40-30, 30-30, 50-30, 40-30] = **[10, 0, 20, 10]**

- Row T3 min = 35: [M-35, 45-35, 40-35, 35-35] = **[M, 10, 5, 0]**
- Row T4 min = 45: [60-45, 55-45, 50-45, 45-45] = **[15, 10, 5, 0]**

### Step 2: Column Reduction

Column minimums: Col 1 min = 10; Col 2 min = 0; Col 3 min = 5; Col 4 min = 0.

- Col 1: [10-10, 10-10, M-10, 15-10] = **[0, 0, M, 5]**
- Col 2: [0, 0, 10, 10]
- Col 3: [M-5, 20-5, 5-5, 5-5] = **[M, 15, 0, 0]**
- Col 4: [20, 10, 0, 0]

### Step 3: Line Covering and Augmentation

Cover zeros with minimum lines: Line 1 = Col 2; Line 2 = Col 4; Line 3 = Row 1 (covers zero at T1-P1). This leaves zeros at (T2, P1) and (T3, P3), (T4, P3) requiring a 4th line. Covering lines = 4 = n. Optimal assignment achieved!

1. Row T1 has a single zero at (T1, P1). Assign **T1 → P1** (50 hrs). Cross out zero at (T2, P1).
2. Now Row T2 has a single surviving zero at (T2, P2). Assign **T2 → P2** (30 hrs).
3. In Column 3, assign **T4 → P3** (50 hrs).
4. Finally assign **T3 → P4** (35 hrs).

| Design Team | Assigned Project | Project Hours |
|---|---|---|
| **T1** | Project P1 | 50 hours |
| **T2** | Project P2 | 30 hours |
| **T3** | Project P4 | 35 hours |
| **T4** | Project P3 | 50 hours |
| **Total** | **Optimal Project Deployment** | **165 Hours (Minimum)** |

## Chapter 7: Real-World Enterprise Applications of the Assignment Model

### 7.1 Aviation Cockpit Crew Rostering

Major commercial airlines deploy advanced variants of the assignment algorithm to schedule thousands of flight captains, first officers, and cabin attendants to scheduled flight segments across international hub networks. The objective function minimizes hotel layover costs and crew deadhead positioning flights while strictly honoring Federal Aviation Administration (FAA) and Directorate General of Civil Aviation (DGCA) flight duty time limitations.

### 7.2 Hospital Nurse Shift Allocation

DegreeLive

Healthcare facilities utilize stochastic assignment models to allocate registered nurses across intensive care units (ICU), emergency trauma wards, pediatric clinics, and general surgical beds. The objective minimizes nurse overtime expenditure while matching specialized clinical certifications to patient acuity levels.

DegreeLive