Master of Science in Mathematics

(M.Sc. Mathematics)

Industrial Mathematics
(DMSMDS101T24)

Self-Learning Material
(SEM 1)

Jaipur National University
Centre for Distance and Online Education

Established by Government of Rajasthan
Approved by UGC under Sec 2(f) of UGC ACT 1956
&
NAAC A+ Accredited

PREFACE

Industrial mathematics stands at the intersection of theory and practice, blending the abstract beauty of mathematical concepts with the pragmatic demands of real-world applications. From optimizing manufacturing processes to designing efficient supply chains, from modeling financial markets to analyzing big data, industrial mathematics plays a pivotal role in shaping the modern world.

This book is a testament to the vibrant and multifaceted nature of industrial mathematics, offering a comprehensive and accessible exploration of its theory, methods, and applications. It is designed to serve as a valuable resource for students, researchers, and practitioners seeking to harness the power of mathematics to solve practical problems in industry and beyond.

TABLE OF CONTENTS

UNIT - 1

Introduction to Finite Difference Schemes

Learning objectives

Differential equation solutions can be approximated numerically using finite difference approaches. Studying finite difference schemes usually aims to teach students about discretization, numerical methods, understanding differential equations, and finite differences, among other things.

Structure

1.1 Basics of Partial Differential Equations

Partial Differential Equations (PDEs) are equations that involve partial derivatives of functions of multiple variables. They find widespread applications in various fields such as physics, engineering, and finance. Here are the key concepts covered in this section:

Definition: PDEs are classified based on their order, linearity, and homogeneity. Common types include elliptic, parabolic, and hyperbolic equations.

Classification of Second Order Equations:

The general linear partial differential equation of the second order in two independent variables is of the form

$$A(x,y)\frac{\partial^2 u}{dx^2}+B(x,y)\frac{\partial^2 u}{\partial x\partial y}+C(x,y)\frac{\partial^2 u}{\partial y^2}+\left(x,y,u\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}\right)=0$$

Such a partial differential equation is said to be

(i) Elliptic when $B^2 - 4AC < 0$

(ii) Parabolic when $B^2 - 4AC = 0$

(iii) Hyperbolic when $B^2 - 4AC > 0$.

Examples of PDEs: Examples include the heat equation, wave equation, and Laplace's equation. Each type of equation represents different physical phenomena and has distinct properties.

1.2 Overview of Finite Difference Method

Finite Difference Method (FDM) is numerical techniques used to solve differential equations by discretizing the domain into a grid and approximating derivatives using finite difference approximations.

Discretization: The continuous domain is divided into discrete points or nodes, forming a grid. Differential operators are approximated using finite difference formulas at these grid points.

Derivation of Finite Difference Equations: By substituting finite difference approximations into the original differential equation, a system of algebraic equations is obtained, which can be solved numerically.

1.3 Explicit and Implicit Schemes

Explicit and Implicit Finite Difference Schemes are two common approaches to solving the discretized equations. They differ in how they treat the unknown values at the next time step. Key points covered in this section include:

Explicit Schemes: In explicit schemes, the value at the next time step is computed explicitly in terms of known values at the current time step. These schemes are easy to implement but may be subject to stability limitations.

Implicit Schemes: Implicit schemes involve solving a system of equations at each time step, where the unknown values at the next time step are implicit functions of both current and future values. Implicit schemes are often unconditionally stable but require solving linear or nonlinear equations, which can be computationally intensive.

The universal first-order equation with degree n has the following form:

1) $a_0(x, y)(y')n + a1(x, y)(y')n - 1 + ... + an-1(x, y)\, y' + an(x, y) = 0$

2) $a_0(x, y)\ p_n + a1(x, y)pn - 1 + ... + an-1(x, y)\, p + an(x, y) = 0$

Example 1:

Form the partial differential equation by eliminating the arbitrary constants

$$z = ax + by + a^2 + b^2.$$

Solution:

Given

$z = ax + by + a^2 + b^2$ ....(1)

Differentiating (1) partially w.r.t. ‘x

$\partial z/ \partial x = a$

i.e., $p = a$ ....(2)

Differentiating (1) partially w.r.t. ‘y’

$\partial z/ \partial y = b$

i.e., $q = b$ ....(3)

From (2) and (3) a = p and b = q

Substituting these values in (1),

We get, $z = px + qy + p^2 + q^2$.

Example 2:

Form the PDE by eliminating the arbitrary constants a and b

$$z = (x + a)(y + b)$$

Solution:

Given

$$z = (x + a)(y + b) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Differentiating (1) partially w.r.t. 'x'

$$\partial z / \partial x = y + b$$

$$\text{i.e., } p = y + b \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Differentiating (1) partially w.r.t. 'y'

$$\partial z / \partial y = x + a$$

$$\text{i.e., } q = x + a \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

From (2) and (3), $x + a = q$ and $y + b = p$

Substituting these values in (1),

We get, $z = pq$.

Formation of partial differential equation by elimination of arbitrary Functions:

The elimination of one arbitrary function from a given relation gives a partial differential equation of first order while elimination of two arbitrary functions from a given relation gives a second or higher order partial differential equation

Example 3:

Form PDE by eliminating arbitrary function f and g

$z = f(x + ay) + g(x - ay)$.

Solution: Given

$$z = f(x + ay) + g(x - ay) \qquad \ldots\ldots\ldots\ldots (1)$$

Differentiating (1) partially w.r.t 'x'

$$\partial z / \partial x = f'(x + ay) + g'(x - ay)$$

$$\text{i.e., } p = f' + g' \qquad \ldots\ldots\ldots\ldots (2)$$

Differentiating (1) partially w.r.t 'y'

$$\partial z / \partial y = af'(x + ay) - ag'(x - ay)$$

$$\text{i.e., } q = af' - ag' \qquad \ldots\ldots\ldots\ldots (3)$$

Differentiating (2) partially w.r.t ‘x’

$$\partial^2 z/\partial x^2 = f'' + g''$$

$$i.e., r = f'' + g'' \ldots\ldots\ldots\ldots (4)$$

Differentiating (3) partially w.r.t ‘y’

$$\partial 2 z/\partial y 2 = a2 f'' + a2 g''$$

$$i.e., t = a2(f'' + g'') \ldots\ldots\ldots (5)$$

Using (4), we get, $t = a^2 r$.

Lagrange’s Linear Equations:

In numerical analysis, Lagrange's linear equations typically refer to a particular method for solving systems of linear equations.

One such method is Gaussian elimination with partial pivoting, often called the Gauss–Jordan method. It's a systematic way to transform a system of linear equations into an equivalent system with a triangular matrix, making it easier to solve.

Here's a basic outline of the Gauss-Jordan method:

1. **Forward Elimination**: Convert the system of equations into triangular form by eliminating variables below the diagonal.
2. **Back Substitution**: Solve for the variables starting from the bottom equation and working upward.

Lagrange's equations might also refer to a formulation of linear equations within the context of optimization problems, where Lagrange multipliers are used to incorporate constraints into the objective function. This method involves setting up a system of linear equations based on the conditions given by the Lagrange multipliers and then solving them to find the optimal solution.

Solution of Partial Differential Equations:

The solution of partial differential equations (PDEs) is a vast and complex topic with applications spanning across various fields such as physics, engineering, and mathematics. Here's a brief overview:

1. **Classification of PDEs**: PDEs can be classified into various types based on their order, linearity, and coefficients. Common types include:

- **Linear vs. Nonlinear**: PDEs are linear if they can be expressed as linear combinations of the dependent variable and its partial derivatives. Otherwise, they are nonlinear.
- **Order**: The order of a PDE is the highest order of derivative present in the equation.
- **Elliptic, Parabolic, and Hyperbolic PDEs**: Depending on the nature of the principal part of the equation, PDEs can be classified into these three categories, each with its own characteristic behavior.

2. **Analytical Methods**: Some PDEs can be solved analytically using techniques such as separation of variables, integral transforms (e.g., Fourier transform, Laplace transform), and method of characteristics. However, analytical solutions are often limited to simple geometries and boundary conditions.
3. **Numerical Methods**: For many practical problems, analytical solutions are not feasible. In such cases, numerical methods are employed. These include:
    - **Finite Difference Method (FDM)**: The domain is discretized, and finite difference approximations are used to discretize the derivatives. The resulting system of algebraic equations is then solved iteratively.
    - **Finite Element Method (FEM)**: The domain is divided into smaller elements, and the PDE is approximated over each element using basis functions. The resulting system of equations is solved numerically.
    - **Finite Volume Method (FVM)**: The domain is divided into control volumes, and the integral form of the PDE is solved over each control volume.

Lagrange Multiplier Method:

The Lagrange multiplier method is a technique used in mathematical optimization to locate the local maxima and minima of a function under equality constraints, or where the values of the variables must precisely satisfy one or more equations. The fundamental concept is to rewrite the problem in a way that preserves the applicability of the derivative test for unconstrained problems. Whether or not stationary points are maxima, minima, or saddle points depends on the definiteness of the bordered Hessian matrix once those points have been determined from the first-order required conditions.

Basic Concept:

Suppose we want to find the extrema (maximum or minimum) of a function f(x) subject to the constraint g(x) = 0. The Lagrange Multiplier Method introduces an auxiliary variable, called the Lagrange multiplier, to transform the constrained optimization problem into a form that can be more easily analyzed.

Steps of the Lagrange Multiplier Method:

1. **Form the Lagrangian**: Define the Lagrangian function $L(x, \lambda)$ as follows:

$L(x,\lambda) = f(x) - \lambda g(x)$

Here, λ is the Lagrange multiplier.

2. **Compute Partial Derivatives**: Calculate the partial derivatives of L with respect to each variable in x and the Lagrange multiplier λ.
3. **Set Equations to Zero**: Set these partial derivatives equal to zero to form a system of equations:

$$\partial L/\partial x_i = 0 \text{ for each i}$$

$$\partial L/\partial \lambda = 0$$

The first set of equations ensures that f(x) is stationary with respect to changes in x, while the second equation enforces the constraint g(x) = 0.

4. **Solve the System of Equations**: Solve this system of equations to find the values of x and λ that satisfy all the conditions.

Let's consider a simple example to illustrate the Lagrange Multiplier Method.

Example 4: Maximize f(x, y)=x^2+y^2 subject to the constraint g(x, y)=x+y−1=0.

1. Form the Lagrangian:

$$L(x, y, \lambda) = x^2+y^2-\lambda(x+y-1)$$

2. Compute Partial Derivatives:

$$\partial L/\partial x = 2x-\lambda = 0$$

$$\partial L/\partial y = 2y-\lambda = 0$$

$$\partial L/\partial \lambda = -(x+y-1) = 0$$

3. Set Equations to Zero:

$2x - \lambda = 0 \Rightarrow \lambda = 2x$

$2y - \lambda = 0 \Rightarrow \lambda = 2y$

$x + y - 1 = 0$

4. Solve the System of Equations:

From $\lambda = 2x$ and $\lambda = 2y$, we get $2x = 2y \Rightarrow x = y$.

Substitute $x = y$ into the constraint $x+y-1 = 0$:

$x+x-1 = 0 \Rightarrow 2x = 1 \Rightarrow x = 1/2, y = 1/2$

Therefore, the maximum value of $f(x, y)=x^2+y^2$ subject to the constraint $x+y = 1$is attained at (1/2,1/2).

1.4 Summary

A numerical method called the Finite Difference Method (FDM) uses difference equations to approximate differential equations and solve them. It entails substituting finite differences for the derivatives and discretizing the continuous domain into a grid of points.

For example, the first derivative f'(x) at a point can be approximated by $(f(x+h)-f(x))/h$, where h is the grid spacing.

1.5 Keywords

- Boundary Conditions
- Grid
- Explicit Scheme
- Implicit Scheme
- Finite Element Method

1.6 Self Assessment Questions

1. What is the fundamental idea behind finite difference schemes?
2. How do finite difference schemes approximate derivatives in differential equations?
3. Explain the significance of discretization in numerical methods.
4. What are the key steps involved in implementing a finite difference scheme computationally?

5. Describe the data structures and algorithms commonly used in finite difference schemes.

1.7 Case Study

Thermal Analysis Using Finite Difference Schemes:

A manufacturer with expertise in creating high-end electrical products. Ensuring that the components of their gadgets work within acceptable temperature ranges to preserve maximum performance and dependability is one of their core problems. They choose to use thermal analysis finite difference techniques as a solution to this problem.

Question: Under different working situations, stimulate the temperature distribution inside the electrical equipment. To optimize the design and cooling solutions, this entails assessing hotspot locations, thermal gradients, and heat dissipation.

1.8 References

1. "Numerical Recipes: The Art of Scientific Computing" by William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery.
2. "Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems" by Randall J. LeVeque.

UNIT - 2

Finite Difference Schemes for Initial and Boundary Value Problems

Learning objectives

For starting and boundary value issues, learning objectives for finite difference schemes usually include both theoretical knowledge and practical abilities. Knowing Finite Difference Methods (FDM), Numerical Analysis, Techniques for Discretization, and their application to various problems.

Structure

2.1 Forward-Time Central-Space (FTCS) Scheme:

Finite difference schemes are numerical methods used to solve differential equations by approximating the derivatives with finite differences. These schemes are widely used for solving initial value problems (IVPs) and boundary value problems (BVPs) in various fields of science and engineering.

Forward-Time Central-Space (FTCS) is a numerical method used to solve partial differential equations, particularly the heat equation or other parabolic PDEs. This method discretizes time using a forward difference and space using a central difference. Here's a detailed explanation of the FTCS method:

Problem Context

Consider the one-dimensional heat equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

where $u = u(x, t)$ is the temperature distribution, α is the thermal diffusivity, x is the spatial coordinate, and t is time.

Forward-Time Discretization

Using the forward difference for the time derivative, we have:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}$$

Central-Space Discretization

Using the central difference for the second spatial derivative, we have:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}$$

FTCS Scheme

Combining these discretizations, the heat equation becomes:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}$$

Solving for u_i^{n+1}, we get the FTCS update formula:

$$u_i^{n+1} = u_i^n + \frac{\alpha \Delta t}{(\Delta x)^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

Stability Considerations

The stability of the FTCS scheme is determined by the Courant-Friedrichs-Lewy (CFL) condition. For the heat equation, the stability condition is:

$$\frac{\alpha . \Delta t}{(\Delta x)^2} \leq \frac{1}{2}$$

This condition must be satisfied to ensure that the numerical solution remains stable.

The Forward-Time Central-Space (FTCS) method is a straightforward and widely used numerical method for solving parabolic partial differential equations like the heat equation. While simple to implement, care must be taken to ensure stability through appropriate choices of the time step and spatial step sizes.

Here's an overview of some common finite difference schemes for both types of problems:

Initial Value Problems (IVPs):

Forward Difference Method: The Forward Difference Method approximates the derivative of a function f(x) using the values of f at discrete points. For a function sampled at points x_i with a uniform spacing h, the forward difference approximation of the first derivative at x_i is given by:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{h}$$

Here, h is the step size or the difference between consecutive points x_i and x_{i+1}.

Formulae:

1. First Derivative:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{h}$$

2. Second Derivative: The second derivative can be approximated using the forward difference twice:

$$f''(x_i) \approx \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2}$$

Example 1:

Let's solve the simple ODE $\frac{dy}{dx} = y$ with the initial condition $y(0) = 1$ using the Forward difference method.

Solution:

Forward Difference Method:

The Forward Difference Method for ODEs is a numerical method where the derivative $\frac{dy}{dx}$ is approximated using a forward difference:

$$\frac{dy}{dx} \approx \frac{y_{i+1} - y_i}{\Delta x}$$

Given the ODE $\frac{dy}{dx} = y$, we can write

$$y_{i+1} = y_i + \Delta x . y_i$$

This simplifies to:

$$y_{i+1} = y_i (1 + \Delta x)$$

Explanation:

1. **Parameters and Initialization:**
   - `x₀` and `y₀` are the initial conditions x = 0 and y = 1.
   - `xf` is the final x value up to which we want to solve the ODE.
   - `dx` is the step size for x.
   - `n_steps` is the number of steps required to reach `xf` from `x0` with step size `dx`.
2. **Arrays:**
   - x is an array of x values from `x₀` to `xf` with `n_steps` points.
   - y is an array to store the numerical solution of y at each x.
3. **Forward Difference Method:**
   - Iterate over the number of steps.
   - Apply the forward difference method to compute y at each step.
4. **Analytical Solution:**

   For comparison, compute the analytical solution $y = e^x$.
5. **Plotting:**

   Plot the numerical solution and the analytical solution for comparison.

Conclusion

This example shows how to use the Forward Difference Method to solve the simple ODE $\frac{dy}{dx}=y$. The numerical solution closely follows the analytical solution $y=e^x$, demonstrating the accuracy of the method for small step sizes.

Discretize the Domain: Choose a step size h and points

$x_0, x_1, \ldots, x_n$ where $x_{i+1} = x_i + h$

Apply the Forward Difference Method:

$$y_{i+1} = y_i + hy_i = y_i(1+h)$$

Starting with $y_0 = 1$ and $x_0 = 0$, iterate to find the values of y at subsequent points:

$$\begin{aligned} y_1 &= y_0(1+h) \\ y_2 &= y_1(1+h) = y_0(1+h)^2 \\ &\vdots \\ y_n &= y_0(1+h)^n \end{aligned}$$

Backward Difference Method:

The Backward Difference Method is another numerical approach to solve differential equations. When combined with the Forward-Time Central-Space (FTCS) scheme for spatial Discretization, it can be used for solving parabolic partial differential equations (PDEs).

To illustrate, let's consider solving the heat equation using a combination of the FTCS method for spatial discretizations and the Backward Difference Method for time discretizations.

Heat Equation

The heat equation in one dimension is given by:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

where $u = u(x,t)$ is the temperature distribution, and α is the thermal diffusivity.

1. **Time Discretization (Backward-Time)**:

- We discretize the time domain using the backward difference method.
- Let Δt be the time step size, and define time levels $t^n = n\Delta t$ for $n = 0,1,2,\ldots,M$.
- The time derivative is approximated as:

$$\frac{\partial u}{\partial t} \approx \frac{u_i^n - u_i^{n-1}}{\Delta t}$$

Combined FTCS with Backward-Time Scheme

Combining these discretizations, the heat equation becomes:

$$\frac{u_i^n - u_i^{n-1}}{\Delta t} = \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}$$

Rearranging for u_i^n, we get:

$$u_i^n - \frac{\alpha \Delta t}{(\Delta x)^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) = u_i^{n-1}$$

This can be rewritten as a system of linear equations:

$$-\frac{\alpha \Delta t}{(\Delta x)^2} u_{i-1}^n + \left(1 + 2\frac{\alpha \Delta t}{(\Delta x)^2}\right) u_i^n - \frac{\alpha \Delta t}{(\Delta x)^2} u_{i+1}^n = u_i^{n-1}$$

Explanation

1. **Parameters and Initialization**:
   - `alpha`, ‘`L`’, ‘`T`’, ‘`nx`’, and ‘`nt`’ define the physical parameters and the resolution of the problem.
   - ‘`dx`’ and ‘`dt`’ are the spatial and time step sizes.
   - The initial condition is set as a heat pulse in the center.
2. **Matrix Assembly**:
   - Matrix ‘A’ represents the coefficients of the linear system derived from the discretizations scheme.
   - Boundary conditions are implemented by setting the first and last diagonal elements of A to 1.
3. **Time-Stepping Loop**:
   - For each time step, solve the linear system Au = b where b is the solution from the previous time step.
4. **Plotting**:
   - Plot the final temperature distribution after the last time step.

Conclusion

The Forward-Time Central-Space Backward Difference Method combines the accuracy of central differences for spatial discretizations with the stability of backward differences for time discretizations. This method is particularly useful for solving parabolic PDEs like the heat equation, providing stable and accurate solutions.

Central Difference Method: This method uses the average of forward and backward difference approximations to approximate the derivative, providing a more accurate approximation compared to forward or backward differences alone.

Runge-Kutta Methods:

Introduction: These are higher-order finite difference methods that use weighted averages of function values at multiple points in time to improve accuracy. The most commonly used are the RK2 and RK4 methods.

A series of iterative techniques for estimating the solutions of ordinary differential equations (ODEs) is known as the Runge-Kutta methods. Because of their stability and simplicity of use, they are more accurate than straightforward techniques like Euler's method and are frequently employed.

.

Runge-Kutta Methods or R K Methods:

The general form of an ODE is:

$$\frac{dy}{dx}=f(x,y)$$

With the initial condition $y(x_0)=y_0$

Second-Order Runge-Kutta Method (RK2)

One of the simplest RK methods is the second-order RK method (RK2), also known as the midpoint method. It is given by:

1. Calculate the intermediate slope:

$$k_1=f(x_n,y_n)$$

2. Calculate the slope at the mid point:

$$k_2=f\left(x_n+\frac{\Delta x}{2},y_n+\frac{\Delta x}{2}k_1\right)$$

3. Update the value of y:

$$y_{n+1} = y_n + \Delta x . k_2$$

Fourth-Order Runge-Kutta Method (RK4)

The fourth-order Runge-Kutta method (RK4) is the most commonly used and provides a good balance between accuracy and computational effort. It is given by:

1. Calculate the initial slope:

$$k_1 = f(x_n, y_n)$$

2. Calculate the slope at the midpoint using k_1:

$$k_2 = f\left(x_n + \frac{\Delta x}{2}, y_n + \frac{\Delta x}{2} k_1\right)$$

3. Calculate another slope at the midpoint using k_2**:**

$$k_3 = f\left(x_n + \frac{\Delta x}{2}, y_n + \frac{\Delta x}{2} k_2\right)$$

4. Calculate the slope at the next point using k_3:

$$k_4 = f(x_n + \Delta x, y_n + \Delta x k_3)$$

5. Update the value of y:

$$y_{n+1} = y_n + \frac{\Delta x}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Example 2:

Solve $\frac{dy}{dx} = y$ with the initial condition $y(0) = 1$ using Runge-Kutta Fourth Order Method**.**

Solution: Let's solve the ODE $\frac{dy}{dx} = y$ with the initial condition $y(0) = 1$ using the RK4 method.

For the solution of ODE $\frac{dy}{dx} = y$ by using the RK4 method is given by:

Calculates y at x_{n+1} from y at x_n using the following steps:

1. Compute the value of y at $x = 0.1$ with a step size $\Delta x = 0.1$.

2. Initial conditions: $x_0 = 0$ and $y_0 = 1$
3. Calculate y at $x = 0.1$

$$k_1 = f(x_0, y_0) = y_0 = 1$$

$$k_2 = f\left(x_n + \frac{\Delta x}{2}, y_n + \frac{\Delta x}{2} k_1\right)$$

$k_2 = f(x_0 + \frac{0.1}{2}, y_0 + \frac{0.1}{2}.1)$

$k_2 = f(1 + 0.05, 1 + 0.05.1)$

$k_2 = f(1.05, 1.05) = 1.05$

$k_3 = f(x_1 + \frac{\Delta x}{2}, y_0 + \frac{\Delta x}{2}. k2)$

$k_3 = f(1.05, 1.1025) = 1.1025$

$k_4 = f(x_0 + \Delta x, y_0 + \Delta x . k_3)$

$k_4 = f(1 + 0, 1.1 \cdot 1.1025)$

$k_4 = f(1, 1.1 .10525) = 1.1125$

$y_{n+1} = y_n + \frac{\Delta x}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

$y_{0+1} = y_0 + \frac{\Delta x}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

$y_1 = y_0 + \frac{0.1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

$y_1 = 1 + 0.16(1 + 2 \cdot 1.05 + 2 \cdot 1.0525 + 1.1125)$

$y_1 = 1 + \frac{0.1}{6}(6.4175)$

$y_1 = 1 + 0.1069583333$

$y_1 = 1.1069583333$

We can repeat these steps to find y for other values of x.

Boundary Value Problems (BVPs):

Finite Difference Method (FDM): FDM is commonly used to discretize the spatial domain of a BVP and then solve the resulting system of algebraic equations. Various finite difference approximations can be used for spatial derivatives, such as central differences.

Shooting Method: In this method, the BVP is converted into an initial value problem by guessing initial values for the unknown boundary conditions and then solving the resulting IVP using a suitable finite difference scheme.

Finite Element Method (FEM): While not strictly a finite difference method, FEM discretizes the domain into elements and approximates the solution within each element using piecewise polynomial basis functions. Finite difference approximations may still be used to solve the resulting system of algebraic equations.

Relaxation Methods: These iterative methods, such as the Jacobi or Gauss-Seidel methods, can be used to solve BVPs by iteratively updating the solution until convergence to the correct boundary conditions is achieved.

The Forward-Time Central-Space (FTCS) scheme is a numerical method used for solving partial differential equations (PDEs), particularly in the context of computational fluid dynamics (CFD) and heat transfer problems.

In essence, FTCS is a finite-difference method where both time and space derivatives are approximated using forward differences for time and central differences for space.

Here's how it typically works:

Discretization: The domain of the problem is discretized in both space and time. This means that the continuous spatial and temporal dimensions are divided into discrete intervals or grid points.

Forward Difference in Time: The time derivative term in the PDE is approximated using a forward difference scheme, meaning that the value of the function at the next time step is approximated based on its current value and the rate of change.

Central Difference in Space: The spatial derivative term in the PDE is approximated using a central difference scheme, which considers the values of the function at neighbouring points on the grid.

Combination: These discretization's are then combined to form a numerical scheme that allows the evolution of the solution over time to be computed iteratively.

The FTCS scheme is simple and straightforward to implement, but it's important to note that it may not always be stable or accurate for all types of problems. In particular, it can be subject to stability constraints that limit the size of the time step that can be used, and it may suffer from numerical diffusion, where sharp gradients in the solution are smoothed out over time.

2.2 Backward Euler Scheme

The Backward Euler scheme is a numerical method commonly used to solve ordinary differential equations (ODEs) or partial differential equations (PDEs) in time-dependent problems. It belongs to the family of implicit finite difference methods and is particularly useful for stiff problems, where the explicit schemes might be unstable or inefficient due to the small time steps required.

In the Backward Euler scheme:

Discretization in Time: The time domain is discretized into time steps of equal size, denoted by Δt.

Implicit Formulation: Unlike explicit schemes where the future value of the solution depends only on the current time step, in the Backward Euler scheme, the future value depends on both the current and future time steps.

Approximation of Derivatives: The derivative terms in the differential equation are approximated using backward differences, where the value at the next time step is approximated in terms of the future value of the solution.

Algebraic Equation: This results in an algebraic equation involving the solution values at multiple time steps. The equation is usually nonlinear and must be solved iteratively, typically using methods like Newton's method.

Mathematically, for an ODE of the form $\frac{dy}{dt}=f(t,y)$, the Backward Euler scheme can be written as:

$$y_{n+1}=y_n+\Delta t\cdot(t_{n+1},y_{n+1})\,y_{n+1}=y_n+\Delta t\cdot f(t_{n+1},y_{n+1})$$

Here, yn yn and $yn+1$ $yn+1$ are the values of the solution at time steps tn tn and $tn+1$ $tn+1$, respectively.

Similarly, for a simple linear ODE $\frac{dy}{dt}=ay$, the Backward Euler scheme becomes:

$$y_{n+1}=y_n-a\,\Delta t\,y_{n+1}=1-a\Delta t\,y_n$$

The Backward Euler scheme is unconditionally stable, meaning it can handle large time steps without stability issues. However, it introduces implicitness, requiring the solution of nonlinear equations at each time step, which can increase computational cost compared to explicit schemes. Nonetheless, it's a valuable tool for stiff problems where stability and accuracy are crucial.

2.3 Cranks-Nicolson Scheme

A numerical technique for solving time-dependent partial differential equations (PDEs) is the Crank-Nicolson scheme. This finite difference approach provides second-order temporal precision by combining aspects of the forward Euler and backward Euler schemes. The approach is generally applicable to diffusion and convection-diffusion problems, especially when stability and precision are crucial.

Here's how the Crank-Nicolson scheme works:

Discretization in Time and Space: Like other finite difference methods, the domain of the problem is discretized both in time and space. Time is divided into equally spaced intervals (Δt), while space is divided into equally spaced grid points.

Central Difference for Time: Unlike the forward Euler or backward Euler schemes, the Crank-Nicolson scheme uses a central difference approximation for the time derivative. This results in a second-order accurate approximation in time.

Central Difference for Space: Similarly, central difference approximations are used for spatial derivatives.

Implicitness: The Crank-Nicolson scheme is semi-implicit, meaning it evaluates the time derivative terms at both the current and next time steps. This results in a system of linear equations, which is typically easier to solve compared to the nonlinear equations encountered in fully implicit methods.

Mathematically, for a simple linear diffusion equation $\partial u \partial t = D \partial 2u \partial x2 \partial t \partial u = D \partial x2 \partial 2u$, the Crank-Nicolson scheme can be written as:

$uin+1-uin\Delta t=D2(ui-1n+1-2uin+1+ui+1n+1\Delta x2+ui-1n-2uin+ui+1n\Delta x2)\Delta tuin+1$
$-uin=2D(\Delta x2ui-1n+1-2uin+1+ui+1n+1+\Delta x2ui-1n-2uin+ui+1n)$

This equation represents a tridiagonal system of linear equations, which can be efficiently solved using techniques such as the Thomas algorithm.

The Crank-Nicolson scheme offers several advantages, including second-order accuracy in time, unconditional stability for linear problems, and reduced numerical diffusion compared to explicit methods. However, it may require more computational resources due to the solution of a linear system at each time step. Overall, it's a versatile and widely used scheme for time-dependent PDEs in various fields, including heat transfer, fluid dynamics, and quantum mechanics.

2.4 Alternating Direction Implicit (ADI) Methods

The Alternating Direction Implicit (ADI) method is a numerical technique used to solve partial differential equations (PDEs), particularly those describing parabolic or elliptic problems. It's an iterative approach that decomposes the problem into smaller, one-dimensional problems, which are easier to solve, and alternates between solving them in different directions.

Here's how the ADI method typically works:

Problem Decomposition: The PDE problem is decomposed into a sequence of one-dimensional problems along different coordinate directions. For example, in a two-dimensional problem, the PDE is solved along the x-direction and then the y-direction in alternating steps.

Implicit Time Integration: Within each direction, implicit time integration is applied, typically using a backward Euler scheme or Crank-Nicolson scheme. This ensures stability, particularly for stiff problems.

Alternating Directions: After each time step, the solution is updated alternately in the x-direction and y-direction. This alternating approach helps to decouple the equations and simplifies the solution process.

Tridiagonal Matrix Solution: At each time step in each direction, the resulting system of equations is typically tridiagonal, making it relatively straightforward to solve using efficient numerical techniques such as the Thomas algorithm.

Iteration: The process of updating the solution in alternating directions is repeated until convergence is achieved. Convergence criteria can be based on reaching a certain tolerance or a maximum number of iterations.

The ADI method offers several advantages:

Stability: The implicit time integration along with the alternating direction approach ensures stability, even for stiff problems.

Efficiency: By decomposing the problem into smaller one-dimensional problems, the computational cost is often reduced compared to solving the full problem in one go.

Accuracy: ADI methods can provide high accuracy, particularly when combined with higher-order time integration schemes or spatial discretizations methods.

However, there are also some limitations:

Dimensionality: ADI methods are most commonly used for problems in two or three dimensions. Extension to higher dimensions can become computationally expensive.

Boundary Conditions: Handling boundary conditions can be more complex, especially when they vary in different directions.

Overall, the ADI method is a powerful tool for solving certain types of PDEs, particularly those with parabolic or elliptic behaviour, and it's widely used in various fields, including computational fluid dynamics, heat transfer, and quantum mechanics.

2.5 Summary

The discretizations of differential equations using finite difference approximations into algebraic equations are a problem. Here's a succinct rundown:

- Discretization: A grid in both space and time is used to discretize differential equations. The forward, backward, and centre differences finite difference formulae are used to approximate spatial derivatives. Discrete techniques such as Crank-Nicolson, forward Euler, and reverse Euler are used to discretize time derivatives.
- Distribution: The equations that finite difference schemes solve can be either partial differential equations (PDEs) or ordinary differential equations (ODEs). PDE systems are classified as parabolic, hyperbolic, or elliptic according to their properties.

2.6 Keywords

1. Finite Difference Method (FDM)
2. Discretization
3. Grid
4. Stability
5. Consistency
6. Boundary Conditions
7. Initial Conditions
8. Parabolic Equations
9. Hyperbolic Equations
10. Elliptic Equations

2.7 Self Assessment Questions

1. What is the fundamental principle behind finite difference schemes?
2. Explain the process of discretization in finite difference methods?
3. How spatial derivatives are approximated in finite difference schemes?
4. What are the main types of finite difference methods used for solving differential equations?
5. Define stability, consistency, and convergence in the context of finite difference schemes?
6. Describe the importance of boundary conditions in finite difference simulations?

2.8 Case Study

Finite Difference Simulation of Heat Conduction in a Metal Rod

Introduction: In this case study, we'll explore how finite difference schemes can be used to simulate heat conduction in a metal rod. Heat conduction is a fundamental process encountered in various engineering applications, such as thermal management in electronic devices, heat exchangers, and material processing.

Problem Statement: Consider a metal rod of length L with known thermal conductivity k, cross-sectional area A, and initial temperature distribution T0(x). The rod is insulated along its length, except for its ends. At each end, the temperature is fixed at T left and T right.

Question: Simulate the temperature distribution along the length of the rod over time using finite difference methods. We'll discretize both space and time and solve the resulting system of algebraic equations to obtain the temperature profile at each spatial grid point and time step.

2.9 References

1. "Numerical Recipes: The Art of Scientific Computing" by William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery
2. "Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems" by Randall J. LeVeque

UNIT- 3

Applications in Fluid Mechanics

Learning objectives

Generally speaking, the learning objectives of Applications in Fluid Mechanics course are to provide students with the information and abilities needed to comprehend and evaluate fluid behaviour in a variety of technical and practical applications. For such a course, the following learning goals are typical:

Comprehending Flexible, Conduct Statics of Fluids, Dynamics of Fluids, Methods for Measuring Flow, Flow in the Pipe

Structure

3.1 Navier-Stokes Equations

The Alternating Direction Implicit (ADI) method finds numerous applications in fluid mechanics, where it's particularly useful for solving the Navier-Stokes equations and related transport phenomena. Here are some specific applications within fluid mechanics:

Incompressible Flow: ADI methods are commonly used to simulate incompressible fluid flow, where the continuity and momentum equations are solved. This includes applications such as flow around obstacles, within pipes, and over surfaces.

Steady-State Flow: ADI methods can be employed to find steady-state solutions of the Navier-Stokes equations. This is useful for understanding the long-term behaviour of fluid systems, such as determining flow patterns in pipes or channels.

Transient Flow: The development of fluid behavior may be properly captured by ADI approaches for transient issues, where the flow conditions change over time. Applications include wave propagation, unstable flow in pipes, and fluid-structure interaction research fall under this category.

Heat Transfer: In addition to fluid flow, ADI methods can be extended to solve coupled fluid flow and heat transfer problems. This includes scenarios such as convective heat transfer in fluid flows, thermal mixing in channels, and heat exchanger analysis.

Turbulent Flow: While direct application of ADI methods to solve turbulent flows may be limited due to the complexity of turbulence models, they can still be used in combination with turbulence models (such as Reynolds-averaged Navier-Stokes or large eddy simulation) to simulate various turbulent flow phenomena.

Multiphase Flows: ADI methods can be extended to handle multiphase flows, where two or more immiscible fluids are present. Applications include oil-water flows, bubble dynamics, and free surface flows.

Boundary Layer Analysis: ADI methods can be used to analyze boundary layer flows, which are important in understanding aerodynamic and hydrodynamic performance. This includes applications such as flow over airfoils, ship hulls, and automotive bodies.

In each of these applications, the ADI method provides a robust and efficient approach for solving the governing equations of fluid mechanics. Its ability to handle complex geometries, transient behaviour, and coupled phenomena makes it a valuable tool for researchers and engineers in the field of fluid mechanics.

Here are the Navier-Stokes equations in their most common form:

Continuity Equation (Conservation of Mass):

$$\frac{\partial \rho}{\partial t} + \nabla . (\rho v) = 0$$

This equation expresses the conservation of mass, where ρ is the fluid density, v is the velocity vector field, and ∇ denotes the divergence operator.

Momentum Equation (Conservation of Momentum):

$$\frac{\partial (\rho v)}{\partial t} + \nabla . (\rho v \, X \, v) = -\nabla p + \nabla . \tau + \rho g \frac{\partial t}{\partial (\rho v)}$$

This equation describes the conservation of momentum, where p is the pressure, τ is the stress tensor, and g is the gravitational acceleration vector. The stress tensor depends on the fluid's viscosity and velocity gradients.

Energy Equation (Conservation of Energy):

$$\partial (\rho E) / \partial t + \nabla \cdot (\rho E v) = -\nabla \cdot (q - p v) + \nabla \cdot (\tau \cdot v) + \rho v \cdot g \, \partial t / \partial (\rho E) + \nabla \cdot (\rho E v) = -\nabla \cdot (q - p v) + \nabla \cdot (\tau \cdot v) + \rho v \cdot g$$

Here, *E* is the total energy per unit volume, including kinetic and internal energy, q is the heat flux vector, and pv represents the work done by the pressure forces. The term $\tau \cdot v$ represents viscous dissipation.

These equations govern the behaviour of fluid flow in a wide range of situations, from simple pipe flows to complex turbulent flows. Solving the Navier-Stokes equations numerically is essential for understanding and predicting fluid behaviour in practical engineering applications, such as aerospace, automotive, and environmental engineering. However, due to the nonlinearity and complexity of these equations, analytical solutions are often limited to simplified cases, and numerical methods, such as finite difference, finite volume, or finite element methods, are commonly used for practical simulations.

3.2 Computational Fluid Dynamics

Computational Fluid Dynamics (CFD) is a branch of fluid mechanics that deals with the numerical simulation of fluid flow and heat transfer phenomena. It involves the use of computer algorithms and numerical methods to solve the governing equations of fluid dynamics, typically partial differential equations (PDEs), which describe the behavior of fluids in motion.

Here are some key aspects of computational fluid dynamics:

1. **Governing Equations**: The fundamental equations governing fluid flow and heat transfer are the Navier-Stokes equations, which are a set of coupled nonlinear PDEs that describe the conservation of momentum and mass for a fluid. Depending on the problem at hand, additional equations such as those for energy conservation (heat transfer) or species transport (e.g., in combustion) may also be included.
2. **Discretization**: The continuous governing equations are discretized using numerical methods to solve them on a discrete grid. Common discretization methods include finite difference, finite volume, and finite element methods. These methods approximate the derivatives in the governing equations using discrete difference or integration schemes.
3. **Solution Techniques**: Once the governing equations are discretized, iterative or direct solution techniques are employed to solve the resulting system of algebraic equations. Iterative techniques such as the Gauss-Seidel method or conjugate gradient method are often used for large-scale problems encountered in CFD simulations.

Applications:

CFD finds applications in a wide range of industries and fields, including aerospace, automotive, energy, environmental engineering, and biomedical engineering. It is used to analyze and optimize the performance of various engineering systems, such as aircraft wings, turbine blades, heat exchangers, and combustion chambers.

3.3 Practical Examples in Industry

Certainly! Computational Fluid Dynamics (CFD) finds extensive application across a wide range of industries, aiding in the design, analysis, and optimization of various systems and processes. Here are some practical examples of CFD applications in industry:

Aerospace:

Aircraft Design: Airflow surrounding aircraft parts, such as wings, fuselages, and engine nacelles, is simulated using computational fluid dynamics (CFD) to maximize aerodynamic performance, minimize drag, and improve fuel economy.

Jet Engine Design: CFD helps in designing and optimizing internal components of jet engines, such as turbine blades, combustion chambers, and nozzles, to improve efficiency, thrust, and reduce emissions.

Automotive:

Vehicle Aerodynamics: CFD is employed to analyze airflow around vehicles, including cars, trucks, and buses, to minimize drag, enhance vehicle stability, and improve fuel economy.
Engine Cooling and Exhaust Systems: CFD is used to optimize the design of cooling systems for engines and exhaust systems to manage heat dissipation, reduce noise, and enhance performance.
Energy:
Wind Turbine Design: CFD simulations are utilized to study wind flow patterns around wind turbine blades, optimize blade shapes, and improve power generation efficiency.
Nuclear Reactor Safety: CFD is applied to model coolant flow and heat transfer within nuclear reactors, aiding in safety analysis, design optimization, and accident mitigation.
Chemical and Process Industries:
Mixing and Reaction Systems: CFD helps in designing and optimizing mixing vessels, reactors, and chemical processes by predicting flow patterns, residence times, and reaction kinetics.
Pollution Control: CFD is used to model gas dispersion and pollutant dispersion in industrial stacks, exhaust systems, and ventilation systems to assess environmental impact and design effective pollution control measures.
Building and Construction:
HVAC Systems: CFD is employed to optimize the design of heating, ventilation, and air conditioning (HVAC) systems in buildings to ensure thermal comfort, indoor air quality, and energy efficiency.
Smoke and Fire Modelling: CFD simulations are used to model smoke movement and fire spread in buildings, aiding in fire safety design and evacuation planning.
Marine and Offshore:
Ship Hydrodynamics: CFD is utilized to study water flow around ships and offshore structures to optimize hull shapes, reduce resistance, and improve manoeuvrability and fuel efficiency.
Offshore Platform Design: CFD simulations help in designing offshore platforms, risers, and mooring systems by analyzing wave loads, wind loads, and hydrodynamic forces.

3.4 Summary

Applications in Fluid Mechanics encompass a diverse range of real-world scenarios where the principles of fluid behaviour are applied to solve engineering problems and optimize systems. Here's a summary:

1. **Hydrodynamics**: Understanding fluid behaviour in motion, including the study of forces, energy transfer, and flow patterns. Applications include the design of ships, submarines, and offshore structures.
2. **Aerodynamics**: Analysis of airflow around objects, crucial in aircraft design, wind turbines, and automotive aerodynamics for fuel efficiency and performance optimization.
3. **Hydraulic Engineering**: Design and management of water systems, including dams, channels, and irrigation networks, to control flooding, provide water supply, and generate hydroelectric power.
4. **Thermal Systems**: Utilizing fluid mechanics principles in the design of heating, ventilation, and air conditioning (HVAC) systems for efficient temperature regulation in buildings and vehicles.
5. **Turbo machinery**: Study and design of turbines, compressors, and pumps used in power generation, propulsion systems, and industrial processes to efficiently transfer energy to or from fluid flow

3.5 Keywords

1. Hydrodynamics
2. Aerodynamics
3. Thermal Systems
4. Biomechanics
5. Fluid Dynamics
6. Pipe Flow

3.6 Self Assessment Questions

1. What are the key differences between laminar and turbulent flow? Provide examples of each.
2. Explain how Bernoulli's equation is applied to analyze fluid flow in a pipe.
3. Describe the concept of boundary layers in aerodynamics and its significance in aircraft design.
4. What factors determine the design of a dam for water storage? Discuss the importance of spillways and outlet structures.

5. How does the Manning's equation differ from the Darcy-Weisbach equation, and when is each used in hydraulic engineering?

3.7 Case Study

Designing an Efficient Cooling System for Data Centers

A technology company is expanding its data center infrastructure to accommodate the increasing demand for cloud services. However, the existing cooling system is struggling to maintain optimal temperatures, leading to potential risks of equipment overheating and decreased efficiency.

Question: Design a new cooling system that effectively manages heat dissipation within the data center while minimizing energy consumption and operational costs.

3.8 References

1. Cengel, Y. A., &Cimbala, J. M. (2017). Fluid Mechanics: Fundamentals and Applications (4th ed.). McGraw-Hill Education.
2. Kundu, P. K., Cohen, I. M., & Dowling, D. R. (2011). Fluid Mechanics (5th ed.). Academic Press.

UNIT - 4

Operational Techniques for Linear Programming Problems

Learning objectives

The learning objectives of Operational Techniques for Linear Programming Problems typically include:

- Understand Linear Programming (LP)
- Understand Graphical Solution Methods
- Applications of Simplex Method
- Understand Sensitivity Analysis
- Applications of Integer Linear Programming (ILP)
- Understand Network Optimization

Structure

4.1 Introduction to Linear Programming

A mathematical optimization method called linear programming (LP) is used to determine which result is optimal for a model with linear dependencies. Different operational strategies are used to effectively tackle linear programming issues. Here are some commonly used techniques:
LPP Formulation

LPP stands for Linear Programming Problem, which is a mathematical optimization technique used to maximize or minimize a linear objective function subject to a set of linear constraints. The general formulation of an LPP is as follows:

Maximize (or Minimize):

$$Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

Subject to:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m$$

Where:

- $x_1, x_2, \ldots, x_n$ are decision variables representing quantities to be determined.
- $c_1, c_2, \ldots, c_n$ are coefficients of the objective function to be maximized or minimized.
- a_{ij} are coefficients of the constraints.
- $b_1, b_2, \ldots, b_m$ are the right-hand side constants of the constraints.
- The inequalities are typically in the form of less than or equal to (≤) constraints.

The Aim of linear programming is to find the values for the choice variables x1, x2,... xn that optimize (maximize or minimize) the objective function while meeting all the requirements.

Linear programming problems can be solved using various algorithms, such as the simplex method, interior-point methods, and the graphical method (for two-variable problems). These algorithms iteratively improve the solution until an optimal solution is found.

Example 1: Let us consider a corporation that manufactures Product A and Product B. The company's labor hours and raw material supply are limited, which affects output. The objective is to maximize profit while taking into account resource limits.

Objective function: Maximize Z=5A+4B

Constraints:

1. Labor hours constraint: The total labor hours available is 200.
   $2A+3B \leq 200$
2. Raw materials constraint: The total raw materials available is 150.
3. $A+2B \leq 150$
4. Non-negativity constraints: $A \geq 0, B \geq 0$.

Here, A and B represent the quantities of Product A and Product B produced, respectively. The objective function represents the total profit obtained from selling A units of Product A and B units of Product B, with coefficients 5 and 4 representing the profit per unit of each product, respectively.

Graphical Analysis of Linear Programming:

Graphical analysis of linear programming involves visually representing the constraints and objective function on a graph to find the optimal solution for the linear programming problem (LPP) when dealing with two decision variables. Here's how it's done:

Step 1: Plot Constraints

1. For each constraint in the LPP, rewrite it in the form $y=mx+b$ by solving for y.
2. Plot each constraint line on the graph.
3. Shade the feasible region, which is the area where all constraints are satisfied. This region is typically a convex polygon.

Step 2: Plot Objective Function

1. Plot the objective function on the graph using its coefficients.
2. Determine whether to maximize or minimize the objective function.

Step 3: Identify Optimal Solution

1. Identify the point within the feasible region that either maximizes or minimizes the objective function. This point is the optimal solution.
2. If the objective function is parallel to one of the constraint lines, the optimal solution may lie on a corner (vertex) of the feasible region.

Example 2: Let's use the example of a company producing Product A and Product B with the following constraints and objective function:

Objective function: Maximize $Z=5A+4B$

Constraints:

1. Labor hours constraint: $2A+3B \leq 2002$
2. Raw materials constraint: $A+2B \leq 150$
3. Non-negativity constraints: $A \geq 0, B \geq 0$

4.2 Computational Procedure of the Simplex Method

The simplex method is a widely used algorithm for solving linear programming problems. Here's a step-by-step computational procedure of the simplex method:

Step 1: Convert the Problem to Standard Form

1. Rewrite the objective function and constraints so that:
   - The objective function is a maximization problem.
   - All constraints are equality constraints.
   - All decision variables are non-negative.

Step 2: Initialize the Simplex Tableau

1. Create the initial simplex tableau by introducing slack variables for each constraint to convert the inequalities into equalities.
2. Include the coefficients of the objective function and the slack variables in the tableau.
3. Identify the basic variables (those corresponding to the slack variables) and non-basic variables (those corresponding to original decision variables).

Step 3: Perform Iterative Steps

1. **Select Pivot Column**: Choose the column with the most negative coefficient in the objective row. This column becomes the pivot column.
2. **Select Pivot Row**: Determine the pivot row by selecting the row with the smallest non-negative ratio (result of dividing the RHS by the corresponding coefficient in the pivot column).
3. **Update the Tableau**: Perform row operations (Gauss-Jordan elimination) to make the pivot element equal to 1 and all other elements in the pivot column equal to 0.
4. **Update Basic and Non-Basic Variables**: Update the basic and non-basic variables based on the pivot column and row.
5. **Repeat Steps 3.1 to 3.4**: Continue iterating until all coefficients in the objective row are non-negative. This indicates that the current solution is optimal.

Step 4: Interpret the Solution

1. Extract the solution from the final tableau:
   - The values of the basic variables (slack variables) give the optimal solution.
   - The objective function value at the optimal solution is obtained from the tableau.
2. If necessary, convert the solution back to the original form of the problem.

Step 5: Sensitivity Analysis

1. Perform sensitivity analysis to understand how changes in the problem's coefficients affect the optimal solution and objective function value.

Step 6: Termination

1. Terminate the algorithm once an optimal solution is found.

Basics of Simplex Method

The basic of simplex method is explained with the following linear programming problem.

Maximize: $60X_1+70X_2$

Subject to: $2X_1 + X_2 \leq 300$;

$3X_1+4X_2 \leq 509$;

$X_1+7X_2 \leq 812$;

$X_1, X_2 \geq 0$

First we introduce the variables $S_3, S_4, S_5 \geq 0$

So that the constraints becomes equations, thus,

$2X_1+ X_2+ 1S_3+ 0S_4+0S_5=300$

$3X_1+ 4X_2+ 0S_3+1S_4+0S_5=509$

$4X_1+ 7X_2+ 0S_3+ 0S_4+1S_5= 812$

the variables S_3, S_4, S_5 are called as slack variables.

Example 3: Solve by simplex method

Maximize: $60X_1+70X_2$

Subject to: $2X_1 + X_2 + 1S_3 + 0S_4+0S_5 = 300$;

$3X_1 +4X_2+0S_3+1S_4+0S_5=509$;

$4X_1+7X_2+0S_3+0S_4+1S_5=812$

and $X_1, X_2, S_3, S_4, S_5 \geq 0$

Solution:

Simplex Table 1

CB	Basic Variable	Cj	60	70	0	0	0
		Xb	x_1	x_2	s_3	s_4	s_5
0	s_3	300	2	1	1	0	0
0	s_4	509	3	4	0	1	0
0	s_5	812	4	7	0	0	1
Z			-60	-70	0	0	0

Simplex Table 2

CB	Basic Variable	Cj	60	70	0	0	0
		Xb	x_1	x_2	s_3	s_4	s_5
0	s_3	184	10/7	0	1	0	-1/7
0	s_4	45	5/7	0	0	1	-4/7
70	X_2	116	4/7	1	0	0	1/7
Zj - Cj			-140/7	0	0	0	70/7

Hence X1 should become a basic variable at the next iteration.

Minimum ratio:

$$\text{Min}\left(\frac{184}{10/7}, \frac{45}{5/7}, \frac{116}{4/7}\right) = \text{Min}\left(\frac{644}{5}, 63, 203\right) = 63$$

Simplex Table 3

$$\text{Min}\left(\frac{94}{1}, \frac{80}{\frac{3}{5}}\right) = 94$$

C_B	Basic Variables	C_j	60	70	0	0	0
		X_B	x_1	x_2	s_3	s_4	s_5
	s3	94	0	0	1	-2	1
	x1	63	1	0	0	7/5	-4/5
	x2	80	0	1	0	-4/5	3/5
	z_j-c_j		0	0	0	28	-6

Minimum ratios:

Simplex Table 4

C_B	Basic Variables	C_j X_B	60 x_1	70 x_2	0 s_3	0 s_4	0 s_5
0	s_5	94	0	0	1	-2	1
60	x_1	691/5	1	0	4/5	-1/5	0
70	x_2	118/5	0	1	-3/5	2/5	0
	z_j-c_j		0	0	6	16	0

So, $z_j - c_j \geq 0$ for all j,

Thus, the objective function is maximized for x_1 = 691/5 and x_2=118/5 and

Hence maximum Z= 9944

4.3 Two-Phase Simplex Method

The Two-Phase Simplex Method is an extension of the Simplex Method used to solve linear programming problems that may have initial infeasible solutions. It involves two phases:

Phase 1:

1. Introduce artificial variables to convert the constraints to equations.
2. Solve the auxiliary problem to find an initial basic feasible solution (IBFS) by minimizing the sum of artificial variables.
3. If the solution to the auxiliary problem has a non-zero objective value, the original problem is infeasible, and the process stops.
4. If the solution to the auxiliary problem has a zero objective value, remove the artificial variables and continue to Phase 2.

Phase 2:

1. Use the simplex method to solve the original problem with the artificial variables removed.
2. Iteratively improve the current basic feasible solution until an optimal solution is found.
3. If the optimal solution to the original problem has a non-zero objective value, the problem is unbounded.

Example 4:

Solve the problem by Two phase method

Minimize $12.5X_1+14.5X_2$

Subject to: $X_1+X_2\geq 2000$

$0.4X_1+0.75X_2\geq 1000$

$0.075X_1+0.1X_2\leq 200$

$X_1,X_2\geq 0$

Solution:

Minimize $12.5X_1+14.5X_2$

Subject to: $X_1+X_2\geq 2000$

$0.4X_1+0.75X_2\geq 1000$

$0.075X_1+0.1X_2\leq 200$

$X_1,X_2\geq 0$

First Convert minimization to maximization

Maximize: $-12.5X_1-14.5X_2 = -25/2X_1-29/2X_2$

Subject to: $X_1+X_2-S_3=2000$

$40X_1+75X_2-S_4=100000$

$75X_1+100X_2+S_5=200000$

$X_1, X_2, S_3, S_4, S_5\geq 0$

Phase I: Maximize: $-A_6-A_7$

Subject to: $X_1+X_2-S_3+A_6=2000$

$40X_1+75X_2-S_4+A_7=100000$

$75X_1+100X_2+S_5=200000$

and $X_1, X_2, S_3, S_4, S_5, A_6, A_7\geq 0$

The initial basic feasible solution is $A_6=2000$, $A_7=100000$ and $S_5=200000$.

Find A_6 and A_7 become zero.

C_B	Basic variables	C_j X_B	0 x_1	0 x_2	0 s_3	0 s_4	0 s_5	-1 A_6	-1 A_7
-1	A_6	2000	1	1	-1	0	0	1	0
-1	A_7	100000	40	75	0	-1	0	0	1
0	S_5	200000	75	100	0	0	1	0	0
		z_j-c_j	-41	-76	1	1	0	0	0

Table 1

C_B	Basic variables	C_j X_B	0 x_1	0 x_2	0 s_3	0 s_4	0 s_5	-1 A_6
-1	A_6	2000/3	7/15	0	-1	1/75	0	1
0	X_2	4000/3	8/15	1	0	-1/75	0	0
0	S_5	200000/3	65/3	0	0	4/3	1	0
		z_j-c_j	-1/15	0	1	-1/75	0	0

TABLE 2

C_B	Basic variables	C_j X_B	0 x_1	0 x_2	0 s_3	0 s_4	0 s_5
0	x_1	10000/7	1	0	-15/7	1/35	0
0	x_2	4000/7	0	1	8/7	-1/35	0
0	s_5	250000/7	0	0	325/7	16/21	1
		z_j-c_j	0	0	0	0	0

Table 3

X_1 = 10000/7 X_2 = 4000/2 S_5=250000/7

Phase II:

C_B	Basic variables	C_j / X_B	-25/2 / x_1	-29/2 / x_2	0 / s_3	0 / s_4	0 / s_5
-25/2	x_1	10000/7	1	0	-15/7	1/35	0
-29/2	x_2	4000/7	0	1	8/7	-1/35	0
0	s_5	250000/7	0	0	325/7	5/7	1
		z_j-c_j	0	0	143/14	2/35	0

Table 1

all Zj-Cj≥0

X_1=10000/7=1428 X_2=4000/7= 571.4

Minimum Z : 26135.3

4.4 Big-M Method

Maximize:-12.5X1–14.5X2

Subject to: X1+ X2–S3= 2000

40X1+75X2-S4=100000

75X1 + 100X2 +S5 = 200000 and X1,X2,S3,S4,S5≥0.

Introduce artificial variables A6 and A7.

Maximize:-12.5X1–14.5X2–M (A6+A7)

Subject to: X1+X2–S3+A6=2000

40X1+75X2-S4+A7=100000

75X1+100X2+S5= 200000

and X1, X2, S3, S4, S5, A6, A7 ≥0

C_B	Basic variables	C_j X_B	-12.5 x_1	-14.5 x_2	0 s_3	0 s_4	0 s_5	-M A_6	-M A_7
-M	A_6	2000	1	1	-1	0	0	1	0
-M	A_7	100000	40	75	0	-1	0	0	1
0	S_5	200000	75	100	0	0	1	0	0
		z_j-c_j	-41M +12.5	-76M +14.5	M	M	0	0	0

Table 1

C_B	Basic variables	C_j X_B	-12.5 x_1	-14.5 x_2	0 s_3	0 s_4	0 s_5	-M A_6
-M	A_6	2000/3	7/15	0	-1	1/75	0	1
-14.5	X_2	4000/3	8/15	1	0	-1/75	0	0
0	S_5	200000/3	65/3	0	0	4/3	1	0
		z_j-c_j	-7/15M +143/30	0	M	-M/75 +29/150	0	0

Table 2

C_B	Basic variables	C_j X_B	-12.5 x_1	-14.5 x_2	0 s_3	0 s_4	0 s_5
-12.5	x_1	10000/7	1	0	-15/7	1/35	0
-14.5	x_2	4000/7	0	1	8/7	-1/35	0
0	s_5	250000/7	0	0	325/7	16/21	1
		z_j-c_j	0	0	143/14	2/35	0

Table 3

Hence,

X_1=10000/7

X_2=4000/7

Minimum Z= 26135.3

4.5 Summary

Operational Techniques for Linear Programming Problems equips learners with the skills to model, solve, and analyze optimization problems in various contexts, enabling efficient decision-making and resource allocation. Operational Techniques for Linear Programming Problems is a course designed to equip learners with the skills and knowledge necessary to tackle optimization challenges within constraints.

4.6 Keywords

- Linear Programming
- Optimization
- Constraints
- Decision Variables
- Objective Function
- Feasible Region
- Graphical Solution
- Simplex Method

4.7 Self Assessment Questions

1. What is the fundamental objective of linear programming?
2. Describe the components of a linear programming model.
3. Explain the difference between feasible solutions and optimal solutions in linear programming.
4. How does the graphical method assist in solving linear programming problems?
5. Outline the steps involved in the simplex method for solving linear programming problems.
6. How does integer linear programming differ from linear programming?

4.8 Case Study

Optimizing Production at Genentech Manufacturing

GreenTech Manufacturing is a company specializing in eco-friendly household products. One of its best-selling items is a multi-purpose cleaner, produced using a blend of natural ingredients. The production process involves mixing, bottling, labelling, and packaging. The company aims to maximize its profits while adhering to various constraints, including ingredient availability, production capacity, and market demand.

Question: GreenTech Manufacturing wants to optimize its production process for the multi-purpose cleaner to meet customer demand while minimizing costs. The company needs to determine the optimal production plan considering factors like ingredient availability, production capacity, and market demand variations.

4.9 References

1. Hillier, F. S., & Lieberman, G. J. (2018). Introduction to Operations Research. McGraw-Hill Education.
2. Taha, H. A. (2016). Operations Research: An Introduction. Pearson.

UNIT - 5

Duality in Linear Programming

Learning objectives

Duality in linear programming is a fundamental concept with several important learning objectives:

- Understanding the Concept of Duality
- Recognizing Weak and Strong Duality
- Dual Feasibility and Primal Feasibility
- Interpreting the Dual Problem
- Applications in Economics and Operations Research
- Optimality Conditions
- Formulation and Solving of Dual Problems

Structure

5.1 Dual Problems

Duality in linear programming refers to the relationship between a given linear programming problem (referred to as the primal problem) and its associated dual problem. The duality concept provides valuable insights into the structure and properties of linear programming problems and plays a crucial role in optimization theory. Here's an overview of duality in linear programming:

Primal Problem:

The fundamental linear programming issue that we are trying to solve is known as the primal problem. Usually, an objective function must be maximized or minimized while taking into account a number of linear restrictions. The following is a formulation of the fundamental problem:

Maximize (or Minimize):Tx

Subject to:

$Ax \leq bx \geq 0$

Dual Problem:

Derived from the primal problem, the dual problem offers an alternative viewpoint on the same optimization problem. It entails maximizing or decreasing a separate objective function while adhering to a pair of complementary restrictions. One way to formulate the dual problem is as follows:

Minimize (or Maximize):$bT\,y$

Subject to:

$AT\,y \geq c$

$y \geq 0$

Where:

y is the vector of dual variables.

AT is the transpose of the constraint coefficient matrix AA.

Relationship between Primal and Dual:

Duality theory establishes a strong relationship between the primal and dual problems. Specifically:

Weak Duality: The following inequality holds for each viable solution xx of the primal issue and any feasible solution yy of the dual problem: $cT\,x \leq bT$y.Weak duality is the name given to this quality.

Strong Duality: If both the primal and dual problems have optimal solutions, their objective function values are equal: If $cTx*=bTy*$, then both $x*$ and $y*$ are optimal solutions. This property is known as strong duality.

Interpretation:

The primal problem seeks to optimize resource allocation (e.g., maximize profit or minimize cost) subject to constraints.

The dual problem provides insights into the value of resources (e.g., prices or shadow prices) and can be interpreted as maximizing the lower bound (minimizing the upper bound) on the objective function.

Duality in linear programming is a fundamental concept with wide-ranging applications in optimization theory, economics, game theory, and operations research. It provides a powerful tool for understanding the structure of optimization problems and deriving useful insights into their properties.

Maximize (or Minimize): cTx

Subject to:

$Ax \leq b$

$x \geq 0$

Dual Problem:

The dual problem is derived from the primal problem using the concept of duality in linear programming. It involves minimizing or maximizing a different objective function, subject to a set of dual constraints. The dual problem can be formulated as follows:

Minimize (or Maximize): $b^T y$

Subject to:

$A^T y \geq c$

$y \geq 0$

Where:

y is the vector of dual variables.

A^T is the transpose of the constraint coefficient matrix A.

Relationship Between Primal and Dual:

The primal and dual problems are closely related through duality theory, which establishes the following relationships:

Weak Duality: The following inequality holds for each possible solution x of the primal issue and any feasible solution y of the dual problem: $cTx \leq bTy$ Weak duality is the name given to this quality.

Strong Duality: If both the primal and dual problems have optimal solutions, their objective function values are equal: If $cTx* = bTy*$, then both $x*$ and $y*$ are optimal solutions.

Interpretation:

The primal problem represents the optimization of resource allocation, while the dual problem provides insights into the value of resources (e.g., prices).

5.2 Duality Theorems

Certainly! Let's discuss the Weak Duality Theorem and the Strong Duality Theorem in linear programming along with their proofs.

1. Weak Duality Theorem:

The objective function value of the primal problem is always smaller than or equal to the objective function value of the dual problem for any practical solutions of the primal and dual issues.

Proof:

Consider the primal problem (P) in standard form:

Maximize $\mathbf{Z_P}$

Subject to $Ax \leq b$,

$Xi \geq 0$

and its associated dual problem (D):

Minimize $\mathbf{Z_D}$

Subject to $A^T y \geq c$,

$y \geq 0$

Let the primal problem have a viable solution, x, and the dual problem have a feasible solution, y. Next, we have

:

$A x \leq b$

$A^T y \geq c$

Taking the inner product of these inequalities, we get:

$cTx \leq yTAx \leq yTb$

Since $y \geq 0$, we have $y T b = b T y$, and thus:

$cTx \leq bTy$

This proves the Weak Duality Theorem.

2. Strong Duality Theorem:

Statement: The optimal objective function values of the dual and primal linear programming problems must equal if they have workable solutions.

Proof:

Let $x*$ represent the optimum solution for the primal problem, where the ideal objective function value is $z*=cTx*$, and $y*$ represent the optimal solution for the dual problem, where the optimal objective function value is $w*=bTy*$.

From the Weak Duality Theorem, we know that $cTx* \leq bTy*$. Additionally, since $x*$ and $y*$ are feasible solutions to the primal and dual problems, respectively, we have:

$\mathbf{c}T\mathbf{x}* \geq \mathbf{c}T\mathbf{x}*; \ \mathbf{b}T\mathbf{y}* \geq \mathbf{b}T\mathbf{y}*$

Combining these inequalities, we get:

$\mathbf{c}T\mathbf{x}* = \mathbf{b}T\mathbf{y}*$

This proves the Strong Duality Theorem.

These duality theorems are fundamental in linear programming, providing essential theoretical results that are used extensively in optimization theory and applications.

5.3Interpretation and Application in Optimization

The duality theorems in linear programming have significant interpretation and application in optimization theory.

Weak Duality Interpretation:

The optimal value of the dual problem, which is the primal problem's optimal value, is lower limited by the weak duality theorem.

It implies that the solution to the primal problem cannot exceed the solution to the dual problem, providing valuable insights into the relationship between the primal and dual problems.

Strong Duality Interpretation:

The strong duality theorem establishes a strong relationship between the primal and dual problems, indicating that they both have optimal solutions if one of them is feasible.

It guarantees that the optimal objective function values of the primal and dual problems are equal, providing a powerful theoretical result that simplifies optimization analysis.

Application in Optimization:

Sensitivity Analysis:

Duality theorems are used in sensitivity analysis to assess the impact of changes in problem parameters (e.g., coefficients, constraints) on the optimal solution.

By analyzing the dual variables, which represent the shadow prices or resource values, sensitivity analysis provides insights into the robustness of the optimal solution.

Algorithm Design:

Duality theorems play a crucial role in the design of optimization algorithms, such as the simplex method and interior point methods.

They provide theoretical foundations for algorithm development and guide the formulation of efficient computational procedures.

Optimization Modelling:

Duality theorems influence the formulation of optimization models by providing insights into the structure and properties of optimization problems.

They aid in the selection of appropriate objective functions, constraints, and decision variables to achieve desirable optimization outcomes.

Resource Allocation:

In economic applications, duality theorems help in resource allocation decisions by providing information about the value of resources (e.g., prices) derived from the dual problem.

They guide decision-making processes by quantifying trade-offs between different resources and objectives.

Game Theory:

Duality theorems are applied in game theory to analyze strategic interactions and equilibrium solutions in competitive environments.

They provide a framework for understanding optimal strategies and payoffs in various types of games.

In summary, duality theorems have broad applications in optimization theory and practice, influencing algorithm design, sensitivity analysis, optimization modelling, resource allocation decisions, and game theory. They provide essential insights into the structure and properties of optimization problems, enabling informed decision-making and efficient problem-solving techniques.

5.4 Summary

There is an underlying link between primal and dual optimization issues that is revealed by the deep idea of duality in linear programming. Fundamentally, duality states that there is a problem known as its twin for any linear programming issue.

Whereas the dual problem entails maximizing or minimizing a separate objective function subject to its own set of constraints, the primal problem focuses on maximizing or minimizing an objective function subject to specific restrictions. It's amazing how closely related these two issues are, with answers to one offering understanding of the other.

5.5 Keywords

- Primal problem
- Dual problem
- Primal feasibility
- Optimality conditions
- interpretation
- Transformation
- Optimization theory
- Linear programming
- Operations research
- Objective function

- Constraints
- Lagrange multipliers
- Optimality.

5.6 Self Assessment Questions

1. Define duality in the context of linear programming and explain its significance.
2. What is the difference between weak duality and strong duality? Provide examples to illustrate each.
3. Explain the concept of dual feasibility and its importance in the context of linear programming.
4. Describe the conditions under which strong duality holds in linear programming.
5. How do complementary slackness conditions contribute to verifying optimality in linear programming problems?

5.7 Case Study

Supply Chain Optimization

Imagine a retail company that operates a network of warehouses and distribution centers across the country. The company's goal is to efficiently distribute its products from manufacturing facilities to retail stores while minimizing transportation costs.

Primal Problem: In this case, the primary issue is figuring out the best routes for transportation and how much inventory to send from warehouses to retail locations. The aim is to save transportation expenses while guaranteeing that the demand of every store is satisfied and that the warehouse's capacity limitations are not surpassed. With limitations on capabilities, requests, and transportation costs, this issue may be expressed as a linear programming problem.

5.8 References

1. Hillier, F. S., & Lieberman, G. J. (2018). Introduction to Operations Research. McGraw-Hill Education.
2. Taha, H. A. (2016). Operations Research: An Introduction. Pearson.

UNIT - 6

Assignment Models

Learning objectives

Learning Objectives of Assignment Models:

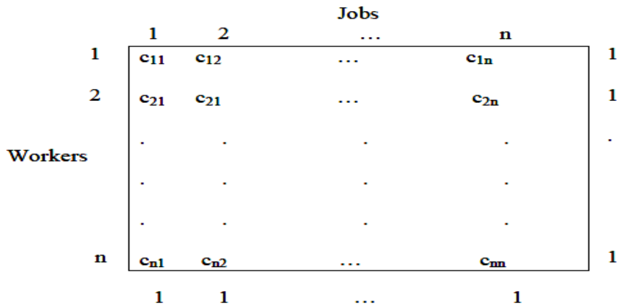
- Conceptual Understanding
- Model Formulation
- Optimization Techniques
- Application Skills
- Solution Methods
- Interpretation and Analysis

Structure

6.1 Mathematical Formulation

The structure of the Assignment problem is similar to a transportation problem, is as follows:

Figure 6.1.1 Structure of the Assignment problem



The effectiveness measure when ith worker is assigned jth job is represented by the element Cij. Presume that the goal is to reduce the total effectiveness measure.
The number of ith people allocated to the jth task is represented by the element Xij. A person can be allocated using the following: Xi1+Xi2+... + Xin= 1, wherei= 1,2, n
A person may only be allocated to one job, and only one job at a time.
maximizing or decreasing an alternative goal function while adhering to specific limitations. It's amazing how closely related these two issues are, with answers to one offering understanding of the other.
we have $X1j+X2j+\ldots\ldots\ldots\ldots\ldots+Xnj=1$, where j = 1, 2, ... , n and the objective function is formulated as Minimize

$$C11\,X11+C12\,X12+\ldots\ldots\ldots.+Cnn\,Xnn \wedge X\,ij \geq 0$$

The assignment problem is actually a special case of the transportation problem where m =n and $ai=bj=1$.

6.2 Hungarian Method

The assignment challenge is to determine the most effective way to allocate a collection of tasks to a group of agents or computers. The Hungarian Method is a combinatorial optimization method that achieves this goal. Known alternatively as the Kuhn-Munkres algorithm, it was created by Harold Kuhn in 1955 and further improved by James Munkres in 1957. Especially when there aren't many tasks and agents, the Hungarian Method offers a productive solution to the assignment problem.

Here's an outline of the Hungarian Method:

1. **Setup:** Given a cost matrix C of size n×n, where n is the number of tasks or agents, and each element c_{ij} represents the cost of assigning the j-th task to the i-th agent.
2. **Initialization:** Convert the cost matrix C into a matrix of potentials P using a method such as the row reduction method. This involves subtracting the minimum value of each row from all elements in that row and subtracting the minimum value of each column from all elements in that column.
3. **Step 1: Zero Assignment:** Identify the smallest element in the matrix of potentials P. Subtract this value from all other elements of the matrix such that the smallest element becomes zero.
4. **Step 2: Row and Column Reduction:** Adjust the matrix P to ensure at least one zero is present in each row and column. This may require subtracting the smallest element in each row from all other elements of the row and subtracting the smallest element in each column from all other elements of the column.
5. **Step 3: Covering Zeros:** Cover all zeros in the matrix using the minimum number of lines (rows or columns) possible. If the number of lines equals n, proceed to Step 7; otherwise, continue to Step 4.
6. **Step 4: Finding Minimum Number of Lines:** Determine the minimum number of lines required to cover all zeros using the minimum number of lines algorithm. Adjust the potentials PPP accordingly.
7. **Step 5: Assigning Tasks:** Assign tasks to agents based on the uncovered zeros. If a zero is uncovered, it represents a potential assignment. Choose any zero and mark its row and column as assigned. If there are no uncovered zeros, proceed to Step 6.
8. **Step 6: Adjusting Potentials:** Adjust the potentials P to create additional zeros while preserving the existing assignments. Then return to Step 3.
9. **Step 7: Final Assignment:** The final assignment is obtained by selecting the zeros corresponding to the assigned tasks. These zeros form a set of non-attacking pairs, representing the optimal assignment.

Example 1:

Four people are available to work on the four occupations in a work shop. Each job may only have one worker at a time. The cost of allocating each individual to each task is displayed in the following table. The goal is to allocate workers to tasks in a way that minimizes the overall cost of the assignment.

		Jobs			
		1	2	3	4
Persons	A	20	25	22	28
	B	15	18	23	17
	C	19	17	21	24
	D	25	23	24	24

Solution:

Step 1: Determine the cost table based on the provided problem. Keep in mind that a fake origin or destination has to be inserted if the number of origins and destinations is not equal.

Step 2: Determine which element in each table row has the lowest cost, then deduct that element from each element in that row. so that the new table's rows will all have a minimum of one zero. The First Reduced Cost table is the name of this new table.

Step 3: Determine which element in each table column has the lowest cost, then deduct that element from each element in that column. This means that there is at least one zero element in every row and column. The Second Reduced Cost Table is the name of this new table.

		Jobs			
		1	2	3	4
Persons	A	0	5	1	7
	B	0	3	7	1
	C	2	0	3	6
	D	2	0	0	0

Step 4: Determine an Assignment

In Step 3, we look at row A of the table and see that there is only one zero (cell A1). We then box this zero and cross off the other zeros in the column that is boxed. This allows us to get rid of cell B1.Upon closer inspection, we can see that row C has one zero (cell C2). We may box this zero and cross out, or remove, the zeros in the boxed column. This is the process that eliminates cell D2. The third column has one zero. Consequently, cell D3 is boxed, which allows us to remove cell D4.

		Jobs			
		1	2	3	4
	A	[0]	5	1	7
Persons	B	0	3	7	1
	C	X	[0]	3	6
	D	2	X	[0]	X

Jobs
1 2 3 4

		Jobs			
		1	2	3	4
	A	0	4	0	6
Persons	B	0	2	6	0
	C	3	0	3	6
	D	3	0	0	0

Persons		Jobs 1	Jobs 2	Jobs 3	Jobs 4
	A	[0]	4	~~0~~	6
	B	0	2	6	[0]
	C	~~3~~	[0]	3	6
	D	3	~~0~~	[0]	~~0~~

The total cost : 78

A1 +B4+ C2+ D3=20+ 17+ 17+24= 78

Unbalanced Assignment Problem:

We assumed in the last section that there would be an equal number of tasks and people to allocate. We refer to this type of assignment problem as a balanced assignment problem. Assume that an assignment problem is deemed imbalanced if the number of employees differs from the number of positions.

Some of them won't be able to find employment if there are fewer jobs than there is people. Therefore, in order to convert the imbalanced assignment problem into a balanced assignment problem, we must add one or more fake tasks of zero duration.

Workers		Jobs 1	Jobs 2	Jobs 3	Jobs 4	Jobs 5	Jobs 6
	A	5	2	4	2	5	0
	B	2	4	7	6	6	0
	C	6	7	5	8	7	0
	D	5	2	3	3	4	0
	E	8	3	7	8	6	0
	F	3	6	3	5	7	0

Example 2:

Determine the shortest time needed to complete all of the tasks in the following unbalanced assignment.

		Jobs				
		1	**2**	**3**	**4**	**5**
	A	5	2	4	2	5
	B	2	4	7	6	6
Workers	**C**	6	7	5	8	7
	D	5	2	3	3	4
	E	8	3	7	8	6
	F	3	6	3	5	7

Solution:

Step1: The cost table

		Jobs					
		1	**2**	**3**	**4**	**5**	**6**
	A	5	2	4	2	5	0
Workers	**B**	2	4	7	6	6	0
	C	6	7	5	8	7	0
	D	5	2	3	3	4	0
	E	8	3	7	8	6	0
	F	3	6	3	5	7	0

Step2: Find the First Reduced Cost Table

		Jobs					
		1	2	3	4	5	6
	A	5	2	4	2	5	0
	B	2	4	7	6	6	0
Workers	C	6	7	5	8	7	0
	D	5	2	3	3	4	0
	E	8	3	7	8	6	0
	F	3	6	3	5	7	0

Step 3: Find the second Cost Reduced table:

		Jobs					
		1	2	3	4	5	6
	A	3	0	1	0	1	0
	B	0	2	4	4	2	0
Workers	C	4	5	2	6	3	0
	D	3	0	0	1	0	0
	E	6	1	4	6	2	0
	F	1	4	0	3	3	0

Step 4:

		Jobs					
		1	2	3	4	5	6
	A	3	~~0~~	1	[0]	1	~~0~~
	B		2	4	4	2	0
Workers	C	[0]	5	2	6	3	~~ ~~
	D	3	~~0~~	~~0~~	1	[0]	~~0~~
	E	6	1	4	6	2	[0]
	F	1	4		3	3	0

Jobs

Workers		1	2	3	4	5	6
	A	3	~~0~~	1	[0]	1	1
	B	[0]	2	4	4	2	1
Workers	C	~~0~~	4	1	5	2	[0]
	D	3	~~0~~	~~0~~	1	[0]	1
	E	5	[0]	3	5	1	0
	F	1	4	[0]	3	3	1

Jobs

Workers		1	2	3	4	5	6
	A	3	0	1	0	1	1
	B	0	2	4	4	2	1
Workers	C	3	4	1	5	2	0
	D	3	0	0	1	0	1
	E	5	0	3	5	1	0
	F	1	4	0	3	3	1

The total 14,

that is A4+B1+D5+E2+F3=2+2 +4+3+3= 14

Example 3:

A computer centre has five jobs to be done and has five computer machines to perform them.

		Jobs 1	2	3	4	5
Computer Machines	1	70	30	X	60	30
	2	X	70	50	30	30
	3	60	X	50	70	60
	4	60	70	20	40	X
	5	30	30	40	X	70

The cost of processing of each job on any machine is shown in the table below.

Solution:

Step 1:

		Jobs 1	2	3	4	5
Computer Machines	1	70	30	500	60	30
	2	500	70	50	30	30
	3	60	500	50	70	60
	4	60	70	20	40	500
	5	30	30	40	500	70

Step3:Find the Second Reduced Cost Table

Jobs

Computer Machines	1	2	3	4	5
1	40	0	470	30	0
2	470	40	20	0	0
3	10	450	0	20	10
4	40	50	0	20	480
5	0	0	10	470	40

Step 4:

Jobs

Computer Machines	1	2	3	4	5
1	40	[0]	470	30	~~0~~
2	470	40	20	0	0
3	10	450	0	[0]	10
4	40	50	~~0~~	20	480
5	[0]	~~0~~	[0]	470	40

Step 5:

Jobs

Computer Machines	1	2	3	4	5
1	~~40~~	[0]	~~470~~	~~30~~	~~0~~
2	470	40	20		0
3	~~10~~	~~450~~		[0]	~~10~~
4	40	50	~~0~~	20	480
5	[0]	~~0~~	[0]	~~470~~	~~40~~

Step 6: Now, go to Step 4 and repeat the procedure until we arrive at an optimal solution (assignment).

The minimum assignment cost is:170

6.3 Variations and Extensions

Variations and extensions in computational methods encompass a wide range of specialized techniques and approaches that build upon foundational concepts to address specific challenges or cater to particular application domains. Here are some notable variations and extensions:

Sparse and Structured Learning:

Sparse learning techniques aim to identify and exploit the inherent sparsity in high-dimensional data, leading to more efficient and interpretable models.

Structured learning methods incorporate domain-specific structural constraints into the learning process, such as graph regularization or manifold learning, to improve generalization performance.

Transfer Learning and Domain Adaptation:

Transfer learning enables knowledge transfer from a source domain to a target domain, where labeled data may be limited or unavailable.

Domain adaptation techniques aim to bridge the gap between source and target domains by aligning their feature distributions or learning domain-invariant representations.

Bayesian Inference and Probabilistic Graphical Models:

Bayesian inference methods provide a principled framework for incorporating prior knowledge and uncertainty into statistical modeling and decision-making.

Probabilistic graphical models, such as Bayesian networks and Markov random fields, capture complex dependencies among variables and enable efficient inference and reasoning.

Reinforcement Learning and Decision Making:

Reinforcement learning algorithms learn optimal decision-making policies through trial and error interactions with an environment, often in the context of sequential decision-making tasks.

Markov decision processes (MDPs) and partially observable Markov decision processes (POMDPs) formalize decision-making problems under uncertainty and are central to reinforcement learning.

Temporal and Sequential Data Analysis:

Time series analysis techniques, such as autoregressive models, recurrent neural networks (RNNs), and long short-term memory (LSTM) networks, are tailored for modeling and forecasting temporal data.

Sequential data analysis methods, including sequence alignment, sequence mining, and sequence generation, extract patterns and insights from ordered sequences of data.

Geospatial and Spatial-Temporal Analysis:

Geospatial analysis techniques leverage spatial data structures and algorithms to analyze and visualize geographical information, such as geographic information systems (GIS) and spatial databases.

Spatial-temporal analysis methods integrate spatial and temporal dimensions to model and analyze dynamic phenomena, such as climate patterns, transportation networks, and epidemiological trends.

Meta-Learning and Self-Supervised Learning:

Meta-learning algorithms enable models to learn how to learn across multiple tasks or domains, facilitating rapid adaptation to new learning scenarios.

Self-supervised learning techniques leverage unsupervised learning signals, such as data augmentation or pretext tasks, to learn useful representations from unlabeled data.

Causal Inference and Counterfactual Reasoning:

Causal inference methods aim to identify causal relationships and estimate causal effects from observational or experimental data, enabling causal reasoning and policy evaluation.

Counterfactual reasoning techniques model hypothetical scenarios to estimate the potential outcomes of different interventions or actions, aiding decision-making under uncertainty.

These variations and extensions in computational methods cater to diverse application domains and research areas, offering specialized tools and techniques to address complex challenges and unlock new opportunities for innovation and discovery. By exploring these advanced methods, researchers can push the boundaries of computational science and contribute to the advancement of knowledge and technology.

Exercise

Question 1: Solve the following transportation problem:

	W1	W2	W3	supply
F1	16	20	12	200
F2	14	8	18	160
F3	26	24	16	90
Demand	180	120	150	

Question 2: Solve the following job sequencing problem :

Jobs	Machines				
	A	B	C	D	E
1	30	37	40	28	40
2	40	24	27	21	36
3	40	32	33	30	35
4	25	38	40	36	36
5	29	62	41	34	39

Question 3:. Find the sequence of jobs that minimizes elapsed time to complete the jobs.

Jobs	Processing Time		
	Machine A	Machine B	Machine C
1	8	3	8
2	3	4	7
3	7	5	6
4	2	2	9
5	5	1	10
6	1	6	9

6.4 Summary

In a variety of fields, including project management, workforce scheduling, logistics, and transportation, assignment models are mathematical strategies that are used to improve the allocation of resources to tasks or activities. Assigning resources in the most effective way to minimize expenses, enhance revenues, or accomplish other predetermined goals is the main goal of assignment models, subject to limitations.

6.5 Keywords

- Assignment Problem
- Optimization

- Decision Variables
- Objective Function
- Constraints
- Hungarian Algorithm
- Transportation
- Logistics
- Resource Allocation

6.6 Self Assessment Questions

1. What is the primary objective of assignment models?
2. Explain the difference between linear programming and integer programming in the context of assignment models.
3. What are decision variables in an assignment model, and how are they used?
4. Describe the key components of formulating an assignment problem as a mathematical model.
5. What are some common applications of assignment models in real-world scenarios?
6. Briefly explain the Hungarian algorithm and its significance in solving assignment problems.

6.7 Case Study

Optimizing shift scheduling for its nursing staff is a difficulty for a major healthcare center that operates around the clock and consists of many departments. The facility seeks to minimize overtime expenses and retain employee satisfaction while always ensuring appropriate staffing levels.The healthcare institution must distribute nursing personnel throughout shifts in several departments while taking into account a number of variables, including skill requirements, employee preferences, labor laws, and patient care requirements.

Question: Establish a fair and effective shift plan that satisfies staff needs, reduces overtime costs, and encourages for work.

6.8 References

1. Hillier, F. S., & Lieberman, G. J. (2018). Introduction to Operations Research. McGraw-Hill Education.
2. Taha, H. A. (2016). Operations Research: An Introduction. Pearson.

UNIT - 7

Travelling Salesman Problem

Learning objectives

- Gain a comprehensive understanding of the Travelling Salesman Problem (TSP) and its significance in combinatorial optimization and operations research.
- Translate real-world scenarios into mathematical representations suitable for solving TSP.
- Analyze the computational complexity of TSP and its variants, including time complexity, space complexity, and the relationship between problem size and algorithm performance.

Structure

7.1 Introduction

One of the most well-known optimization problems in computer science and operations research is the Traveling Salesman Problem (TSP). Its main goal is to determine the quickest path that enables a salesman to travel to a group of locations precisely once and then return to the starting point.

The travelling salesman problem can be described as follows:

TSP $=\{(G, f, t): G=(V, E)$ a complete graph,

F is a function $V \times V \rightarrow Z$,

$T \in Z$,

G is a graph that contains a travelling salesman tour with cost that does not exceed}.

Example 1:

Consider the following set of cities:

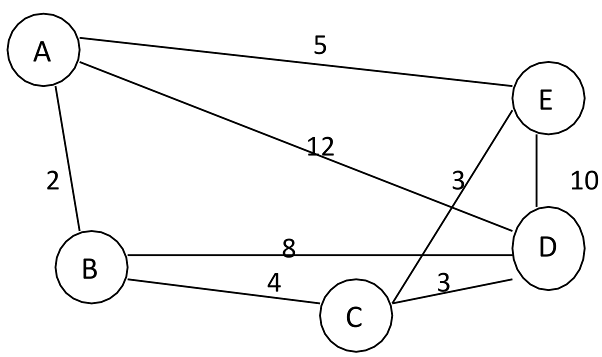


Figure 7.1 Travelling Salesman Path

The issue lies in finding a negligible way passing from all vertices once. For instance Path1 {A, B, C, D, E, A} and Path2 {A, B, C, E, D, A} pass all the vertices however Way 1 has an all out length of 24 and Way 2 has an all out length of 31.

Hamiltonian Cycle:

Example 2:

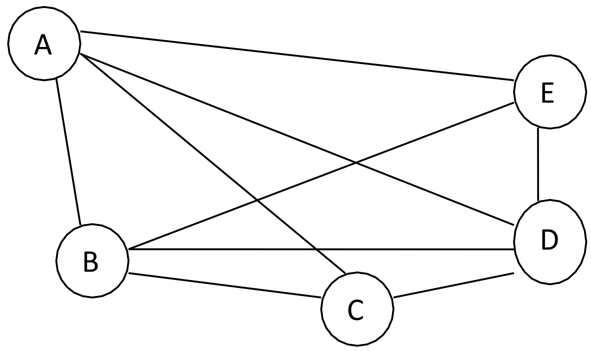


Figure 7.2 A graph with various Hamiltonian paths.

Let P ={A, B,C,D, E}is a Hamiltonian cycle.

7.2 Methods to solve the travelling salesman problem

In the event that for the arrangement of vertices a, b, c e V, The facts confirm that t (a, c) 5 t(o, b) + t(b, c) where t is the expense capability, we say that it fulfils the triangle disparity.

To start with, we make a base traversing tree the heaviness of which is a lower bound on the expense of an ideal mobile sales rep visit. Utilizing this base traversing tree, we will make a visit the expense of which is all things considered twice the heaviness of the spreading over tree. We present the calculation that plays out these calculations utilizing the MST-Demure calculation.

7.3 Trying to solve the travelling salesman problem using greedy algorithms

Consider the case of the asymmetric travelling salesman. We utilize the notation (Kn,c), where n is the number of vertices and c is the weight function. We assume that the definition of the symmetric travelling salesman problem is the same, with K_n denoting a full undirected graph. When employing heuristics to discover an approximate solution to an NP-hard problem, computational experiments are required to compare the results. A metric known as the dominance number is used to compare how well heuristics work. It is preferable to use a heuristic with a greater domination number rather than one with a lower one.

Definition: The domination number for the TSP of a heuristic A is an integer such as that for each instance l of the TSP on n vertices A produces a tour T that is now worse than at least d(n) tours in l including T.

When comparing the nearest neighbour and greedy algorithms for the TSP, we discover that while they perform well for the Euclidean TSP, they perform poorly for the symmetric and asymmetric TSPs. Using the dominance number, we examine below how well the nearest neighbour and greedy algorithms perform.

7.4 Summary

In summary, TSP represents a challenging optimization problem with widespread applications and continuous research efforts aimed at developing efficient algorithms and solving practical instances effectively.

7.5 Keywords

- Travelling Salesman Problem (TSP)
- Combinatorial Optimization
- Operations Research
- Mathematical Formulation
- Exact Algorithms

7.6 Self Assessment Questions

1. What is the objective of the Travelling Salesman Problem?
2. How is the Travelling Salesman Problem formulated mathematically?
3. Name one exact algorithm used to solve the Travelling Salesman Problem.
4. What is the complexity class of the Travelling Salesman Problem?
5. Provide an example of a heuristic algorithm used to approximate solutions for TSP.
6. What are some real-world applications of the Travelling Salesman Problem?
7. What are the challenges associated with solving large instances of the Travelling Salesman Problem?
8. How does the complexity of the Travelling Salesman Problem scale with the number of cities?

9. What interdisciplinary fields often collaborate to address challenges related to TSP?
10. What are some optimization software tools commonly used to solve the Travelling Salesman Problem?

7.7 Case Study

A courier company operates a fleet of delivery vehicles to transport parcels between multiple locations within a city. The company aims to minimize travel time and fuel costs by optimizing the routes taken by its vehicles. This case study demonstrates how the Travelling Salesman Problem (TSP) can be applied to optimize delivery routes efficiently.

Objectives:

Minimize Travel Distance: Develop optimal routes for delivery vehicles to minimize the total distance travelled while visiting all delivery locations exactly once.

Maximize Efficiency: Improve operational efficiency by reducing travel time and fuel consumption for delivery vehicles.

Ensure Timely Deliveries: Ensure timely deliveries to customers by optimizing routes to minimize delays and maximize vehicle utilization.

Adaptability: Design routes that can adapt to changes in delivery demand, traffic conditions, and vehicle availability throughout the day.

Cost Reduction: Reduce operational costs associated with vehicle maintenance, fuel expenses, and driver hours by optimizing delivery routes.

7.8 References

1. Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (2006). The Travelling Salesman Problem: A Computational Study. Princeton University Press.
2. Lawler, E. L., Lenstra, J. K., RinnooyKan, A. H. G., &Shmoys, D. B. (1985). The Travelling Salesman Problem. Wiley.

UNIT - 8

Transportation Models

Learning objectives

- Identify the key components of transportation models, including sources, destinations, supply, demand, and transportation costs.
- Formulate transportation problems as linear programming models, considering constraints on supply and demand, as well as transportation costs.
- Perform sensitivity analysis to evaluate the impact of changes in supply, demand, and transportation costs on the optimal solution.

Structure

8.1 Mathematical Formulation

Transportation problems belong to a unique family of linear programming problems where the goal is to satisfy supply constraints and demand requirements while minimizing the cost of moving a product from multiple sources (like factories) to multiple destinations (like warehouses). In addition to the direct transportation of a good, other domains that can be covered by the transportation model are inventory management, staff assignment, and work scheduling. There are other variables and restrictions in the problem. Therefore, the simplex method cannot be used to solve such a problem. This is the reason why solving the transportation problem requires a unique computational process.

The basic mathematical formulation:

Parameters:

m: Number of suppliers (sources).

n: Number of demand locations (destinations).

c: Cost of shipping one unit from supplier i to destination j.

si Supply available at supplier i.

: Demand at destination j.

Decision Variables:

x: Quantity of goods shipped from supplier i to destination j.

Objective Function:

Minimize the total transportation cost:

Minimize

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \text{cij} \cdot \text{xij}$$

Constraints:

Supply constraint: Ensure that the total quantity shipped from each supplier does not exceed its supply capacity.

$$\sum_{j=1}^{n} x_{ij} \leq s_i \quad \text{, for } i=1,2,\ldots,m$$

Demand constraint: Ensure that the total quantity received at each destination meets its demand.

$$\sum_{i=1}^{n} x_{ij} \leq d_j \text{ , for } j=1, 2, \ldots, n$$

Non-negativity constraint: Ensure that the quantities shipped are non-negative.

$$x_{ij} \geq 0, \text{ for } i=1, 2, ..., m \text{ and } j=1, 2, ..., n$$

Explanation:

The goal of the objective function is to minimize the total cost, which is calculated by multiplying the quantity delivered by the total cost of shipping from each source to each destination.

The supply restriction makes sure that each supplier's overall shipment volume stays within its supply capacity.

The demand constraint makes sure that each destination's total quantity received satisfies its demand.

The amounts sent are guaranteed to be non-negative by the non-negativity criterion.

This formulation allows for solving the transportation problem using various optimization techniques, such as the simplex method, transportation simplex method, or specialized algorithms like the North-West Corner Method, Vogel's Approximation Method, or the Minimum Cost Method.

8.2 Initial Basic Feasible Solution

In the transportation problem, finding an initial basic feasible solution (IBFS) is crucial for starting iterative optimization algorithms like the transportation simplex method. Here are a few methods to obtain an IBFS:

1. North-West Corner Method:

This method starts by allocating shipments from the top-left (North-West) corner of the cost matrix and iterates through each cell by moving South or East, prioritizing the cell with the smallest supply or demand.

2. Minimum Cost Method (or Least Cost Method):

This method selects the cell with the smallest transportation cost and allocates shipments until either the supply or demand of the corresponding row or column is exhausted.

3. Vogel's Approximation Method (VAM):

Vogel's method finds the difference between the two smallest costs in each row and each column and selects the largest difference. Then, it allocates shipments in the corresponding cell and adjusts the supply and demand accordingly.

Top of Form

Transportation Algorithm:-The simplex algorithm and the transportation algorithm have identical steps, which are as follows:

Step 1: Using any one of the following three techniques, identify a simple, workable solution

1. The North West Corner Approach
2. The Lowest Possible Cost Approach
3. The Vogel Approximation Technique

Step 2: Apply the following approach to find the best answer.

1. The UV Method or MODI (Modified Distribution Method).

One of the following three approaches can be used to secure a non-artificial fundamental viable solution due to the unique nature of the transportation problem.

- North West Corner Method (NWCM)
- Least Cost Method (LCM)
- Vogel Approximation Method (VAM)

The quality of the first basic viable solution produced by each of these three approaches varies, with a better initial solution producing a smaller objective value. Though the North West Corner Method requires fewer computations, it typically yields the worst first basic viable solution, whereas the Vogel Approximation Method produces the best.

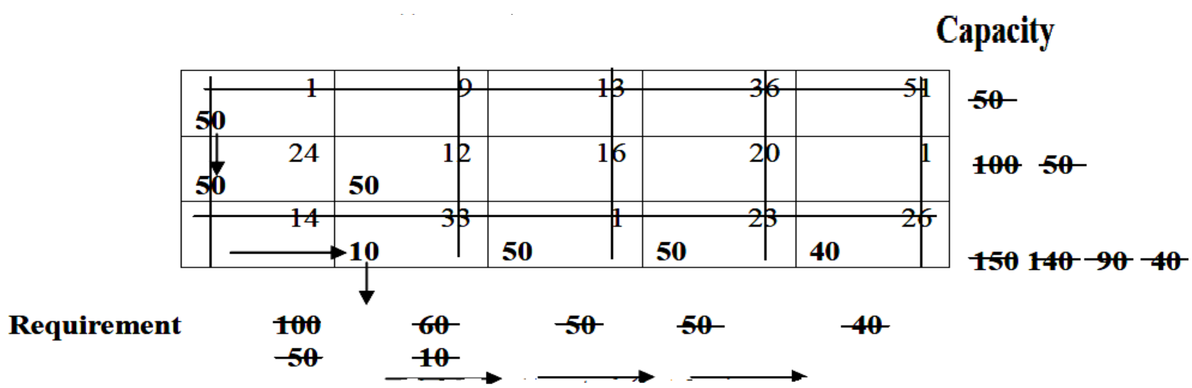
North West Corner Method:

Example 1:

To demonstrate how to apply the North West Corner Method for identifying a basic workable solution, let's look at the problem presented in Example 1.1.

	Retail Agency					
Factories	1	2	3	4	5	Capacity
1	1	9	13	36	51	50
2	24	12	16	20	1	100
3	14	33	1	23	26	150
Requirement	100	60	50	50	40	300

Solution:



The arrows indicate the generation sequence of the allotted (bolded) quantities. The initial fundamental solution is provided as

X11=50,X21=50,X22=50X32=10,X33= 50,X34=50,X35=40

The corresponding transportation cost is

50x1+50x24+50x12+10x33+50x1+50x23+ 40x26=4420

Least Cost Method:

Since we seek out the row and column that correspond to the value of Cij that is minimum, the LCM is sometimes referred to as the matrix minimum technique. The cheapest paths are the focus of this strategy, which finds a superior initial fundamental workable answer. As in the NWC-Method, we begin the allocation by giving as greatly as we can to the cell with the lowestone cost rather than beginning with the northwest cell. We ought to choose the row and the column if there are two or more minimum expenses. being in line with the row with the lower number.

Only one of the arrows and the column is crossed out if they are satisfied at the same time. Once we have exactly one uncrossed-out row or column at the end, we search for the uncrossed-out cell with the lowest unit cost and repeat the process.

Example 2:

Determine the initial basic feasible solution using Least Cost Method Problem

	Retail Agency					
Factories	1	2	3	4	5	Capacity
	1 **50**	9	13	36	51	**~~50~~**
	24	12 **60**	16	20	1 **40**	**~~100~~ ~~60~~**
	14 **50**	33	1 **50**	23 **50**	26	**~~150~~ ~~100~~ ~~50~~**
Requirement	**~~100~~ ~~50~~**	**~~60~~**	**~~50~~**	**~~50~~**	**~~40~~**	

Solution:

In order for the LCM's fundamental workable solution to have a transportation cost,

1x50+ 12x60+ 1x40+ 14x50+ 1x50+ 23x50= 2710

Vogel Approximation Method (VAM):

The least cost approach has been enhanced by VAM, which typically yields superior results. The procedure in this approach is:

Example 3:

Solve the following transportation problem

Origin	Destination				a_i
	1	2	3	4	
1	20	22	17	4	120
2	24	37	9	7	70
3	32	37	20	15	50
b_j	60	40	30	110	240

Solution:

Origin	Destination				a_i	Column Penalty
	1	2	3	4		
1	20	22	17	4	120	13
2	24	37	9	7	70	2
3	32	37	20	15	50	5
b_j	60	40	30	110	240	
Row Penalty	4	15	8	3		

Origin	Destination				a_i	Column Penalty
	1	2	3	4		
1	20	22 **40**	17	4	80	13
2	24	37	9	7	70	2
3	32	37	20	15	50	5
b_j	60	40	30	110	240	

Origin	Destination				a_i	Column Penalty
	1	2	3	4		
1	20	22 **40**	17	4 **80**	0	13
2	24	37	9	7	70	2
3	32	37	20	15	50	5
b_j	60	40	30	110	240	
Row Penalty	4	15	8	3		

Origin	Destination				a_i	Column Penalty
	1	2	3	4		
1	20	22 **40**	17	4 **80**	0	13
2	24	37	9 **30**	7	40	17
3	32	37	20	15	50	17
b_j	60	40	30	110	240	
Row Penalty	8	15	8	8		

Origin	Destination				a_i Column Penalty
	1	2	3	4	
1	20	22 40	17	4 80	0 13
2	24	37	9 30	7 30	10 17
3	32	37	20	15	50 17
b_j	60	40	30	110	240

Cost =22x40+ 4x80+ 9x30+ 7x30+ 24x10+ 32x50= Rs.3520

MODI Method:

The u-v strategy, normally known as the MODI technique or the Altered Dissemination method, offers an ideal arrangement at the most minimal conceivable expense for the transportation issue.

Question: Solve the transportation problem and optimize the solution.

						Supply
	1	9	13	36	51	50
	24	12	16	20	1	100
	14	33	1	23	26	150
Demand	**100**	**70**	**50**	**40**	**40**	**300**

Solution:-

Hint: Finding the most fundamentally workable answer must come first. The simplest workable solution that applies the least-cost strategy is

X11=50, X22=60, X25=40, X31=50, X32=10, X33=50 and X34=40

Where C12–U1–V2=9+13-33=-11.

8.3 Degeneracy and Unbalanced Problems

Unbalanced transportation problems occur when the total supply and demand are not equal. If the overall supply exceeds the total demand in the unbalanced transportation problem, we add an extra column to show the excess supply with zero transportation cost. Likewise, in the event that the aggregate demand surpasses the aggregate supply, a new row is appended to the transportation table, signifying unfulfilled demand at no cost of transportation.

Example 4 :

Evaluate the solution of unbalanced transportation problem given below as

Plant	w_1	w_2	w_3	Supply
X	20	17	25	400
Y	10	10	20	500
Demand	400	400	500	

Solution: -

Hint: In this situation, there is 900 total supply and 1300 total demand. To represent the unmet demand, we will now add a new row with zero transportation cost.

	Warehouses			
Plant	w_1	w_2	w_3	Supply
X	20	17	25	400
Y	10	10	20	500
Unsatisfied Demand	0	0	0	400
Demand	400	400	500	1300

Degenerate Transportation Problem:-

Degeneracy in a Transportation Problem arises when the number of filled cells in the transportation table (basic variables) is less than $m+n-1$, where m is the number of supply points and n is the number of demand points.

Exercise:

1. In a transportation problem, what do you mean by degeneracy?

2. Provide a mathematical solution to the transportation problem.

3. Apply Vogel's approximation technique to derive a transportation problem's first undamentally workable solution and then identify the best one.

8.4 Summary

Transportation models play a crucial role in optimizing transportation operations, improving efficiency, reducing costs, and enhancing sustainability in logistics and supply chain management.

8.5 Keywords

- Transportation models
- Linear programming
- Optimization
- Logistics
- Supply chain management

8.6 Self Assessment Questions

1. What is the primary objective of transportation models in operations research?
2. How are transportation problems formulated mathematically?
3. Name one optimization technique used to solve transportation models.
4. What is the purpose of sensitivity analysis in transportation modeling?
5. Provide an example of a real-world application of transportation models.
6. How do transportation models contribute to supply chain management?
7. What types of constraints are typically included in transportation models?
8. What software tools are commonly used to solve transportation problems?
9. Explain the significance of decision support systems in transportation modeling.
10. What emerging trends are influencing the development of transportation models?

8.7 Case Study

Urban freight transportation plays a crucial role in supplying goods to businesses and consumers in densely populated areas. However, it also poses significant challenges related to congestion, pollution, and inefficiency. This case study illustrates how transportation models can be applied to optimize urban freight transportation.

Objectives: Minimize Transportation Costs: Develop a transportation plan that minimizes transportation costs for delivering goods within urban areas.

8.8 References

1. Geoffrion, A. M., & Graves, G. W. (1974). Multicommodity distribution system design by benders decomposition. Management Science, 20(5), 822-844.
2. Ballou, R. H., &Pazer, H. L. (1985). Modeling the costs of urban freight distribution. Transportation Science, 19(4), 362-383.

UNIT - 9

Advanced Game Theory

Learning objectives

- Explores equilibrium concepts such as sub game perfect equilibrium, Bayesian Nash equilibrium, and evolutionary stable strategies.
- Analyzes strategic interactions involving sequential decision-making and timing considerations.
- Investigates solution concepts like the core, Shapley value, and bargaining solutions.

Structure

9.1 Introduction to Game Theory

The different definition of games is given by

"Game theory, more properly the theory of games of strategy, is a mathematical method of analyzing a conflict. The alternative is not between this decision or that decision, but between this strategy or that strategy to be used against the conflicting interest".

"Game theory is a mathematical technique helpful in making decisions in situations of conflicts, where the success of one part depends at the expense of others, and where the individual decision maker is not in complete control of the factors influencing the outcome".

"The 'Game' is simply the totality of the rules which describe it. Every particular instance at which the game is played – in a particular way – from beginning to end is a 'play'. The game consists of a sequence of moves, and the play of a sequence of choices".

"A game is a competitive situation where two or more persons pursue their own interests and no person can dictate the outcome. Each player, an entity with the same interests, make his own decisions. A player can be an individual or a group".

Game theory aids in determining a company's optimal course of action in light of predicted countermoves from rival companies. If the following characteristics are true, a competitive scenario is a competitive game:

1. Let's imagine there are a finite number of competitors, N.
2. Each of the N contestants has a limited number of options for how to proceed.
3. Each contestant chooses a plan of action from the options presented to him, and this results in a play of the game. The idea that each player chooses their course of action simultaneously is crucial to game theory. Consequently, no rival will be in a position to ascertain the decisions made by his rivals.
4. A play's conclusion is determined by the specific actions that each participant does. Every possible result has an associated set of payments, one for each player, and might be zero, negative, or positive.

Saddle point:

A seat point in a game is the area in the prize framework where the base of the section maxima and the limit of the column minima are equivalent. The worth of the game is the result at the seat point, and the matching strategies are known as the unadulterated methodologies.

Strength: Something like one of the excess players' procedures might be better than one of different players'. It is said that the better methodologies offset the more terrible ones.

Types of Games:

Games for two players and several players: There are precisely two participants in two-player games, and each player can only use a limited number of methods. A game is referred to as an n-person game if there are more players than two.

Zero sum and non-zero sum games: A game is referred to as a zero sum game if the total amount of payments made to all players for every possible result in the game is zero. A game is referred to as a non-zero sum game if the total payoffs from every play are positive or negative but not zero.

There are two types of games: perfect information games and imperfect information games. In a perfect information game, every player knows the opponent's plan. However, a game with imperfect information is one in which a player can only move forward in his game by making educated guesses about the opponent's strategy because no player can predict it beforehand.

Games for a limited number of players or moves, as well as games with an infinite number of moves: A game in which each player's number of moves is set before the play begins is known as a finite number of moves game. Conversely, we refer to a game as having an infinite number of moves if it may be played for a long time and there are no restrictions on how many moves any player can make..

Constant-sum games: We refer to a game as constant-sum if the total of the payouts to both players in each scenario is constant but the game's sum is not zero.

9.2 Nash Equilibrium

The main idea in game theory is called Nash Equilibrium, after the mathematician and Nobel laureate John Nash. In a strategic encounter, it denotes a situation in which no player has any motivation to unilaterally alter their approach.

Formal Definition: In a game with N players, where each player i has a strategy set Si and utility function $ui(s1,s2,...,sN)$, a Nash Equilibrium is a set of strategies $(s1*,s2*,...,sN*)$ such that for each player i:

$$ui(s1*,s2*,...,sN*) \geq ui(s1,s2*,...,sN*)$$

for all $s1$ in $S1$, and similarly for all other players j.

Characteristics:

Stability: In a Nash Equilibrium, no player can improve their outcome by changing their strategy unilaterally.

Mutual Best Response: Every player's strategy is their best reaction to every other player's selected plan.

No Regret: Players have no regret for choosing their strategies once a Nash Equilibrium is reached.

Applications:

- Nash Equilibrium is widely used in economics, political science, biology, and other fields to analyze strategic interactions.
- It helps in understanding the behaviour of firms in markets, negotiation strategies, and even evolutionary stability in biological systems.

In summary, Nash Equilibrium is a fundamental concept that provides insight into strategic decision-making in competitive situations where each actor's decision affects others' outcomes.

9.3 Mixed Strategy Equilibrium

Mixed Strategy Equilibrium is a concept in game theory where players, instead of choosing a single pure strategy, choose a probability distribution over their possible pure strategies. In a mixed strategy equilibrium, each player's strategy because considering the other players' strategies, no player has an incentive to unilaterally stray from their selected course of action.

Characteristics:

Randomized Strategies: Players randomize their choices based on probability distributions over their pure strategies.

No Dominant Strategy: Regardless of the strategies used by other players, there isn't a dominating method among them that offers a larger payout.

Expected Utility Maximization: Players maximize their expected utility given the strategies of the other players.

Example 1:

Imagine a straightforward penny matching game in which Player 1 selects heads (H) or tails (T) at the same time as Player 2 does the same. Player 1 earns $1 if both players match—that is, if both select H or T—and Player 2 wins $1 if not.

In this game, neither player has a dominant strategy. Therefore, a mixed strategy equilibrium can be achieved where both players randomize their choices with equal probability: $p1(H)=p2(H)=p$ and $p1(T)=p2(T)=1-p$. In this equilibrium, neither player has an incentive to deviate from their strategy because changing their probability distribution would not improve their expected payoff.

Applications:

Mixed strategy equilibria are used to analyze complex games where pure strategies alone may not capture the strategic interactions effectively.

They are applied in various fields such as economics, political science, biology, and evolutionary game theory to model behaviour in uncertain or strategic environments.

9.4 Summary

Advanced Game Theory provides a sophisticated framework for understanding and analyzing strategic interactions in complex, dynamic environments. By exploring advanced concepts, techniques, and applications, researchers can tackle diverse challenges and contribute to cutting-edge research at the intersection of economics, computer science, and other disciplines.

9.5 Keywords

- Equilibrium concepts
- Subgame perfect equilibrium
- Bayesian Nash equilibrium
- Evolutionary stable strategies

- Dynamic games

9.6 Self AssessmentQuestions

1. What are some equilibrium concepts beyond Nash equilibrium?
2. How do dynamic games differ from static games in game theory?
3. What is the significance of subgame perfect equilibrium in sequential decision-making?
4. How does cooperative game theory differ from non-cooperative game theory?
5. What are some solution concepts used in cooperative game theory?
6. What is the role of algorithmic game theory in addressing computational complexity in strategic interactions?
7. How are empirical methods used to validate theoretical predictions in game theory?
8. What are some applications of game theory in multi-agent systems and artificial intelligence?
9. How does strategic reasoning contribute to decision-making in complex environments?
10. What interdisciplinary applications does game theory have in fields such as economics, computer science, and political science?

9.7 Case Study

Governments often allocate radio frequency spectrum to telecommunications companies through auctions. However, designing auctions that encourage competitive bidding while maximizing revenue presents challenges. This case study demonstrates the application of advanced game theory concepts in designing spectrum auctions.

Objectives: Maximize Revenue: Design an auction mechanism that maximizes government revenue from spectrum allocation.

9.8 References

1. Nisan, N., Roughgarden, T., Tardos, É.,&Vazirani, V. V. (Eds.). (2007). Algorithmic Game Theory. Cambridge University Press.
2. Myerson, R. B., &Satterthwaite, M. A. (1983). Efficient Mechanisms for Bilateral Trading. Journal of Economic Theory, 29(2), 265-281.

UNIT - 10

Game Problems using Graphical Method

Learning Objectives

- Define the fundamental concepts of game theory, including players, strategies, payoffs, and equilibrium.
- Learn how to represent game problems graphically using payoff matrices or strategic form games.
- Develop skills to solve game problems using graphical methods, including iterative elimination of dominated strategies and graphical inspection of payoff matrices.

Structure

10.1 Graphical Solution Techniques

The graphical method is often used to solve two-player, zero-sum games, particularly those with a finite number of strategies for each player. In this method, a graphical representation called a payoff matrix is constructed, for every cell represents the payoff to the row player and the negative of the payoff to the col. player.

Graphical solution techniques are often used to analyze and solve two-player, zero-sum games, particularly those with a small number of strategies for each player. These techniques provide a visual representation of the game's payoff structure and help identify optimal strategies for both players. Here are some graphical solution techniques commonly used:

1. Payoff Matrix:

$$\begin{array}{c} \\ A_1 \\ A_2 \\ \cdot \\ \cdot \\ A_m \end{array} \begin{array}{c} \begin{array}{cccc} B_1 & B_2 & \cdots\cdots & B_n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \cdots\cdots & a_{1n} \\ a_{21} & a_{22} & \cdots\cdots & a_{2n} \\ \cdot & & & \\ \cdot & & & \\ a_{m1} & a_{m2} & \cdots\cdots & a_{mn} \end{bmatrix} \end{array}$$

2. Graphical Representation:

Make a grid with rows standing for the strategies of the row players and columns for the strategies of the column players. Plot the payoffs from the payoff matrix in the corresponding cells.

3. Dominance Analysis:

Determine which player's dominated strategies are. When one method consistently yields a larger reward than the opponent's strategy, it is said to be dominated.

Take dominated strategies out of the equation.

4. Optimal Strategy Identification:

For the row player, identify the maximum payoff in each row and choose the row with the highest maximum payoff.

For the column player, identify the minimum payoff in each column and choose the column with the lowest minimum payoff.

5. Graphical Solution:

Once dominated strategies are eliminated and optimal strategies are identified, mark the optimal strategies on the graphical representation.

The intersection of the optimal strategies indicates the optimal outcome of the game.

6. Interpretation:

Interpret the result in the context of the game. The optimal outcome represents the best response for both players given the available strategies.

Here's how you can approach solving game problems using the graphical method:

Steps:

Construct the Payoff Matrix:

Identify the strategies available to each player.

Fill in the payoffs for each combination of strategies, where the row player's payoff is positive and the column player's payoff is negative.

Plot the Payoff Matrix:

Make a grid with the strategies for the row player represented by the rows and the column player by the columns. Plot the payoffs from the payoff matrix in the corresponding cells.

Find the Optimal Strategy:

If there are no dominated strategies, find the optimal strategy for each player.

For the row player, identify the maximum payoff in each row and choose the row with the highest maximum payoff.

For the column player, identify the minimum payoff in each column and choose the column with the lowest minimum payoff.

Interpret the Result:

Once the optimal strategies are identified, interpret the result in the context of the game.

Example 1:

Consider the following payoff matrix for a simple game:

Construct the Payoff Matrix:
Fill in the payoffs for each combination of strategies.

Plot the Payoff Matrix:

Draw a grid and plot the payoffs in the corresponding cells.

Identify Dominance:

Look for any dominated strategies for each player.

Find the Optimal Strategy:

Identify the maximum payoff in each row and the minimum payoff in each column.

Choose the row with the highest maximum payoff).

Choose the column with the lowest minimum payoff.

Interpret the Result:

This process allows you to systematically analyze and solve game problems using the graphical method, providing insights into optimal strategies and payoffs for each player.

10.2 Application in Various Scenarios

Graphical solution techniques in game theory have wide-ranging applications across various scenarios, including economics, business, politics, and social interactions. Here are some examples of how these techniques are applied in different contexts:

1. Economics:

- **Market Competition:** Analyzing strategies of firms in oligopolistic markets to determine optimal pricing and output decisions.
- **Resource Allocation:** Evaluating bidding strategies in auctions to maximize utility or profit.
- **Labor Negotiations:** Studying wage bargaining between labor unions and firms to understand optimal negotiation strategies.

2. Business Strategy:

- **Product Positioning:** Determining optimal product positioning strategies by analyzing competitive interactions and consumer preferences.

- **Advertising and Promotion:** Assessing the effectiveness of advertising and promotional strategies by modelling competitive advertising games.
- **Strategic Alliances:** Analyzing strategic alliances and partnerships to identify optimal collaboration strategies in competitive markets.

3. Politics and International Relations:

- **Negotiation and Diplomacy:** Understanding strategic interactions between countries in international relations to analyze negotiation and conflict resolution strategies.
- **Election Campaigns:** Modeling electoral competition between political parties to study campaign strategies and voter behaviour.
- **Policy Making:** Analyzing policy decisions and their impact on various stakeholders to inform optimal policy choices.

4. Social Interactions:

- **Network Formation:** Studying the formation of social networks and the evolution of cooperation and competition within networks.
- **Peer Influence:** Analyzing peer effects and social influence in decision-making processes, such as adoption of new technologies or behaviours.
- **Resource Sharing:** Examining cooperative behaviours and strategies for resource sharing in social dilemmas, such as the tragedy of the commons.

5. Sports and Games:

- **Sports Strategy:** Evaluating game strategies in sports competitions, such as soccer or basketball, to optimize player positions and tactics.
- **Board Games:** Analyzing optimal strategies in board games like chess, poker, or tic-tac-toe to improve game play and decision-making.

Graphical solution techniques provide a powerful framework for analyzing strategic interactions and decision-making in diverse scenarios. By visualizing payoff structures and identifying optimal strategies, these techniques help individuals and organizations to make better decisions and achieve better outcomes in competitive environments.

10.3 Summary

Graphical methods offer a powerful tool for analyzing game problems, providing a visual framework for understanding strategic interactions, identifying optimal strategies, and predicting equilibrium outcomes. By leveraging graphical representations, decision-makers can make informed strategic choices and navigate complex decision environments effectively.

10.4 Keywords

- Game theory
- Strategic interactions
- Players
- Strategies
- Payoffs

10.5 Self Assessment Questions

1. What is the primary goal of using graphical methods in analyzing game problems?
2. Define dominance in the context of game theory.
3. How are best responses identified graphically in game problems?
4. What is a Nash equilibrium, and how is it located using graphical methods?
5. Explain the concept of mixed strategies in game theory.
6. How do graphical methods aid in identifying optimal strategies in game problems?
7. Describe one real-world application where graphical methods are used to analyze strategic interactions.
8. What are the advantages of using graphical representations in solving game problems?
9. How does iterative elimination of dominated strategies contribute to finding equilibrium outcomes?
10. What role does visualization play in understanding and interpreting solutions to game problems?

10.6 Case Study

A retail chain operates multiple stores in a competitive market where pricing decisions directly impact profitability and market share. The chain wants to optimize its pricing strategy to maximize revenue while considering the reactions of competitors. This case study demonstrates the use of game theory and graphical methods to analyze and optimize pricing strategies.

Objectives: Maximize Revenue: Determine pricing strategies that maximize revenue for the retail chain.

10.7 References

1. Osborne, M. J., & Rubinstein, A. (1994). A Course in Game Theory. MIT Press.
2. Binmore, K. (2007). Playing for Real: A Text on Game Theory. Oxford University Press.

UNIT - 11

Queuing Models

Learning objectives

- Define the basic components of queuing models, including arrival processes, service processes, queue disciplines, and system capacity.
- Differentiate between various types of queuing models such as M/M/1, M/M/c, M/G/1, G/G/1, and queuing networks.
- Develop simulation models to analyze complex queuing systems that cannot be solved analytically.

Structure

11.1 Queuing Theory

Queuing theory is a significant area within operations research that focuses on the analysis of waiting lines or queues. This theory is widely applicable in various fields, including telecommunications, traffic engineering, computing, and the service industry, to optimize the process and manage the flow of items or individuals through a system.

Here's a basic overview of the key concepts and applications of queuing theory:

Key Concepts

1. **Queue Components**:
    - **Arrival Process**: Describes how customers arrive at the queue. Commonly modeled as a Poisson process.
    - **Service Process**: Describes how customers are served. Often modeled with exponential service times.
    - **Number of Servers**: The number of parallel service channels available.
    - **System Capacity**: The maximum number of customers that can be in the system (both waiting and being served).

2. **Performance Metrics**:
    - **Average Waiting Time**: The expected time a customer spends in the queue.
    - **Average Queue Length**: The expected number of customers in the queue.
    - **Utilization Factor**: The fraction of time servers are busy.
    - **Probability of n Customers in the System**: The likelihood of having a specific number of customers in the system.

3. **Common Queue Models**:
    - **M/M/1**: A single-server queue with Poisson arrivals and exponential service times.
    - **M/M/c**: A multi-server queue with Poisson arrivals and exponential service times.

- **M/G/1**: A single-server queue with Poisson arrivals and general service time distribution.
- **G/G/1**: A single-server queue with a general arrival process and general service time distribution.

Applications

1. **Telecommunications**: Managing data packets in a network, ensuring minimal delays and efficient data transmission.
2. **Healthcare**: Scheduling patients in hospitals, optimizing staff allocation, and reducing patient wait times.
3. **Manufacturing**: Controlling production lines, minimizing downtime, and managing inventory.
4. **Service Industry**: Managing customer service operations in banks, call centers, and retail stores to reduce wait times and improve service efficiency.
5. **Traffic Engineering**: Optimizing traffic flow, reducing congestion, and improving signal timings at intersections.

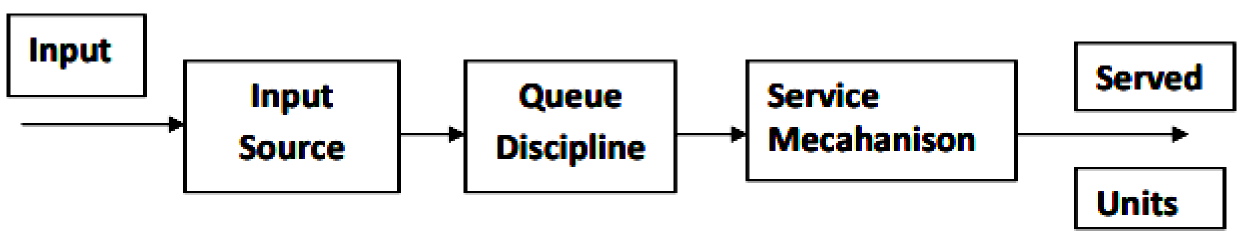


Fig 11.1.1 : Structure of Queuing Theory

11.2 Birth-Death Processes

Birth-death processes are a special type of Markov process that are particularly useful in modeling queuing systems. In these processes, "births" represent arrivals to the system, and "deaths" represent departures from the system. This framework allows for a structured analysis of queues where the state of the system changes due to these arrivals and departures.

Key Concepts of Birth-Death Processes

1. **States**: Represent the number of customers in the system.

2. **Transition Rates**:

- **Birth Rate (λn\lambda_nλn)**: The rate at which customers arrive when the system is in state nnn.
- **Death Rate (μn\mu_nμn)**: The rate at which customers are served and leave when the system is in state nnn.

Common Birth-Death Queue Models

1. M/M/1 Queue

- **Arrival Rate (λ\lambdaλ)**: Constant for all states.
- **Service Rate (μ\muμ)**: Constant for all states.

Equilibrium Probabilities: The probability PnP_nPn of having nnn customers in the system is given by:

$$P_n = P_0 \left(\frac{\lambda}{\mu}\right)^n$$

Performance Metrics:

- **Average number of customers in the system (L)**:

$$L = \frac{\lambda}{\mu - \lambda}$$

- **Average time a customer spends in the system (W)**:

$$W = \frac{1}{\mu - \lambda}$$

2. M/M/c Queue

- **Arrival Rate (λ)**: Constant for all states.
- **Service Rate (μ)**: Each of the c servers works at rate μ, so the total service rate depends on the number of customers n in the system.

Equilibrium Probabilities: The probability P_0 that the system is empty is given by:

$$P_0 = \left[\sum_{n=0}^{c-1} \frac{1}{n!} \left(\frac{\lambda}{\mu} \right)^n + \frac{1}{c!} \left(\frac{\lambda}{\mu c} \right)^c \frac{1}{1 - \frac{\lambda}{\mu c}} \right]^{-1}$$

Performance Metrics:

- **Average number of customers in the system (L)**:

$$L = \sum_{n=0}^{c-1} n P_n + \sum_{n=c}^{\infty} n P_n$$

- **Average time a customer spends in the system (W)**:

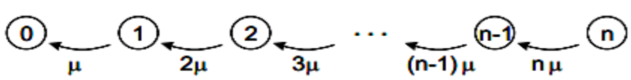
$$W = \frac{L}{\lambda}$$

Example 1. Pure death process

$$\begin{cases} \lambda_i = 0 \\ \mu_i = i\mu \end{cases} \quad i = 0, 1, 2, \ldots \qquad \pi_i(0) = \begin{cases} 1 & i = n \\ 0 & i \neq n \end{cases}$$

all individuals have the same mortality rate μ

the system starts from state n



State 0 is an absorbing state, other states are transient

$$\begin{cases} \frac{d}{dt}\pi_n(t) &= -n\mu\pi_n(t) & \Rightarrow \quad \pi_n(t) = e^{-n\mu t} \\ \frac{d}{dt}\pi_i(t) &= (i+1)\mu\pi_{i+1}(t) - i\mu\pi_i(t) & i = 0, 1, \ldots, n-1 \end{cases}$$

$$\frac{d}{dt}(e^{i\mu t}\pi_i(t)) = (i+1)\mu\pi_{i+1}(t)e^{i\mu t} \quad \Rightarrow \quad \pi_i(t) = (i+1)e^{-i\mu t}\mu \int_0^t \pi_{i+1}(t')e^{i\mu t'}dt'$$

$$\pi_{n-1}(t) = ne^{-(n-1)\mu t}\mu \int_0^t \underbrace{e^{-n\mu t'}e^{(n-1)\mu t'}}_{e^{-\mu t'}} dt' = n\, e^{-(n-1)\mu t}(1 - e^{-\mu t})$$

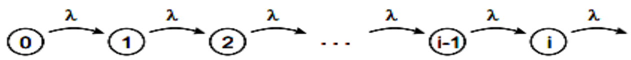
Recursively $$\boxed{\pi_i(t) = \binom{n}{i}(e^{-\mu t})^i(1 - e^{-\mu t})^{n-i}}$$

Binomial distribution: the survival probability at time t is $e^{-\mu t}$ independent of others

Example 2. Pure birth process (Poisson process)

$$\begin{cases} \lambda_i = \lambda \\ \mu_i = 0 \end{cases} \quad i = 0, 1, 2, \ldots \qquad \pi_i(0) = \begin{cases} 1 & i = 0 \\ 0 & i > 0 \end{cases}$$

birth probability per time unit is constant λ — initially the population size is 0



All states are transient

$$\begin{cases} \frac{d}{dt}\pi_i(t) &= -\lambda\pi_i(t) + \lambda\pi_{i-1}(t) & i > 0 \\ \frac{d}{dt}\pi_0(t) &= -\lambda\pi_0(t) & \Rightarrow \quad \pi_0(t) = e^{-\lambda t} \end{cases}$$

$$\frac{d}{dt}(e^{\lambda t}\pi_i(t)) = \lambda\pi_{i-1}(t)e^{\lambda t} \quad \Rightarrow \quad \pi_i(t) = e^{-\lambda t}\lambda\int_0^t \pi_{i-1}(t')e^{\lambda t'}dt'$$

$$\pi_1(t) = e^{-\lambda t}\lambda\int_0^t \underbrace{e^{-\lambda t'}e^{\lambda t'}}_{1}dt' = e^{-\lambda t}(\lambda t)$$

Recursively $\boxed{\pi_i(t) = \frac{(\lambda t)^i}{i!}e^{-\lambda t}}$ Number of births in interval $(0, t) \sim \text{Poisson}(\lambda t)$

11.3 Markovian Queuing Models

Markovian queuing models are a subset of queuing theory where the arrival and service processes follow an exponential distribution, which means they exhibit the memoryless property. These models are described using Markov chains and are characterized by their simplicity and analytical tractability. Here, we will discuss several common Markovian queuing models, their characteristics, and how they are analyzed.

Common Markovian Queuing Models

1. M/M/1 Queue
2. M/M/c Queue
3. M/M/∞ Queue
4. M/M/c/K Queue
5. M/M/c/c Queue

1. M/M/1 Queue

Characteristics:

- **Arrival Process**: Poisson process with rate λ

- **Service Process**: Exponential distribution with rate μ
- **Number of Servers**: 1
- **System Capacity**: Infinite

2. M/M/c Queue

Characteristics:
- **Arrival Process**: Poisson process with rate λ
- **Service Process**: Exponential distribution with rate μ
- **Number of Servers**: c
- **System Capacity**: Infinite

3. M/M/∞ Queue

Characteristics:
- **Arrival Process**: Poisson process with rate λ
- **Service Process**: Exponential distribution with rate μ
- **Number of Servers**: Infinite
- **System Capacity**: Infinite

4. M/M/c/K Queue

Characteristics:
- **Arrival Process**: Poisson process with rate λ
- **Service Process**: Exponential distribution with rate μ
- **Number of Servers**: c
- **System Capacity**: K (maximum number of customers in the system)

5. M/M/c/c Queue (Erlang B model)

Characteristics:
- **Arrival Process**: Poisson process with rate λ
- **Service Process**: Exponential distribution with rate μ
- **Number of Servers**: c
- **System Capacity**: c (no waiting space, customers are blocked if all servers are busy)

Question: Landings in a phone both are viewed as Poisson at a typical season of 8 min between our appearance and the following. The length of the calls circulated dramatically, with 'a mean of 4 min, Decide:

(a) Normal part of the day that the telephone will be being used.

(b) Expected number of units in the line Anticipated holding up time in the line.

(c) Anticipated number of units in the framework.

(e) Anticipated holding up time in the framework

(f) Expected number of units in line that now and again.

(g) What is the likelihood that an appearance should sit tight in line for administration?

(h) What is the likelihood that precisely 3 units are in framework

(1) What is the likelihood that an appearance won't need to sit tight in line for administration?

(j) What is the likelihood that there are at least 3 units in the framework?

(k) What is the likelihood that an appearance should stand by in excess of 6 min in line for administration?

(l) What is the likelihood that in excess of 5 units in framework

(m) What is the likelihood that an appearance should stand by in excess of 8 min in framework?

(n)Telephone Organization will introduce a second corner when persuaded that an appearance would have to hang tight for authenticate 6 min in line for telephone.

Solution:

The mean arrival rate λ

$$\lambda = \frac{1}{8} X\, 60 = 7.5\, per\ hour$$

The mean service rate (μ)

$$\mu = \frac{1}{4} X\, 60 = 15\, per\, hour$$

(a) Fraction of the day that the phone will be in use

$$\rho = \frac{\lambda}{\mu} = \frac{7.5}{15} = 0.5$$

(b) The expected number of units in the queue

$$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)} = \frac{7.5^2}{15(15 - 7.5)}$$

$L_q = 0.5$ (Units) person

(c) Expected waiting time in the queue

$$W_q = \frac{L_q}{\lambda} = \frac{0.5}{7.5} = 0.066 \text{ Hours}$$

(d) Expected number of unit in the system

$$L = L_q + \frac{\lambda}{\mu} = 0.5 + 0.5 = 1 \text{ Person}$$

(e) Expected waiting time in the system

$$W = W_q + \frac{1}{\mu} = 0.066 + \frac{1}{15} = 0.133$$

(f) Expected number of units in the queue that form from time to time

$$D = \frac{\mu}{\mu - \lambda} = \frac{15}{15 - 7.5} = \frac{15}{7.5} = 2 \text{ Person}$$

(g) Probability that an arrival will have to wait in the system:-

$$P_{ro} = 1 - P_o$$

$$P_o = 1 - \frac{\lambda}{\mu}$$

$$= 1 - \left(1 - \frac{\lambda}{\mu}\right)$$

$$P_{ro} = \frac{\lambda}{\mu} = \frac{7.5}{15} = 0.5$$

(h) The Probability that exactly zero waits in the system:-

$$P_o = 1 - \frac{\lambda}{\mu}$$

$$= 1 - 0.5 = 0.5$$

(i) The probability that exactly 3 units in the system:-

$$P_n = P_o - \left(\frac{\lambda}{\mu}\right)^n \qquad n = 3$$

$$P_3 = 0.5(0.5)^3 = 0.0625$$

(j) Probability that an arrival will not have to wait for service:-

$$P_o = 1 - \frac{\lambda}{\mu}$$

$$= 0.5$$

(k) Probability that 3 or more units in the system:-

$$P_{n\ or\ more} = \left(\frac{\lambda}{\mu}\right)^n \qquad n = 3$$

$$P_{n\ or\ more} = 0.5^3 = 0.125$$

(l) Probability that an arrival will have to wait more than 6mins in queue for service

$$P_{ro} = \left(\frac{\lambda}{\mu}\right) e^{(\lambda-\mu)\omega}$$

$$\omega = 6\min = \frac{6}{60} hrs$$

$$P_{ro} = 0.5\, e^{(7.5-15)\frac{6}{60}}$$

$$P_{ro} = 0.236$$

(m) Probability that more than 5 units in the system

$$P_{ro} = \left(\frac{\lambda}{\mu}\right)^n \qquad n = 6$$

$$P_{ro} = 0.5^6 = 0.015$$

(n) Probability that an arrival will directly enter for service

P_o= 0.5

11.4 Summary

Queuing models provide valuable insights into system performance and behaviour, aiding in the design, analysis, and optimization of various processes and systems. By understanding queuing theory and applying appropriate models, organizations can improve efficiency, customer satisfaction, and resource utilization across a wide range of applications.

11.5 Keywords

- Queues
- Queuing theory

- Arrival processes
- Service processes
- Queue disciplines

11.6 Self Assessment Questions

1. What is the primary purpose of queuing models?
2. Define the arrival process in queuing theory.
3. Explain the significance of service processes in queuing models.
4. Differentiate between the M/M/1 and M/M/c queuing models.
5. What is Little's Law, and how is it applied in queuing theory?
6. What role does queuing theory play in optimizing resource allocation?
7. How does dynamic routing contribute to managing queues in computer networks?
8. Describe one real-world application where queuing models are used.
9. What are the main performance metrics used to evaluate queuing systems?
10. How can simulation software be beneficial in analyzing complex queuing systems?

11.7 Case Study

An international airport faces challenges with long wait times and congestion at its security screening checkpoints, particularly during peak travel hours. Passengers often experience frustration and delays, impacting their overall travel experience. The airport authority aims to optimize the security screening process to improve efficiency and enhance passenger satisfaction.

1. Minimize average passenger wait time at security checkpoints.
2. Maximize throughput of passengers through security screening.
3. Optimize resource allocation, including staffing and equipment usage.
4. Maintain security standards while enhancing the passenger experience.

11.8 References

1. Kleinrock, L. (1975). Queuing Systems, Volume 1: Theory. Wiley-Interscience.
2. Gross, D., Shortle, J. F., Thompson, J. M., & Harris, C. M. (2008). Fundamentals of Queuing Theory (4th ed.). Wiley-Interscience.

UNIT - 12

Advanced Queuing Models

Learning objectives

- Understand the basic principles of queuing theory, including the concepts of queues, servers, arrival processes, and service processes.
- Differentiate between various types of queuing systems (e.g., single-server vs. multi-server, finite vs. infinite capacity).
- Understand the assumptions, limitations, and appropriate applications of each model.

Structure

12.1 Queuing Networks

M/G/1 Queuing Mechanisms There is a general distribution for service times. The remaining M/M/1 presumptions are kept ramifications We are no longer able to depend on service times' memory less feature. The number of customers at time t must be shown in each state if the state transition diagram approach is to be used, and A(t) indicates the amount of time the customer has been served by the server up until that point. Could you explain why M/M/ (N(t), A(t)), N(t) doesn't require

Queuing networks are queuing systems composed of interconnected queues, where entities move between queues based on predefined routing rules. These networks are used to model complex systems with multiple stages of service, such as manufacturing processes, computer networks, telecommunications systems, and service facilities like hospitals and banks. Here's an overview of Queuing networks:

Components of Queuing Networks:

1. **Nodes (Queues):**
    - Each node represents a queue where entities (such as customers, packets, or jobs) wait for service.
    - Queues may have different characteristics, such as arrival rates, service rates, and queue capacities.
2. **Servers:**
    - Servers are responsible for serving entities in the queues.
    - Each queue may have one or more servers dedicated to providing service.
3. **Routing Rules:**
    - Routing rules determine how entities move between queues in the network.
    - Routing may be deterministic (fixed paths) or stochastic (probabilistic).
4. **Network Topology:**
    - The arrangement of queues and connections between them forms the network topology.
    - Queuing networks may have simple linear topologies or complex interconnected structures.

Types of Queuing Networks:

1. **Jackson Networks:**
    - Jackson networks are a class of Queuing networks with Poisson arrivals, exponential service times, and deterministic routing.
    - Nodes in Jackson networks are interconnected, and entities move between nodes according to predetermined routes.
2. **BCMP Networks:**
    - BCMP (Baskett, Chandy, Muntz, Palacios) networks are a generalization of Jackson networks that allow for more complex routing and service time distributions.
    - BCMP networks can model non-exponential service times and more flexible routing options.
3. **Closed and Open Networks:**
    - Closed Queuing networks have a fixed number of entities circulating within the system, while open networks allow entities to enter and exit the system over time.

Performance Measures:

1. **Throughput:**
    - The rate at which entities are processed by the network.
2. **Utilization:**
    - The fraction of time servers are busy serving entities.
3. **Queue Length:**
    - The average number of entities waiting in each queue.
4. **Response Time:**
    - The mean duration of an entity within the system, encompassing waiting and servicing times.
5. **Blocking Probability:**
    - The probability that an entity cannot enter a queue due to queue capacity constraints.

Applications:

- Telecommunications Networks: Modelling data packet routing in computer networks, internet traffic management.
- Manufacturing Systems: Analyzing production lines, supply chain logistics, and inventory management.
- Service Operations: Optimizing service facilities such as hospitals, call centers, and banks.

Queuing networks provide a powerful framework for analyzing and optimizing complex systems with multiple stages of service. By studying the behaviour of entities as they move through interconnected queues, decision-makers can gain insights into system performance, resource allocation, and bottlenecks, leading to more efficient operations and improved customer satisfaction.

12.2 Multi-Server Systems

Multi-server systems are queuing systems where multiple servers are available to serve customers or entities. These systems are widely used in various applications, including telecommunications, computer networks, service operations, and manufacturing. Here's an overview of multi-server systems:

Characteristics:

1. **Number of Servers:**
    - Multi-server systems have more than one server available to provide service to customers.
    - The number of servers (C) can vary based on system requirements and capacity.
2. **Service Process:**
    - Servers in multi-server systems may operate independently or collaboratively to serve customers.
    - Service times may follow various distributions, such as exponential, deterministic, or general distributions.
3. **Queuing Discipline**:
    - Each server may have its queue or share a common queue with other servers.

- Queuing disciplines describe the order in which clients are served, such as first-come-first-served, priority-based, or shortest-job-next.

4. **Routing Rules:**
    - Multi-server systems may have fixed or dynamic routing rules that determine how customers are assigned to servers.
    - Routing rules can optimize system performance by balancing the workload across servers.

Performance Measures:

1. **Utilization (ρ):**
    - The average fraction of time servers are busy serving customers.
    - Calculated as $\rho = \lambda C \cdot$, where λ is the arrival rate and μ is the service rate per server.
2. **Queue Length:**
    - The average number of customers waiting in the queue.
    - Queue length depends on the number of servers, arrival rate, service rate, and queue capacity.
3. **Waiting Time:**
    - The typical amount of time patrons must wait in line before receiving service.
    - Waiting time is affected by system load, arrival rate, service rate, and Queuing discipline.
4. **Response Time:**
    - The average time clienteles spend in the scheme, including service and waiting time.
    - Response time is influenced by Queuing discipline, service time distribution, and system load.

Strategies for Optimization:

1. **Load Balancing:**
    - Distributing customer arrivals evenly across servers to balance the workload and reduce Queuing delays.

2. **Dynamic Routing:**
   - Dynamically assigning customers to servers based on current server loads and customer characteristics.
3. **Server Synchronization:**
   - Synchronizing the service processes of multiple servers to optimize resource utilization and minimize idle time.
4. **Queue Management:**
   - Implementing effective Queuing disciplines and capacity planning strategies to minimize queue lengths and waiting times.

Applications:

- Telecommunications Networks: Call centers, data centers, and internet service providers.
- Service Operations: Hospitals, banks, airports, and customer service centers.
- Manufacturing Systems: Assembly lines, production facilities, and supply chain logistics.

Multi-server systems play a crucial role in various industries, providing efficient and reliable service to customers while optimizing resource utilization and operational efficiency. By understanding the characteristics and performance measures of multi-server systems, organizations can design and manage queuing systems that meet their service level objectives and customer demands.

12.3 Queuing Theory in Practice

Queuing theory is widely applied in practice across various industries and domains to analyze, design, and optimize systems involving waiting lines or queues. Here are some common applications of Queuing theory in practice:

1. Telecommunications Networks:

- Call Centers: Queuing theory helps optimize call center operations by analyzing call arrival patterns, staffing requirements, and service levels.
- Data Networks: It is used to design and manage data networks, including routers, switches, and packet-switched networks, to minimize packet loss and latency.

2. Service Operations:

- Healthcare: Queuing theory is applied in hospitals and clinics to optimize patient flow, reduce waiting times, and allocate resources efficiently.
- Retail: It helps retailers manage checkout lines, staffing levels, and customer service operations to enhance customer satisfaction.
- Transportation: Queuing theory is used in transportation systems such as airports, train stations, and bus terminals to manage passenger flows and minimize congestion.

3. Manufacturing and Supply Chain:

- Production Systems: It is applied in manufacturing facilities to optimize production lines, inventory management, and material handling processes.
- Supply Chain Logistics: Queuing theory helps optimize distribution centers, warehouses, and logistics networks to streamline order full filament and reduce lead times.

4. Computer Systems and Networks:

- Computer Networks: It is used to analyze and optimize network performance, including packet-switched networks, cloud computing systems, and content delivery networks.
- Operating Systems: Queuing theory helps optimize resource allocation and scheduling algorithms in operating systems to improve system throughput and response times.

5. Retail and Customer Service:

- Supermarkets: It helps supermarkets and retail stores optimize checkout processes, queue management, and staffing levels to reduce waiting times and improve customer satisfaction.
- Banking: Queuing theory is applied in banks and financial institutions to manage customer queues, service times, and teller staffing levels to enhance customer service.

6. Traffic Engineering:

- Road Traffic: It is used to analyze traffic flows, congestion patterns, and traffic signal timings to optimize traffic management and reduce travel times.
- Public Transportation: Queuing theory helps optimize bus routes, train schedules, and passenger boarding processes to improve public transportation efficiency.

7. Emergency Services:

- Emergency Departments: It is applied in hospitals' emergency departments to manage patient triage, resource allocation, and treatment processes during peak demand periods.

- Emergency Call Centres: Queuing theory helps optimize emergency call routing, dispatcher allocation, and response times in emergency call centres.

Queuing theory provides valuable insights and tools for analyzing and improving the performance of systems involving waiting lines or queues. By applying Queuing theory principles and techniques, organizations can enhance efficiency, reduce costs, and improve customer service in various real-world applications.

12.4 Summary

In summary, Advanced Queuing Models equip individuals with the analytical and practical skills needed to understand, design, and optimize complex queuing systems in various real-world applications, driving efficiency and performance improvements.

12.5 Keywords

- Arrival Rate (λ)
- Service Rate (μ)
- Queue Discipline
- Traffic Intensity (ρ)
- Little's Law

12.6 Self-Assessment questions

1. What is Little's Law in queuing theory?
2. How does an M/M/1 queue differ from an M/M/c queue?
3. What role does the Poisson process play in queuing models?
4. Explain the concept of traffic intensity (ρ) and its significance.
5. What is the primary difference between M/G/1 and G/G/1 queuing models?
6. Describe a scenario where a queuing network might be used.
7. What is meant by the term "queue discipline"? Give an example.
8. How can simulation be used in analyzing complex queuing systems?
9. What are heavy traffic approximations, and when are they used?

10. What optimization techniques can be applied to improve the performance of a queuing system?

12.7 Case Study

A major metropolitan bank faces significant customer service challenges during peak hours. Customers experience long wait times, leading to dissatisfaction and potential loss of clientele. The bank has multiple service counters but struggles with effectively managing customer flow and service efficiency.

1. Reduce average customer wait time.
2. Improve customer satisfaction.
3. Optimize resource allocation (e.g., the number of tellers).
4. Implement a system for better queue management during peak hours.

12.8 References

1. Gross, D., Shortle, J. F., Thompson, J. M., & Harris, C. M. (2008). Fundamentals of Queuing Theory (4th ed.). Wiley - Interscience.
2. Kleinrock, L. (1975). Queuing Systems, Volume 1: Theory. Wiley-Interscience.