SUBJECT: DECISION SCIENCE

Module 2: Transportation and Assignment Problems

Comprehensive University-Level Study Notes • Units 7 – 11

7 Transportation Problem - Initial Basic Feasible Solution

Mathematical Structure of the Transportation Problem

The Transportation Problem (TP) is a special class of Linear Programming problems focused on logistics and physical distribution networks. The objective is to determine the most cost-effective distribution plan for transporting a single homogenous commodity from a set of *m* origins (sources/ factories) with known supply capacities to a set of *n* destinations (warehouses/markets) with known demand requirements. The underlying assumption is that transportation costs are directly proportional to the number of units shipped along any given route.

The mathematical objective function is formulated as follows:

$$\text{Minimize } Z = \Sigma_{i=1}^{m} \Sigma_{j=1}^{n} c_{ij} x_{ij}$$

Subject to the capacity and requirement constraints:

$$\Sigma_{j=1}^{n} x_{ij} = a_i \text{ (Supply constraints for source i)}$$

$$\Sigma_{i=1}^{m} x_{ij} = b_j \text{ (Demand constraints for destination j)}$$

$$x_{ij} \geq 0 \text{ (Non-negativity constraint for all routes)}$$

Where c_{ij} represents the unit shipping cost from origin *i* to destination *j*, and x_{ij} represents the quantity of units allocated to that specific route.

Balanced vs. Unbalanced Transportation Problems

- **Balanced TP:** Occurs when total supply exactly matches total demand ($\Sigma a_i = \Sigma b_j$). A direct solution is feasible immediately.
- **Unbalanced TP:** Occurs when total supply does not equal total demand ($\Sigma a_i \neq \Sigma b_j$). To solve, the matrix must be balanced structurally:
  - If Supply > Demand, introduce a *Dummy Destination* column with demand equal to the excess supply and unit shipping costs set to zero.
  - If Demand > Supply, introduce a *Dummy Origin* row with supply equal to the deficit and unit shipping costs set to zero.

[Image of Transportation Problem Matrix Layout]

Algorithms for Initial Basic Feasible Solution (IBFS)

Before optimizing a transportation network, an initial baseline matrix configuration must be established using heuristic algorithms. We evaluate two key approaches:

I. North West Corner Rule (NWCR)

A purely structural approach that completely ignores transportation costs, prioritizing systematic cell saturation starting from the top-left coordinate of the matrix.

Step 1: Begin at the extreme top-left cell (Row 1, Column 1) of the allocation matrix.

Step 2: Allocate the maximum possible units to this cell, limited by the smaller of the remaining supply or demand capacities: $x_{11} = \min(a_1, b_1)$.

Step 3: Deduct the allocated units from the respective supply and demand parameters. Cross out the saturated row or column.

Step 4: If row capacity is depleted, move one cell downward into the same column. If column demand is satisfied, move one cell to the right into the same row. If both are satisfied simultaneously, move diagonally. Repeat until all allocations are finalized.

II. Vogel's Approximation Method (VAM)

An advanced heuristic algorithm that considers costs by evaluating the economic penalty of making a sub-optimal allocation choice. VAM consistently yields an IBFS very close to, or identical to, the absolute optimum solution.

> **Step 1: Penalty Calculation:** For every row and column, identify the two lowest unit costs and compute their absolute mathematical difference. This difference represents the "penalty cost" of missing the lowest-cost route.
>
> **Step 2: Selection:** Scan all calculated row and column penalties and select the single largest penalty value. If a tie occurs, break it by selecting the row or column with the lowest absolute unit cost.
>
> **Step 3: Allocation:** Inside the selected row or column with the highest penalty, identify the cell containing the minimum unit cost. Allocate the maximum possible units to this cell: $x_{ij} = \min(a_i, b_j)$.
>
> **Step 4: Elimination & Iteration:** Adjust remaining supply and demand parameters, cross out the fully saturated row or column, and recompute all row and column penalties for the remaining matrix. Repeat the process until all units are allocated.

8 Test for Optimality (The Modified Distribution (MODI) Method)

Prerequisite Condition: Non-Degeneracy

Before evaluating an initial solution for absolute optimality, it must be verified as non-degenerate. A transportation solution is non-degenerate if it satisfies two strict criteria:

1. The total number of occupied (allocated) cells must be exactly equal to **m + n – 1**, where *m* represents the number of rows and *n* represents the number of columns.
2. The allocated cells must be located in independent positions, meaning it is impossible to trace a closed loop linking only occupied cells.

Resolving Degeneracy: If the allocated cells are fewer than $m + n - 1$, degeneracy exists. To resolve this, allocate an infinitesimally small dummy quantity, represented as epsilon (ε), into an unallocated, independent cell that has the lowest available unit shipping cost. This allows the MODI algorithm to execute correctly.

The Modified Distribution (MODI) Method Algorithm

The MODI method (also known as the U-V method) calculates shadow prices for rows (u_i) and columns (v_j) to determine if shifting allocations to unoccupied cells can further reduce total distribution costs.

Step 1: Compute Row and Column Dual Variables: For every occupied (allocated) cell, set up the linear equation:

$\mathbf{u_i + v_j = c_{ij}}$

To begin solving the system of equations, assign an initial value of zero to the row variable ($u_i = 0$) that contains the largest number of allocations. Compute the remaining u_i and v_j values across the matrix.

Step 2: Calculate Opportunity Costs for Unoccupied Cells: For every unallocated cell, evaluate its net evaluation index (opportunity cost), denoted as Δ_{ij}, using the formula:

$\mathbf{\Delta_{ij} = c_{ij} - (u_i + v_j)}$

Step 3: Optimality Evaluation Criterion: Evaluate the calculated Δ_{ij} indices:

• If **all $\Delta_{ij} \geq 0$**, the current distribution plan is **optimal** and cannot be improved further.

• If any **$\Delta_{ij} = 0$**, an **alternative optimal solution** exists, meaning allocations can be shifted without changing total cost.

• If any **$\Delta_{ij} < 0$**, the current plan is sub-optimal. Shifting units to the cell with the most negative opportunity cost will reduce total expenditures.

Step 4: Closed-Loop Adjustment: Identify the unallocated cell with the largest negative Δ_{ij} value as the entering cell. Construct a closed loop that starts at this entering cell, shifts horizontally and vertically to turn only on occupied cells, and returns to the start. Alternatingly assign plus (+) and minus (–) signs to the corners of the loop, starting with a plus sign at the entering cell. Find the minimum allocation among the negative-signed corners, denote it as θ, add θ to all positive corners, and subtract θ from all negative corners. Retest for optimality.

9 Assignment Problem – Introduction, Solution Methods (Hungarian Method)

Introduction to the Assignment Problem

The Assignment Problem (AP) is a specialized subset of the Transportation Problem where the objective is to assign a set of n resources (often workers or machines) to an equal number of n tasks (jobs) on a one-to-one basis. The goal is to optimize a performance metric, such as minimizing total operational time, execution costs, or maximizing total yield efficiency. The problem structure assumes a perfectly square matrix layout where the supply capacity at every source is exactly 1, and the demand requirement at every destination is exactly 1.

The Hungarian Method Algorithm

The Hungarian Method is a highly efficient combinatorial optimization algorithm that uses reduced matrix costs to determine an optimal assignment pattern without evaluating all possible permutations.

Step 1: Row Reduction: Identify the absolute minimum element within each individual row of the assignment cost matrix. Subtract this minimum value from every element located within that same row. This guarantees that every row contains at least one zero.

Step 2: Column Reduction: Examine the newly modified matrix resulting from Step 1. Identify the minimum element within each individual column. Subtract this minimum value from every single element located within that column. This ensures that every row and column contains at least one zero.

Step 3: Line Optimization Test: Draw the absolute minimum number of straight horizontal and vertical lines required to completely cover all zero elements present across the matrix workspace.

• Let the total number of covering lines be denoted as L, and the order of the square cost matrix be denoted as N.

• If **L = N**, an optimal assignment path is available. Skip directly to Step 5.

• If **L < N**, the current zero distribution cannot yield an optimal assignment. Proceed to Step 4.

Step 4: Matrix Modification: Scan the uncovered elements (those not crossed out by any line) and locate the smallest value among them, denote it as k.

• Subtract k from all completely uncovered elements across the matrix.

• Add k to elements located precisely at the intersections of horizontal and vertical covering lines.

• Leave all other elements that are covered by a single line unchanged. Return to Step 3.

Step 5: Final Assignment Allocation: Locate rows or columns containing exactly one zero element. Box that zero to make an assignment, and cross out all other zeros located in its respective row and column. Repeat systematically until every resource is uniquely assigned to a single task.

10 Maximization in Assignment Problem – Unbalanced Assignment Problem

Real-world optimization problems often deviate from standard minimization formats. Decision scientists use specific structural transformations to resolve these special variations:

I. Unbalanced Assignment Problems

An assignment problem is unbalanced when the number of available resources (rows) does not equal the number of required tasks (columns), meaning the cost matrix is rectangular ($m \neq n$).

Solution Methodology: The Hungarian Method strictly requires a square matrix structure. To resolve an unbalance, introduce dummy rows or dummy columns with cell values set to zero. This balances the matrix format without distorting the underlying cost logic of the problem.

• If Rows < Columns, introduce dummy rows with zero operational costs until a square structure is achieved.

• If Columns < Rows, introduce dummy columns with zero operational costs.

II. Maximization in Assignment Problems

This variation occurs when the matrix values represent positive returns rather than costs—such as maximizing sales revenue, corporate profit margins, or employee productivity metrics.

Solution Methodology: The Hungarian method is engineered exclusively to find the minimum value path. To solve a maximization problem, it must be converted into an equivalent minimization problem using an *Opportunity Loss Matrix* transformation:

> **Step 1: Profit Inversion:** Scan the entire maximization matrix and locate the single absolute largest value present.
>
> **Step 2: Matrix Transformation:** Subtract every individual element in the matrix from this maximum value. This transforms the parameters into relative opportunity losses.
>
> **Step 3: Execution:** Apply the standard Hungarian Method algorithm to this transformed matrix to find the optimal assignment plan.

11 Travelling Salesman Problem

Core Concept and Nature

The Travelling Salesman Problem (TSP) is a classic combinatorial optimization problem in network analysis. A salesman must visit a specific set of *n* cities exactly once and return to the original starting city, structured such that the total travel distance, time, or transit cost is minimized. The problem can be modeled as a specialized variation of the Assignment Problem, but it introduces an extra routing constraint: the final solution must form a single closed loop (a Hamiltonian cycle) and explicitly avoid sub-tours.

[Image of Travelling Salesman Tour vs Sub-tours Network]

Mathematical Constraints and Matrix Modifications

To prevent the optimization model from assigning a salesman to stay in the same city, the diagonal cells of the distance matrix (representing travel from City *i* to City *i*) are set to an infinite cost parameter, denoted as $\mathbf{c_{ii} = \infty}$.

Solution Approach via the Assignment Model

The standard solution procedure follows a structured branching approach:

1. Treat the distance matrix as a standard assignment problem and solve it using the Hungarian Method.
2. Examine the resulting assignment paths to see if they form a continuous, unbroken loop covering all cities (e.g., City 1 → City 3 → City 2 → City 4 → City 1). If a single loop is formed, the solution is optimal.
3. If the solution breaks down into separate sub-tours (e.g., a closed loop between 1 → 2 → 1 and another between 3 → 4 → 3), it is invalid for a traveling salesman.
4. **Sub-tour Elimination:** To break an invalid sub-tour, locate the next smallest non-zero element in the modified matrix that connects the separate sub-tours. Force an allocation into that alternative route by assigning an infinite cost parameter (∞) to the current sub-tour links. Re-evaluate the matrix until a single continuous tour is established.

End of Module 2 • Subject: Decision Science