Rashtreeya Sikshana Samithi Trust
RV INSTITUTIONS

RV Institute of Management ®

Autonomous Institution
Affiliated to Bengaluru City University

APPLICATION OF OPERATIONS RESEARCH IN BUSINESS

23MBA421

NOTES

AUTHORS

Dr. Vinay K S
Dr. Santhosh M
Prof. Pooja Takalkar

OPTIMIZE DECISIONS | IMPROVE EFFICIENCY | ENHANCE PERFORMANCE | DRIVE EXCELLENCE

RV INSTITUTE OF MANAGEMENT
Master of Business Administration | Semester II | 23MBA421 | 4 Credits

APPLICATION OF OPERATIONS RESEARCH IN BUSINESS

23MBA421

Authors : Dr. Vinay K S | Dr. Santhosh M | Prof. Pooja Takalkar

MODULE 1	MODULE 2	MODULE 3	MODULE 4	MODULE 5
Linear Programming (LPP)	Transportation & Assignment	Sequencing & Replacement	Game Theory & Queuing	Network Analysis & Simulation
12 Hrs	10 Hrs	10 Hrs	8 Hrs	8 Hrs

Reference: N.D. Vohra — Quantitative Techniques in Management | K.K. Chawla et al. | Hamdy A. Taha — OR | Hiller & Lieberman

MODULE 1: Linear Programming Problem (LPP) (12 Hours)

Operations Research (OR) is a discipline that deals with the application of advanced analytical methods to help make better decisions. It arrives at optimal or near-optimal solutions to complex decision-making problems by applying scientific methods to allocate scarce resources. The main purpose of OR is to provide a rational basis for decision-making even in the absence of complete information.

1.1 Introduction to Operations Research

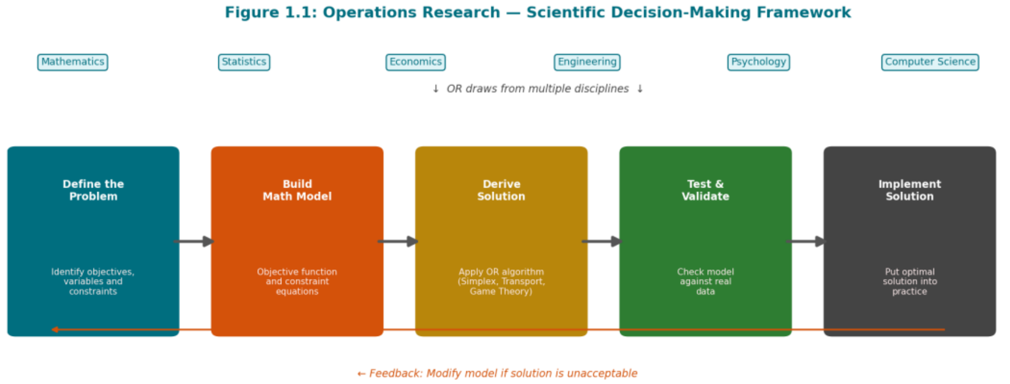

Figure 1.1: Operations Research — Scientific Decision-Making Framework (5-step process with feedback)

Definition

OR is a scientific approach to decision-making that seeks to best design and operate a system, usually under conditions requiring the allocation of scarce resources. — Morse & Kimball

Characteristics of OR

- Interdisciplinary — Uses mathematics, statistics, engineering, economics and behavioural science
- Systematic approach — Follows a structured scientific process from problem definition to implementation
- Uses mathematical models — Represents real-world problems as mathematical equations and inequalities
- Goal-oriented — Seeks optimal or near-optimal solutions (maximise profit or minimise cost)
- Decision-making under uncertainty — Provides rational basis even with incomplete information

Scope of OR

Area	OR Application	Example
Finance	Portfolio optimisation; capital budgeting; cash management	Minimise risk at target return
Marketing	Media selection; sales allocation; distribution routing	Max reach within ad budget
Production	Product mix; scheduling; inventory management	Max profit from machines
Logistics	Transportation routing; warehouse location; assignment	Min transport cost

HR	Staff scheduling; manpower planning	Optimise shift assignments
Project Mgmt	CPM/PERT scheduling and control	Min project completion time

1.2 Types of Models in OR

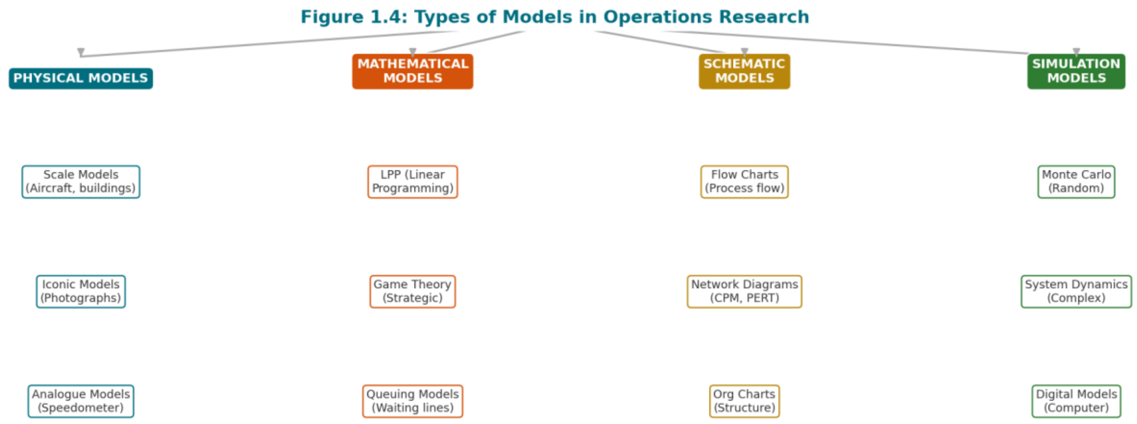

Figure 1.4: Types of Models in Operations Research — Physical, Mathematical, Schematic, Simulation

1.3 Linear Programming Problem (LPP)

Linear Programming is a mathematical optimisation technique used to determine the best outcome (maximum profit or minimum cost) given a set of linear constraints. Both the objective function and all constraints must be linear.

Components of an LPP

TERM / CONCEPT	MEANING & EXPLANATION
Decision Variables	Unknown quantities to be determined (e.g., X_1 = units of Product A, X_2 = units of Product B)
Objective Function	Linear function to be maximised (profit) or minimised (cost): Max/Min $Z = c_1X_1 + c_2X_2 + ...$
Constraints	Linear inequalities or equations representing limitations on resources (raw material, machine hours, labour)
Non-negativity	$X_1 \geq 0$, $X_2 \geq 0$ — decision variables cannot be negative (physical quantities)

Assumptions of LPP

- Linearity — Both objective function and constraints are linear (proportionality and additivity hold)
- Divisibility — Decision variables can take fractional (non-integer) values
- Certainty — All coefficients (c_j, a_{ij}, b_i) are known with certainty
- Non-negativity — All decision variables are ≥ 0

Standard LPP Format — Maximization:

- Max $Z = c_1X_1 + c_2X_2 + ... + c_nX_n$
- Subject to: $a_{11}X_1 + a_{12}X_2 + ... \leq b_1$ (Resource 1 constraint)

- $a_{21}X_1 + a_{22}X_2 + ... \le b_2$ (Resource 2 constraint)
- And $X_j \ge 0$ for all j = 1, 2, ..., n

1.4 LPP Formulation — Worked Examples

Example: Product Mix Problem

A company produces two products P and Q. Each unit of P requires 2 hours of machine time and 1 kg of material. Each unit of Q requires 1 hour of machine time and 3 kg of material. Available: 100 machine hours and 150 kg of material. Profit: ₹5 per unit of P and ₹4 per unit of Q. Maximise profit.

Let X_1 = units of P, X_2 = units of Q

Max $Z = 5X_1 + 4X_2$

Subject to: $2X_1 + X_2 \le 100$ (Machine time)

$X_1 + 3X_2 \le 150$ (Material)

$X_1, X_2 \ge 0$

1.5 Graphical Solution Method

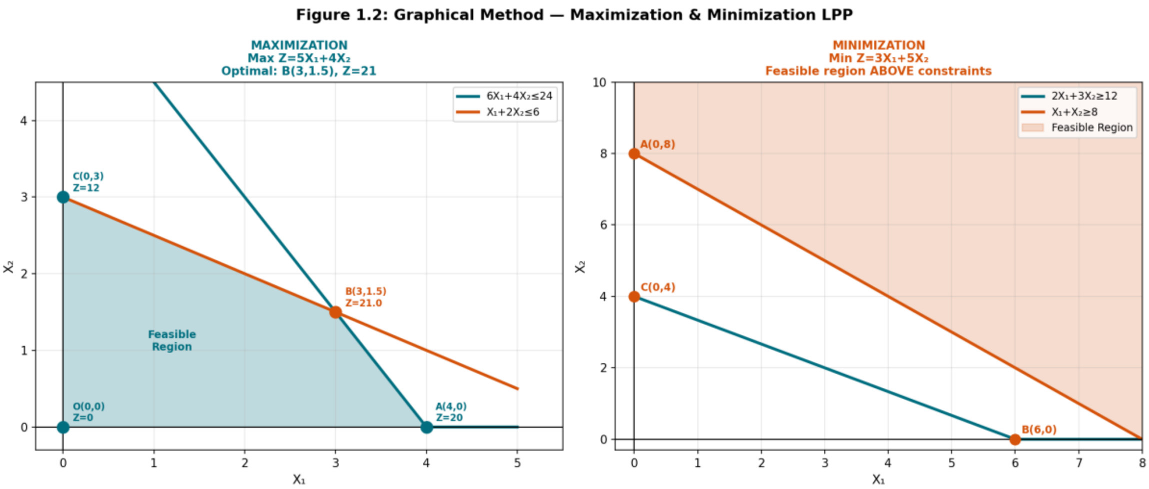

Figure 1.2: Graphical Method — Maximization (left) and Minimization (right) LPP solutions

Steps for Graphical Method

1. Plot each constraint as an equality (line) on the X_1-X_2 plane
2. Identify the feasible region (satisfies ALL constraints including non-negativity)
3. For maximization: Feasible region is below/left of all ≤ constraints
4. For minimization: Feasible region is above/right of all ≥ constraints
5. Find all corner points (vertices) of the feasible region
6. Calculate Z at each corner point
7. The optimal solution is the corner point with maximum (or minimum) Z value

Special Cases in Graphical Method:

- Unique Optimal Solution — Optimal at a single corner point (most common case)

- Multiple Optimal Solutions — Objective function parallel to a constraint; optimal on entire edge
- Unbounded Solution — Feasible region is unbounded in direction of optimisation (no maximum)
- Infeasible — No point satisfies all constraints simultaneously; constraints are contradictory
- Redundant Constraint — A constraint that does not affect the feasible region

1.6 Simplex Method

Figure 1.3: Simplex Method — Step-by-step algorithmic flowchart

The Simplex Method is an algebraic procedure for solving LPPs with any number of variables. It moves from one corner point (Basic Feasible Solution) to an adjacent one with a better objective function value, until the optimal is reached.

Converting to Standard Form

- Add SLACK variables (s_1, s_2...) for ≤ constraints to convert inequalities to equalities
- Add SURPLUS variables and ARTIFICIAL variables for ≥ constraints (Big-M method)
- The initial BFS sets all slack variables as basic variables (in the basis) with value = RHS

Simplex Tableau Structure

Basis (BV)	CB	X_1	X_2	s_1	s_2	b (RHS)
s_1	0	2	1	1	0	100
s_2	0	1	3	0	1	150
Z_j		**0**	**0**	**0**	**0**	**0**
$C_j - Z_j$		**5**	**4**	**0**	**0**	

Simplex Rules — Key Formulas

- Entering Variable: Column with MOST POSITIVE ($C_j - Z_j$) value — called the Key Column
- Departing Variable: Row with MINIMUM RATIO = b_i / a_{ij} (where a_{ij} > 0) — called Key Row
- Pivot Element: Element at intersection of Key Row and Key Column
- New Key Row = Old Key Row ÷ Pivot Element

- Other Rows: New Row = Old Row − (Key Column Element × New Key Row)
- Optimality: All $C_j - Z_j \leq 0$ for maximisation (solution is optimal)

$$Z_j = \Sigma C_{\beta i} \times a_{ij} \quad \text{(for each column j)}$$
$$C_j - Z_j = c_j - Z_j \quad \text{(Net evaluation row)}$$

1.7 Duality in LPP

Every LPP (called the Primal) has a corresponding dual problem. The primal and dual provide the same optimal value. Duality is important for sensitivity analysis and economic interpretation (shadow prices).

PRIMAL Problem	DUAL Problem
Maximisation	Minimisation
m constraints; n variables	n constraints; m variables
≤ constraints → ≥ in dual	≥ constraints → ≤ in dual
Objective coefficients (c) → RHS in dual	RHS (b) → Objective in dual
Constraint matrix A → A^T in dual	Transpose of constraint matrix used

Dual Formulation Rules:

- If Primal: Max $Z = cX$, s.t. $AX \leq b$, $X \geq 0$
- Dual: Min $W = b^TY$, s.t. $A^TY \geq c$, $Y \geq 0$
- Where Y = dual variables (shadow prices / opportunity costs)
- Strong Duality: Optimal Z (primal) = Optimal W (dual)
- Shadow price of a constraint = dual variable value = marginal value of relaxing that constraint by 1 unit

MODULE 2: Transportation and Assignment Models (10 Hours)

Transportation and Assignment models are special-structure LPPs that can be solved more efficiently using dedicated algorithms. They deal with distributing goods from sources to destinations (Transportation) and assigning agents to tasks (Assignment) optimally.

2.1 Transportation Model

The Transportation Model determines how to distribute a product from m supply sources (factories, warehouses) to n demand destinations (markets, retail stores) at minimum total transportation cost, subject to supply and demand constraints.

TERM / CONCEPT	MEANING & EXPLANATION
Supply (a_i)	Quantity available at source i (i = 1,2,...,m)
Demand (b_j)	Quantity required at destination j (j = 1,2,...,n)
Unit Cost (c_{ij})	Cost of transporting one unit from source i to destination j
Allocation (x_{ij})	Amount transported from source i to destination j (decision variable)
Balanced Problem	Total Supply = Total Demand: $\Sigma a_i = \Sigma b_j$ (all supply and demand satisfied)
Unbalanced Problem	Supply ≠ Demand; add dummy row (if supply<demand) or dummy column (if supply>demand)
Feasibility Condition	A BFS exists only if $\Sigma a_i = \Sigma b_j$ (or after balancing with dummy)

$$\text{Min } Z = \Sigma\Sigma\, c_{ij}\, x_{ij} \quad (i=1 \text{ to } m;\ j=1 \text{ to } n)$$
$$\text{Subject to: } \Sigma_j\, x_{ij} = a_i \quad \forall i \quad \text{(Supply constraints)}$$
$$\Sigma_i\, x_{ij} = b_j \quad \forall j \quad \text{(Demand constraints)}$$
$$x_{ij} \geq 0 \quad \forall i,j$$

2.2 Methods for Initial Basic Feasible Solution (IBFS)

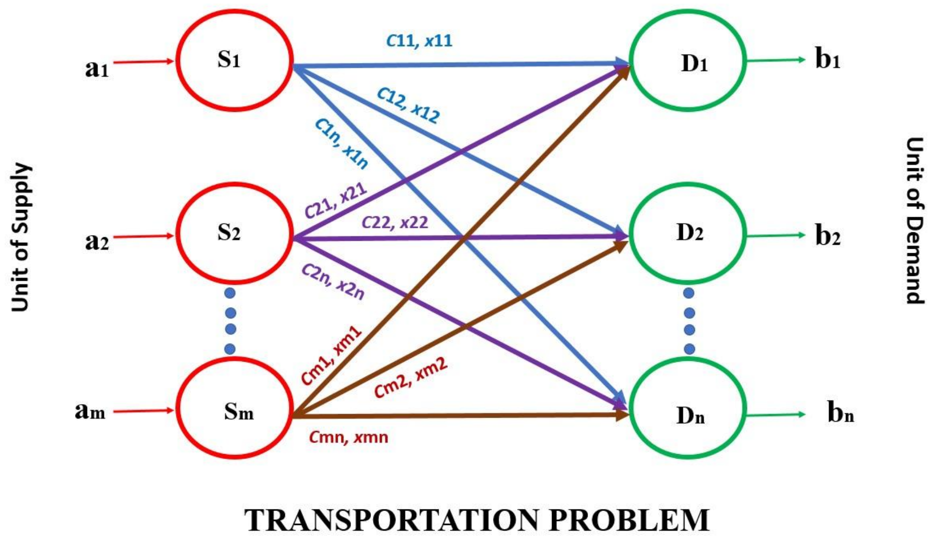

Figure 2.1: Transportation — NWCR vs VAM Initial Basic Feasible Solutions (highlighted cells = allocations)

Method 1: North-West Corner Rule (NWCR)

Start allocation from the top-left (north-west) cell and move right or down based on which supply/demand is exhausted first. Simple but gives poor initial solution (ignores costs).

8. Start at cell (1,1) — the top-left corner
9. Allocate as much as possible: min(supply of row 1, demand of col 1)
10. If supply is exhausted — move to next row; if demand is met — move to next column
11. Continue until all supply and demand are satisfied
12. Number of allocations = m+n−1 (for non-degenerate solution)

Method 2: Least Cost Method (LCM) / Matrix Minima

Always allocate to the cell with the lowest unit cost first. Gives a better initial solution than NWCR as it considers transportation costs.

13. Find the cell with the minimum cost in the entire table
14. Allocate as much as possible: min(supply of that row, demand of that column)
15. Cross out the exhausted row or column; adjust remaining supply/demand
16. Repeat with the next minimum cost cell in the remaining table
17. Continue until all allocations are made (m+n−1 allocations)

Method 3: Vogel's Approximation Method (VAM) — Best IBFS

VAM is the most efficient method for finding IBFS. It uses 'penalties' (opportunity costs) to guide allocations and typically gives the closest initial solution to the optimal.

18. Calculate PENALTY for each row and column = difference between the two lowest costs
19. Select the row or column with the HIGHEST penalty
20. In that row/column, allocate maximum possible to the cell with the LOWEST cost
21. Cross out exhausted row/column; recalculate penalties for remaining rows/columns
22. Repeat until all allocations are complete

Comparison of IBFS Methods:

- NWCR: Simplest; ignores costs; farthest from optimal; use only when speed matters
- LCM: Moderate complexity; considers costs; better than NWCR; may miss optimal
- VAM: Most complex; uses opportunity cost penalties; closest to optimal; RECOMMENDED for exams
- IBFS is only a STARTING POINT — optimal solution is found using MODI method thereafter

2.3 Optimal Solution — MODI Method (Modified Distribution Method)

The MODI method (also called the u-v method) tests whether the IBFS is optimal and improves it if not. It is based on the concept of opportunity costs for unallocated cells.

Steps of MODI Method

23. Assign values u_i (row numbers) and v_j (column numbers) for allocated cells using: $u_i + v_j = c_{ij}$; set $u_1 = 0$
24. Calculate opportunity cost for each unallocated (empty) cell: $\Delta_{ij} = c_{ij} - u_i - v_j$
25. OPTIMALITY TEST: If all $\Delta_{ij} \geq 0$ → solution is OPTIMAL (stop here)
26. If any $\Delta_{ij} < 0$ → solution can be improved; select the most negative cell as the entering cell
27. Trace a CLOSED LOOP: start from entering cell; alternate + and − through allocated cells

28. Find minimum allocation among negative-sign cells = θ (the amount to reallocate)
29. Add θ to positive cells; subtract θ from negative cells; update table and repeat

Maximization in Transportation

To solve maximisation transportation problems, convert profit matrix to opportunity loss matrix by subtracting all values from the maximum value. Then solve as minimisation.

$$\text{Opportunity Loss} = \text{Max}(c_{ij}) - c_{ij} \quad \text{for all } i, j$$

2.4 Assignment Model

The Assignment model is a special case of transportation where m workers (sources) are assigned to n jobs (destinations) on a 1-to-1 basis. The objective is to minimise total cost (or maximise total profit).

Hungarian Method (Minimisation)

30. Row Reduction: Subtract the minimum value of each row from all elements of that row
31. Column Reduction: Subtract the minimum value of each column from all elements of that column
32. Optimality Test: Draw minimum number of horizontal and vertical lines to cover all zeros
33. If number of lines = n (matrix order) → OPTIMAL assignment; go to step 5
34. If lines < n → Subtract smallest uncovered element from all uncovered elements; add it to elements covered twice; go to step 3
35. Make assignments at zero positions (one per row, one per column)

Assignment — Special Cases:

- Maximisation: Subtract all elements from max value → get opportunity cost matrix → solve as min
- Unbalanced (m≠n): Add dummy row(s) or column(s) with zero costs to balance
- Multiple Optimal Solutions: When more than one set of zero assignments possible
- Prohibited Assignment: Replace prohibited cell with large M (penalty) value
- Travelling Salesman Problem (TSP): Each city visited exactly once; assign city-to-city trips — NP-hard for large n; only theory in syllabus
- Crew Assignment: Assign flight crew to routes; complex regulatory constraints — theory only

MODULE 3: Sequencing and Replacement Problems (10 Hours)

3.1 Sequencing — Introduction

Sequencing involves determining the optimal order in which a set of jobs should be processed on a set of machines to minimise total elapsed time (makespan), idle time, or other objectives. It is widely used in scheduling production operations.

TERM / CONCEPT	MEANING & EXPLANATION
Processing Time (t_{ij})	Time required to process job i on machine j
Elapsed Time	Total time from start of first job to completion of last job (makespan)
Idle Time	Time a machine sits idle waiting for the next job
No Passing	Standard assumption: jobs processed in same order on all machines
In-Out Table	Table showing start time (in) and finish time (out) for each job on each machine

3.2 Processing n Jobs Through 2 Machines (Johnson's Algorithm)

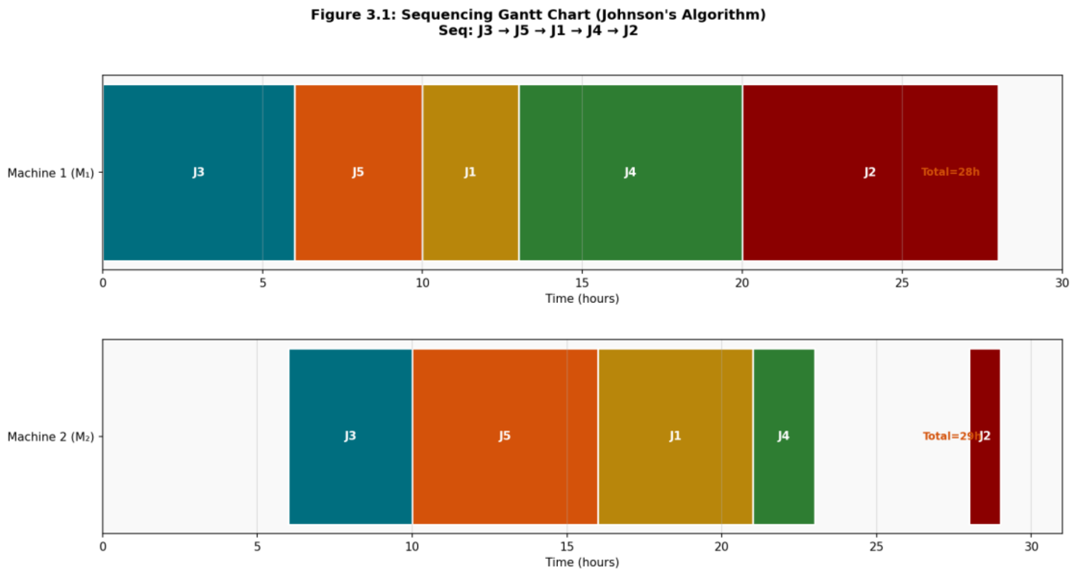

Figure 3.1: Sequencing Gantt Chart — Optimal sequence using Johnson's Algorithm (M_1 and M_2 schedules)

Johnson's Algorithm finds the optimal sequence for n jobs on 2 machines that minimises total elapsed time (makespan). The algorithm is based on the principle of minimising idle time on Machine 2.

Johnson's Algorithm — Steps

36. List the processing times of all jobs on Machine 1 (M_1) and Machine 2 (M_2)
37. Find the job with the MINIMUM processing time across both machines
38. If minimum is on M_1 → place that job FIRST in the sequence
39. If minimum is on M_2 → place that job LAST in the sequence
40. Remove that job from the list; repeat steps 2-4 for remaining jobs
41. Build the In-Out table using the optimal sequence to find elapsed time and idle times

Solving the In-Out Table:

- M_1 Start Time: = M_1 finish time of previous job (no idle time on M_1)
- M_1 Finish Time = M_1 Start Time + Processing time on M_1
- M_2 Start Time = max(M_1 Finish Time of current job, M_2 Finish Time of previous job)
- M_2 Finish Time = M_2 Start Time + Processing time on M_2
- Total Elapsed Time = M_2 finish time of the last job in sequence
- Idle Time on M_1 = Total elapsed time − ΣProcessing times on M_1
- Idle Time on M_2 = M_2 start of first job + gaps between consecutive jobs on M_2

3.3 Processing n Jobs Through 3 Machines

Johnson's algorithm can be extended to 3 machines (M_1, M_2, M_3) using the following conversion approach:

Condition for Converting 3-Machine to 2-Machine Problem

The optimal sequence found for the equivalent 2-machine problem is also optimal for the 3-machine problem, IF either of these conditions hold:

$$\text{Min}(M_1) \geq \text{Max}(M_2) \quad \text{OR} \quad \text{Min}(M_3) \geq \text{Max}(M_2)$$

Conversion Steps

42. Create equivalent Machine G: $G_i = M_{1i} + M_{2i}$ (sum of M_1 and M_2 for each job)
43. Create equivalent Machine H: $H_i = M_{2i} + M_{3i}$ (sum of M_2 and M_3 for each job)
44. Apply Johnson's algorithm on the G-H problem to find optimal sequence
45. Use this sequence to build the 3-machine In-Out table

3.4 Processing N Jobs in M Machines

For N jobs on M machines (M > 3), there is no simple optimal algorithm. The problem is NP-hard for M ≥ 3. Common approaches:

- Heuristic methods — Shortest Processing Time (SPT), Longest Processing Time (LPT) rules
- Branch and Bound — Systematic enumeration; finds optimal but computationally intensive
- Johnson's Extension — Can be applied when dominance conditions hold
- TORA / Software — Most practical approach for large-scale problems

3.5 Replacement Theory — Introduction

Replacement theory deals with the problem of deciding WHEN to replace an asset that is deteriorating with age or may fail suddenly. The objective is to minimise total cost over time.

TERM / CONCEPT	MEANING & EXPLANATION
Capital (Purchase) Cost	Initial cost of the machine; reduces as machine ages (depreciation / resale value)
Running / Operating Cost	Maintenance, repair, and operating costs that INCREASE as machine ages
Total Cost	Sum of capital cost and cumulative running costs over the asset's life
Average Annual Cost	Total cost / Number of years; the OPTIMAL replacement period minimises this

Optimal Replacement Period	Year in which average annual cost is minimum — replace the machine at this age

3.6 Replacement of Items That Deteriorate (Case 1 — Value Does Not Change)

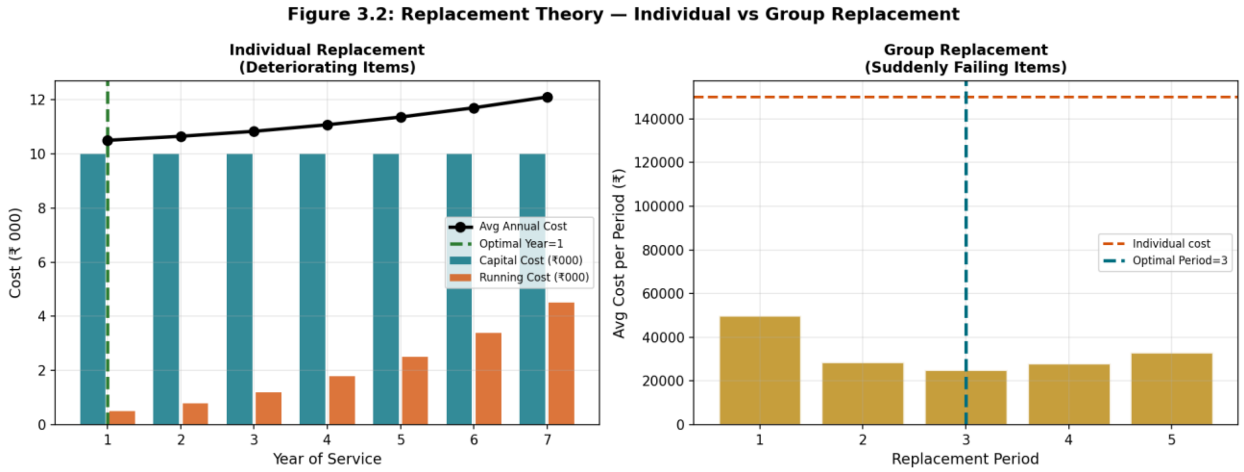

Figure 3.2: Replacement Theory — Individual (left: deteriorating items) and Group Replacement (right: failing items)

When an item deteriorates gradually with time and the resale value is assumed to be ZERO (or constant), find the replacement period that minimises average annual total cost.

$$\text{Annual Cost for year n: } C_n = \text{Capital Cost} + \text{Running Cost for year n}$$
$$\text{Total Cost after n years: } TC(n) = C + R_1 + R_2 + \ldots + R_n$$
$$\text{Average Annual Cost: } A(n) = TC(n) / n = [C + \Sigma R_i] / n$$

Optimal replacement period = value of n at which A(n) is minimum. In practice: replace when this year's running cost exceeds the previous average annual cost.

Decision Rule for Individual Replacement:

- Calculate Average Annual Total Cost A(n) for n = 1, 2, 3, ...
- Optimal period (n*) = year where A(n) is minimum
- Alternative Rule: Replace if Current Year Running Cost > Average Annual Cost of previous year
- Both methods give the same optimal replacement period

3.7 Replacement of Items That Fail Completely

Some items (bulbs, electronic components) do not deteriorate gradually but fail suddenly. The decision is whether to replace each item individually when it fails, or replace ALL items at fixed intervals (group replacement) regardless of whether they have failed or not.

Individual Replacement Policy

Replace each item only when it fails. Cost per period = (expected number of failures per period) × (individual replacement cost).

Group Replacement Policy

Replace ALL items at fixed intervals of t periods, plus replace individual failures as they occur during the interval.

$$\text{Cost/period for group replacement at period } t = [N \times p_0 \times Cg + \Sigma(\text{expected failures} \times C_i)] / t$$

Where: N = total number of items, p_0 = probability of surviving the full period, Cg = group replacement cost per item, C_i = individual replacement cost

Optimal group replacement period = period at which average cost is minimum AND is less than the individual replacement cost per period.

MODULE 4: Theory of Games and Queuing Theory (8 Hours)

Game theory analyses strategic interactions between rational decision-makers (players) competing against each other. Queuing theory models and analyses waiting line systems to optimise service performance.

4.1 Theory of Games — Introduction

TERM / CONCEPT	MEANING & EXPLANATION
Game	Competitive situation with two or more players, each choosing strategies to maximise their gain
Player	Each participant in the game making strategic decisions
Strategy	Complete plan of action for a player; may be pure (fixed choice) or mixed (probability)
Pay-off	Outcome (gain or loss) of each combination of strategies; shown in a Pay-off Matrix
Zero-Sum Game	One player's gain = other player's loss; total pay-off = 0
Two-Person Zero-Sum	Simplest game: only two players; most studied in OR; Σ pay-offs = 0
Saddle Point	Element that is simultaneously row minimum and column maximum (equilibrium point)
Value of Game (V)	Expected pay-off when both players play optimally; V = saddle point if it exists

4.2 Pure Strategy — Saddle Point Method

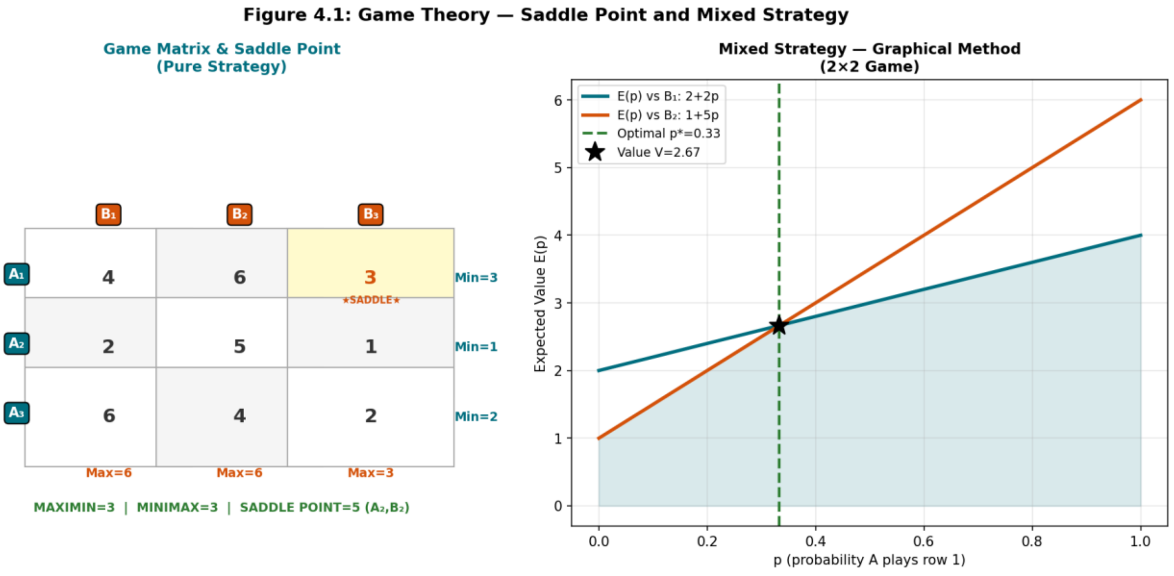

Figure 4.1: Game Theory — Pay-off Matrix with Saddle Point (left) and Mixed Strategy Graphical Solution (right)

A pure strategy game has a unique optimal strategy for each player. The solution is found using the MINIMAX-MAXIMIN criterion: Player A maximises their minimum gain (maximin); Player B minimises their maximum loss (minimax).

Finding Saddle Point — Steps

46. Find the MINIMUM value in each row (row minimums) — from Player A's perspective
47. Find the MAXIMUM value in each column (column maximums) — from Player B's perspective
48. MAXIMIN (A's optimal) = Maximum of row minimums
49. MINIMAX (B's optimal) = Minimum of column maximums
50. If MAXIMIN = MINIMAX → SADDLE POINT exists; this is the optimal pure strategy
51. Value of Game V = Saddle point value; both players play corresponding pure strategies

> **Saddle Point Rules:**
> - Saddle Point = element that is both minimum in its row AND maximum in its column
> - If Maximin = Minimax → Saddle point exists → Pure strategy game
> - If Maximin < Minimax → No saddle point → Mixed strategy required
> - The player whose minimax strategy is optimal DOMINATES the other in some sense

4.3 Mixed Strategy — Algebraic Formula Method (2×2 Games)

When no saddle point exists, players must randomise their strategies (play each with a probability). The optimal mixed strategy is the one that maximises the expected pay-off.

For a 2×2 Game Matrix [[a,b],[c,d]]:

```
p = (d−c) / (a+d−b−c)    [Probability of Player A playing Row 1]
q = (d−b) / (a+d−b−c)    [Probability of Player B playing Column 1]
V = (ad−bc) / (a+d−b−c)    [Value of the game]
```

For Player A: $p_1 = p$ (Row 1), $p_2 = 1-p$ (Row 2)

For Player B: $q_1 = q$ (Col 1), $q_2 = 1-q$ (Col 2)

4.4 Graphical Method (2×n or m×2 Games)

The graphical method is used when one player has only 2 strategies and the other has more. Plot expected payoffs as lines vs. probability p, find the highest point of the lower envelope (maximin point) for Player A.

52. For each of Player B's strategies, write E(p) as a linear function of p (Player A's probability)
53. Plot all lines on a graph with p on X-axis (0 to 1) and E on Y-axis
54. Identify the HIGHEST POINT of the LOWER ENVELOPE (lowest line at each p)
55. This highest point gives the optimal p and value of game V
56. The lines passing through this optimal point are Player B's optimal strategies

4.5 Principle of Dominance

A strategy is said to dominate another if it gives a better (or equal) payoff regardless of what the opponent does. Dominated strategies can be eliminated to simplify the game matrix.

ROW Dominance (Player A)	COLUMN Dominance (Player B)
If all elements of Row i ≥ Row j → Row j is DOMINATED	If all elements of Col j ≤ Col k → Col k is DOMINATED

Dominated row (j) is eliminated	Dominated column (k) is eliminated
Player A will never choose strategy j	Player B will never choose strategy k
Reduces matrix size; repeat until saddle point or 2×2	Reduces matrix size for formula method

4.6 Queuing Theory

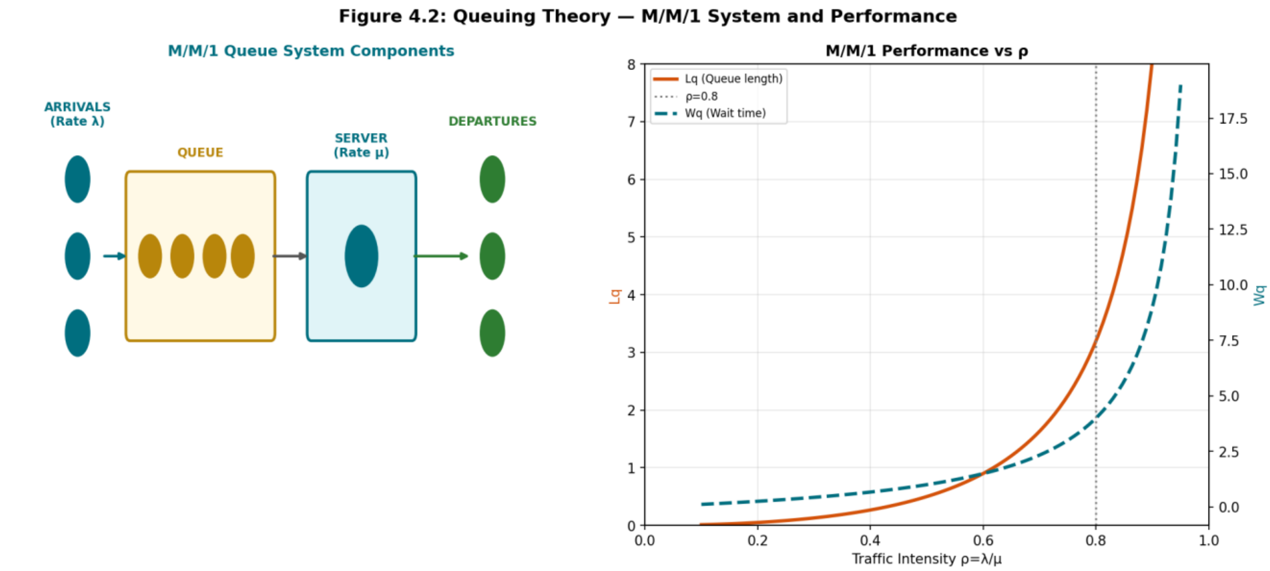

Figure 4.2: M/M/1 Queue System Components and Performance Measures vs Traffic Intensity ρ

Queuing theory provides mathematical models for analysing waiting line systems (queues). The goal is to balance the cost of providing service (more servers) vs. the cost of waiting (customer impatience, lost business).

Kendall's Notation: A/B/c/K/m/D

- A = Arrival distribution (M = Markovian/Poisson, D = Deterministic, G = General)
- B = Service time distribution (M = Exponential, D = Deterministic, G = General)
- c = Number of servers
- K = System capacity (max customers; default ∞)
- m = Population size (default ∞)
- D = Queue discipline (FCFS default; LCFS; Priority)

4.7 Single Server / Single Queue Model (M/M/1)

M/M/1: Poisson arrivals (rate λ), exponential service times (rate μ), single server, infinite capacity, FCFS discipline.

Condition for Stability

$$\rho = \lambda/\mu < 1 \quad \text{(Traffic Intensity; utilisation factor)}$$

If $\rho \geq 1$, queue grows indefinitely (system unstable).

M/M/1 Performance Measures — All Formulas

Measure	Formula	Meaning
Ls — Avg customers in system	$Ls = \lambda/(\mu-\lambda)$	Total in queue + being served
Lq — Avg customers in queue	$Lq = \lambda^2/[\mu(\mu-\lambda)]$	Waiting in queue (not being served)
Ws — Avg time in system	$Ws = 1/(\mu-\lambda)$	Total time: waiting + service
Wq — Avg time in queue	$Wq = \lambda/[\mu(\mu-\lambda)]$	Time waiting before service begins
P_0 — Prob server idle	$P_0 = 1 - \rho = 1 - \lambda/\mu$	Probability no customers present
P_n — Prob n in system	$P_n = (1-\rho) \times \rho^n$	Probability of exactly n customers
ρ — Server utilisation	$\rho = \lambda/\mu$	Fraction of time server is busy

Little's Law — Fundamental Relationship:

- $Ls = \lambda \times Ws$ (Customers in system = Arrival rate × Time in system)
- $Lq = \lambda \times Wq$ (Customers in queue = Arrival rate × Wait time in queue)
- $Ws = Wq + 1/\mu$ (System time = Queue time + Service time)
- These relationships hold for ALL stable queue disciplines and models (not just M/M/1)

MODULE 5: Network Analysis and Simulation (8 Hours)

Network analysis (CPM and PERT) provides systematic methods for planning, scheduling, and controlling complex projects consisting of interrelated activities. Simulation (Monte Carlo) models complex systems using random numbers to analyse uncertainty.

5.1 Networking Concepts

TERM / CONCEPT	MEANING & EXPLANATION
Activity	A specific task or job that takes time and resources; represented by an arrow →
Event (Node)	The start or end point of an activity; a point in time; no duration
Network Diagram	Arrow diagram showing all activities and their relationships (precedences)
Predecessor Activity	Activity that must be completed before a given activity can start
Successor Activity	Activity that can start only after the given activity is completed
Dummy Activity	Fictitious activity (zero duration) to show logical dependency without consuming resources; dotted arrow
AOA / AON	Activity-On-Arrow (traditional) vs Activity-On-Node (precedence diagram) representations

Rules for Drawing Network Diagrams

- Each activity is represented by ONE arrow; direction = left to right
- Each event (node) is represented by a circle; numbered uniquely
- Network must have ONE start event and ONE end event
- No two activities can share both the same start and end events (use dummy if needed)
- No loops or cycles are permitted in the network
- Dummy activities (dotted arrows, zero duration) used to show logical dependency without shared events

5.2 Critical Path Method (CPM)

Figure 5.1: CPM Network — Critical Path and Float Calculation

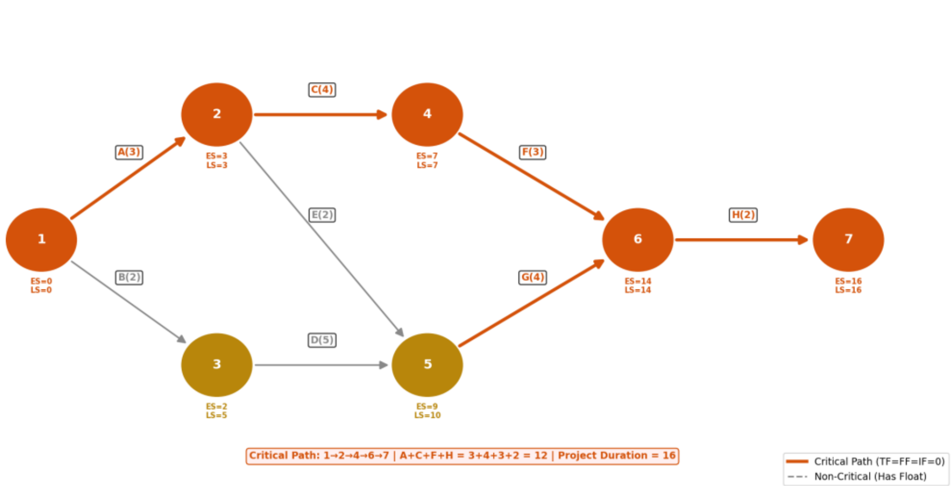

Figure 5.1: CPM Network — Critical path (red), non-critical activities (dashed), ES/LS values at each node

CPM identifies the CRITICAL PATH — the longest path through the network that determines the minimum project duration. Any delay in a critical activity directly delays the project.

Time Calculations — Forward and Backward Pass

Forward Pass (Earliest Times):

```
ES(j) = max[ES(i) + Duration(i,j)]   for all activities ending at j
ES of first event = 0;  EF = ES + Duration
```

Backward Pass (Latest Times):

```
LS(i) = min[LS(j) − Duration(i,j)]   for all activities starting at i
LS of last event = ES of last event (project duration)
```

Float (Slack) Calculations

Float Type	Formula	Meaning
Total Float (TF)	`TF = LS(j) − ES(i) − Duration(ij) = LS − ES`	Max delay without delaying project
Free Float (FF)	`FF = ES(j) − ES(i) − Duration(ij)`	Delay without affecting successor's ES
Independent Float (IF)	`IF = ES(j) − LS(i) − Duration(ij) (≥0)`	Delay not affecting any other activity

Critical Path Properties:

- Critical activities have TF = FF = IF = 0
- There can be MORE than one critical path in a network
- The critical path is the LONGEST path from start to end event
- Project duration = ES/LS of the last event = length of critical path
- Any delay in a critical activity increases project duration by same amount

5.3 PERT (Program Evaluation and Review Technique)

Figure 5.2: PERT — Time Estimates and Probability of Project Completion

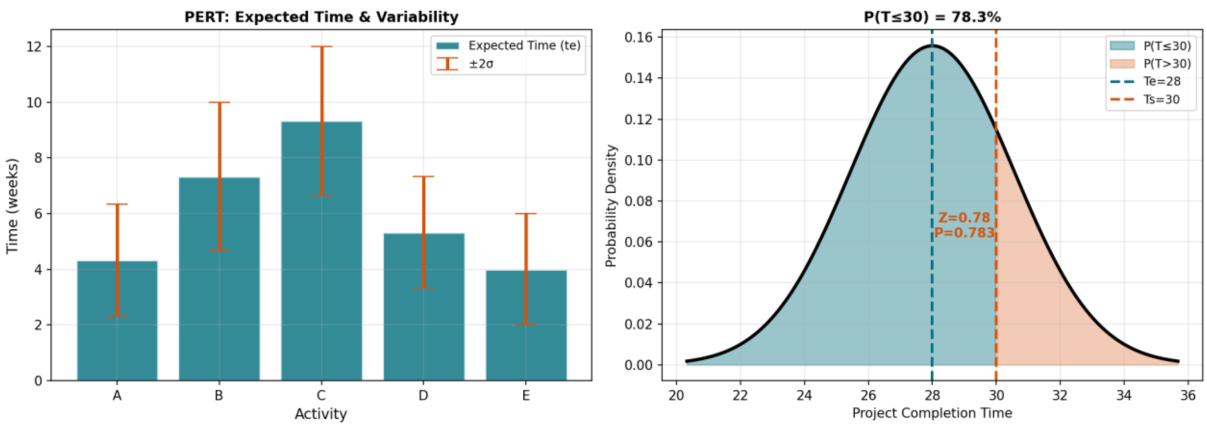

Figure 5.2: PERT — Expected Time with variability (left) and Probability of meeting scheduled date (right)

PERT handles uncertainty in activity durations using THREE time estimates: optimistic (t_o), most likely (t_m), and pessimistic (t_p). This is appropriate for research, defence, and other projects with uncertain durations.

Three Time Estimates and Expected Time

t_o = Optimistic time (fastest possible; everything goes perfectly)

t_m = Most Likely time (normal conditions; most probable duration)

t_p = Pessimistic time (worst case; maximum likely time if everything goes wrong)

Expected Time: $te = (t_o + 4t_m + t_p) / 6$ [Beta distribution formula]

Variance: $\sigma^2 = [(t_p - t_o) / 6]^2$

Probability of Meeting Scheduled Date

Since the project duration Te follows approximately a Normal distribution (sum of independent activities on critical path):

$Te = \Sigma\, te$ (critical path activities) [Expected project duration]

$\sigma^2_p = \Sigma\, \sigma^2_i$ (critical path activities) [Project variance]

$Z = (Ts - Te) / \sigma_p$ [Standard normal statistic]

$P(T \leq Ts) = \Phi(Z)$ [From standard normal table]

Where Ts = scheduled completion date, Φ(Z) = cumulative normal probability.

CPM	PERT
Used for well-defined, repetitive projects	Used for new, R&D, uncertain projects
SINGLE time estimate per activity	THREE time estimates (t_o, t_m, t_p) per activity
Deterministic (no uncertainty)	Probabilistic (handles uncertainty)
Focuses on time-cost tradeoff (crashing)	Focuses on probability of completion
Examples: Construction, maintenance	Examples: Software dev., R&D, defence
Float/Slack calculation important	Variance and probability calculation important

5.4 Simulation — Monte Carlo Method

Figure 5.3: Monte Carlo Simulation — Steps and Results

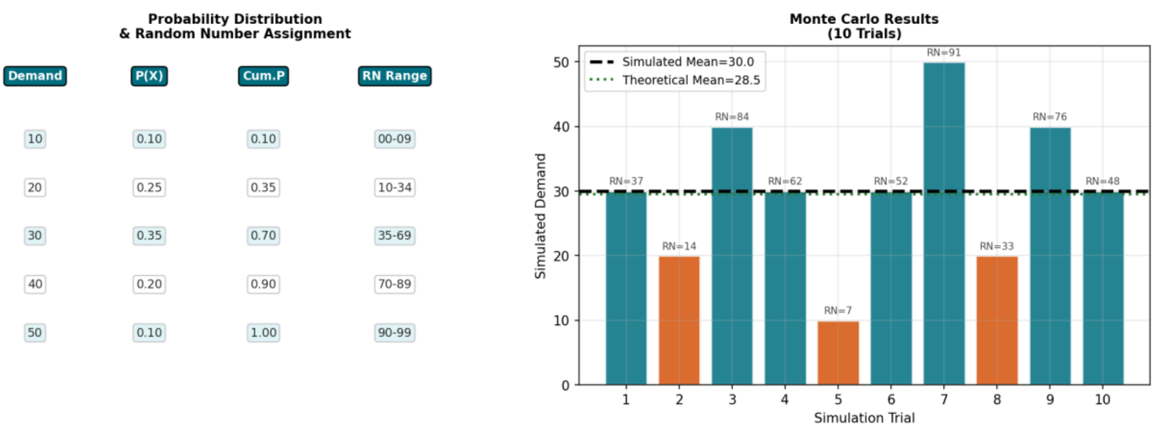

Figure 5.3: Monte Carlo Simulation — Probability table setup and simulation results (10 trials)

Simulation is a technique that creates an artificial environment to imitate the operations of a real-world system. Monte Carlo simulation uses RANDOM NUMBERS to simulate probabilistic events and study system behaviour over time.

Steps in Monte Carlo Simulation

57. Define the probability distribution for the key variable (demand, service time, etc.)
58. Build a cumulative probability distribution (CDF) from the frequency or probability table
59. Assign random number ranges to each outcome proportionally (e.g., P=0.25 → 25 numbers)
60. Generate or select random numbers (from random number table, computer, or dice)
61. Map each random number to its corresponding outcome using the RN assignment table
62. Record the simulated outcome for each trial; repeat for n trials
63. Analyse results: calculate simulated mean, compare with theoretical mean, draw conclusions

Random Number Assignment Rule:

- Proportional assignment: If P(X=10) = 0.10 → assign 10 random numbers (00–09)
- If P(X=20) = 0.25 → assign 25 random numbers (10–34)
- Random numbers from table: use two-digit (00–99) for precision
- The larger the number of trials (n), the closer simulated results approach theoretical values
- Monte Carlo gives APPROXIMATE answers — accuracy improves with more simulations

Applications of Monte Carlo Simulation

- Inventory Management — Simulate demand and lead time variability to optimise stock levels
- Project Management — Simulate activity durations for probabilistic project completion
- Financial Analysis — Simulate stock prices, interest rates, and portfolio returns
- Queuing Systems — Simulate arrivals and service to study congestion before implementing
- Risk Analysis — Evaluate likelihood of different project outcomes under uncertainty

5.5 Summary — OR Techniques at a Glance

Module	Technique	Business Application
1 — LPP	Linear Programming (Graphical + Simplex)	Product mix; resource allocation; profit maximisation
1 — LPP	Duality (Formulation)	Shadow prices; sensitivity analysis
2 — Transport	NWCR, LCM, VAM, MODI	Logistics; distribution network optimisation
2 — Assignment	Hungarian Method	Job scheduling; staff-to-task allocation; routing
3 — Sequencing	Johnson's Algorithm (2M, 3M)	Production scheduling; job shop management
3 — Replacement	Individual & Group Replacement	Capital budgeting; maintenance management
4 — Game Theory	Saddle Point, Mixed Strategy	Pricing; competitive strategy; bidding
4 — Queuing	M/M/1 Model (all 7 formulas)	Bank counters; call centres; service design
5 — Network	CPM (TF, FF, IF)	Construction; IT project scheduling

5 — Network	PERT (3 estimates, probability)	R&D projects; uncertainty management
5 — Simulation	Monte Carlo (random numbers)	Demand forecasting; risk analysis; inventory