Master of Science in Mathematics
(M.Sc. Mathematics)

Mathematical Programming

(OMSMCO203T24)

Self-Learning Material
(SEM II)

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

Jaipur National University

Course Code: OMSMCO203T24
Mathematical Programming

TABLE OF CONTENTS

EXPERT COMMITTEE

1. **Dr. Vikas Gupta**
Dean
Department of Mathematics
LNMIIT, Jaipur

2. Dr. Nawal Kishor Jangid
Department of Mathematics
SKIT, Jaipur

COURSE COORDINATOR
Dr. Sanju Jangid
Dept. of Basic Sciences
JNU, Jaipur

UNIT PREPARATION

Unit Writers	**Assisting & Proof Reading**	**Unit Editor**
Mr. Praveen Kumar Dept. of Basic Sciences JNU, Jaipur Unit 1 - 5	Prof. Hoshiyar Singh Dept. of Basic Sciences JNU, Jaipur	Mr. Rakesh Mishra Dept. of Basic Sciences JNU, Jaipur
Dr. Yogesh Khandelwal Dept. of Basic Sciences JNU, Jaipur Unit 6 - 8		
Mr. Mohammed Asif Dept. of Basic Sciences JNU, Jaipur Unit 9 - 11		

Secretarial Assistance:
Mr. Mukesh Sharma

COURSE INTRODUCTION

Mathematical Programming is a discipline that uses advanced analytical methods to help make better decisions. It applies techniques from mathematical modeling, statistics, and optimization to solve complex problems and improve decision-making in various industries such as logistics, finance, manufacturing, healthcare, and many others. It combines scientific methods with practical applications to improve processes, resource allocation, and strategic planning.

The course is of four credits and divided into 11 units. Each unit is divided into sub topics. There are sections and subsections in each unit. Each unit starts with a statement of objectives that outlines the goals we hope you will accomplish.

Course Outcomes:

At the completion of the course, a student will be able to:

1. Recall the concept of linear programming problems.
2. Explain PERT and CPM methods.
3. Apply the basic concepts of Game theory.
4. Analyze the significance of the notions of Duality.
5. Evaluate dynamic programming.
6. Develop the applications of different methods.

Acknowledgements:

The content we have utilized is solely educational in nature. The copyright proprietors of the materials reproduced in this book have been tracked down as much as possible. The editors apologize for any violation that may have happened, and they will be happy to rectify any such material in later versions of this book.

UNIT -1

Non-Linear Programming Problem

Learning Objectives

- Gaining an understanding of nonlinear optimization and seeing its importance in solving problems in real life where there are nonlinear interactions between variables.
- Making educated selections requires an understanding of how modifications to the problem parameters effect the best solution and the interpretation of sensitivity analysis findings.

Structure

1.1 Introduction

The Linear Programming Problem (LPP) which can be review as to

$$Maximize \;\; Z = \sum_{j=1}^{n} c_j x_j$$

$$subject\ to \sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad for \quad i = 1, 2, ..., m$$

$$and \;\; x_j \geq 0 \qquad for \quad j = 1, 2, ..., m$$

The word 'non-linear programming' generally refers to the trouble in which the OF(OF) (1) becomes nonlinear, or one or extra of the constraint inequality (2) have nonlinear or both.

Eg. 1.1.1

Examine the following issue:

$$Maximize(\ Minimize\ )\ Z = x_1^2 + x_2^2 + x_3^3$$

$$subject\ to \qquad x_1 + x_2 + x_3 = 4 \quad and \quad x_1\ , x_2\ , x_3 \geqslant 0$$

1.2 Graphical Solution

The ideal solution in a LPP was typically found at one of the extremes of the convex region that the problem's constraints and OFcreated. However, it is not required to solve the NLPP at the outermost points of its viable domain.

Eg. 1.2.1

Solve the following problem graphically.

$$Maximize \quad Z = 2x_1 + 3x_2 \qquad (1)$$

$$subject\ \ to \quad x_1^2 + x_2^2 \leqslant 20, \qquad (2)$$

$$x_1 x_2 \leqslant 8 \quad and \quad x_1,\ x_2 \geqslant 0 \qquad (3)$$

Solution:

In this problem OF is linear and the constraints are non-linear.

${x_1}^2 + {x_2}^2 = 20$ symbolizes circle and$x_1 x_2 = 0$depicts hyperbola. Asymptotes are represented by $X - axis$ and $Y - axis$.

Cracking eq^n (2) and (3), we get$x_1 = -2, -4, 2, 4$. But $x_1 = -2, -4$ are impossible$(x_1 \geq 0)$

Take $x_1 = 2$ and 4 in eq^n (2) and (3), then we get $x_2 = 4$ and 2 respectively. So, the points are (2,4) or (4,2). Shaded non-convex region of OABCD (Figure 1.2.1) is called the feasible region. Now, we maximize the OFi.e $2x_1+3x_2 = K$ lines for different constant values of K and stop the process when a line touches the extreme boundary point of the feasible region for some value of K.

At (2,4), K = 16 which touches the extreme boundary point. We have boundary point of like (0,0), (0,4), (2, 4), (4,2), (4,0). Where the value of Z is maximum at point (2,4).

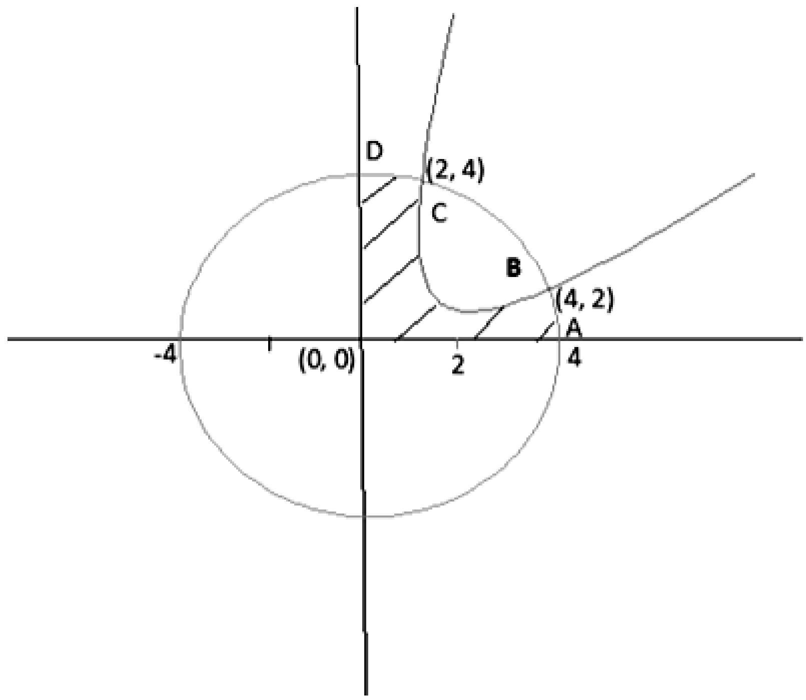


Figure 1.2.1 Graph to maximize Z

Max. Z=16.

1.3 Single-variable optimization

Single-variable optimization, also known as univariate optimization, involves locating the maximum (minimum)rate of a function of a single variable. This type of optimization is fundamental in calculus and mathematical analysis, and it is widely used in various fields such as economics, engineering, and operations research.

Key Concepts:

1. **OF**: The function f(x) that you want to optimize (maximize or minimize).

2. **Domain**: The set of all possible values that the variable x can take. It is important to define the domain because the optimal solution must lie within this range.

3. **Critical Points**: Points where the derivative of the function f'(x) is zero/undefined. These points are potential candidate for local maxima/minima, and saddle points.

4. **Global vs. Local Optima:**

- Local Optima: A point x = a is a local maximum (or minimum) if f(a) is greater (or less) than f(x) for all x in some small nbd around a.
- Global Optima: A point x=b is a global maximum/minimum if f(b) is greater (or less) than f(x) for all x in the entire domain.

5. **First Derivative Test**: If f'(x) changes sign around a critical point, it indicates a local maximum/minimum.

- If f'(x)modify from “+ve’ to “-ve’, it’s a local maximum.
- If f'(x)modify from “-ve’ to “+ve’, it’s a local minimum.

6. **Second Derivative Test**: If f''(x) is “+ve at a critical point, it indicates a local minimum; if “-ve, a local maximum. If f''(x)=0, the test is inconclusive.

Steps in Single-Variable Optimization:

1. Define the OF: Clearly specify the function f(x) to be optimized.
2. Determine the Domain: Identify the range of x over which optimization is to be performed.
3. Find Critical Points:
   - calculate the f'(x).
   - Crack f'(x) = 0 to get critical points.
   - Check where f'(x) is indeterminate, if applicable.
4. Examine Critical Points:
   - Utilize the 1st/2nd derivative test to classify each critical point.
   - Evaluate the OFat the critical points.
5. Evaluate Endpoints (if the domain is closed and bounded): Check the values of f(x) at the boundaries of the domain.
6. Compare Values: Identify the maximum or minimum values among the critical points and endpoints.

Eg.1.3.1

Suppose we want to minimize the function $f(x) = x^2 - 4x + 4$.

Solution:

1. I Derivation: $f'(x) = 2x - 4$.
2. Find Critical Points : Solve $f'(x) = 0$.
   - $2x - 4 = 0 \Rightarrow x = 2$.
3. Second Derivative: $f''(x) = 2$.
   - Since $f''(x) > 0$, $x = 2$is a local minimum.
4. Evaluate the Function: $f(2) = 2^2 - 4(2) + 4 = 0$.

Thus, the function$f(x) = x^2 - 4x + 4$has a minimum value of 0 at $x = 2$.

1.3.1 Bisection Method

The bisection method is a easy and robust numerical technique for discovering roots of a continuous function $f(x)$. It is working by repetitivelytapering down an interval $[a, b]$ that contains a root.

Steps of the Bisection Method:

1. **Initial Interval**: Start with an interval [a, b] such that f(a) and f(b) have reverse signs (f(a).f(b)<0), indicating a root exists between a and b (by the Intermediate Value Theorem).
2. **Midpoint Calculation**: Calculate the middle point c={a + b}/{2} .
3. **Evaluate the Function**: Calculate f(c).
   - If f(c)=0, c is the root.
   - If f(a)*f(c) < 0, the root be positioned in [a, c], so b=c .
   - If f(b)*f(c) < 0, the root be positionedin [c, b], so a=c .
4. **Repeat**: Continue this process until the interval [a, b] is sufficiently small, i.e., |b - a| is less than a pre-defined tolerance level.

Eg. 1.3.2

Let's use both methods to find a root of the function $f(x) = x^2 - 2$.

Solution:

1. **Initial Interval**: Choose [a, b] = [1, 2].
   - f(1) = 1 - 2 = -1
   - f(2) = 4 - 2 = 2
   - f(1).f(2) < 0 , so a root exists between 1 and 2.
2. **Iteration**:
   - c = {1 + 2}/{2} = 1.5
   - $f(1.5) = 1.5^2 - 2 = 0.25$
   - Since f(1)*f(1.5) < 0 , set b = 1.5 .
3. Continue this process until the interval is sufficiently small. The root will converge to $\sqrt{2} \approx 1.414$.

1.3.2 Newton's Method

Newton's method is an efficient iterative method for discoveringconsecutively better estimate to the roots of a real-valued function.

Steps of Newton's Method:

1. **Initial Guess**: Start with an initial guess x_0 .

2. **Iterative Formula**: Compute the next approximation using the formula:

$$x\{n+1\} = xn - [\{f(xn)\}/\{f'(xn)\}]$$

here $f'(x_n)$ is the derivative of $f(x)$ at x_n.

3. **Convergence Check**: Continue the iterations until $|x_{\{n+1\}}-x_n|$ is less than a pre-defined tolerance level.

Eg. 1.3.3

Let's use both methods to find a root of the function $f(x) = x^2 - 2$.

Solution:

1. **Initial Guess**: Choose $x0 = 1.5$.

2. **Iteration**:
   - $f(x_0) = 1.5^2 - 2 = 0.25$
   - $f'(x_0) = 2*(1.5) = 3$
   - $x_1 = 1.5 - \{0.25\}/\{3\} \approx 1.4167$

3. Continue this process until the difference between successive approximations is sufficiently small. The root will converge to $\sqrt{2} \approx 1.414$.

Note

- Bisection Method: Slower but more robust and guaranteed to converge if the initial interval is chosen correctly.

- Newton's Method: Faster but requires a good initial guess and can fail to converge if the function or its derivative is not well-behaved.

1.4 Multivariable Optimization without Constraints

A non-linear multi-variable optimization without constraints:

$$Maximize \quad f(x_1, x_2, ..., x_n)$$

$$with \quad x_1, x_2, ..., x_n \geqslant 0$$

Local Maxima and Minima

Local Maximum: A point x in the domain of a function f(x) is a local maximum if there exists a neighborhood approximately x∗ such that $f(x*) \geq f(x)$ for every x in that neighborhood.

Local Minimum: A point x∗ in the domain of a function f(x) is a local minimum if there exist a neighborhood around x∗ such that $f(x*) \leq f(x)$ for all x in that neighborhood.

Global Maxima and Minima

Global Maximum: A point x∗ in the domain of a function f(x) is a global maximum if $f(x*) \geq f(x)$ for all x in the entire domain.

Global Minimum: A point x∗ in the domain of a function f(x) is a global minimum if $f(x*) \leq f(x)$ for every x in the entire domain.

Unconstrained optimization

Unconstrained optimization engages finding the optimum of a function without any restrictions or constraints on the variables. This is a fundamental problem in various fields such as machine learning, economics, and engineering. The goal is to find a point x∗ that optimizes the OFf(x).

We have to optimize

$$f(x_1, x_2, \ldots, x_n)$$

In unconstrained type of function we determine the extreme points

$$\frac{\partial f}{\partial x_1} = 0$$
$$\frac{\partial f}{\partial x_2} = 0$$
$$\vdots$$
$$\frac{\partial f}{\partial x_n} = 0$$

For one value

$$\frac{\partial^2 f}{\partial x^2} > 0 \qquad \text{Then } f \text{ is minimum.}$$
$$\frac{\partial^2 f}{\partial x^2} < 0 \qquad \text{Then } f \text{ is maximum.}$$
$$\frac{\partial^2 f}{\partial x^2} = 0 \qquad \text{Then further investigation needed.}$$

For two variables

$rt - s^2 > 0$ Then the function is minimum.

$rt - s^2 < 0$ Then the function is maximum.

$rt - s^2 = 0$ Further investigation needed.

Where $r = \frac{\partial^2 f}{\partial {x_1}^2}, s = \frac{\partial^2 f}{\partial x_1 x_2}, t = \frac{\partial^2 f}{\partial {x_2}^2}$

For ‘n’ variables

Hessian Matrix

$$\begin{bmatrix} \frac{\partial^2 f}{\partial {x_1}^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_1 x_2} & \frac{\partial^2 f}{\partial {x_2}^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 f}{\partial x_1 x_n} & \frac{\partial^2 f}{\partial x_2 x_n} & \cdots & \frac{\partial^2 f}{\partial {x_n}^2} \end{bmatrix}$$

$| H | > 0$ at p_1, f is attains minimum at p_1.

$| H | < 0$ at p_1, f is attains maximum at p_1.

Convex Functions

A function f: $R^n \rightarrow R$ is convex if, for all $x, y \in Rn$ and $\lambda \epsilon [0, 1]$,

$$f(\lambda x_1 + (1-\lambda)x_2) \leqslant \lambda f(x_1) + (1-\lambda)f(x_2) \quad where \ \ 0 \leqslant \lambda \leqslant 1$$

Properties of Convex Functions

- Local Minima are Global Minima: If a convex function has a local minimum, it is also a global minimum.
- First-order Condition: A differentiable function f is convex if its gradient ∇f satisfies $f(y) \geq f(x) + \nabla f(x)T(y - x) \, for \, all \, x, y.$
- Second-order Condition: A twice time-differentiable function f is convex ifits Hessian matrix $Hf(x)$ is positive semidefinite for all x.

Eg. 1.4.1

- Linear functions $f(x) = a^T x + b$
- Quadratic functions $f(x) = x^T A x + b^T x + c$ where A is positive semi-definite
- Exponential functions $f(x) = e^x$

Concave Functions

A function $g:R^n \rightarrow R$ is concave if, for all $x, y \in Rn$ and $\lambda \epsilon [0, 1]$,

$$f(\lambda x_1 + (1-\lambda)x_2) \geqslant \lambda f(x_1) + (1-\lambda) f(x_2) \quad where \;\; 0 \leqslant \lambda \leqslant 1$$

Properties of Concave Functions

- Local Maxima are Global Maxima: If a concave function has a local maximum, it is also a global maximum.
- First-order Condition: A differentiable function g is concave ifits gradient ∇g satisfy$g(y) \leq g(x) + \nabla g(x) T(y - x)\ for\ all\ x, y.$
- Second-order Condition: A twice-differentiable function g is concave if and only if its Hessian matrix $H_g(x)$ is negative semi-definite for all x.

Eg. 1.4.2

- Linear functions $g(x) = a^T x + b$
- Quadratic functions $g(x) = -x^T A\ x + bTx + c$ where A is positive semi-definite
- Logarithmic functions g(x) = log(x)

1.5 Multivariable Optimization with Constraints

General NLPP

Let Q be a real-valued function of n-variables specifyby:

(a) $Q = f(x1, x2, \ldots, xn)$.

Let $(b1, b2, \ldots, bm)$ be a set of constraints:

$$(b) \quad g_1\ (x_1, x_2, \ldots, x_n)\ [\ \leqslant\ or\ \geqslant\ or\ =\ ]\ b_1$$

$$g_2\ (x_1, x_2, \ldots, x_n)\ [\ \leqslant\ or\ \geqslant\ or\ =\ ]\ b_2$$

$$g_3\ (x_1, x_2, \ldots, x_n)\ [\ \leqslant\ or\ \geqslant\ or\ =\ ]\ b_3$$

$$g_m\ (x_1, x_2, \ldots, x_n)\ [\ \leqslant\ or\ \geqslant\ or\ =\ ]\ b_m$$

Where $g1$ are real-valued functions of n variables, $x_1, x_2, \ldots, x_n$.

lastly, let (c) $x_j > 0$ where j = 1, 2, ..., n. →"+ve" Non-negativity constraint.

1.5.1 Lagrange Multipliers

In this instance, the continuous function optimization problem was covered. The NLPP (NLPP) is made up of equality side constraints and certain differentiable OF; Lagrange multipliers can be used to optimize the system. The sensitivity of the OF's optimal value to changes in the problem's specified constraints is measured by a Lagrange multiplier. Examine the issue of figuring out the worldwide optimum of

$$Z = f(x_1, x_2, ..., x_n)$$

s.t. the

$$g_i\ (x_1, x_2, ..., x_n) = b_i\ ,\ i = 1, 2, ..., m.$$

Let us first formulate the Lagrange function L defined by:

$$L(x_1, x_2, ..., x_n; \lambda_1, \lambda_2, ..., \lambda_m) = f(x_1, x_2, ...x_n) + \lambda_1\ g_1(x_1, x_2, ..., x_n) + \lambda_2\ g_2(x_1, x_2, ..., x_n) + ... + \lambda_m g_m(x_1, x_2, ..., x_n)$$

where $\lambda_1, \lambda_2, \ldots\ldots, \lambda_m$ are Lagrange Multipliers.

For the stationary points

$$\frac{\partial L}{\partial x_j} = 0 \quad , \quad \frac{\partial L}{\partial \lambda_i} = 0 \quad \forall\ j = 1(1)n \quad \forall\ i = 1(1)m$$

cracking the ^{eqns} to get stationary points.

Eg. 1.5.1

A rectangular box with an opening on top must hold 32 cubic meters of space. Determine the box's dimensions that will use the least amount of material to build.

Solution:

Let $x1, x2, x3$ be the faces of the rectangular face with $x1, x3$in the faces of the bottom faces and s be its surface, then

$$s = 2x_1x_2 + 2x_2x_3 + x_1x_3$$

subject to

$x_1x_2x_3$=32 and Let x_1, x_2, x_3> 0.

Form the Lagrangian function

$$L(x_1, x_2, x_3, \lambda) = 2x_1x_2 + 2x_2x_3 + x_1x_3 + \lambda(x_1x_2x_3 - 32)$$

The stationary points are the solutions of the followings:

$$\frac{\partial L}{\partial x_1} = 2x_2 + x_3 + \lambda x_2 x_3 = 0 \qquad (4)$$

$$\frac{\partial L}{\partial x_2} = 2x_1 + 2x_3 + \lambda x_1 x_3 = 0 \qquad (5)$$

$$\frac{\partial L}{\partial x_3} = 2x_2 + x_1 + \lambda x_1 x_2 = 0 \qquad (6)$$

$$\frac{\partial L}{\partial \lambda} = x_1 x_2 x_3 - 32 = 0 \qquad (7)$$

from eq^n (1) and (2) we obtain

$$x1 = 2x2$$

and by eq^ns (1) & (3) we obtain $x2 = x3$

put these value in eq^n (4) we get

$$x_1 = 4, \ x_2 = 2, \ x_3 = 4$$

$Min.S = 48$.

1.5.2 Kuhn-Tucker Conditions (KT Conditions)

Kuhn-Tucker Necessary Conditions

Max. f(X), X = (x_1, x_2,...,x_n) s.t. constraints

$g_i(X) \leqslant b_i$, $i = 1, 2, ..., m,$

Counting the "+ve" constraints X≥0, the necessary conditions for a local maxima at $\bar{X}$ are

$$(i) \ \frac{\partial L(\bar{X}, \bar{\lambda}, \bar{s})}{\partial x_j} = 0, \ j = 1, 2, ..., n,$$

$$(ii) \ \bar{\lambda}_i \, [\, g_i(\bar{X}) - b_i \,] = 0,$$

$$(iii) \ g_i(\bar{X}) \leqslant b_i \, , \quad (iv) \ \bar{\lambda}_i \geqslant 0 \, , \ i = 1, 2, ..., m.$$

KT Sufficient Conditions

The KT circumstances which are necessary circumstances are also enough if f(x) is concave and the feasible space is convex, i.e. if $f(x)$ is strictly concave and $gi(x), i = 1,2, \ldots, m$ are convex.

Problem 1.5.2

$Max.Z = 10x_1 + 4x_2 - 2x_1^2 - 3x_2^2,$

subject to

$$2x_1 + x_2 \leq 5$$

$$x_1, x_2 \geq 0$$

Solution:

We have,

$f(X) = 10x_1 + 4x_2 - 2x_1^2 - x_2^2$

$h(X) = 2x_1 + x_2 - 5$

The Kuhn-Tucker conditions are

$$\frac{\partial f}{\partial x_1} - \lambda \frac{\partial h}{\partial x_1} = 0$$

$$\frac{\partial f}{\partial x_2} - \lambda \frac{\partial h}{\partial x_2} = 0$$

$\lambda h(X) = 0,$

$h(X) \leq 0, \quad \lambda \geq 0.$

Applying these conditions, we get

$10 - 4x_1 - 2\lambda = 0$ (i)

$4 - 2x_2 - \lambda = 0$ (ii)

$\lambda(2x_1 + x_2 - 5) = 0$ (iii)

$2x_1 + x_2 - 5 \leq 0$ (iv)

$\lambda \geq 0$ (v)

By (iii)

$$\lambda = 0 \; or \; 2x1 + x2 - 5 = 0$$

When $\lambda = 0$, the outcome of (i) and (ii) provide $x1 = 2.5$ and $x2 = 2$ which doesn't suit the eq^n (iv). Hence $\lambda = 0$ doesn't yield a feasible outcome.

When $2x1 + x2 - 5 = 0$ and $\lambda \neq 0$, the outcome of (i), (ii) and (iii) yields, x_1=11/6, x_2=4/3, λ=4/3, which suit all the necessary conditions.

It can be established that the OF is concave in X, as the constraint is convex in X. therefore these necessary circumstances are also the sufficient circumstances of maximize f(X).

Therefore the optimal outcome is

$$x_1^* = \frac{11}{6}, x_2^* = \frac{4}{3},$$

which gives

$$Z_{max} = \frac{91}{6}$$

1.6 Summary

Nonlinear Programming (NLP) involves optimizing a function s.t. non-linear constraints. Unlike linear programming, where relationships are linear, NLP deals with more complex relationships between variables. NLP finds applications in diverse fields such as engineering, economics, finance, and machine learning.

1.7 Keywords

- Non-Linear Programming Problem
- Single-Variable Optimization
- Lagrange multipliers
- Kuhn-Tucker

1.8 Self Assessment Questions

1. What distinguishes Non-Linear Programming (NLP) from Linear Programming (LP)?
2. What is the objective of an NLP?
3. Can you give an Eg. of a real-world problem that can be modeled as an NLP?
4. What are the key components of an NLP?
5. How do you differentiate between global and local optima in NLPs?
6. Name two solution methods used for solving NLPs.
7. What role do constraints play in NLPs?

1.9 Case study

1. A manufacturing company uses its limited resources to make two different kinds of items. How can production limits and demand changes be taken into account when applying non-linear programming to maximize profits?
2. An investor seeks to maximize profits and minimize risk in their investment portfolio optimization. In light of different investment possibilities, expected returns, and risk

considerations, how might non-linear programming be used to create an ideal portfolio?

1.10 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.

UNIT -2

Quadratic Programming Problem

Learning Objectives

- Acquiring the knowledge of expressing real-world issues as quadratic programming models and determining decision variables
- Recognizing quadratic programming specific instances, including convex and non-convex quadratic programs.
- Knowing what local and global optima in quadratic programming issues mean

Structure

2.1 Introduction

Quadratic programming deals with the NLPP of maximizing or minimizing) the quadratic OF s.t. a set of linear inequality constraints.

The general **QPP** can be specified as tracks:

$$Maximize\ Z = CX + \frac{1}{2}X^T QX$$

subject to

$$AX \leqslant B \quad and \quad X \geqslant 0$$

where

$$X = (x_1, x_2, ..., x_n)^T$$

$$C = (c_1, c_2, ..., c_n) \quad , \quad B = (b_1, b_2, ..., b_m)^T$$

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

$$Q = \begin{pmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \vdots & \vdots \\ q_{n1} & \cdots & q_{nn} \end{pmatrix}$$

2.2 Wolfe's modified simplex method

Let the QPP be:

$$Maximize \quad Z = f(X) = \sum_{j=1}^{n} c_j x_j + \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} c_{jk} x_j x_k$$

s.t. the constraints:

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \; x_j \geqslant 0 (i = 1, ..., m, j = 1, ..., n)$$

Here

$C_{jk} = C_{kj}$ for all j and k, $b_i > 0$ for all i = 1,2,...,m.

Also, assume that the quadratic form

$$\sum_{j=1}^{n} \sum_{k=1}^{n} c_{jk} x_j x_k$$

be -ve semi-definite.

after that, the Wolfe's iterative process may be delineate in the next steps:

Step 1. First, convert the inequality constraints into equation by introducing slack- variable q_i^2 in the ith constraint (i = 1,...,m) and the slack variable r_j^2 the jth non-negative constraint (j = 1, 2,...,N)

Step 2. Then, make the Lagrangian function

$$L(X, \mathbf{q}, \mathbf{r}, \lambda, \mu) = f(X) - \sum_{i=1}^{m} \lambda_i [\sum_{j=1}^{n} a_{ij} x_j - b_i + q_i^2] - \sum_{j=1}^{n} \mu_j [-x_j + r_j^2]$$

Where $X = (x_1, x_2, ..., x_n)$, $\mathbf{q} = (q_1^2, q_2^2, ..., q_m^2)$, $\mathbf{r} = (r_1^2, r_2^2, ..., r_n^2)$, and $\mu = (\mu_1, \mu_2, ..., \mu_n)$,

$\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)$,

Step 3. Now initiate the '+ve' artificial variable v_j, j = 1,2,...,n in the KT conditions

$$c_j + \sum_{k=1}^{n} c_{jk} x_k - \sum_{i=1}^{m} \lambda_i a_{ij} + \mu_j = 0$$

for j = 1, 2,...,n and to make an OF

$Z_v = v_1 + v_2 + ... + v_n$

Step 4. In this step, find the primary basic feasible outcome to the next LPP:

$$\sum_{k=1}^{n} c_{jk}x_k - \sum_{i=1}^{m} \lambda_i a_{ij} + \mu_j + v_j = -c_j \quad (j = 1, ...n)$$

$$\sum_{j=1}^{n} a_{ij}x_j + q_i^2 = b_i \quad (i = 1, ..., m)$$

$$v_j, \lambda_j \mu_j, x_j \geqslant 0 \quad (i = 1, ..., m; j = 1, ..., n)$$

and satisfying the complementary slackness condition:

$$\sum_{j=1}^{n} \mu_j x_j + \sum_{i=1}^{m} \lambda_i s_i = 0, \quad (where\ s_i = q_i^2)$$

Or

$$\lambda_i s_i = 0 \quad and \quad \mu_j x_j \; = \; 0 \quad (for\ i = 1, ..., m; j = 1, ..., n).$$

Step 5. Now, use the two-phase simplex approach as normal to get the best solution for the LP issue that was created in. Step 4: The solution needs to meet the complimentary slackness. condition mentioned previously.

Step 6. The optimum outcome thus obtained in Step 5 provide the optimum outcome of provide QPP also.

Eg. 2.2.1

$Maximize\ \ 2x_1 + x_2 - x_1{}^2$

subject to

$$2x_1 + 3x_2 \leq 6$$

$$2x_1 + x_2 \leq 4 \qquad and \quad x_1,\ x_2\ \geq\ 0$$

Solution:

Since the OF is convex and each constraint is convex, so the particular NLPP is a CNLPP.

Now

$$L(X, \overline{\lambda}) = (-2x_1 - x_2 + x_1{}^2) + \lambda_1(2x_1 + 3x_2 - 6) + \lambda_2(2x_1 + x_2 - 4)$$

Therefore the Khun-Tucker condition are

$$\therefore \ \frac{\partial L}{\partial x_j} \geq 0 \Rightarrow -2 + 2x_1 + 2\lambda_1 + 2\lambda_2 \geq 0$$

$$\Rightarrow -2 + 2x_1 + 2\lambda_1 + 2\lambda_2 - \mu_1 = 0$$

$$-1 + 3\lambda_1 + \lambda_2 \geq 0$$

$$\Rightarrow -1 + 3\lambda_1 + \lambda_2 - \mu_2 = 0$$

$$\therefore \quad \frac{\partial L}{\partial \lambda_i} \leq 0 \Rightarrow 2x_1 + 3x_2 - 6 \leq 0$$

$$\Rightarrow 2x_1 + 3x_2 - 6 + S_1 = 0$$

$$2x_1 + x_2 - 4 \leq 0$$

$$\Rightarrow 2x_1 + x_2 - 4 + S_2 = 0$$

$$\therefore \quad x_j \frac{\partial L}{\partial x_j} = 0 \Rightarrow x_1\mu_1 = 0, x_2\mu_2 = 0 \qquad (1)$$

$$\therefore \quad \lambda_i \frac{\partial L}{\partial \lambda} = 0 \Rightarrow \lambda_1 S_1 = 0, \lambda_2 S_2 = 0 \qquad (2)$$

The above system of equation can be written as

$$\begin{aligned} &2x_1 + 2\lambda_1 + 2\lambda_2 - \mu_1 = 2 \\ &3\lambda_1 + \lambda_2 - \mu_2 = 1 \\ &2x_1 + 3x_2 + S_1 = 6 \\ &2x_1 + x_2 + S_2 = 4 \qquad (3) \\ &x_1, x_2, \lambda_1, \lambda_2, S_1, S_2, \mu_1, \mu_2 \geq 0 \\ &x_1\mu_1 = 0, x_2\mu_2 = 0, \lambda_1 S_1 = 0, \lambda_2 S_2 = 0. \end{aligned}$$

This eq^n (3) is a LPP without an OF . To discover the outcome we can note down (3) as the next LPP.

$$\begin{aligned} &max.Z = -R_1 - R_2 \\ &\text{subject to} \\ &2x_1 + 2\lambda_1 + 2\lambda_2 - \mu_1 = 2 \\ &3\lambda_1 + \lambda_2 - \mu_2 = 1 \\ &2x_1 + 3x_2 + S_1 = 6 \\ &2x_1 + x_2 + S_2 = 4 \end{aligned}$$

Now crack this by the two phase simplex method. The conclusion of the phase (1) describes the feasible outcome of the problem

The optimal outcome of the QPP is

$$x_1 = \frac{2}{3}, x_2 = \frac{14}{9}, \lambda_1 = \frac{1}{3}, S_2 = \frac{10}{9}.$$

2.3 Summary

1. The QP OF is quadratic, implying that it includes both quadratic and perhaps linear elements related to the decision variables. Reducing or increasing this function is the aim.

2. Constraints: QP may be s.t. in equality or linear equality restrictions. These restrictions stand for limitations or specifications that the solution needs to meet.
3. Quadratic programming (QP) can be solved using a variety of techniques, such as numerical optimization algorithms designed specifically for QP problems and analytical techniques like the Lagrangian approach, active-set methods, and interior-point methods.
4. Convexity: A QP may be convex or non-convex depending on the limitations and the type of OF. Convex QPs have exceptional behavior, frequently exhibiting distinct global optima.

2.4 Keywords

- Quadratic Programming Problem
- Wolfe's modified simplex method
- Beale's Method

2.5 Self Assessment Questions

1. What distinguishes Quadratic Programming (QP) from linear programming?
2. What type of function characterizes the OF in QP?
3. Can you provide an Eg. of a real-world problem that can be modeled as a QP?
4. What are the possible types of constraints in a Quadratic Programming Problem?
5. Name one analytical method used to solve QP.
6. How does the convexity of a QP affect its optimization?
7. What are some common applications of Quadratic Programming?

2.6 Case study

1. A financial institution seeks to maximize returns and minimize risk in its investment portfolio optimization. In light of different investment possibilities, expected returns, and risk considerations, how can one use quadratic programming to create an ideal portfolio?
2. A machine learning researcher wants to train a support vector machine for a classification task. How can Quadratic Programming be applied to optimize the SVM's hyperplane parameters and kernel function to achieve the best classification performance?

2.8 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.

UNIT -3

Separable Programming

Learning Objective

- It is important for students to understand how to recognize separable structures in optimization issues.
- For separable programming problems, it is crucial to comprehend the optimality conditions.
- Algorithms and problem-solving strategies specific to separable programming should be taught to students.

Structure

3.1 Introduction

3.2 Separable Function

3.3 Separable Programming Problem

3.4 Piece-wise Linear Approximation of Non-linear Function

3.5 Reduction of separable programming problem to linear programming problem

3.6 Summary

3.7 Keywords

3.8 Self Assessment Questions

3.9 Case Study

3.10 References

3.1 Introduction

Such non-linear programming issues, where the objective function and constraints are separable, are dealt with in this programming. An ideal solution can be obtained by applying the standard simplex approach to reduce an NLPP to an LPP.

3.2 Separable Function

Definition 3.2.1

A function $f(x1, x2, ..., xn)$ is said to be separable if it can be expressed as the sum of n single valued functions $f_1(x_1)$, $f_2(x_2)$,..., $f_n(x_n)$, i.e.

$$f(x_1, x_2, ..., x_n) = f_1(x_1) + f_2(x_2) + ... + f_n(x_n)$$

Eg.:

$g(x1, x2, \ldots, xn{:}\ = \ c1x1 + \ldots + C2n$ where c's are constants, is a separable function.

$g(x_1, x_2, x_3) = x_1^3 + x_2^2 \sin(x_1) + \log(x_1^6)$

is not a separable function.

3.3 Separable Programming Problem (SPP)

A separable programming problem may be defined as a NLPP in which the OF may be written as a linear combination of multiple separate single variable functions, some of which or all of which are non-linear. Convex programming is a subset of non-linear programming that deals with the convex set of points and the challenge of minimizing a convex objective function (or maximizing a concave OF). We often accept non-linear limitations.

Separable Convex Programming Problem

Where the separate functions are all convex in a separable programming problem it may be described as a separable convex programming problem with separate OF . i.e

If $f(x) = f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n)$

where $f_1(x_1), f_2(x_2), \ldots, f_n(x_n)$ are all convex.

Eg. 3.3.1

$f(x) = 3x_1^3 + 2x_3^2 - x_1 - 3x_3$

So, let

$f_1(x_1) = 3x_1^3 - x_1$ and $f_2(x_2) = 2x_3^2 - 3x_3$

3.4 Piece-wise Linear Approximation of Non-linear Function

Considering the non-linear OF

$$Maximize z = \sum_{j=1}^{n} f_j(x_j)$$

s.t. constraints

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \ , \quad i = 1, 2, \ldots, m \quad and \quad x_j \geqslant 0 \quad ; for\ all \ \ j$$

In which $f_j(x_j)$= non-linear function in x_j. The point (a_k, b_k), k = 1, 2, ..., K are known as the breaking points amalgamation the linear segments approximating the f(x). If w_k denotes the +ve weight linked with the k^{th} breaking point as

$$\sum_{k=1}^{K} w_k = 1$$

$$w_{k'} = (a_{k'}, b_{k'}) \quad and \quad w_{k'+1} = (a_{k'+1}, b_{k'+1})$$

$$then \quad f(x) = \sum_{k=1}^{k} b_k w_k \quad , \quad where \quad x = \sum_{k=1}^{k} a_k w_k$$

s.t. the necessary additional constraints are

$$0 \leqslant w_1 \leqslant y_1$$

$$0 \leqslant w_2 \leqslant y_1 + y_2$$

$$\vdots$$

$$0 \leqslant w_{k-1} \leqslant y_{k-1} + y_{k-2}$$

$$0 \leqslant w_k \leqslant y_{k-1}$$

$$\sum_{k=1}^{k} w_k = 1, \quad \sum_{k=1}^{k-1} y_k = 1, \quad y_k = 0 \ or \ 1 \ for \ all \ k$$

Suppose $y_{k'} = 1$

Above all other $y_k = 0$

$$0 \leqslant w_{k'} \leqslant y_{k'}$$

$$0 \leqslant w_{k'+1} \leqslant y_{k'} = 1$$

The left over constraints should thus be $w_k \leq 0$.

All other $w_k = 0$ would therefore be as preferred.

3.5 Reduction of SPP to LPP

Let us take the SPP

$$Max.(or \ \ Min.) \ \ Z = \sum_{j=1}^{n} f_j(x_j)$$

s.t. the constraints

$$\sum_{j=1}^{n} g_{ij}(x_j) \leqslant b_i$$

$x_j \geq 0$ (i = 1, 2, ..., m; j = 1, 2, ..., n) where all or some $g_{ij}(x_j)$, $f_j(x_j)$ are nonlinear.

An equivalent mixed problem is shown below

$$Max.(or\ Min.)z = \sum_{j=1}^{n}\sum_{k=1}^{K_j} f_j(a_{jk})w_{jk},$$

s.t. the constraints

$$\sum_{j=1}^{n}\sum_{k=1}^{K_j} g_{ij}(a_{jk})w_{jk} \leqslant b_i \quad ,i = 1,2,...,m$$

$$0 \leqslant w_{j1} \leqslant y_{j1}$$

$$0 \leqslant w_{jk} \leqslant y_{j.k-1} + y_{jk} \quad , \quad k = 2,3,...,K_{j-1}$$

$$0 \leqslant w_{jk_j} \leqslant y_{j.k_{j-1}}$$

$$\sum_{k=1}^{k_j} w_{jk} = 1, \qquad \sum_{k=1}^{k_{j-1}} y_{jk} = 1$$

$$y_{jk} = 0 \quad or \quad 1 \ ; \qquad k = 1,2,...,K_j \ , \qquad j = 1,2,...,n$$

w_{jk} and y_{jk} are the variables for the approximating problem

The regular simplex method may be used for explain the estimated problem with the additional constraints linking y_{jk}.

Algorithm

Step 1: Where the OF is of minimize form; it is to be converted into maximization.

Step 2: The functions $f_j(x_j)$ and $gij(x_j)$ are tested to see whether they are satisfying the concavity (convexity) conditions required for the maximization(minimization) of NLPP. Where the conditions are not fulfilled, the method isn't applicable, else go to the next step.

Step 3: The interval $0 \le xj \le tj\ (j = 1,2,...,n)$ to be divided into a number of mesh points $ajk\ (k = 1,2,...,Kj)$ such that $a_{j1} = 0$,

$$a_{j1} < a_{j2} < ... < a_{jk} = t_j.$$

Step 4: Compute piece-wise linear approximation for each point a_{jk}, for each $f_j(x_j)$ and $g_{ij}(x_j)$

$$where\ j = 1,2,...,n;\ i = 1,2,...,m.$$

Step5: Calculation of step 4 are then used to, write down the linear approximation piece wise of the given NLPP.

Step 6: The two-phase simplex method may now be used to solve the given NLPP. For this method r $w_{i1}(i = 1, 2,...,m)$ may be considered as artificial variables. , The costs linked with them are assumed to be zero as they are not given. Now, Phase-I is automatically complete for this method., the initial simplex table of Phase-I is optimum and will therefore be the first simplex table for Phase-II.

Step 7: The optimum outcome x_j^* of the original problem is finally obtained by the relations:

$$x_j^* = \sum_{k=1}^{K_j} a_{jk} w_{jk} \quad (j = 1, 2, ..., n)$$

3.6 Summary

- By dividing up difficult optimization issues into smaller, independent sub-problems that can be solved separately and then combined to get the overall best solution, separable programming is a technique for optimization.
- The optimization issue is divided into more manageable sub-problems in separable programming, each of which deals with a subset of variables or constraints.
- Numerous disciplines, including operations research, engineering, economics, logistics, and more, frequently use separable programming.

3.7 Keywords

- Separable Programming
- Optimization problems
- Reduction of separable programming
- Piece-wise Linear Approximation

3.8 Self Assessment Questions

1. What are some common decomposition methods used in separable programming?
2. What are the benefits of decomposing optimization problems into sub-problems?
3. In what types of applications is separable programming particularly useful?
4. How do optimality conditions apply to separable programming?
5. Can you give an e.g. of a real-world problem that can be solved using separable programming?
6. What are some challenges associated with implementing separable programming techniques?

3.9 Case Study

1. Consider a large manufacturing company with multiple production facilities, distribution centers, and transportation networks. How can separable programming be applied to optimize the supply chain, considering factors such as production capacities, inventory levels, transportation costs, and customer demands?

2. Consider a financial institution overseeing a varied assortment of investments across multiple industries and domains. How do risk management, return goals, and regulatory constraints factor into the optimal allocation of funds when using separable programming techniques?

3.10 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.

UNIT -4

Convex Optimization

Learning Objective

- It is essential to become proficient at expressing optimization issues as convex optimization problems.
- Convex optimization issues from practical applications should be recognized by students, and they should also be taught how to formulate them in conventional mathematical forms.
- Lagrange duality, weak and strong duality, and the process of deriving and interpreting dual solutions are all ideas that students should be familiar with.

Structure

4.1 The Problem of optimization

Minimization of the problem, $f_o(z)$, over the optimization variable, x, s.t. constraints.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0 \quad i = 1, \dots, m \\ & h_i(x) = 0 \quad i = 1, \dots, n \end{aligned}$$

x^*, is the solution i.e the optimizer, which satisfies the constraints and has the lowest objective value.

$$fo(x*) \leq fo(x)$$

for all other x that are feasible (i.e. that assure the constraints).

The problem may find local minima and thus be very hard. Convex functions may however, be optimized as there are many well-developed tools for optimizing them Known rates of convergence and provable guarantees exist for this group of functions.

Nice Optimization Problems

Eg. 4.1.1

Least Squares

The least squares problem

$$\underset{x}{\text{minimize}} \quad ||Ax - b||_2^2$$

An analytical solution of the form is:

$$x^* = (A^T A)^{-1} A^T b$$

Though an analytical solution exists for this problem, the same does not exist, for most problems. To get close to the solution let us learn how to use an algorithm.

Linear Program

$$\begin{aligned} &\underset{x}{\text{minimize}} && c^T x \\ &\text{subject to} && a_i^T x \leq b_i \quad i = 1, \ldots, m \end{aligned}$$

4.2 General Convex Optimization Problem

Recall the problem:

$$\begin{aligned} &\underset{x}{\text{minimize}} && f_0(x) \\ &\text{subject to} && f_i(x) \leq 0 \quad i = 1, \ldots, m \end{aligned}$$

where f_i are all convex.

Def. A function is convex if:

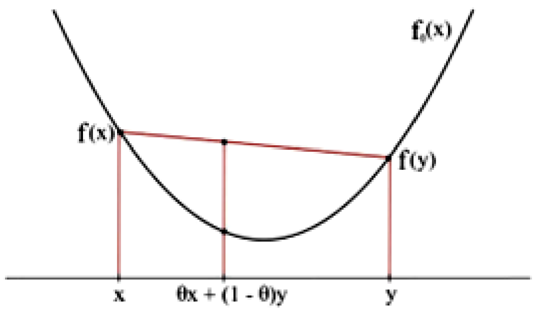


Figure 4.2.1 Graph of General Convex Optimization relate to given f(x)

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$

If $\theta \epsilon [0, 1]$.

If the inequality is strict, a function is strictly convex, and if the Hessian has positive curvature strongly convex. A convex set is defined by the constraints $f_i(x) \leq 0$.

Definition-For any x_1, $x_2 \epsilon C \rightarrow \theta x_1+(1-\theta)x_2 \epsilon C$ for $\theta \epsilon [0,1]$ a set$C \subset X$ is convex if the line between any two points relics in the set

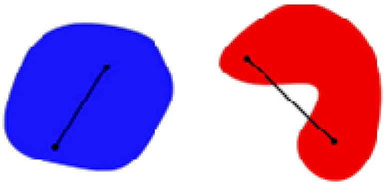


Figure 4.2.2 Convex and non-convex function

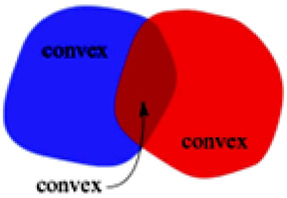


Figure 4.2.3 The intersection of two convex functions

Figure 4.2.2 shows the convex and non-convex function with diagram. And Figure 4.2.3 shows that a convex function is also formed by the intersection of two convex functions.

Fact: For disjoint convex sets C and D, there is a hyperplane a€R^d (Figure 4.2.3) such that:

$$a^T x \leq b \quad x \in C$$
$$a^T x \geq b \quad x \in D$$

This is the separating hyperplane theorem.

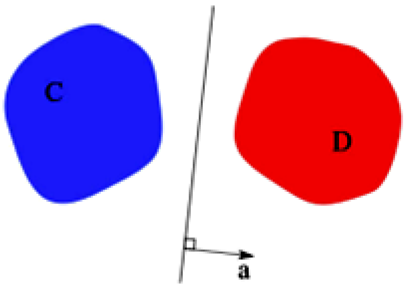


Figure 4.2.4 The separating hyperplane of convex functions

Fact: If C is a convex set, then there is a supporting hyperplane (Figure 4.2.5) at every boundary point x_o.

$$a^T x \leq a^T x_0 \quad \text{for all } x \in C$$

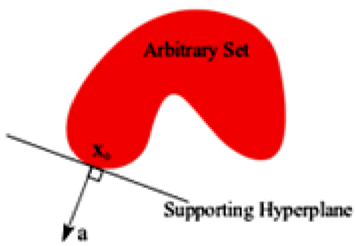


Figure 4.2.5 The supporting hyperplane of convex function

f is convex if

$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$

If z is a general random variable (Eg., $z = x$ w.p. θ, $z = y$ w.p. $1–\theta$)

$f(\mathbb{E}z) \leq \mathbb{E}(f(z))$

for a convex function. This is called **Jenson's Inequality**.

Eg. 4.2.1

x, x^4, 0, e^x, negative entropy (-x log x)

Eg. on R": Norms

$||x||_p = (\sum_{i=1}^{n} |x_i|^p)^{1/p}$

Eg. on $R^{n \times m}$: Matrix Norms

$||X||_2 = \sigma_{max}(X) = \max_v \frac{||Xv||_2}{||v||_2} = f(x)$

Fact: $f: Rn \rightarrow R$ is convex iff $g: R \rightarrow R$

g(t) = f(x+tv) is convex for any v in t. One may check the convexity (see figure 4.2.6) of f by checking 1-D convexity.

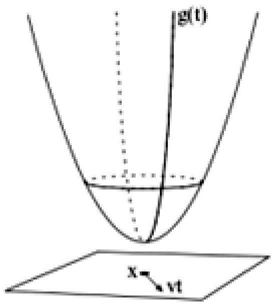


Figure 4.2.6 The convexity of g with f convex function

First Order Condition

f is differentiable if:

$$\nabla f(x) = (\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), ..., \frac{\partial f}{\partial x_n}(x))$$

exists. A differentiable f (with convex domain) is convex iff:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

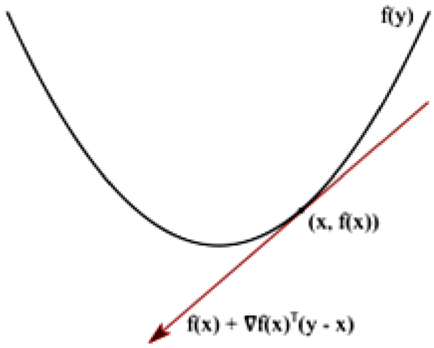


Figure 4.2.7 First order condition graph of faction f

Second Order Condition

f is convex if

$$\nabla^2 f(x)_{ij} = \frac{\partial f}{\partial x_i \partial x_j}(x)\ \nabla f(x) \succeq 0$$

For a positive semi definite Matrix A (A≥ 0) if $x^T Ax \geq 0$ (i.e. all eigen values ≥ 0).

$\nabla^2 f$ measures the curvature at x and is called the **Hessian**. f is strictly convexif the Hessian $\nabla^2 f(x) > 0$. f is strongly convex if $\nabla^2 f(x) > \mu I$.

Positive curvature everywhere

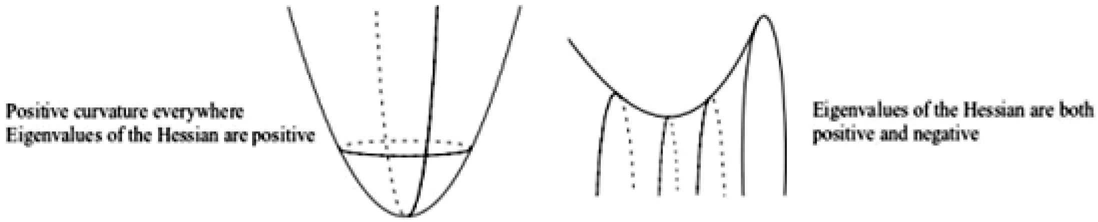


Figure 4.2.8 Eigen values of the Hessian

Def. A point x* if it is the optimal solution

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0 \\ & ||x^* - x|| \leq R \end{aligned}$$

is a local solution of an optimization problem.

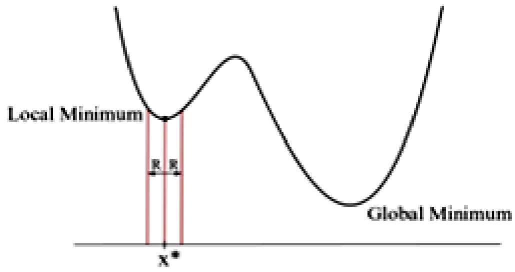


Figure 4.2.8 Local and Global minimum of convex optimization problems

For convex optimization problems (figure 4.2.8)local outcomes are all global outcomes.

4.3 Gradient Descent

This makes the "approximately optimal" point where $f(xk) \approx f(x*)$ after k iterations.

Initialize x_o Set:

$$x_1 = x_0 - \eta \nabla f_0(x_0)$$

where η is the step size and - $\eta \nabla f_o(x_o)$ is the direction of steepest descent.

$$x_{k+1} = x_k - \eta_k \nabla f_0(x_k)$$

always converges to a local minimum, and to a global minimumif the function is convex,. Where it is close to the minimum, in case step size is very large the value may skipbeyond the optimal point and move backward and forward around it. One must think carefully before selecting the size of the step, the easiest of which isη_k=1/k. Convergence may be improvedby using Newtons Method to fit a quadratic approximation.

4.4 Summary

Mathematical optimization with an emphasis on solving problems when the feasible region and the goal function are both convex is known as convex optimization. Finding the least (or greatest) value of a convex function over a convex set is the aim of convex optimization, which covers a broad range of real-world issues from several academic fields.

Operation Research, Machine Learning, Finance, Signal Processing and control theory are some of the many application of Convex optimization. In machine learning, convex optimization is used for training algorithms such as support vector machines, logistic regression, and neural networks. In signal processing, convex optimization techniques are applied to problems such as signal denoising and image reconstruction. In control theory, convex optimization is used for designing optimal control strategies for dynamical systems. In finance, convex optimization is used for portfolio optimization and risk management. In operations research, convex optimization is applied to problems such as transportation planning and resource allocation.

4.5 Keywords

- Convex optimization
- The Optimization Problem
- General Convex Optimization Problem
- Gradient Descent

4.6 Self Assessment Questions

1. What are the key characteristics of convex sets?
2. What distinguishes a convex function from a non-convex one?
3. Why are convex optimization problems often preferred in practice?
4. What are the main applications of convex optimization?
5. What is duality theory in the context of convex optimization?

4.7 Case Study

In a city facing traffic congestion issues, how can convex optimization methods be used to optimize traffic flow and minimize congestion? This could involve optimizing traffic signal timings, route assignments, and toll pricing strategies to improve overall traffic efficiency and reduce travel times.

4.8 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.

UNIT - 5

Numerical optimization techniques

Learning Objective

- Being aware of the optimization algorithms' convergence qualities, such as rate of convergence, convergence criteria, and the effect of problem features on convergence.
- Enhancing one's ability to solve problems by using optimization approaches to tackle real-world issues, analyzing the outcomes, and drawing conclusions from the optimization process.
- With a strong foundation in numerical optimization techniques, students will be prepared to take on a wide variety of optimization issues in a variety of academic fields and business settings.

Structure

5.1 Introduction to Numerical Optimization-

Finding the best solution to a problem from a set of feasible options—where the solutions are represented numerically—is the focus of the mathematical and computer science discipline of numerical optimization. Applications in the fields of science, engineering, finance, machine learning, and many more depend on this topic.

Here's a basic introduction to numerical optimization:

1. **OF** : At the core of any optimization problem lies an OF . This function represents what you want to optimize, whether it's maximizing profit, minimizing cost, fitting a

model to data, or any other goal. Mathematically, an OF can be denoted as $(x)f(x)$, where xx is a vector representing the variables of the problem.

2. **Constraints**: Often, optimization problems come with constraints, which are conditions that the solution must satisfy. Constraints may be of two types equality (e.g., f(x)=0) or inequality constraints (e.g., h(x)≤0). Constraints can significantly affect the feasible solution space.
3. **Optimization Algorithms**: The ways used to find an optimal solution to an optimization problem are called numerical optimization algorithms. They may be broadly be categorized into two types:
   - **Gradient-based methods**: These methods utilize information about the gradient (first derivative) or the Jacobian (matrix of partial derivatives) of the OF to iteratively improve the solution.
   - **Derivative-free methods**: Also known as direct search or black-box optimization methods, these algorithms do not require explicit knowledge of derivatives. They explore the solution space based solely on the OFevaluations.
4. **Global vs. Local Optima**: A key challenge in optimization is distinguishing between global and local optima. A global optimum is the best possible solution across the entire feasible region, while a local optimum is the best solution within a certain neighborhood of the current point. Depending on the problem and the optimization algorithm used, it may be difficult to guarantee finding the global optimum.
5. **Applications**: Numerical optimization finds applications in numerous fields, such as engineering design, financial portfolio optimization, machine learning model training (e.g., parameter tuning), logistics and supply chain management, and many others. Essentially, any problem that involves making decisions or finding the best solution among alternatives can benefit from optimization techniques.

5.2 Line search methods

Line search methods are iterative optimization techniques used to find a local minimum or maximum of a function. They are particularly useful in gradient-based optimization algorithms, such as gradient descent and quasi-Newton methods. The primary idea is to determine an appropriate step size along a given search direction that sufficiently decreases (or increases) the OF.

Basic Steps in Line Search Methods

1. Choose a Search Direction: Determine the direction d_k in which to move from the current point x_k.

2. Perform Line Search: Find an appropriate step size α_k that minimizes (or maximizes) the OF along the search direction d_k.

3. Update the Current Point: Move to the new point $x_{\{k+1\}}=x_k+\alpha_k d_k$.

4. Check Convergence: If the change in the OF or the norm of the gradient is below a specified threshold, terminate the algorithm.

Line Search Criteria

1. Exact Line Search

In exact line search, the step size α_k is chosen to minimize the OF precisely along the search direction:

α_k= arg minα_kf(x_k+αd_k).

This approach is often impractical for complex functions due to the computational effort required.

2. Inexact Line Search

Inexact line search methods seek a step size that sufficiently decreases the OF without the need for exact minimization. Common criteria include:

-**Armijo (Sufficient Decrease) Condition**: Ensures that the step size decreases the OF by a sufficient amount:

$$f(x_k + \alpha d_k) \leq f(x_k) + c_1\alpha\nabla f(x_k)^T d_k,$$

where $0 < c_1 < 1$ is a constant.

-**Wolfe Conditions**: Combine the sufficient decrease condition with a curvature condition to ensure that the slope of the OF is adequately reduced:

$$\nabla f(x_k + \alpha d_k)^T d_k \geq c_2\nabla f(x_k)^T d_k,$$

where $0 < c_1 < c_2 < 1$.

-**Strong Wolfe Conditions**: Strengthen the Wolfe conditions by enforcing the curvature condition with a stricter inequality:

$$|\nabla f(x_k + \alpha d_k)^T d_k| \leq c_2|\nabla f(x_k)^T d_k|.$$

5.3 Gradient methods

Gradient methods, also known as gradient descent methods, are a class of optimization algorithms used to minimize (or maximize) an objective function by iteratively moving in the direction of the negative gradient of the function. These methods are widely used in various fields, including machine learning, numerical optimization, and engineering. Here's an overview of gradient methods:

1. Gradient Descent

Gradient Descent is a first-order iterative optimization algorithm for finding the minimum of a function. The basic idea is to take steps proportional to the negative of the gradient of the function at the current point.

Algorithm:

1. Initialize x_0 (starting point).

2. Repeat until convergence:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$$

where α is the step size (learning rate) and $\nabla f(x_k)$ is the gradient of f at x_k.

2. Stochastic Gradient Descent (SGD): This is a variant of gradient descent where the gradient is computed using a randomly selected subset of data points (mini-batch) instead of the entire dataset. This can significantly speed up the optimization process for large datasets.

Algorithm:

1. Initialize x_0 (starting point).

2. Repeat for each mini-batch B_k:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f_{B_k}(\mathbf{x}_k)$$

where $\nabla fB_k(x_k)$ is the gradient computed using the mini-batch B_k.

careful tuning of step size and mini-batch size.

3. Momentum

Momentum is an extension of gradient descent that helps accelerate convergence by adding a part of the earlier update vector to the present update vector. This helps in smoothing the optimization path and can help escape local minima.

Algorithm:

1. Initialize $x_0$and set $v_0 = 0$ (initial velocity).

2. Repeat until convergence:

$$\mathbf{v}_{k+1} = \beta \mathbf{v}_k + \alpha \nabla f(\mathbf{x}_k)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{v}_{k+1}$$

where β is the momentum parameter (usually between 0.5 and 0.9).

4. Nesterov Accelerated Gradient (NAG)

Nesterov Accelerated Gradient is a variant of momentum that computes the gradient at the predicted next position, leading to a more accurate and potentially faster convergence.

Algorithm:

1. Initialize x_0 and set $v_0 = 0$ (initial velocity).
2. Repeat until convergence:

$$\mathbf{y}_k = \mathbf{x}_k - \beta \mathbf{v}_k$$

$$\mathbf{v}_{k+1} = \beta \mathbf{v}_k + \alpha \nabla f(\mathbf{y}_k)$$

5.4 Summary

1. Types of Optimization Problems: Under certain constraints, an OF must be maximized or minimized. They fall into two categories: restricted optimization, which requires satisfying extra constraints, and unconstrained optimization, which just takes the OF into account.
2. The optimization process utilizes a range of algorithms and techniques, such as gradient-based approaches (like Newton's method and gradient descent), stochastic approaches (like simulated annealing and genetic algorithms), and metaheuristic approaches (like particle swarm optimization and ant colony optimization).
3. Complexity Analysis and Convergence: It's important to comprehend how optimization algorithms converge. We examine convergence rates, convergence criteria, and the effect of issue features on convergence. In evaluating the effectiveness of various algorithms, computational complexity is also taken into account.

5.5 Keywords

- Numerical Optimization
- Line search methods

- Gradient methods
- Optimization Problems
- Convergence

5.6 Self Assessment Questions

1. Name two popular gradient-based optimization algorithms.
2. What is the role of constraints in constrained optimization problems?
3. What is convergence in the context of optimization algorithms?
4. Name a stochastic optimization method and briefly explain how it works.
5. How do metaheuristic approaches differ from traditional optimization algorithms?
6. What programming languages are commonly used for implementing optimization algorithms?
7. Give an Eg. of a real-world application where optimization techniques are commonly used.
8. What is the significance of computational complexity in optimization algorithms?

5.7 Case Study

A leading aircraft manufacturer is developing a new model with the objective of enhancing fuel efficiency while maintaining structural integrity and aerodynamic performance. The design team is tasked with optimizing the shape and structure of the aircraft wing to achieve these objectives.

Problem Statement: Optimize the design of the aircraft wing to minimize fuel consumption while satisfying constraints related to structural strength, aerodynamic performance, and manufacturing feasibility.

5.8 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.

UNIT -6

Newton's method

Learning Objective

- Students should be able to implement Newton's method algorithmically to solve optimization problems.
- Recognizing that Newton's method may converge to local optima rather than global optima in certain cases is important.
- Students should understand strategies for addressing this limitation, such as using multiple starting points or incorporating global optimization techniques.

Structure

6.1 Single variables optimizations:-

When the maximum and minimum of a function is found with respect to one variable it is called single variable optimization in mathematical programming. This type of optimization is fundamental and can be applied in various fields such as economics, engineering, and data science.

Key Concepts in Single Variable Optimization

1. **OF**: The function f(x) that needs to be maximized or minimized.
2. **Domain**: The set containing all values of x for which the function f(x) is defined.
3. **Critical Points**: Points where f''(x) is zero or undefined. These points could be the local maxima, minima, or saddle points.
4. **Endpoints**: In cases where the domain is a closed interval, the endpoints of the interval need to be checked as well since the global maximum or minimum can occur there.

5. **Second Derivative** Test: A method to find out whether a critical point is a local maximum/minimum/saddle point:

- If $f''(x) > 0$ at a critical point, it is a local minimum.
- If $f''(x) < 0$ at a critical point, it is a local maximum.
- If $f''(x) = 0$, the test is indecisive.

Steps in Single Variable Optimization

1. **Define the OF**: Clearly state the function f(x) to be optimized.
2. **Compute the First Derivative**: Find f'(x) to identify critical points.
   f'(x) = 0
3. Solve for Critical Points: Solve the equation f'(x) = 0 to find the critical points.
4. Compute the Second Derivative: Find f''(x) to apply the second derivative test.
5. Evaluate the OF at Critical Points and Endpoints: Calculate the function values at all critical points and endpoints to determine the global maximum or minimum.

Eg. 6.1.1

To evaluate the maximum and minimum of $f(x) = -2x^3 + 3x^2 + 12x + 5$ in the interval [0, 2].

Solution:

1. **OF** : $f(x) = -2x^3 + 3x^2 + 12x + 5$
2. **First Derivative**: $f'(x) = -6x^2 + 6x + 12$
3. **Solve for Critical Points**: $-6x^2 + 6x + 12 = 0$

Solving this quadratic equation gives the critical points.

4. **Second Derivative**: $f''(x) = -12x + 6$

Use the second derivative test on the critical points to classify them.

5. **Evaluate at Critical Points and Endpoints**:
   - Calculate f(x) at each critical point within the interval [0, 2].
   - Calculate f(0) and f(2) .

Eg. 6.1.2

Maximize the profit $P(x) = -5x^2 + 150x - 1000$.

Solution:

1. **Define the OF** :
   $P(x) = -5x^2 + 150x - 1000$

2. **Find the Derivative**:

$P'(x) = \{d/dx\}(-5x^2 + 150x - 1000) = -10x + 150$

3. **Set the Derivative to Zero**:

$-10x + 150 = 0 \Longrightarrow x = 15$

4. **Second Derivative Test**:

$P''(x) = \{d^2/dx^2\}(-5x^2 + 150x - 1000) = -10$

Since $P''(15) = -10 < 0$, the function has a maximum at $x = 15$.

5. **Evaluate Boundaries**: (If applicable, otherwise skip this step).

6. **Compare Values**: The maximum profit occurs at $x = 15$.

Thus, the value of x that maximizes the profit is $x = 15$.

6.2 Summary

1. It is essential to comprehend how to use Newton's approach algorithmically. This entails figuring out convergence criteria, updating variables, and computing derivatives.
2. Rather than leading to global optima, Newton's technique might converge to local ones. This restriction can be eased by applying tactics like utilizing several starting points or global optimization strategies.
3. Sensitivity analysis assesses the impact of altering the parameters of the problem on the convergence and solution found by Newton's approach. This makes it easier to comprehend how resilient solutions are to changes in the input data.

6.3 Keywords

- Newton's method
- Convergence
- Local Optima
- Operations research
- Optimization problems

6.4 Self Assessment Questions

1. How does Newton's method differ from other optimization techniques?
2. What role does the concept of root-finding play in Newton's method?
3. How does Newton's method handle local optima in optimization problems?

4. Can you explain the iterative process of Newton's method in brief?
5. What are the key factors that influence the convergence of Newton's method in optimization problems?
6. How is sensitivity analysis used to evaluate the robustness of solutions obtained through Newton's method?

6.5 Case Study

A multinational retail corporation is facing challenges in optimizing its supply chain network. The company operates multiple warehouses and distribution centers globally, with the goal of minimizing transportation costs while meeting customer demand efficiently. However, the existing network configuration is suboptimal, leading to high transportation expenses and inefficient inventory management.

Objective: The objective is to redesign the supply chain network to minimize transportation costs while ensuring timely delivery to customers.

6.6 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.

UNIT -7

Hessian Matrix

Learning Objective

- Students should understand the fundamental concepts of CPM and PERT, including their objectives, methodologies, and terminologies
- Students should understand the concept of the Hessian matrix in the context of optimization problems. This includes recognizing its role in determining the curvature and behavior of the OF around a given point.
- Understanding how PERT incorporates uncertainty and risk management into project scheduling through probabilistic estimates and sensitivity analysis.
- Students should learn techniques for analyzing and mitigating risks in project networks.

Structure

7.1 Introduction of Hessian Matrix

The Hessian matrix is a key concept in multivariable calculus and optimization, used to analyze the curvature of a function with respect to multiple variables. It extends the idea of the second derivative from single-variable calculus to functions of multiple variables, providing important information about the function's local behavior.

Definition 7.1.1

For a twice-differentiable scalar function f: $R^n \rightarrow R$, the Hessian matrix is a square matrix of 2^{nd}-order partial derivatives. If f{x} = (x_1, x_2,…., x_n) represents the vector of variables, the Hessian matrix H of f is defined as:

$$H(f)(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Each entry H_{ij} in the Hessian matrix represents the second partial derivative of f with respect to x_i and x_j

Properties

1. **Symmetry**: If f is twice continuously differentiable, the Hessian matrix is symmetric, meaning $H_{ij}=H_{ji}$.
2. **Quadratic Form**: The Hessian matrix is used to form a quadratic approximation of f near a point x_0:

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T(\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T H(f)(\mathbf{x}_0)(\mathbf{x}-\mathbf{x}_0)$$

3. Critical Points: The Hessian matrix helps determine the nature of critical points (points where the gradient $\nabla f=0$):

Local minimum - where $H > 0$&definite at a critical point.

Local maximum - where $H < 0$&definite at a critical point..

Saddle point –Where H is indefinite (has both > 0 and < 0 eigenvalues).

Eg. 7.1.1

Consider the function $f(x, y) = x^3 + y^3 - 3xy$.

Solution:

1. First Partial Derivatives:

$$\frac{\partial f}{\partial x} = 3x^2 - 3y, \quad \frac{\partial f}{\partial y} = 3y^2 - 3x$$

2. Second Partial Derivatives:

$$\frac{\partial^2 f}{\partial x^2} = 6x, \quad \frac{\partial^2 f}{\partial y^2} = 6y, \quad \frac{\partial^2 f}{\partial x \partial y} = -3, \quad \frac{\partial^2 f}{\partial y \partial x} = -3$$

3. Hessian Matrix:

$$H(f)(x,y) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 6x & -3 \\ -3 & 6y \end{bmatrix}$$

4. Critical Points: Solve ∇f = 0:

$3x^2 - 3y = 0 \Longrightarrow y = x^2$

$3y^2 - 3x = 0 \Longrightarrow x = y^2$

Substituting y = x^2 into x = y^2 , we get x = (x^2)^2 = x^4 , leading to x(x^3 - 1) = 0 . Thus, x = 0 or x = 1.

- For x = 0, y = 0.
- For x = 1, y = 1.

The critical points are (0,0) and (1,1) .

5. Nature of Critical Points:

- At (0,0):

The eigen values of this matrix are \pm 3i , indicating a saddle point.

$$H(f)(0,0) = \begin{bmatrix} 0 & -3 \\ -3 & 0 \end{bmatrix}$$

- At (1,1):

$$H(f)(1,1) = \begin{bmatrix} 6 & -3 \\ -3 & 6 \end{bmatrix}$$

The eigen values of this matrix are 9 and 3, both positive, indicating a local minimum.

7.2 Applications of Hessian Matrix

The Hessian matrix finds applications across various fields, particularly in optimization, numerical analysis, physics, and engineering. Here are some specific applications:

1. **Optimization**:
    - **Constrained Optimization**: In constrained optimization problems, the Hessian matrix helps determine the nature of critical points and the feasibility of solutions. It is essential for algorithms such as the method of Lagrange multipliers and interior-point methods.

- **Unconstrained Optimization**: In unconstrained optimization, the Hessian matrix is used to classify critical points as minima, maxima, or saddle points. It plays a central role in Newton's method and quasi-Newton methods for finding the minimum or maximum of a function.

2. **Machine Learning**:
   - **Second-Order Optimization**: In certain machine learning algorithms, such as Newton's method for training logistic regression models or neural networks, the Hessian matrix can be used to accelerate convergence and improve optimization efficiency. However, its computation can be expensive, so approximations like the Gauss-Newton method or the BFGS algorithm are often used.
   - **Hessian-Based Regularization**: The Hessian matrix can also be used to compute regularization terms in machine learning models. For Eg., Hessian-based L2 regularization penalizes large weights in a neural network to prevent over fitting.
3. **Physics**
   - **Quantum Mechanics**: In quantum mechanics, the Hessian matrix plays a role in understanding the stability and behavior of quantum systems. For Eg., in the study of molecular vibrations, the Hessian matrix characterizes the potential energy surface around an equilibrium geometry.
   - **Solid Mechanics**: In solid mechanics, the Hessian matrix is used to analyze the stability of mechanical systems and to determine critical points in stress or strain fields.
4. **Engineering**:
   - **Structural Analysis**: The Hessian matrix is used in finite element analysis (FEA) and structural optimization to evaluate the stability and performance of engineering structures under various loading conditions.
   - **Control Systems**: In control theory, the Hessian matrix can be used in the design and analysis of control systems, particularly in optimal control and state estimation problems.
5. **Financial Modeling**:
   - **Portfolio Optimization**: In finance, the Hessian matrix is used in portfolio optimization to determine the optimal allocation of assets in an investment

portfolio. It helps investors balance risk and return by considering the covariance structure of asset returns.

6. **Image Processing**:
   - **Image Registration**: In medical imaging and computer vision, the Hessian matrix is used in image registration algorithms to align and match images from different modalities or viewpoints. It helps quantify image similarity and guide the optimization process.

7.3 Project Management

Project management in mathematical programming involves using mathematical techniques and optimization models to plan, execute, and control projects efficiently. These techniques can help optimize resource allocation, schedule tasks, manage costs, and ensure timely completion of projects. Key areas in project management where mathematical programming is applied include scheduling, resource allocation, cost management, and risk analysis.

1. Scheduling

Project scheduling involves determining the optimal start and finish times for project activities. Mathematical programming techniques like linear programming (LP), integer programming (IP), and mixed-integer programming (MIP) are used to solve scheduling problems.

Critical Path Method (CPM)

This is a step wise technique for project management to recognize actions on the critical path. The critical path is the series of activities that determines minimum project duration.

Steps:

1. Required activities for project completion to be listed.
2. Dependence between activities to be found.
3. Construct a project network diagram.
4. Estimate the Earliest Start (ES) and Earliest Finish (EF) times for every activity.
5. Estimate the Latest Start (LS) and Latest Finish (LF) times for every activity.
6. Estimate the critical path by finding the longest path through the network diagram.

Formulation:

Minimize project duration T

Subject to:

$$
\begin{aligned}
&ES(i) \geq EF(j) && \forall (j \to i) \in \text{dependencies} \\
&EF(i) = ES(i) + \text{duration}(i) && \forall i \\
&LS(i) = LF(i) - \text{duration}(i) && \forall i
\end{aligned}
$$

Program Evaluation and Review Technique (PERT)

PERT is a statistical tool utilized to examine and signify the tasks concerned in finishing a project. It includes uncertainty by using three time estimates for every activity: most likely, pessimistic, and optimistic.

Formulation:

Expected time for each activity:

TE = (O + 4M + P)/6

If M is the most likely time, O is the optimistic time and P is the pessimistic time.

2. Resource Allocation

Resource allocation involves assigning available resources to project activities in the most efficient way. Mathematical programming can help optimize resource utilization while respecting constraints like resource availability and activity durations.

Linear Programming for Resource Allocation

Linear programming can be used to allocate resources to activities such that the total cost or time is minimized.

Formulation:

Minimize: $\sum_{i=1}^{n} c_i x_i$

Subject to:

$$
\sum_{i=1}^{n} a_{ij} x_i \leq b_j \quad \forall j
$$

$$
x_i \geq 0 \quad \forall i
$$

where c_i is the cost of resource i, x_i is the amount of resource i allocated, a_{ij} is the amount of resource i required for activity j, and b_j is the available amount of resource j.

3. Cost Management

Cost management involves planning and controlling the budget of a project. Mathematical programming can help in budgeting, forecasting costs, and optimizing expenditures.

Cost Minimization Model

A cost minimization model can be formulated to minimize the total project cost s.t. constraints on resources and project duration.

Formulation:

Minimize: $\sum_{i=1}^{n} c_i x_i$

Subject to:

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j \qquad \forall j$$
$$\sum_{i=1}^{n} d_i x_i \leq D \quad \text{(Project duration constraint)}$$
$$x_i \geq 0 \qquad \forall i$$

where d_i is the duration of activity i, and D is the maximum allowed project duration.

4. Risk Analysis

Risk analysis involves identifying, assessing, and prioritizing risks associated with a project. Mathematical programming can help in quantifying risks and determining the optimal risk mitigation strategies.

Risk Optimization Model

A risk optimization model can be used to minimize the impact of risks on the project while considering the cost of mitigation measures.

Formulation:

Minimize: $\sum_{i=1}^{n} (R_i x_i + C_i x_i)$

Subject to:

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j \qquad \forall j$$
$$\sum_{i=1}^{n} d_i x_i \leq D \quad \text{(Project duration constraint)}$$
$$x_i \geq 0 \qquad \forall i$$

where R_i is the risk associated with activity i, C_i is the cost of mitigating the risk, and x_i is the decision variable indicating whether to mitigate the risk for activity i.

7.4 Historical Development of CPM/PERT

PERT/CPM/Network Analysis developed along two streams, i.e. industrial and military.

CPM (Critical Path Method) In 1957 this method was discovered by Mr. M.R. Walker and J.E. Kelly. They developed the method for UNIVAC-I. it was first tested in 1958. In March 1959, it was used at Du Pont works in Louisville, Kentucky. To reduce Unproductive time 93 Hrs from 125.

Project Evaluation and Review Technique (PERT) was designed in 1958 for the POLARIS missile program of the US Navy.

The above are **network-oriented techniques which** use the same principle and decide the time schedule for a project, CPM is deterministic while PERT is probabilistic. These techniques are known as project **scheduling** techniques.

The Framework for PERT and CPM

The six steps used in both techniques are:

I. Definition of project and listing of its required activities. Each Project (made up of several tasks) may have only one start activity one finish activity.

II. Defining relationships between the activities. Sequence of the activity is decided.

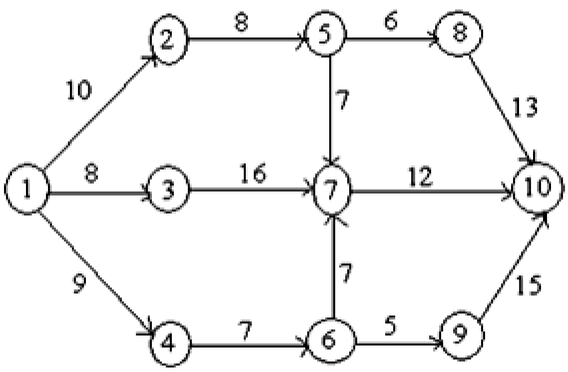
III. Create the "Network" linking all the activities. Each Activity should have exclusive event numbers. Fake arrows are used when needed so that each activities has a unique identity.

IV. Allot time and/or cost estimates to every activity

V. Calculate the longest time path through the network. This is called the critical path.

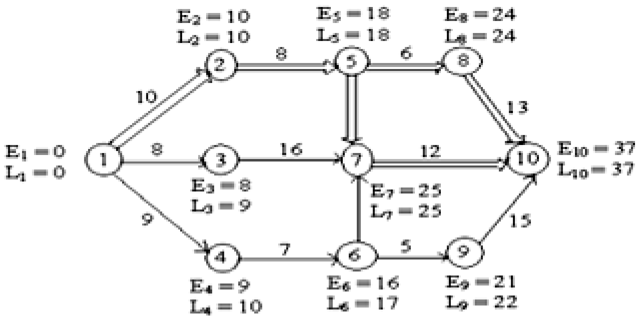
Eg. 7.4.1

Find the early start and late start in respect of all node points and recognize critical path for the next network.



Solution:

Computation of E and L. for each node is exposed in the network



Activity(i, j)	Normal Time (D_{ij})	Earliest Time		Latest Time		Float Time (L_i - D_{ij}) - E_i
		Start (E_i)	Finish (E_i + D_{ij})	Start (L_i - D_{ij})	Finish (L_i)	
(1, 2)	10	0	10	0	10	0
(1, 3)	8	0	8	1	9	1
(1, 4)	9	0	9	1	10	1
(2, 5)	8	10	18	10	18	0
(4, 6)	7	9	16	10	17	1
(3, 7)	16	8	24	9	25	1
(5, 7)	7	18	25	18	25	0
(6, 7)	7	16	23	18	25	2
(5, 8)	6	18	24	18	24	0
(6, 9)	5	16	21	17	22	1
(7, 10)	12	25	37	25	37	0
(8, 10)	13	24	37	24	37	0
(9, 10)	15	21	36	22	37	1

Network Analysis Table

By the table, the critical nodes are (1, 2), (2, 5), (5, 7), (5, 8), (7, 10) and (8, 10).

By the table, there are two feasible critical paths

$$i.\ 1 \to 2 \to 5 \to 8 \to 10$$

$$ii. 1 \to 2 \to 5 \to 7 \to 10$$

7.5 Summary

- Project management techniques like CPM and PERT are commonly employed to schedule and oversee project activity.
- PERT uses probabilistic estimations to account for uncertainty, whereas CPM is deterministic and depends on set activity periods.
- The minimal project time is determined by the critical route, which is identified by CPM. PERT offers insight into the uncertainty around project duration.
- PERT is more suited for projects with significant uncertainty or variable in the duration of the activities, whereas CPM is more suited for those with clearly defined activity durations.
- Both methods aid in efficient project scheduling, planning, and administration, ensuring that projects are finished on time and under budget.

7.6 Keywords

- PERT
- CPM
- Project management
- Hessian matrix

7.7 Self Assessment Question

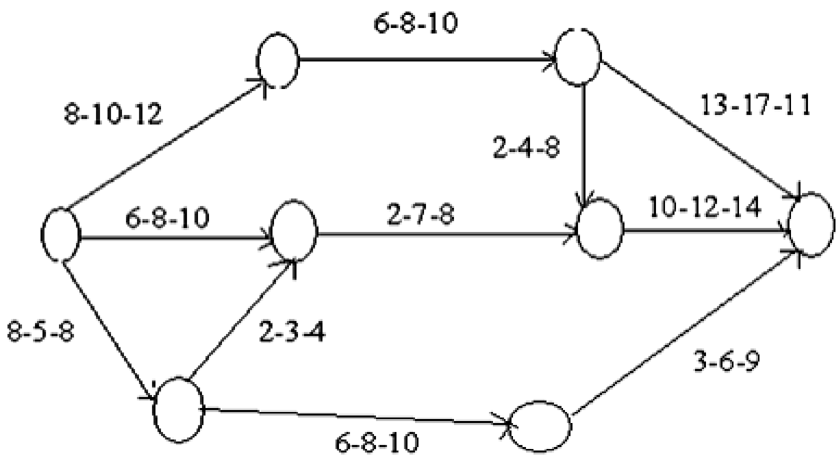
1. What is PERT?

2. For the next data, draw network. Discover the critical path, slack time after calculate the earliest probable time and the latest acceptable time

Activity	Duration	Activity	Duration
(1 – 2)	5	(5 – 9)	3
(1 – 3)	8	(6 – 10)	5
(2 – 4)	6	(7 – 10)	4
(2 – 5)	4	(8 – 11)	9
(2 – 6)	4	(9 – 12)	2
(3– 7)	5	(10 – 12)	4
(3 – 8)	3	(11 – 13)	1
(4 – 9)	1	(12 – 13)	7

[Ans. Critical path: 1→3→7→10→12→13]

3. Calculate the variance and the expected time for each activity



7.8 Case Study

Imagine you are managing a construction project to build a new office complex. The project involves multiple activities such as site preparation, foundation laying, structural construction, interior finishing, and landscaping. Each activity has a different duration and dependencies on other activities.

Using the concepts of Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT), outline how you would approach scheduling and managing this construction project. Discuss the steps you would take to create project networks, identify critical activities and the critical path, estimate project duration, and manage project risks effectively. Additionally, explain the benefits of using both CPM and PERT in this scenario.

7.9 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.

UNIT -8

Integer programming

Learning Objective

- Gaining knowledge on how to formulate practical issues as models for integer programming.
  Knowing the algorithms that underpin solvers for integer programming.
- Recognizing the computational intricacy involved in solving integer programming issues.
- Pupils ought to be able to distinguish between issues that can be solved in polynomial time.

Structure

8.1 Introduction of Integer programming

Variables in linear programming can take any real or fractional value but fractional values of variables have no significance in real life problems e.g. 28.5 workers or machines are required for a project.

The problem is overcome by rounding off all variables to the nearest integer, however, this approach may not satisfy all constraints, thus the OF values obtained may not be optimal.
Integer programming method is used to avoid this problem. In these problems some or all variables are limited to integer values only (discrete values).

Applications:-

Some of areas of application of integer programming are: shifting schedule and routing, capacity extension fixed charges, Capital Budgeting, batch size, construction scheduling, plant location and size, etc.

Integer Programming Problems - Types

These problems are of 3 types:

(i) **Pure (all) IPP** here all variables are constrained to integer values.

(ii) **Mixed IPP** here some variables are constrained to integer values.

(iii) **Zero-one IPP** here all decision variables are constrained to 1 / 0 only.

Standard form of Pure IPP

The pure IPP in its standard form can be written as follows

$$\text{Max } Z = c_1x_1 + c_2x_2 + \ldots . c_nx_n,$$

S.t. the constraints

$$a_{11}x_1 + a_{12}x_2 + \ldots\ldots\ldots\ldots a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \ldots\ldots\ldots\ldots a_{2n}x_n = b_2$$

$$a_{31}x_1 + a_{32}x_2 + \ldots\ldots\ldots\ldots a_{3n}x_n = b_3$$

$$;$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots\ldots\ldots\ldots a_{mn}x_n = b_m$$

and $x1, x2, x3, xn \geq 0$ and are integers.

Cut

A cut is a linear constraint insert to the specified LP Problem, it is also called additional linear constraint. (or fractional cut)

Eg. 8.1.1

Considering the next linear IPP

$\text{Max } Z = 14x_1 + 16x_2$

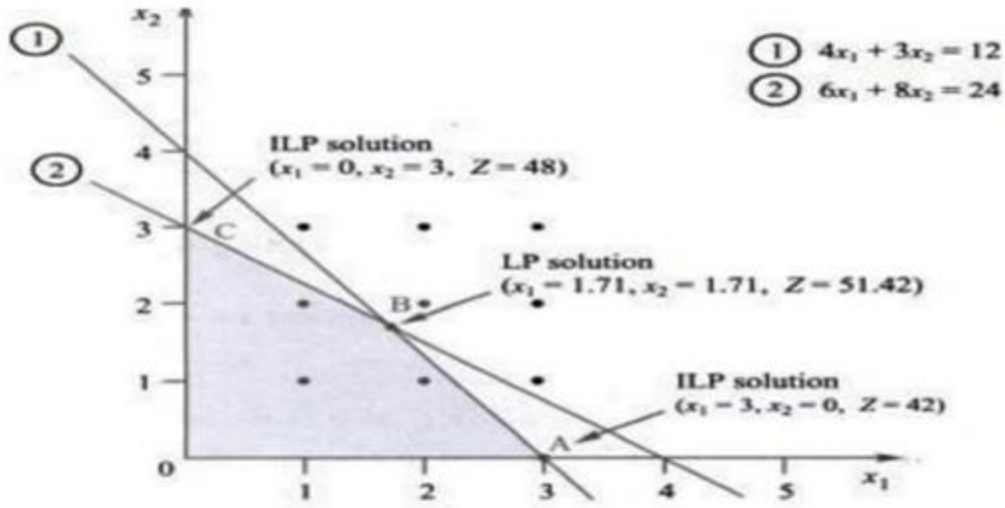
Subject to the constraints

$4x_1 + 3x_2 \leq 12$ and $6x_1 + 8x_2 \leq 24$

and $x_1, x_2 \geq 0$ and are integers.

Solution:-

Graphical Method



feasible region →OABC.

We obtain the next outcome $x1 = 1.71$ and $x2 = 1.71, Max\ Z = 51.42$.

This outcome doesn't satisfy the integer necessities of the variables $x1$ and $x2$.

Rounding off the outcome, we obtain$x1 = 2$ and $x2 = 2, Max\ Z = 60$, But it isn't feasible outcome, since it does not satisfy the specified inequality

$$4x1 + 3x2 = 4(2) + 3(2) = 14 \text{ is not less than } 12$$

$$6x1 + 8x2 = 6(2) + 8(2) = 28 \text{ is not less than } 24,$$

As such, the solution is not workable. The graph's dots indicate potential integer solutions that fall inside the LP problem's feasible zone. Another name for the dots is lattice points.

8.2 Gomory's algorithm for all and mixed IPP

Gomory's all integer cutting plane method

The Gomory all integer cutting plane algorithm is a methodical process that produces cuts (extra linear restrictions) to guarantee the integer outcome to the LP problem in a finite number of steps.

Property of Gomory's Algorithm

• The part of the initial viable solution space containing a workable integer solution to the original issue is never closed off by further linear restrictions.

• The existing non-integer optimum solution to the linear programming problem is eliminated by each fresh set of additional restrictions (hyper planes).

Steps of Gomory's Algorithm

Step 01

Use the simplex approach to solve the given LP issue, ignoring the integer restrictions.

Step 02

The present solution is the best one if every variable in the final simplex table's XB column is a non-negative integer. If not, move on to the next action.

Step 03

In the event that a few of the fundamental variables lack non-negative integer values, Gomory's linear constraints (cut) are generated. These additional linear constraints are then added to the bottom of the optimum simple table, rendering the solution unfeasible.

Step 04

If the optimal solution found using the dual simplex approach is integer, the new issue is then solved which is the necessary one; if not, repeat Step 3 until all of the fundamental variables have non-negative integer values.

Eg. 8.2.1

Crack the next IPP via Gomory's Cutting plane method, Max $Z = x1 + x2$,s.t.the circumstances$3x1 + 2x2 \leq 5, x2 \leq 2$ and $x1, x2 \geq 0$ and integer.

Solution:-

Step:-01

Let's use the simplex approach to choose the best solution, omitting the integer requirements.

Simple method

Adding slack or excess variables is the first step in converting the inequality constraints to equality constraints.

$$3x1 + 2x2 \leq 5, Add\ slack\ variable\ si \geq 0, we\ have\ 3x1 + 2x2 + s1 = 5,$$

$$x2 \leq 2, Add\ slack\ variable\ s2 \geq 0, we\ have\ x2 + s2 = 2,$$

So $Max\ Z = x1 + x2 + 0s1 + 0s2$

Initial Basic feasible solution

Let $x1 = 0$ and $x2$ =0 replacement in the particular problem, we have $s1 = 5, s2 = 2$ and Max z=0.

first Simple table

CB_j	C_j Basic variables	1 x_1	1 x_2	0 s_1	0 s_2	Minimum of ratios	
0	$s_1=5$	[3]	2	1	0	5/3	←
0	$s_2=2$	0	1	0	1	---	
z=0	Z_j	0	0	0	0		
	Z_j-C_j	-1	-1	0	0		
		↑					

Arrows symbolizes chosen row and column.

First row values for the new table

$\frac{5}{3}$ $\frac{3}{3}=1$ $\frac{2}{3}$ $\frac{1}{3}$ $\frac{0}{3}=0$

Second row values for the new table

$2-\frac{(5)(0)}{3}=2,\quad 0-\frac{(3)(0)}{3}=0,\quad 1-\frac{(2)(0)}{3}=1,\quad 0-\frac{(1)(0)}{3}=0,\quad 1-\frac{(0)(0)}{3}=1$

First simplex table

CB_j	C_j Basic variables	1 x_1	1 x_2	0 s_1	0 s_2	Minimum of ratios	
1	$x_1=5/3$	1	2/3	1/3	0	5/2	
0	$s_2=2$	0	[1]	0	1	2	←
z=5/3	Z_j	1	2/3	1/3	0		
	Z_j-C_j	0	-1/3	1/3	0		
			↑				

Max $Z=5/3+0+0s_1+0s_2=5/3$

x_2 the basis and s_2 leave the basis, the diagonal variable is 1.

First row values for the new table

$\frac{5}{3}-\frac{(2)(2/3)}{1}=\frac{1}{3}\quad 1-\frac{(0)(2/3)}{1}=1\quad \frac{2}{3}-\frac{(2/3)(1)}{1}=0\quad \frac{1}{3}-\frac{(2/3)(0)}{3}=\frac{1}{3}\quad 0-\frac{(2/3)(1)}{1}=\frac{-2}{3}$

Second row values for the new table

$\frac{2}{1}=2 \quad \frac{0}{1}=0 \quad \frac{1}{1}=1 \quad \frac{0}{1}=0 \quad \frac{1}{1}=1$

	C_j	1	1	0	0	
CB_j	Basic variables	x_1	x_2	s_1	s_2	Minimum of ratios
1	$x_1=1/3$	1	0	1/3	-2/3	
1	$x_2=2$	0	1	0	1	
z=7/2	Z_j	1	1	1/3	1/3	
	Z_j-C_j	0	0	1/3	1/3	

Max $Z=1/3+2+0s_1+0s_2=7/2$

$Since\ Zj-C\ \geq 0\ for\ all\ j=1{,}2{,}3{,}4$, consequently by the simplex method

The over table is optimum, and the optimum key is specified by $x1=1/3, x2=2\ and\ Max\ z=7/2$.

Step-02

In the current optimal outcome all the basic variables in the basis are not integer.

So the outcome isn't desirable.

Step-03

We take the first row to be the source row for the Gomory fractional cut since x is the sole basis variable whose value is a non-negative fraction.

$0+1/3 = x_1+0x_2+(1/3)s_1-(2/3)s_2$

$0+1/3 = (1+0)\ x_1+(0+0)\ x_2+(0+1/3)s_1+(-1+1/3)s_2$

Therefore the Gomory's fractional cut -I is written as follows

$(1/3) \leq (1/3)\ s_1+(1/3)s_2$

$(1/3)= (1/3)\ s_1+(1/3)s_2 - s_{g1}$

$(-1/3)= (-1/3)\ s_1-(1/3)s_2 + s_{g1}$

(ADD surplus variable S_{g1})

1st Dual Simples Table

CB_j	C_j	1	1	0	0	0	
	Basic variables	x_1	x_2	s_1	s_2	s_{g1}	
1	$x_1=1/3$	1	0	1/3	-2/3	0	
1	$x_2=2$	0	1	0	1	0	
0	$s_{g1}=-1/3$	0	0	[-1/3]	-1/3	1	←
z=7/2	Z_j	1	1	1/3	1/3	0	
	$C_j - Z_j$	0	0	-1/3	-1/3	0	
	Ratio	-	-	1	1	-	
				↑			

$Ratio = (Cj - Z))/y3), calculate\ for\ all\ y_{3j} < 0.$

The preceding table is not optimal since the numbers in the ratio row are positive. The row containing Gomory's slack variable and the column with the lowest positive ratio value must always be chosen. The diagonal element is -1/3, Sg1 exits the basis, and Si enters the basis.

First row of next table

$$1/3 - \frac{(-1/3)(1/3)}{(-1/3)} = 0 \quad 1 - \frac{(0)(1/3)}{(-1/3)} = 1 \quad 1/3 - \frac{(-1/3)(1/3)}{(-1/3)} = 0 \quad -2/3 - \frac{(-1/3)(1/3)}{(-1/3)} = -1$$

$$0 - \frac{(1/3)(1)}{(-1/3)} = 1$$

Second row of new table

$$2 - \frac{(0)(1/3)}{(-1/3)} = 2 \quad 0 - \frac{(0)(0)}{(-1/3)} = 0 \quad 1 - \frac{(0)(0)}{(-1/3)} = 1 \quad 1 - \frac{(0)(-1/3)}{(-1/3)} = 1 \quad 0 - \frac{(0)(1)}{(-1/3)} = 0$$

Last row of next table

$$\frac{(-1/3)}{(-1/3)} = 1 \quad \frac{0}{(-1/3)} = 0 \quad \frac{0}{(-1/3)} = 0 \quad \frac{(-1/3)}{(-1/3)} = 1 \quad \frac{(-1/3)}{(-1/3)} = 1 \quad \frac{1}{(-1/3)} = -3$$

Second Gomory's cut table

	C_j	1	1	0	0	0
CB_j	Basic variables	x_1	x_2	s_1	s_2	S_{g1}
1	$x_1=0$	1	0	0	-1	1
1	$x_2=2$	0	1	0	1	0
0	$s_1=1$	0	0	1	1	-3
z=2	Z_j	1	1	0	0	1
	C_j -Z_j	0	0	0	0	-1

Since $Cj - Zj \geq 0$ for each $j = 1, 2, 3, 4, 5, 6$.So the table is optimum.
therefore the optimum integer outcome is $x1 = 0, x2 = 2, and\ Max\ Z = 2$.

Mixed Integer Programming Algorithm

Step-01

Use the simplex approach to solve the given LP issue, disregarding the variables' integer requirements.

Step -02

The present method is ideal if all of the integer-restricted fundamental variables have integer values. If not, move to step 3.

Step:-03

prefer a row r matching to a basic variable xr that has the maximum fractional value f, and make a cutting plane as underneath

$$s_g = -f_r + \sum a_{rj} x_j + \left(\frac{f_r}{f_r - 1}\right) \sum a_{rj} x_j \text{ where } 0<f_r<1.$$

insert this cutting plane at the base of the optimum simplex table

Step:-04

Proceed to step 02 after utilizing the dual simplex approach to determine the new optimum solution. Until all of the limited variables are integers, the process is repeated.

8.3 Branch and Bound Algorithms cutting plan algorithm

Branch and Bound is a widely used optimization technique that systematically searches for the best solution to a problem by exploring the entire solution space. It's commonly used in combinatorial optimization problems such as the Cutting Stock Problem, where you want to minimize material waste by cutting pieces from larger sheets.

Prodecure of Branch and Bound Method

In the context of the Cutting Stock Problem, the algorithm typically works as follows:

1. Initialization: Start with an initial solution. For the Cutting Stock Problem, this could involve taking all the pieces to be cut and arranging them in a way that minimizes waste. This could be a simple solution like placing each piece on a separate sheet.
2. Branching: At each step, the algorithm selects one variable (or decision) to branch on. In the Cutting Stock Problem, this could involve selecting a piece and deciding how to cut it from the current sheet. Each branch represents a possible decision or choice.
3. Bounding: Calculate a lower bound for each branch. This is an estimate of the minimum possible value that the objective function can take if we follow that branch. For the Cutting Stock Problem, this might involve calculating the remaining space on the sheet after making a cut.
4. Pruning: If the lower bound of a branch exceeds the current best solution, we prune that branch. This means we don't need to explore it further because it won't lead to a better solution than the ones we already have.
5. Optimization: Repeat steps 2-4 until all branches have been explored or pruned. At each step, update the current best solution if a better one is found.
6. Termination: Stop when all branches have been explored or pruned, and return the best solution found.

Eg. 8.3.1

Crack the next all IPP via branch& bound method,

$$Max\ Z = 2x1 + 3x2,$$

S.t.the conditions

$$6x1 + 5x2 \leq 25,$$

$$x1 + 3x2 < 10$$

$$and\ \ x1, x2\ \geq 0\ and\ integers$$

Solution:-

Step:-01

calming the integer form, Let us discover the optimal result to the above LP problem via graphical method

Graphical Method

Let us draw the line $6x_1+4x_2=25$-----(1)

Let $x_1=0$, $6(0)+5x_2=25$ $\Rightarrow x_2=25/4$ i.e (0,5)

Let $x_2=0$, $6x_1+5(0)=25$ $\Rightarrow x_1=25/6$ i.e(4.16,0)

Let us draw the line $x_1+3x_2=10$------(2)

Let $x_1=0$, $(0)+3x_2=10$ $\Rightarrow x_2=10/3$ i.e (0,3.33)

Let $x_2=0$, $x_1+3(0)=10$ $\Rightarrow x_1=10$ i.e(10,0)

To find the intersection point

(1)x1=> $6x_1+5x_2=25$

(2)x6=> $6x_1+18x_2=60$

(subtract)________

$-13x_2=-35$ $\Rightarrow x_2=35/13$

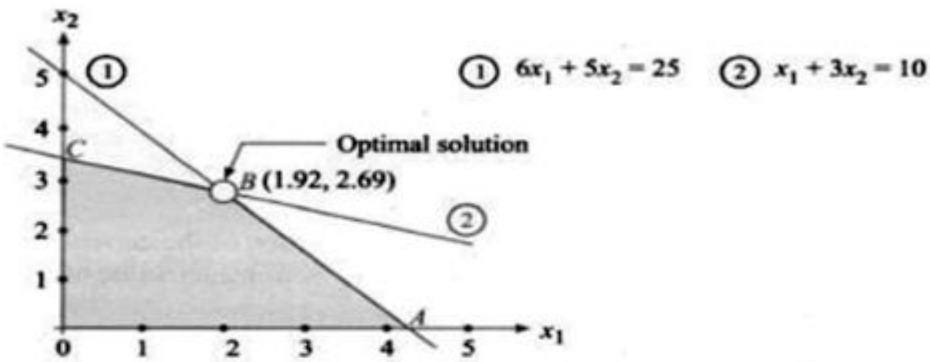
$$Sub\ x2\ values\ in\ equation\ (2), we\ have\ x1 = 25/13$$

i.e. the intersection point is (1.92, 2.69)

$$At\ (4.16,0) \quad Max\ Z = 2(4.16) + 3(0) = 8.32$$

$$At\ (0,3.33) \quad Max\ Z = 2(0) + 3(3.33) = 9.99$$

$$At\ (1.92,2.69) \quad Max\ Z = 2(1.92) + 3(2.69) = 11.91$$



The optimum result is $x1 = 1.92, x2 = 2.69\ and\ Max\ Z1 = 11.91$.

Let $Zu = 11.91$ be the initial upper bound.

The feasible result by rounding off the result provide the initial lower bound $ZL = 11\ (rounding\ off\ x1 = 1, x2 = 3)$. The optimal result lies between these twice bounds.

Step:-02 (Branching step)

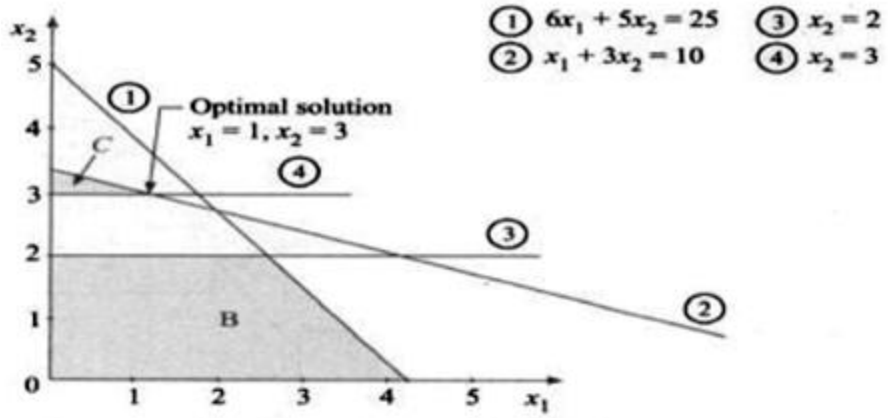
Where both $X1\ and\ X2$aren't integers, so we can choose variable for branching with maximum fractional value, x_2 has more fractional value compare to x_1.

Crack the x_2 for branching, split the specified problem into twice sub-problem A and B to remove the fractional part of $x2 = 2.69$,

The new constraints to be added are $x2 \leq 2$ *and* $x2 \geq 3$.

LP sub problem B	LP sub problem C
Max $Z=2x_1+3x_2$, Subject to the conditions $6x_1+5x_2 \leq 25$, $x_1+3x_2 \leq 10$ $x_2 \leq 2$ and $x_1, x_2 \geq 0$ and integers *Draw the line x_2=2-------(3)* It is the line parallel to x_1 axis and passes through (0,2)	Max $Z=2x_1+3x_2$, Subject to the conditions $6x_1+5x_2 \leq 25$, $x_1+3x_2 \leq 10$ $x_2 \geq 3$ and $x_1, x_2 \geq 0$ and integers *Draw the line x_2=3--------(4)* It is a line parallel to x_1 axis and passes through (0,3)

The corresponding graph is given by



The solution to the sub problem B is given by equation (1) and (3) Sub x_2=2 in equation (1), we have $6x_1+5(2)=25$ =>$x_1=5/2$ i.e (2.5, 2) and *Max $Z_2=2(2.5)+3(2)=11$*	The solution to the sub problem C is given by equation (2) and (4) Sub x_2=3 in equation (2), we have $x_1+3(3)=10$ =>$x_1=1$ i.e (1, 3) and *Max $Z_3=2(1)+3(3)=11$*

Step:-03 (Bound Step)

Here the greatest integer answer is $x_1 = 1, x2 = 3$ and $Max\ Z3 = 11$

$$ZL = 11$$

Step:-04(Fathoming step)

Both the sub-problems B and C give same Z.

In the sub-problem C both the variables $x1$ and $x2$ are integer, so there is no require to branch the sub-problem C.

$$Zu = 11.$$

Since $ZL \leq Z3 < ZU$ $(11 \leq 11 < 11.91)$, $t$herefore the new upper bound is

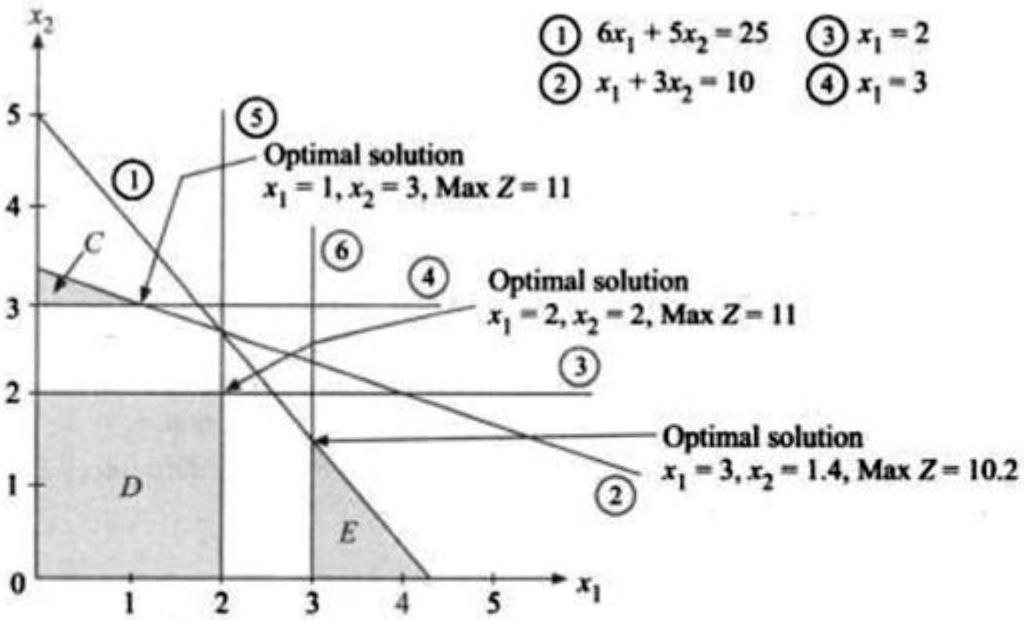
In the sub-problem B, x1 is not a fractional and since $ZL \leq\ Z2(11 \leq 11)$.

Step:-05 (Branch step)

Let us split the sub-problem B into two sub-problems that is D and E by take$x1 = 2.5$.

The new constraints to be added are $x1 \leq 2$ and $x1 \geq 23$.

LP sub problem D	LP sub problem E
Max $Z=2x_1+3x_2$,	**Max $Z=2x_1+3x_2$,**
Subject to the conditions	**Subject to the conditions**
$6x_1+5x_2 \leq 25$,	**$6x_1+5x_2 \leq 25$,**
$x_1+3x_2 \leq 10$	**$x_1+3x_2 \leq 10$**
$x_2 \leq 2$	$x_2 \leq 2$
$x_1 \leq 2$	$x_1 \geq 3$
and $x_1, x_2 \geq 0$ and integers	**and $x_1, x_2 \geq 0$ and integers**
Draw the line x_1=2-------(5)	*Draw the line x_1=3------(6)*
It is the line parallel to x_2 axis and passes through (2,0)	**It is the line parallel to x_2 axis and passes through (3,0)**



The solution to the sub problem D is given by equation (3) and (5) i.e x_1=2 and x_2=2 *Max Z_4=2(2)+3(2)=10*	The solution to the sub problem E is given by equation (1) and (6) Sub x_1=3 in equation (1), we have 6(3)+5x_2=25 =>x_2=1.4 i.e (3, 1.4) and *Max Z_5=2(3)+3(1.4)=10.2*

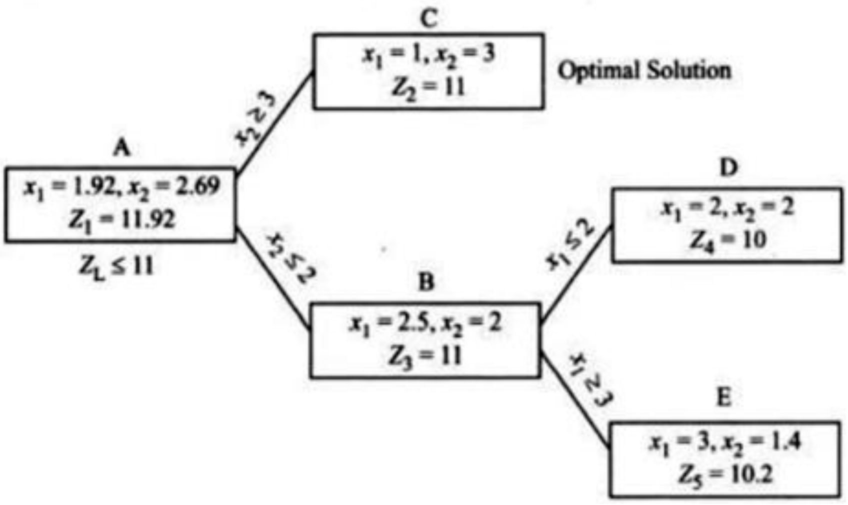
Step:-05 (Bound step)

The best integer results give the lower bound.

Here the top integer result is $x1 = 2, x2 = 2 \; and \; Max \; Z4 = 10$

Step:-06(Bound step)

The LP sub-problem D has an integer feasible solution, but Z4< ZL (10<11), and the lower bound's value (ZL=11) stays intact. Consequently, additional branching is not contemplated for the sub-problem D. Sub-problem E cannot be further branched using variable x2 since its answer, x2, is not an integer. However, as Z5 the ideal answer is thus the integer solution that matches the present lower bound. Therefore, the integer optimum solution, which corresponds to sub-problem C, is Max Z=11, x1=1, and x2= 3. The following enumeration tree may be used to illustrate the full branch and bound operation. Every node is a smaller issue.



8.4 Summary

- Optimization of a linear OF under linear constraints is the main objective of integer programming, which additionally takes into account the requirement that some or all of the decision variables be integers.
- Numerous strategies are commonly employed to address problems involving integer programming, including heuristic approaches like greedy algorithms, metaheuristics, and constraint programming, as well as accurate approaches like branch and cut and branch and bound.
- Numerous industries where discrete decisions must be made, such as engineering, finance, telecommunications, logistics, supply chain management, production planning, scheduling, and telephony, use integer programming.

8.5 Keywords

- Optimization
- Integer programming
- Branch and bound.

8.6 Self Assessment Question

1. Name one exact method used to solve Integer Programming problems.
2. How does sensitivity analysis contribute to understanding solutions in Integer Programming?
3. Provide one real-world application where Integer Programming is commonly used.
4. What role does optimization software play in solving Integer Programming problems?

5. How does Integer Programming integrate with other optimization techniques?
6. What are the key challenges associated with solving NP-hard Integer Programming problems?
7. Briefly explain the concept of feasibility in Integer Programming.

8.7 Case Study

Imagine you are a logistics manager for a distribution company responsible for optimizing delivery routes for a fleet of vehicles. The company needs to deliver goods to multiple customers located in different cities while minimizing transportation costs and ensuring timely deliveries. Each vehicle has a limited capacity, and certain customers have specific time windows during which deliveries must be made. Formulate an Integer Programming model to address this optimization problem and outline the steps you would take to solve it. Additionally, discuss the potential challenges and considerations involved in implementing the optimal solution in a real-world logistics operation.

8.8 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.

UNIT -9

Dynamic programming

Learning Objective

- Developing problem-solving skills by learning how to identify problems that can be solved using dynamic programming techniques.
- Students should grasp the fundamental concept of dynamic programming, which involves breaking down complex problems into simpler sub-problems and solving them recursively.
- They should understand the principles of optimal substructure and overlapping sub-problems that underlie dynamic programming.
- Learning optimization techniques to improve the efficiency of dynamic programming algorithms, such as reducing redundant calculations and optimizing space usage.

Structure

9.1 Introduction

Dynamic programming is another popular approach for solving optimization problems like the Cutting Stock Problem. It involves breaking down a problem into simpler sub-problems and explaining each sub-problem only once, accumulating the outcomes to keep away from redundant computations. This approach is mainly useful when the problem can be separated into overlie sub-problems, and the optimal solution can be efficiently constructed from the solutions of the sub-problems.

In the context of the Cutting Stock Problem, dynamic programming can be used to find the optimal cutting plan by considering the problem recursively. Here's how it could work:

1. **Define the Sub-problem**: Break down the original problem into smaller sub-problems. In the Cutting Stock Problem, a sub-problem might involve deciding how to cut a subset of pieces from a given sheet of material.
2. **Formulate the Recurrence Relation**: Define a recursive relationship between the original problem and its sub-problems. This relationship should express how the optimal solution to the original problem depends on the solutions to its sub-problems.
3. **Memoization or Tabulation**: Implement the dynamic programming solution using either memoization (top-down approach) or tabulation (bottom-up approach).
    - **Memoization**: Store the results of solved sub-problems in a data structure (e.g., a table or a dictionary) to avoid redundant computations.
    - **Tabulation**: Build up the solutions to sub-problems iteratively, starting from the smallest sub-problems and gradually solving larger ones.
4. **Reconstruct the Solution**: Once all sub-problems are solved, reconstruct the optimal solution to the original problem using the results stored during the dynamic programming process.

9.2 Bellman principle of optimality

The Bellman Principle of Optimality, named after mathematician Richard Bellman, is a fundamental concept in dynamic programming. It status that an optimal policy has the assets that, whatever the first state and first decision are, the remain decisions must comprise an optimal policy with observe to the state resultant from the first decision.

In simpler terms, it suggests that if you have an optimal outcome to a larger problem, you can break it down into smaller sub-problems, and the solutions to these sub-problems must also be optimal. This principle is crucial in dynamic programming because it permits us to crack complex problems efficiently by recursively breaking them down into simpler sub-problems.

Mathematically, the Bellman Principle of Optimality can be expressed as follows:

Let π^* be an optimal policy for a given problem. Then, for any state s and any decision a that leads to a state s', the value of the optimal solution from state s onward must satisfy:

$$V^*(s) = \max_a [R(s,a) + \gamma V^*(s')]$$

where:

- $V^*(s)$ is the optimal value function, showing the maximum estimated cumulative reward achievable from state s onward.
- R(s, a) is the immediate reward obtained by taking action a in state s.
- γ is the discount factor that trades off immediate and future rewards.
- s' is the status resulting from taking action a in state s.

This equation essentially states that the optimal value function $V^*(s)$ for a given state s is the maximum expected immediate reward R(s, a) plus the expected optimal value function $V^*(s')$ of the resulting state s', averaged over all possible actions a that can be taken in state s.

By applying the Bellman Principle of Optimality, we can construct efficient dynamic programming algorithms such as value iteration and policy iteration to find optimal solutions to problems with overlapping sub-problems.

9.3 Solution of problems with finite number stages

(i) Explain via forward recursion procedure

Optimum outcome for Stage '1'

Only beginning node 1 may access the final nodes in stage one, which are 2, 3, and 4. The shortest path between nodes 1 and 2 is 7, the shortest path between nodes 1 and 3 is 8, and the short path between nodes 1 and 4 is 5.

Optimum resolution for Step '2'

The last nodes in stage one are 5, 6. In this case, nodes 2, 3, and 4 can all reach node 5, and nodes 2, 3, and 4 can reach node 6. According to the following, the shortest distances between nodes 1 and node 5 are 7+12=19, 8+8=16, and 5+7=12.

g routes 1-2-5, 1-3-5, 1-4-5. Here the minimum digit is 12 corresponding to the routes 1-4-5

The short path between node 1 and node 6 is 8+9=17, 5+13=18, which corresponds to routes 1-3-6 and 1-4-6. The routes 1-3-6 correspond to a minimum value of 17.
At Stage '3', the **Optimum** number for stage one ending nodes is 7. Here, three nodes—5, 6, and 7—can connect to node 7. The shortest paths, 1-4-5-7 and 1-3-6-7, are 12+9=21 and 17+6=23, respectively, from node 1 to node 7. In this case, the routes 1-4-5-7 equate to a minimum value of 21. As a result, stage 3 demonstrates that 21 miles is the shortest path between nodes 1 and 7.

(ii) Use backward word recursion to solve. i=1,2,3 since there are three phases, and j=1,2,3,4,5,6,7 because there are seven states. Phase every stage of the problem where a choice needs to be taken.

Eg. 9.3.1

Consider a company that sells a product over a period of, say, 10 days. Each day, the company must decide on the price to set for the product. The goal is to maximize total revenue over the 10-day period. This problem can be solved using dynamic programming, where each stage corresponds to a day, and the decision at each stage is the price set for that day.

Eg. 9.3.2

A retailer must decide how much inventory to order each week over a finite planning horizon, say 8 weeks. The goal is to minimize costs while ensuring that demand is met. This problem can be formulated as a dynamic programming problem, where each stage corresponds to a week, and the decision at each stage is the order quantity for that week.

Eg. 9.3.3

An investor has a sum of money to invest over a period of, say, 5 years. At the beginning of each year, the investor must decide how to allocate the money among different investment options (e.g., stocks, bonds, real estate). The goal is to maximize the total return on investment over the 5-year period. This problem can be solved using dynamic programming, where each stage corresponds to a year, and the decision at each stage is the allocation of funds to different investment options.

Eg. 9.3.4

Use dynamic programming method to solve the following problem

$$\text{Min } Z = y_1^2 + y_2^2 + y_3^2$$

S.t. the constraints

$y_1+y_2+y_3 > 15$ and $y_1, y_2, y_3 \geq 0$

Solution:-

as there are three choice variables $y1, y2, y3$.

so the particular problem is to be resolve in 3 stages.

$$Let\ S3 = y1 + y2 + y3 - -(1)$$

$$S2 = y1 + y2\ (= S3 - y3) \qquad - -(2)$$

$$S1 = y1\ (= S2 - y2) - -(3)$$

The functional relation is given by

$$f_1(s_1) = \min_{0 \le y_1 \le s_1} y_1^2 \qquad \text{--------}(4)$$

$$f_2(s_2) = \min_{0 \le y_2 \le s_2} y_1^2 + y_2^2 = \min_{0 \le y_1, y_2 \le s_2} \{f_1(s_1) + y_2^2\} \qquad \text{--------}(5)$$

$$f_3(s_3) = \min_{0 \le y_3 \le s_3} y_1^2 + y_2^2 + y_3^2 = \min_{0 \le y_3 \le s_3} \{f_2(s_2) + y_3^2\} \qquad \text{--------}(6)$$

To find the minimum value of $f_2(S_2)$

$f_2(s_2)=y_1^2+y_2^2$

$f_2(s_2)=(s_2-y_2)^2+y_2^2$ (equation (3))

Differentiate with respect to 'y' and assume equal to zero, we have

$2(s_2-y_2)(0-1)+2y_2=0$

$=> y_2=s_2/2$ -------(7)

Sub the value of y_2 in $f_2(S_2)$, we have

$f_2(s_2)= y_1^2+y_2^2= y_1^2+(s_2/2)^2$ (by equation (5))

$f_2(s_2)= (s_2-y_2)^2+(s_2/2)^2$ (by equation (3))

$f_2(s_2)= (s_2- s_2/2)^2+(s_2/2)^2$

$f_2(s_2)= (s_2/2)^2+(s_2/2)^2=2s_2^2/4$

$f_2(s_2)=s_2^2/2.$ --------(8)

To find the minimum value of $f_3(S_3)$

$f_3(s_3)=f_2(s_2)+y_3^2$ (by equation (3))

$f_3(s_3)= s_2^2/2+y_3^2$ (by equation (8))

$f_3(s_3)= (s_3-y_3)^2/2+y_3^2$ (by equation (2))

Differentiate with respect to 'y_3' and equate it to zero, we have

$2(s_3-y_3)(0-1)/2+2y_3=0$

$2(15-y_3)(0-1)/2+2y_3=0$ (Sub $s_3=15$, data in given problem)

$(15-y_3)(-1)+2y_3=0$

$-15+y_3+2y_3=0$

$\Rightarrow y_3=5$ --------(9)

Substitute y_3 values in $f_3(S_3)$

$f_3(s_3)=(15-5)^2/2+5^2=100/2+25=75$

Optimal solution

Since $s_3=15$

$(2)\Rightarrow s_2=s_3-y_3=15-5=10$ $f_2(s_2)=s_2^2/2=(10)^2/2$

$(3)\Rightarrow s_1=s_2-y_2=10-5=5$ $(y_2=s_2/2)$

$(3)\Rightarrow y_1^*=s_1=5$

$(7)\Rightarrow y_2^*=s_2/2=10/2=5$

$(9)\Rightarrow y_3^*=5$

Hence the optimal policy is (5, 5, 5) with $f_3(15)=75$.

9.4 Summary

- Optimization problems can be solved using dynamic programming (DP), a method of problem-solving that divides complex problems into smaller ones. Each sub-problem is only solved once, and the solutions are stored to prevent duplicate computations.
- Based on their features, dynamic programming problems can be divided into several groups, including matrix chain multiplication, 0/1 knapsack, longest common subsequence, and shortest path. Different approaches to dynamic programming can be needed for each category.

9.5 Keywords

- Dynamic programming problems
- Optimization problems
- Complexity analysis
- Bellman principle of optimality

9.6 Self Assessment

1. What is the main principle behind Dynamic Programming?
2. Can you name one key characteristic of problems that make them suitable for Dynamic Programming solutions?
3. What are the two main approaches to implementing Dynamic Programming algorithms?
4. How does Dynamic Programming differ from other problem-solving techniques like Greedy Algorithms?
5. Provide an Eg. of a problem commonly solved using Dynamic Programming.
6. What does the term "optimal substructure" mean in the context of Dynamic Programming?
7. How does Dynamic Programming address the issue of overlapping sub-problems?
8. Describe one method used to optimize Dynamic Programming algorithms.
9. What is the time complexity analysis typically performed for Dynamic Programming algorithms?
10. How does understanding Dynamic Programming contribute to problem-solving in various domains?

9.7 Case Study

Imagine you are a software engineer tasked with developing an algorithm to optimize the selection of projects in a project portfolio. Each project has a certain profit value and duration, and there are constraints on the total duration and budget of the portfolio. Formulate this problem as a Dynamic Programming (DP) model and outline the steps you would take to solve it. Discuss the key considerations in designing the DP algorithm, including identifying sub-problems, defining the state space, and implementing the memoization or tabulation approach. Additionally, explain how your DP solution contributes to maximizing the overall profit while satisfying the project duration and budget constraints.

9.8 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.

UNIT -10
Duality

Learning Objective

Students should grasp the concepts of primal and dual problems and how they relate to each other. This involves understanding the formulation of both types of problems, their constraints, objectives, and decision variables.

- Understanding the optimality conditions associated with both primal and dual problems is crucial.
- Duality also plays a significant role in sensitivity analysis. Students should learn how changes in problem parameters affect the optimal solutions of both primal and dual problems, and how duality facilitates sensitivity analysis in optimization.

Structure

10.1 Introduction of Duality

Each LPP known as the primal has a corresponding LPP known as the dual. One of the issues is primary, while the other is dual. The knowledge on the optimum solution of the other problem may be found in the optimal solution of either problem.

Let the primal problem be

$$\text{Max } Z_x = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

S.t. restrictions

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2$$

.

.

.

$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_n$

$x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0$

The corresponding dual is defined as

$\text{Min } Z_w = b_1w_1 + b_2w_2 + \ldots + b_mw_m$

S.t. restrictions

$a_{11}w_1 + a_{21}w_2 + \ldots + a_{m1}w_m \geq c_1$

$a_{12}w_1 + a_{22}w_2 + \ldots + a_{m2}w_m \geq c_2$

.

.

.

$a_{1n}w_1 + a_{2n}w_2 + \ldots\ldots\ldots\ldots + a_{mn}w_m \geq c_n$

$w_1, w_2, \ldots, w_m \geq 0$

Matrix Notation

Primal

Max $Z_x = CX$

Subject to

$AX \leq b$ and $X \geq 0$

Dual

$\text{Min } Z_w = b^T W$

Subject to

$A^T W \geq C^T$ and $W \geq 0$

10.2 Advantages and Application of Duality

Duality is a fundamental concept in optimization theory that establishes a relationship between two optimization problems, known as the primal and the dual. The primal problem seeks to maximize/minimize an OF subject to certain constraints, while the dual problem is derived from the primal and involves maximizing (or minimizing) a different objective function under certain constraints. Here are some advantages and applications of duality:

Advantages:

1. **Insight into the Problem**: Duality provides a different perspective on the original problem, which can often offer insights into its structure and properties. By examining the dual problem, one can gain a deeper understanding of the primal problem and its solution.

2. **Sensitivity Analysis**: Duality allows for sensitivity analysis, which involves examining how changes in the parameters of the problem affect the optimal solution. By analyzing the dual problem, one can determine how changes in the constraints or objective function coefficients of the primal problem impact its optimal solution.

3. **Algorithm Design**: Duality can be used to design efficient algorithms for solving optimization problems. For example, the dual problem may have a simpler structure than the primal problem, making it easier to solve. Additionally, certain algorithms, such as interior-point methods, exploit the duality relationship to efficiently solve both the primal and dual problems simultaneously.

4. **Bounding Solutions**: Duality provides bounds on the optimal value of the primal and dual problems. These bounds can be useful for verifying the optimality of a solution or for determining how close the primal and dual solutions are to each other.

5. **Economic Interpretation**: In some optimization problems arising in economics, duality has a natural economic interpretation. For example, in linear programming, the dual variables can represent prices or shadow prices, providing insights into the economic implications of the optimal solution.

Applications:

1. **Linear Programming**: Duality plays a central role in linear programming, where it is used to derive important results such as the weak and strong duality theorems. The dual problem in linear programming often has economic interpretations, making it useful for modeling and analyzing various economic systems.

2. **Convex Optimization**: Duality is also important in convex optimization, where it helps establish relationships between primal and dual problems for a wide range of convex

optimization problems. This relationship is exploited in algorithm design and theoretical analysis.

3. Game Theory: Duality is used in game theory to analyze and solve various types of games, including zero-sum games and non-zero-sum games. The concept of duality helps identify equilibrium strategies and analyze the payoffs of different players.

4. **Scheduling and Resource Allocation**: Duality is applied in scheduling and resource allocation problems to optimize the allocation of resources subject to various constraints. By formulating the primal and dual problems, one can efficiently allocate resources while satisfying operational constraints.

Procedure for a Standard Primal Form

Step- 1- modify the OF to Maximize

Step- 2 - If the constraints have an inequality then multiply both sides by -1 and renovate the inequality to '≤'.

Step- 3-If the constraint has an '=' then substitute it by twice constraints concerning the inequality obtainable in opposite commands. For Eg. $x1 + 2x2 = 4\ is$write as

$$x1 + 2x2 \leq 4$$

$$x + 2x2 \geq 24\ (using\ step2) \rightarrow x - 2x2 \leq -4$$

Step- 4 - each unrestricted variable is replace by the dissimilarity of two non-negative variables.

Step- 5 - We obtain the standard primal form of the given LPP in which.

- every constraints have '≤, where the OF is of maximize.
- every constraints have '≥', where the OF is of minimize.

Eg.10.2.1

Write the dual of the given problems

Min $Z_x=2x_2+5x_3$

Subject to

$x_1+x_2 \geq 2$

$2x_1+x_2+6x_3 \leq 6$

$x_1 - x_2 +3x_3 = 4$

$x_1, x_2, x_3 \geq 0$

Solution

Primal

$\text{Max } Z_x' = -2x_2 - 5x_3$

Subject to

$-x_1 - x_2 \leq -2$

$2x_1 + x_2 + 6x_3 \leq 6$

$x_1 - x_2 + 3x_3 \leq 4$

$-x_1 + x_2 - 3x_3 \leq -4$

$x_1, x_2, x_3 \geq 0$

Dual

$\text{Min } Z_w = -2w_1 + 6w_2 + 4w_3 - 4w_4$

Subject to

$-w_1 + 2w_2 + w_3 - w_4 \geq 0$

$-w_1 + w_2 - w_3 + w_4 \geq -2$

$6w_2 + 3w_3 - 3w_4 \geq -5$

$w_1, w_2, w_3, w_4 \geq 0$

Eg.10.2.2

$\text{Min } Z_x = 3x_1 - 2x_2 + 4x_3$

Subject to

$3x_1 + 5x_2 + 4x_3 \geq 7$

$6x_1 + x_2 + 3x_3 \geq 4$

$7x_1 - 2x_2 - x_3 \geq 10$

$x_1 - 2x_2 + 5x_3 \geq 3$

$4x_1 + 7x_2 - 2x_3 \geq 2$

$x_1, x_2, x_3 \geq 0$

Solution

Primal

$\text{Max } Z_x' = -3x_1 + 2x_2 - 4x_3$

Subject to

$-3x_1 - 5x_2 - 4x_3 \leq -7$

$-6x_1 - x_2 - 3x_3 \leq -4$

$-7x_1 + 2x_2 + x_3 \leq -10$

$-x_1 + 2x_2 - 5x_3 \leq -3$

$-4x_1 - 7x_2 + 2x_3 \leq -2$

$x_1, x_2, x_3 \geq 0$

Dual

Min $Z_w = -7w_1 - 4w_2 - 10w_3 - 3w_4 - 2w_5$

Subject to

$-3w_1 - 6w_2 - 7w_3 - w_4 - 4w_5 \geq -3$

$-5w_1 - w_2 + 2w_3 + 2w_4 - 7w_5 \geq 2$

$-4w_1 - 3w_2 + w_3 - 5w_4 + 2w_5 \geq -4$

$w_1, w_2, w_3, w_4, w_5 \geq 0$

Duality and Simplex Method

Eg. 10.2.3

Crack the particular primal problem via simplex method. Hence inscribe the result of its dual

Max $Z = 30x_1 + 23x_2 + 29x_3$

Subject to

$6x_1 + 5x_2 + 3x_3 \leq 26$

$4x_1 + 2x_2 + 6x_3 \leq 7$

$x_1 \geq 0, x_2 \geq 0$

Solution

Primal form

Max $Z = 30x_1 + 23x_2 + 29x_3$

Subject to

$6x_1 + 5x_2 + 3x_3 \leq 26$

$4x_1 + 2x_2 + 6x_3 \leq 7$

$x_1 \geq 0, x_2 \geq 0$

SLPP

Max $Z = 30x_1 + 23x_2 + 29x_3 + 0s_1 + 0s_2$

Subject to

$6x_1 + 5x_2 + 3x_3 + s_1 = 26$

$4x_1 + 2x_2 + 6x_3 + s_2 = 7$

$x_1, x_2, s_1, s_2 \geq 0$

	$C_j \rightarrow$		30	23	29	0	0	
Basic Variables	C_B	X_B	X_1	X_2	X_3	S_1	S_2	Min Ratio X_B / X_K
s_1	0	26	6	5	3	1	0	26/6
s_2	0	7	4	2	6	0	1	7/4→
	Z = 0		↑ -30	-23	-29	0	0	←Δ_j
s_1	0	31/2	0	2	-6	1	-3/2	31/4
x_1	30	7/4	1	1/2	3/2	0	1/4	7/2→
	Z = 105/2		0	↑ -8	16	0	15/2	←Δ_j
s_1	0	17/2	-4	0	-12	1	-5/2	
x_2	23	7/2	2	1	3	0	1/2	
	Z =161/2		16	0	40	0	23/2	←Δ_j

$\Delta \geq 0$ so the optimal result is $Z = 161/2, x1 = 0, x2 = 7/2, x3 = 0$.

The optimal result to the dual of the given problem will be

$$ZW *= 161/2, w1 = \Delta 4, = 0, w2 = \Delta 5 = 23/2$$

In this way, we can discover the outcome to the dual without actually solve it.

10.3Dual simplex method

Computational Procedure of Dual Simplex Method

The Dual Simplex Method is an algorithm utilized to resolve LPP by iteratively moving from one feasible solution to another along improving directions. It's particularly useful when the primal problem is infeasible or unbounded, or when the dual problem is easier to solve than the primal problem. Here's a computational procedure for the Dual Simplex Method:

Step 1: Initialization

1. Formulate the Dual Problem: Start with the primal linear programming problem in standard form:

$$\text{Maximize } \mathbf{c}^T\mathbf{x} \text{ subject to } \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$$

Then, formulate its dual problem:

$$\text{Minimize } \mathbf{b}^T\mathbf{y} \text{ subject to } \mathbf{A}^T\mathbf{y} \geq \mathbf{c}, \mathbf{y} \geq \mathbf{0}$$

2. Initialize Feasible Dual Solution: Start with a feasible dual solution y satisfying the dual constraints

$$A^T y \geq c.$$

Step 2: Iteration

Repeat until an optimal solution is found or the problem is resolute to be infeasible/ unbounded:

1. Compute Reduced Costs: Compute the reduced costs

$$r = c - A^T y.$$

2. Select Entering Variable: Choose a variable with a negative reduced cost as the entering variable. This variable will enter the basis.
3. Compute Direction: Compute the direction d to move along by solving the system Ad=Ai where Ai is the i^{th} column of A corresponding to the selected entering variable.
4. Check for Infeasibility: If there is no positive component in d, the problem is infeasible. Terminate the algorithm.
5. Select Exiting Variable: Choose a variable to leave the basis. This is typically the variable that reaches zero value first along the direction d.
6. Update Basis: Update the basis by replacing the leaving variable with the entering variable.
7. Update Dual Solution: Update the dual solution y by adding the appropriate multiple of d to it.

Step 3: Termination

If the reduced costs r are all +ve, the current solution is optimal. or else, go back to Step 2 and continue the iteration.

Eg. 10.3.1

Minimize $Z = 2x_1 + x_2$

Subject to

$3x_1 + x_2 \geq 3$

$4x_1 + 3x_2 \geq 6$

$x_1 + 2x2 \geq 3$

and X10, X2 ≥ 0

Solution

Step 1 - modify the particular problem

Maximize Z2 – = 'x$_1$ – x$_2$

Subject to

$-3x_1 - x_2 \le -3$

$-4x_1 - 3x_2 \le -6$

$-x_1 \ - 2x_2 \le -3$

$x_1, x_2 \ge 0$

Step 2 - Adding slack variables to each constraint

Maximize Z2 – = 'x$_1$ – x$_2$

Subject to

$-3x_1 - x_2 + s_1 = -3$

$-4x_1 - 3x_2 + s_2 = -6$

$-x_1 - 2x_2 + s_3 = -3$

$x_1, x_2, s_1, s_2, s_3 \ge 0$

Step 3 - Construct the simplex table

	$C_j \rightarrow$		-2	-1	0	0	0	
Basic variables	C_B	X_B	X_1	X_2	S_1	S_2	S_3	
s_1	0	-3	-3	-1	1	0	0	
s_2	0	-6	-4	-3	0	1	0	→ outgoing
s_3	0	-3	-1	-2	0	0	1	
				↑				
	Z0 = '		2	1	0	0	0	$\leftarrow \Delta_j$

Step 4 - To discover the departure vector

$Min\ (-3, -6, -3) = -6$. Therefore $s2$ is departure vector

Step 5 - To find the incoming vector

$Max\ (\Delta_1/X21, \Delta_2/X22) = (2/-4, 1/-3) = -1/3$. So x_2 is inward vector

Step 6-The key element is -3. Progress to first iteration

	$C_j \rightarrow$		-2	-1	0	0	0	
Basic variables	C_B	X_B	X_1	X_2	S_1	S_2	S_3	
s_1	0	-1	-5/3	0	1	-1/3	0	→ outgoing
x_2	-1	2	4/3	1	0	-1/3	0	
s_3	0	1	5/3	0	0	-2/3	1	
			↑					
	Z2- = '		2/3	0	0	1/3	0	$\leftarrow \Delta_j$

Step 7-To find the departure vector

$Min\ (-1, 2, 1)\ =\ -1. Hence\ s,$ is departure vector

Step 8 - To find the incoming vector

$Max\ (\Delta_1/X11, \Delta_4//X14) = (-2/5, -1) = -2/5.$ So $x_1$is inward vector

Step 9-The key element is -5/3. Progress to following iteration

	$C_j \rightarrow$		-2	-1	0	0	0	
Basic variables	C_B	X_B	X_1	X_2	S_1	S_2	S_3	
x1	-2	3/5	1	0	-3/5	1/5	0	
x2	-1	6/5	0	1	4/5	-3/5	0	
s3	0	0	0	0	1	-1	1	
	Z- = 12/5		0	0	2/5	1/5	0	$\leftarrow \Delta_j$

Step-10 - $\Delta_j \geq 0$ and $X_B \geq 0$, so the optimal outcome is $Max\ Z = 12/5, and\ x1 = 3/5, x2 = 6/5.$

10.4 Summary

For a number of reasons, duality is crucial in operations research:

1. Interpretation: It offers decision-making assistance by offering economic interpretations of optimization problems, including costs, pricing, and resource allocations.
2. Optimality Conditions: Duality shows conditions that are necessary for optimal solutions to be verified and analyzed, such as complementary slackness.
3. Sensitivity Analysis: By analyzing how modifications to the problem parameters impact the best solutions for both primal and dual problems, it helps with sensitivity analysis.
4. Duality provides guidance for several solution techniques, including the interior-point method and the simplex method, which enhance computing effectiveness and problem-solving procedures.

10.5 Keywords

- Duality
- Simplex method
- Linear programming
- Dual problem

10.6 Self Assessment Question

1. What is the duality principle in operations research?
2. Define primal and dual problems in optimization.
3. How do primal and dual variables relate to each other?
4. What are the optimality conditions associated with primal and dual problems?
5. Explain the economic interpretation of primal and dual variables.

10.7 Case Study

1. Consider a manufacturing company trying to optimize its production process. Formulate both the primal and dual problems for this scenario, and discuss how the duality principle can be applied to gain insights into the cost and resource allocation aspects of production.
2. A logistics company is faced with the task of minimizing transportation costs while meeting supply and demand constraints. Formulate the primal and dual problems for this transportation scenario, and analyze how duality can help in understanding the allocation of transportation resources and the pricing of goods.

10.8 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.

UNIT -11

Game Theory

Learning Objective

- Students should grasp fundamental concepts of game theory, including players, strategies, payoffs, and equilibrium concepts such as Nash equilibrium.
- Developing skills in formulating games as mathematical models and solving them using appropriate solution concepts and techniques, such as dominance reasoning, backward induction, and mixed strategies.
- Developing strategic thinking skills by understanding how to anticipate and respond to the actions of other decision-makers in competitive or cooperative environments.

Structure

11.1 Introduction:

Game theory is a mathematical framework intended for analyzing competitive situations where the results depend on the actions of many agents, each with potentially conflicting interests. It is widely used in economics, political science, psychology, and computer science to study strategic interactions among rational decision-makers.

Key Concepts in Game Theory

1. **Players**: The decision-makers in the game. Each player aims to maximize their own payoff.
2. **Strategies**: The plan of action or policy a player tracks in a given situation. Strategies can be pure (a specific course of action) or mixed (a probabilistic combination of possible actions).
3. **Payoffs**: The reward received by a player as a result of the chosen strategies. Payoffs are often represented in a payoff matrix.
4. **Games**: The complete setting in which players, strategies, and payoffs interact. Games can be cooperative or non-cooperative, zero-sum or non-zero-sum, simultaneous or sequential.

Types of Games

1. **Cooperative vs. Non-Cooperative Games**:
    - **Cooperative Games**: Players can form binding commitments or coalitions.
    - **Non-Cooperative Games**: No binding agreements; players act independently.
2. **Zero-Sum vs. Non-Zero-Sum Games**:
    - **Zero-Sum Games**: One player's gain is exactly another player's loss.
    - **Non-Zero-Sum Games**: The total payoff to all players can vary; one player's gain isn't necessarily another's loss.
3. **Simultaneous vs. Sequential Games**:
    - **Simultaneous Games**: Players choose their actions simultaneously.
    - **Sequential Games**: Players take turns making their moves, with each player observing the previous moves.

Classic Examples

1. **Prisoner's Dilemma**:
    - Two prisoners must decide independently whether to confess or remain silent.
    - The optimal individual strategy leads to a worse collective outcome.
2. **Battle of the Sexes**:
    - A couple must decide independently on a date activity with different preferences.
    - The goal is to coordinate on a common choice despite differing individual preferences.

3. **Stag Hunt**:
    - Two hunters can either hunt a stag together or individually hunt rabbits.
    - Cooperation yields a higher payoff, but there is a risk if the other doesn't cooperate.
4. **Nash Equilibrium**:
    - A situation where no player can benefit by change their strategy while the others keep their unaffected.
    - It represents a state of mutual best responses.

Conceptual math in game theory

The subsequent the precondition needed to understand game theory

Total of profits and losses

A game is said to be a zero sum game if the total profits and losses for all players in it equal zero, precisely matching the total losses for all other players.

Kinds of games

A variety of important characteristics, the most prominent of which is the player count, may be used to categorize games. A game can therefore be classified as one-person, two-person, or n-person (where n is more than two) with unique characteristics for each group.

Single-player games

Games against nature are another name for one-player games. The player just needs to make a list of all the possibilities and select the best one when there are no opponents. Even though the game may appear more difficult when chance is incorporated, the choice is actually still very straightforward.

Consider someone choosing whether or not to carry an umbrella. There is no conscious opponent, even though this individual could make a poor choice. That is, the player may base his choice on straightforward probability because it is assumed that nature will not be affected in any way by his choice. Game theorists are not very interested in single-player games.

11.2 Two-player zero-sum game

A two-player, zero-sum game (TPZS) in which there are additional participants Games in which more than two players participate are known as (i) team sports in which there are only two teams, but each team consists of many players. (ii) A lot of individuals participate in military conflicts as stand-ins, thus it should not be shocking that a lot of military issues may also be examined as TPZS games.

Games that are not TPZS include: (i) parlor games where participants are unable to tell each other apart into two sides; (ii) poker and Monopoly games that are played by more than two players. (iii) Because there are too many players and their interests aren't entirely at odds, the majority of genuine economic "games" aren't TPZS. **Game with a positive sum** Positive sum scenarios in game theory are those where the sum of the profits and losses above zero. When resources are extended in some way and a strategy is developed to meet the requirements and goals of all parties involved, a positive sum results.

Perfect games

Perfect information games: these are those in which every player always has complete knowledge of the rules. We refer to it as "perfect games." Consider a game like chess, where every participant is always fully informed about the rules. In the game of chess, if both players make the best decisions, one of three outcomes must happen: (i) Whiteswin (has a plan that works against any black strategy), (ii) Black wins, or (iii) White and Black draw.

Imperfect games

Games with incomplete knowledge, where players never know everything there is to know about the game. It goes by the name of imperfect games as well. For instance, in a game of poker, players are unaware of every card held by every opponent.

Finite games

Games in which the number of participants is finite, the options available to each player are finite, and the game cannot continue forever. For example, the majority of parlor games, chess, checkers, and poker, have an end.

Cooperative games

According to game theory, a cooperative game (also known as a coalitional game) is one in which groups of players (referred to as "coalitions") compete with one another because cooperative behavior may be enforced externally (for example, through contract law).

Non cooperative games

They criticize cooperative games where coalitions cannot be formed or where all agreements must be self-enforcing (by means of realistic threats, for example).

Repay

Payoff refers to the effect of a certain decision (or tactic) in the game. It is also expected that the player is aware of the payoff beforehand. It is stated in terms of numerical numbers, such as dollars, market share percentage, or utility.

Pay off matrix

When participants choose their specific tactics, the payoffs in terms of gain/loss can be expressed in the form of a matrix recognized as a payoff matrix. Let A1, A2,... Am be potential player A tactics. Assume that player B's potential tactics are B1, B2,... Bn.

There are m x n possible outcomes in total, and it is believed that each player is aware of both his opponent's list of potential moves and his own. We'll suppose for the sake of convenience that player A is always winning and player B always losing. If player A choose strategy I and player B selects strategy J, then let aij-pay off represent the advantage that player A has over player B.

The pay of matrix player A is shown in a table.

Player A's strategies	Player B's strategies			
	B_1	B_2	....	B_n
A_1	a_{11}	a_{12}	....	a_{1n}
A_2	a_{21}	a_{22}	....	a_{2n}
....			....	
A_m	a_{m1}	a_{m2}	....	a_{mn}

Notes

In zero sum games, a player's gain is equivalent to a player's loss, and vice versa. For example, the payoff tables of two players would have the identical numbers, but with a dissimilar sign for each person. Thus, building a table with pay is sufficient for a single participant.

Strategy

A player's strategy is the list of all conceivable movements or courses of action that he will pursue in order to maximize any potential payoff or outcome. It is considered that the player is aware of every feasible line of action beforehand.

Different Strategies

In game theory, players often employ two kinds of strategies: mixed strategies and pure strategies.

(i) Straight forward strategy:

Pure strategy refers to a specific plan of action that the player chooses (course of action). That is to say, every player is aware of every plan that is available to him beforehand and chooses just one, regardless of the other players' tactics. The player's goal is to maximize gain or minimize loss.

(ii) Mixed strategy:

Mixed strategies are courses of action that have a fixed probability and are to be chosen at a specific time. In other words, the participants' goal in a probabilistic scenario is to maximize projected gains or minimize estimated losses by selecting a pure strategy with defined probability.

In mixed strategy, there exists a set S={p1, p2,....., pn}, where p; is the probability that the pure strategy, j, would be picked and whose sum is unity, if there are 'n' number of pure strategies of the player.

$$\text{i.e } \mathrm{p1} + \mathrm{p2} + \ldots\ldots + \mathrm{pn} = 1 \text{ and } \mathrm{p;} >= 0 \; for\ all\ j = 1,2,\ldots n.$$

Remark:

(i) We refer to a player as employing a "mixed strategy" if they arbitrarily select a pure approach. A player selects an action with certainty in a pure strategy, but in a mixed strategy, he selects a probability distribution among the possible actions.

(ii) The player is considered to have chosen pure strategy if a specific p;=1 and every other value is zero. J.

Optimal strategy

Optimal strategy is the specific tactic (or comprehensive plan) that a player uses to maximize his profits or loses while remaining unaware of the tactics of his rivals.

Value of the game

The value of the game, represented by the letter V, is the anticipated result when participants adhere to their best available strategy.

Basic assumptions of game

(i) The number of strategies that each player may employ is limited. Each player may receive a different list.

(ii) Player B tries to minimize losses while Player A tries to maximize earnings.

(iii) Before the play, each player makes their own decisions without consulting the other.

(iv) In order to prevent any player from benefiting from direct knowledge of the other player's choice, the decisions are made and stated simultaneously.

(v) Each player is aware of potential rewards for both oneself and others.

Minmax-Maxmin principle

The fundamental challenge of playing games is for each player to choose the best course of action without being aware of their opponent's plan. The study's goal is to understand how these players choose their own strategies in order to maximize their rewards. The minmax-maxmin principle is the name given to this type of decision-making criterion.

Remarks

The optimal strategy selection for both players in a pure strategy issue is determined by the Minmax-Maxmin concept.

Saddle point

A saddle (equilibrium) point is reached in the game if minmax value equals maxmin value.

Remark:

(i) The optimal approach is the one that corresponds to the saddle point. (ii) The value of the game is the amount of payoff at a saddle point. (iii) Multiple saddle points are possible in a game. (iv) Some games do not have saddle points. (v) Its name comes from the fact that, in a reward matrix that resembles a saddle, it is the minimum of a row and the maximum of a column. This matrix will be shown in a moment.

Process to find out saddle point

Step: 01

Choose the lowest element in each row of the payoff matrix and record it under the title "row minima." Next, choose the biggest piece from this group and surround it with a rectangle.

Step: 02

Choose the largest or maximum element in each column of the pay matrix and record it under the heading "Column Maximas." Next, pick the lowest element in this group and surround it with a circle.

Step: -03

Determine which elements are the same in the rectangle and the circle, then indicate where those elements should go in the matrix. This component, known as the saddle point, stands for the game's worth.

Games without saddle point

Assume that if a game has no pure strategy solution, then there is no saddle point. In these cases, in order to win the game, both players must figure out the best combinations of tactics to locate the saddle point.

By giving each strategy a probability of selection, it is possible to ascertain the best possible combination for each participant. Mixed strategy is the best plan as assessed by this method.

Fair Game

A game is considered fair if its value is zero, meaning that no participant has gained or lost. When both players' optimal tactics are pure strategies, the game is the simplest kind. This is true if and only if there is a saddle point in the pay-off matrix.

Strictly determinable

If the game's maxmin and minmax values are equal and both equal the game's value, the game is said to be rigorously determinable.

11.3 Game with saddle points

Eg. 11.3.1

For both the player find the optimal plan

	Player-B				
		I	II	III	IV
Player-A	I	-2	0	0	5
	II	4	2	1	3
	III	-4	-3	0	-2
	IV	5	3	-4	2

Solution:-

By principle of maxmin-minmax

	Player-B					Row Minimum
		I	II	III	IV	
Player-A	I	-2	0	0	5	-2
	II	4	2	1	3	1
	III	-4	-3	0	-2	-4
	IV	5	3	-4	2	-6
Column Maximum		5	3	1	5	

From the column maximum values, choose the smallest value.

namely, Minimax = 1 (circled)

Pick the highest value from the row's lowest values.

i.e., the rectangle-marked Maximin = 1.

Player A will select option II, which has a maximum payout of one.

Strategy III will be selected by Player B.

Player A will receive one unit and Player B will lose one unit since the game has a value of 1.

The game has a saddle point and is not fair since the maximin value equals the minimax value, which is 1. (Given that the game's value is non-zero) Additionally, maxmin=minimax=value of the game; as a result, the game may be determined precisely.

The saddle point in this entirely strategic game is (A-II, B-III).

Player A must choose strategy II, and Player B must choose strategy III, which are the best options for both players as determined by pure strategy.

Method for solving a 2x2 game without using a saddle point

If player A's pay-off matrix is provided by

Player B

Player A $a_{11}a_{12}$

$a_{21}a_{22}$

then the value of the game and the best strategies are calculated using the following formulas.

$$p_1 = \frac{a_{22} - a_{21}}{a_{11} + a_{22} - (a_{12} + a_{21})} \quad \& \quad p_2 = 1 - p_1$$

$$q_1 = \frac{a_{22} - a_{12}}{a_{11} + a_{22} - (a_{12} + a_{21})} \quad \& \quad q_2 = 1 - q_1$$

$$V = \frac{a_{11}a_{22} - a_{21}a_{12}}{a_{11} + a_{22} - (a_{12} + a_{21})}$$

Hence the optimal mixed strategy for player A and B given by

$$S_A = \begin{pmatrix} A_1 & A_2 \\ p_1 & p_2 \end{pmatrix} \text{ and } S_B = \begin{pmatrix} B_1 & B_2 \\ q_1 & q_2 \end{pmatrix}$$

Here, player A's p1-probability of choosing a strategy A1

p2 = likelihood that player A will select a strategy A2

Q1 = Player Probability A decide on B1 plan.

Q2 = likelihood that player A will select tactic B2

Eg. 11.3.2

A and B, the two players, match coins. Should the coins align, A will receive two units of value; if they don't align, B will receive two units of value. Ascertain the best tactics for each participant and the game's worth.

Solution:-

Now let's build player A's payoff matrix.

Player B

Player A HT Row minimum

H2-2-2

T-2 2 -2

Column Maximums 2   2

Given that maxmin=-2 and minmax=2, the game's value falls between -2 and 2.

i.e., there is no unique saddle point since maxmin does not equal minmax. Mixed strategies should be used to solve games without a saddle point.

Since the game is 2 x 2 and lacks a saddle point, we apply the following formulas.

$$p_1 = \frac{a_{22} - a_{21}}{a_{11} + a_{22} - (a_{12} + a_{21})} = \frac{(2) - (-2)}{(2+2) - (-2-2)} = 4/8 = 1/2$$

$$p_2 = 1 - p_1 = 1/2$$

$$q_1 = \frac{a_{22} - a_{12}}{a_{11} + a_{22} - (a_{12} + a_{21})} = \frac{2 - (-2)}{2 + 2 - (-2-2)} = 4/8 = 1/2$$

$$q_2 = 1 - q_1 = 1/2$$

As a result, the game is fair.

Thus, the best combined approach for players A and B is provided by

$$S_A = \begin{pmatrix} H & T \\ 1/2 & 1/2 \end{pmatrix} \text{ and } S_B = \begin{pmatrix} H & T \\ 1/2 & 1/2 \end{pmatrix}$$

An algebraic approach to game solving without saddle points

The likelihood that player A will select his strategies A1, A2,..., Am, should be represented by the numbers p1, p2,..., pm, where p1+p2+...+Pm=1.

Assume that q1, q2,..., qn represent the likelihood that player B would select his strategy B1, B2,..., Bn, respectively, and that q1+q2+...+qn=1.

Player A's strategies	Player B's strategies				Probability
	B_1	B_2	....	B_n	p_1
A_1	a_{11}	a_{12}	....	a_{1n}	p_2
A_2	a_{21}	a_{22}	....	a_{2n}	p_3
....			....		
A_m	a_{m1}	a_{m2}	....	a_{mn}	p_m
Probability	q_1	q_2		q_n	

To find S_A

The anticipated profit for player A in the event that player B chooses strategy B1, B2, and so on Player A is a gainer, thus since he anticipates at least V, we have

$a_{11}p_1+a_{12}p_2+......+a_{m1}p_m>=V$

$a_{12}p_1+a_{22}p_2+....+a_{m1}p_m>=V$

......

$a_{1n}p_1+a_{22}p_2+....+a_{mn}p_m>=V$

The aforementioned inequalities are treated as equations and solved for the specified unknowns in order to get the values of pi.

To find S_B

Player A's predicted loss against player B if they use tactics A1, A2,...am Considering that player B is a loser, we have

$a_{11}q_1+a_{12}q_2+......+a_{1n}q_n<=V$

$a_{12q}1+a_{22}q_2+....+a_{m1}q_m>=V$

......

$a_{1n}q_1+a_{2n}q_2+....+a_{mn}q_m>=V$

The aforementioned inequalities are treated as equations and solved for the specified unknowns in order to determine the values of qi.

The value of the game may be found by changing the values of ai and qi in any one of the aforementioned equations.

Remarks:-

When both players have more than two plans, this process gets rather drawn out.

Eg. 11.3.3

A and B, the two players, match coins. Should the coins align, A will receive two units of value; if they don't align, B will receive two units of value. Ascertain the best tactics for each participant and the game's worth.

Solution:-

Now let's build player A's payoff matrix.

	Player B			
Player A	H	T	Row minimum	Probability
H	2	-2	-2	p_1
T	-2	2	-2	p_2
Column Maximums	2	2		
Probability	q_1	q_2		

Given that maxmin=-2 and minmax=2, the game's value falls between -2 and 2. i.e., there is no unique saddle point since maxmin does not equal minmax. Mixed strategies should be used to solve games without a saddle point.

The algebraic technique is applied in this 2x2 game that lacks a saddle point.

To find S_A

$2p_1-2p_2=V$ ----(1) and $-2p_1+2p_2=V$ ----(2)

Therefore we have $2p_1-2p_2 =-2p_1+2p_2$

$=>2p_1-2(1-p_1) =-2p_1+2(1-p_1)$ $\quad$ $[p_1+p_2=1]$

$=>2p_1-2+2p_1 =-2p_1+2-2p_1$

$=>2p_1-2+2p_1 =-2p_1+2-2p_1$

$=>8p_1=4$

$=>p_1=1/2$

$p_2=1-p_1=1-1/2=1/2$

To find S_B

$2q_1-2q_2=V$ ----(3) and $-2q_1+2q_2=V$ ----(4)

Therefore we have $2q_1-2q_2 =-2q_1+2q_2$

$=>2q_1-2(1-q_1) =-2q_1+2(1-q_1)$ $\quad$ $[q_1+q_2=1]$

$=>2q_1-2+2q_1 =-2q_1+2-2q_1$

$=>2q_1-2+2q_1 =-2q_1+2-2q_1$

$=>8q_1=4$

$\Rightarrow q_1=1/2$

Thus $q_2=1-q_1=1-1/2=1/2$

$S_A=\begin{pmatrix} H & T \\ 1/2 & 1/2 \end{pmatrix}$ and $S_B=\begin{pmatrix} H & T \\ 1/2 & 1/2 \end{pmatrix}$

$(1) \Rightarrow 2p_1-2p_2=V$

$\Rightarrow 2(1/2)-2(1/2)=V$

$\Rightarrow V=0$

11.4 Introduction of the rule of dominance

The Rule of Dominance is a concept in decision theory and game theory that helps individuals or organizations make optimal choices when faced with multiple options. It states that if one option is superior to another option across all possible scenarios or criteria, then the dominant option should be chosen.

Here's an introduction to the Rule of Dominance:

1. **Principle**: The Rule of Dominance suggests that when comparing two or more alternatives, if one alternative is better than another in every aspect or under every possible circumstance, then it is considered dominant and should be chosen.
2. **Decision Making**: In decision-making contexts, the Rule of Dominance provides a straightforward method for evaluating and selecting the best option among competing alternatives. By systematically comparing the attributes, benefits, costs, and risks associated with each option, decision-makers can identify dominant options and eliminate inferior ones.
3. **Criteria for Comparison**: The Rule of Dominance requires clear criteria for comparison to assess the superiority of one option over another. These criteria could include factors such as cost, quality, efficiency, effectiveness, risk, and any other relevant considerations depending on the decision context.
4. **Eg.**: For Eg., consider a company evaluating two potential suppliers for a critical component. Supplier A offers higher quality, faster delivery times, and lower costs compared to Supplier B. In this case, Supplier A is dominant because it outperforms Supplier B across all relevant criteria.

11.5 Graphical methods for solving mixed strategy games

Graphical methods for solving mixed strategy games are visual techniques used in game theory to find the optimal strategies and outcomes of two-player zero-sum games, where players choose their strategies probabilistically (i.e., using mixed strategies) to maximize their expected payoffs. These graphical methods provide a geometric representation of the game matrix and help identify the optimal mixed strategies for both players.

One of the most commonly used graphical methods for solving mixed strategy games is the graphical method for 2x2 games. Here's an overview of the process:

1. **Plotting the Payoff Matrix:** Plot the game's payoff matrix first, with Player 1's and Player 2's strategies represented in the rows and columns, respectively. The payoffs to Player 1 and Player 2 for every combination of strategies are contained in each cell of the matrix.

2. **Drawing Best Response Lines**: Determine which mixed strategies will maximize each player's expected payout in light of their opponent's strategies by drawing their best response line. When Player 1 is undecided between their pure strategy, the ideal response line for Player 1 links those points. The optimal response line for Player 2 does the same thing; it links the places at which Player 2 is undecided between their pure plans.

3. **Identifying the Nash Equilibrium**: At the intersection of the optimal response lines, the game's Nash equilibrium is reached. When neither player has an incentive to change their plan in light of the opponent's approach, this point reflects the ideal combination of strategies for both players.

4. **Calculating Probabilities**: The intersection point coordinates can be used to compute the probability related to each mixed strategy. Each player's probability of selecting their own pure strategy is determined by these probabilities.

11.6 Linear programming methods for solving mixed strategy games

Linear programming (LP) methods can be used to solve mixed strategy games, particularly in cases where the game is larger or more complex than a simple 2x2 matrix. Linear programming provides a systematic approach to finding optimal strategies for players in mixed strategy games by formulating the problem as an optimization program with linear constraints.

Here's how linear programming methods can be applied to solve mixed strategy games:

1. **Formulate the LP Model**:
    - Define decision variables representing the probabilities of each player choosing their respective strategies. For Eg., let $xixi$ represent the probability of Player 1 choosing strategy ii, and let $yjyj$ represent the probability of Player 2 choosing strategy jj.
    - Define the OF to maximize or minimize the expected payoff for one of the players. The OF is typically a linear combination of the payoffs associated with each strategy.
    - Formulate constraints based on the requirements of the game. These constraints ensure that the probabilities sum to 1 (since players must choose a strategy with certainty) and that the expected payoffs for both players are consistent with the game matrix.
2. **Solve the LP Model**:
    - Use a linear programming solver to solve the formulated LP model. The solver will find the values of the decision variables (i.e., the probabilities of each player's mixed strategy) that optimize the OF while satisfying the constraints.
    - The optimal solution to the LP model corresponds to the Nash equilibrium of the game, where neither player has an incentive to deviate from their strategy given the opponent's strategy.
3. **Interpret the Solution**:
    - Once the LP solver finds the optimal solution, interpret the results to determine the optimal mixed strategies for both players and the expected payoffs associated with those strategies.
    - Verify that the solution satisfies the conditions of a Nash equilibrium, ensuring that no player can unilaterally improve their payoff by changing their strategy.

11.7 Summary

- In operations research, game theory offers a mathematical framework for examining decision-makers' strategic interactions. It has many uses, ranging from business and economics to politics, biology, and computer science. Fundamentally, the goal of game theory is to comprehend how rational players act in cooperative or competitive settings and how their choices affect the course of events.

- The ideas of equilibrium, players, tactics, and payoffs are important aspects of game theory. A key idea in many game-theoretic analysis is the Nash equilibrium, which occurs when neither player has an incentive to unilaterally stray from their selected strategy. Nevertheless, additional equilibrium ideas that are appropriate for many game kinds and situations are also explored by game theory, including sub-game perfect equilibrium, correlated equilibrium, and evolutionary stable strategies.
- Applications of game theory can be found in many areas, such as negotiations, auctions, pricing schemes, strategic planning, and dispute resolution. It gives important insights into strategic interactions and decision-making processes, empowering decision-makers to foresee and effectively address other people's behaviors.

11.8 Keywords

- Game Theory
- static games
- Saddle Point
- Dominance Rule
- Maxmin and Minmax
- Mixed strategy
- Dominance rule

11.9 Self Assessment Question

1. What is game theory, and what does it study?
2. What are the key components of a game?
3. Define Nash equilibrium and its significance in game theory.
4. How do static games differ from dynamic games?
5. What solution concept is often used to analyze dynamic games?
6. Explain the concept of cooperative game theory.
7. What are some real-world applications of game theory?
8. How does game theory help in understanding strategic interactions?
9. What are some limitations of game theory?

10. How can game theory be used in decision-making processes?

11.10 Case Study

1. Imagine a situation in which an oligopolistic market is dominated by multiple enterprises. Examine the ways in which enterprises' strategic interactions during price-setting can be modeled using game theory. Talk about the effects of various equilibrium theories, such as collusion and Nash equilibrium.
2. Consider a supply chain in which manufacturers, merchants, and suppliers are only a few of the many parties involved. Model the process of negotiating prices, manufacturing quantities, and distribution agreements between various parties using game theory. Examine the ways in which cooperative game theory can be used to accomplish win-win situations.

11.11 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.

UNIT -12

Sequencing Problems

Learning Objective

- Students should grasp the concept of job sequencing problems, which involve scheduling tasks or jobs on machines or in a production line to optimize certain objectives such as minimizing completion time, minimizing lateness, or maximizing resource uses.
- Developing skills in applying optimization techniques to solve job sequencing problems.
- Analyzing the computational complexity of job sequencing problems and understanding the trade-offs between solution quality and computational resources.

Structure

12.1 Introduction of sequencing problems

Sequencing problems arise in various contexts where items or tasks need to be arranged in a specific order to optimize a certain objective. These problems are common in operations research, scheduling, manufacturing, logistics, and computer science. Here are some key types of sequencing problems:

1. **Job Sequencing**:
    - In job sequencing problems, a set of jobs or tasks must be processed on a machine or a set of machines in a specific sequence to minimize the total processing time, completion time, or cost.
    - Eg.s include the flow shop scheduling problem, where each job must be processed sequentially on a series of machines, and the job shop scheduling problem, where each job may require processing on multiple machines in a specific order.
2. **Production Sequencing**:
    - Production sequencing problems involve determining the optimal sequence of production orders on a manufacturing line to minimize setup times, idle times, or production costs.
    - This type of problem is common in manufacturing industries such as automotive, electronics, and aerospace, where efficient production sequencing can lead to cost savings and improved throughput.
3. **Vehicle Routing**:
    - In order to reduce the overall journey distance, time, or cost while meeting a variety of limitations such capacity limits, time windows, and vehicle availability, vehicle routing problems (VRPs) entail figuring out the best order of stops or deliveries for a fleet of vehicles.
    - The vehicle routing problem with time windows (VRPTW), in which each customer has a defined time window during which they must be serviced, and the traveling salesman problem (TSP), in which a single vehicle must visit a set of sites exactly once and return to the beginning point, are two Eg.s.
4. **DNA Sequencing**:
    - In bioinformatics, DNA sequencing problems involve determining the order of nucleotides in a DNA molecule based on experimental data from sequencing techniques.
    - Various algorithms and computational methods are used to assemble short DNA fragments into longer sequences, with the goal of reconstructing the original DNA sequence.
5. **Task Scheduling**:
    - Task scheduling problems involve determining the optimal order and timing of tasks or activities to maximize resource utilization, minimize project duration, or meet project deadlines.

- Eg.s include project scheduling, employee shift scheduling, and classroom scheduling, where tasks or events must be scheduled to avoid conflicts and optimize resource allocation.

ALGORITHM TO RESOLVE THE PROBLEM OF JOB SEQUENCING

The organization of tasks to be completed or processed by a machine in a specific order is known as job sequencing. In the world of computers, job sequencing problems have emerged as the main concern. A finite collection of m machines, each of which can only process one assignment at a time, and a finite set of n tasks, each of which is composed of a series of operations. Our goal is to identify an inventory, or an allocation of the operations to time intervals to machines that has minimal length. Each assignment must be assessed continuously on a specific machine for a certain amount of time.

It can be difficult to get a computer to solve an issue or complete a task. This is because it involves thoughtful preparation, close attention to detail, and a great deal of thinking. It may also be stimulating, demanding, and full of opportunities for creative fulfillment. This is due to the fact that using a computer to solve a problem is like to training a stupid servant how to perform a task. As we've previously seen, the initial version of the task description is prepared as an algorithm, and we provided a definition! However, it is important to utilize that definition carefully. It is important to constantly keep in mind the executing representative's skills and limitations—the computer. Any model's constituent parts can be of

- Decision variables (i.e., model output, what we can modify to improve the system)
- Parameters (i.e., non-modifiable values used as model input)
- OF (e.g., profit, maximizing quality, minimizing expenses, and time).
- Constraints (Select the permissible values for the decision variable)

The following describes three fundamental categories of choice variables.

- Sequence (job permutations)
- Scheduling policy (Determines the next job, given the current state of the system)
- Schedule (Allocation of the work in a more complicated environment)

Auxiliary variables, or dynamic data, are other variables. These are some of the auxiliary variables that are described.

- Time to completion (Cij); tardiness (Lj); tardiness (Tj); and makespan (Mj)

All inventory problems fall under one of three main aims, or goals.

- Tasks ought to be completed on time; there should be no "late" work.
- The length of time a job remains in the system need to be
- Using work centers to the fullest extent possible is avaricious (regarding work center use).

A schedule is a work order that is established for each processing system equipment. The process of deciding how to distribute scarce resources among various tasks throughout time is called scheduling. The beginning and ending timings of each job's operations can be used to describe the timetable. Making a timetable that reduces a specific OF is the goal. In Figure 12.1.1, the scheduling job is displayed.

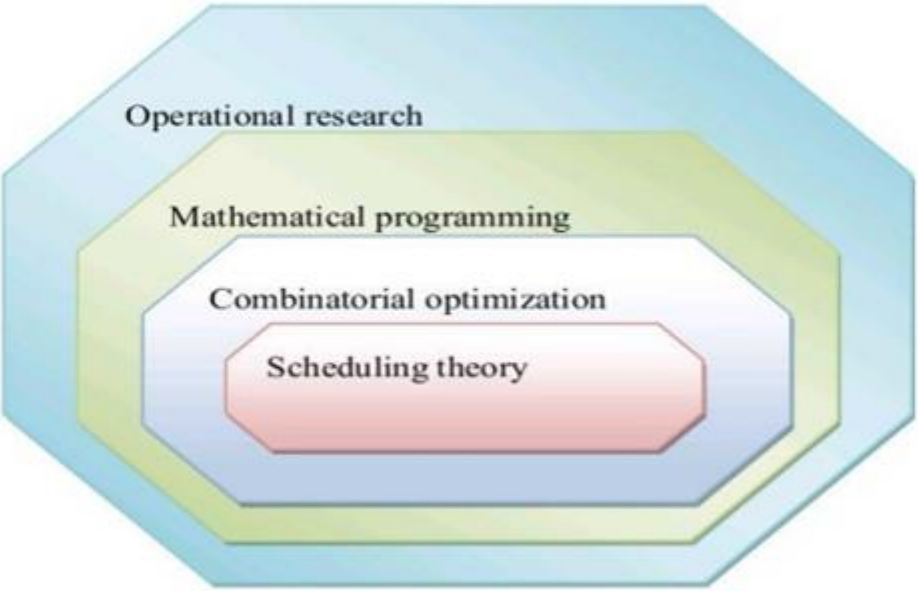


Fig 12.1.1 Scheduling Task

The following are the standard standards for scheduling.

- A machine cannot process two or more jobs at the same time; two or more machines cannot process a job at the same time.

Certain restrictions must be met, depending on the kind of scheduling system (e.g., jobs may be issued at various times, jobs may be permitted to be preempted by other jobs, etc.). The following is a description of the different scheduling challenges.

- Aligning schedules with operational time
- The assignment is how demand is divided up and prioritized.

- Important considerations include whether to schedule in reverse or forward
- The standards for assigning priority

Two different kinds of scheduling exist. The two of them are backward scheduling and forward scheduling.

Forward scheduling begins as soon as the specifications are determined. It generates a workable timetable, although it could miss deadlines. It usually leads to an accumulation of work-in-process inventories.

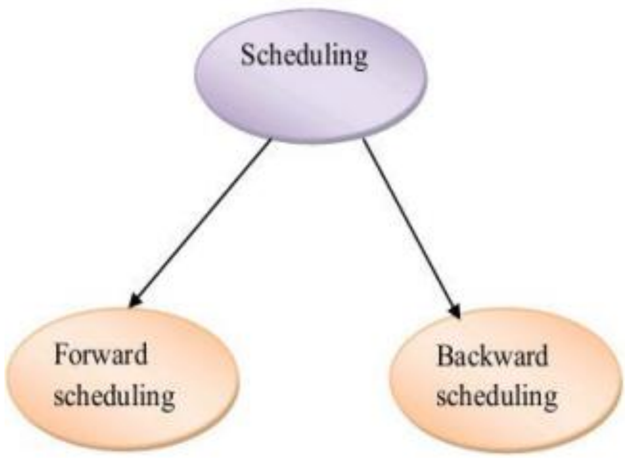


Fig 12.1.2 Scheduling Types

12.2 N tasks are processed by two machines.

The following procedures can be used to sequence the N jobs over two computers. Compute the sequencing problem's time deviation table.

- Complete that task first in the cell when machine 1's two time deviation vectors are zero.

In the event that two vectors in several cells are zero, compute the cumulative deviation of the relevant columns. The biggest sum deviation cell is processed first, and so forth.

- For machine 2, the procedures are the same, except the jobs are completed last.
- Repeat the previous steps, calculating the reduced time deviation table for each non-assigned work.
- If there is a sequence that includes every job, stop the process.
- To acquire the smallest overall amount of elapsed time, reverse order the obtained sequence.

The following example explains why N jobs should be sequenced in two machines.

Eg. 12.2.2

This is the processing of N jobs in two machines.

Jobs/Machines	J1	J2	J3	J4	J5	J6	J7	J8	J9
M1	2	5	4	9	6	8	7	5	4
M2	6	8	7	4	3	9	3	8	11

Solution:

Using equations (1) and (2), the time deviation table for the given issue will first be computed. This is the table.

Jobs/Machines	J1	J2	J3	J4	J5	J6	J7	J8	J9
M1	(7,4)	(4,3)	(5,3)	(0,0)	(3,0)	(1,1)	(2,0)	(4,3)	(5,7)
M2	(5,0)	(3,0)	(4,0)	(7,5)	(8,3)	(2,0)	(8,4)	(3,0)	(0,0)

The following is the assignment for tasks J4 on machine 1 and J9 on machine 2, both of which have vector zero.

J4								J9

The following is the table of decreased time length for the remaining jobs.

Jobs/Machines	J1	J2	J3	J5	J6	J7	J8
M1	2	5	4	6	8	7	5
M2	6	8	7	3	9	3	8

For this sequence, the time deviation table is computed as follows.

Jobs/Machines	J1	J2	J3	J5	J6	J7	J8
M1	(6,4)	(3,3)	(4,3)	(2,0)	(0,1)	(1,0)	(3,3)
M2	(3,0)	(1,0)	(2,0)	(6,3)	(0,0)	(6,4)	(1,0)

Because job J6 on machine 2 contains zero values for both of its vectors, it is categorized as

J4							J6	J9

The time duration table is provided for jobs J1, J2, J3, J5, J7, and J8. As the jobs are allocated to the machines as previously stated, jobs J4, J6, and J9 will not be included in this table.

Jobs/Machines	J1	J2	J3	J5	J7	J8
M1	2	5	4	6	7	5
M2	6	8	7	3	3	8

For the jobs mentioned above, the time deviation table is computed as follows.

Jobs/Machines	J1	J2	J3	J5	J7	J8
M1	(5,4)	(2,3)	(3,3)	(1,0)	(0,0)	(2,3)
M2	(2,0)	(0,0)	(1,0)	(5,3)	(5,4)	(0,0)

The vectors zero for the jobs J7, J2, and J8 in machine 1 and machine 2 are allocated as follows.

J4	J7				J2	J8	J6	J9

The following is the time duration chart for the remaining jobs J1, J3, and J5.

Jobs/Machines	J1	J3	J5
M1	2	4	6
M2	6	7	3

For the jobs mentioned above, the time deviation table is computed as follows.

Jobs/Machines	J1	J3	J5
M1	(4,4)	(2,3)	(0,0)
M2	(1,0)	(0,0)	(4,3)

Here, J5 in machine 1 and J3 in machine 2 are allocated as they both have zero vectors. The remaining J1 is similarly assigned in the manner that follows.

J4	J7	J5	J1	J3	J2	J8	J6	J9

Next, this sequence is put in reverse order, starting with the machine 1 and machine 2 jobs and working down to the most recent job acquired. This is how it is depicted:

J1	J5	J7	J4	J9	J6	J8	J2	J3

For the given problem, the sequence listed above is necessary. For this work sequence, the minimal total elapsed time is computed. Total Elapsed Time = Time between beginning the first task in the optimum sequence on machine 1 and finishing the last work in the optimum sequence on machine M is the formula used to calculate the total elapsed time.

Job sequence	Machine 1		Machine 2	
	Time-in	Time-out	Time-in	Time-out
1	0	2	2	8
5	2	8	8	11
7	8	15	11	14
4	15	24	14	18
9	24	28	18	29
6	28	36	29	38
8	36	41	38	46
2	41	46	46	54
3	46	50	54	61

Thus, 61 hours is the minimal total elapsed time that may be acquired.

12.3 n jobs through 3 machines

The following procedures can be used to sequence the N jobs over three computers. Compute the sequencing problem's time deviation table.

- Complete that task first in the cell when machine 1's two time deviation vectors are zero.

Find the sum of the deviation vectors individually for the cells above and below the zero cell if both of the time deviation vectors in machine 3 are zero, then complete that task in the last machine. If the vectors in machine 2 are zero. Examine the two deviations in comparison.

If the total of the deviation vectors above the cell is smaller than the other, then do that specific task first. If they are identical, finish the task first or last.

- Repeat the previous steps, calculating the reduced time deviation table for each non-assigned work.
- If there is a sequence that includes every job, stop the process.
- To acquire the smallest overall amount of elapsed time, reverse order the obtained sequence.

Eg. 12.3.1

The following list of N jobs to be completed by three machines is provided.

Jobs/Machines	J1	J2	J3	J4	J5	J6	J7
M1	3	8	7	4	9	8	7
M2	4	3	2	5	1	4	3
M3	6	7	5	11	5	6	12

First, using equations (1) and (2), the time deviation table for the given issue will be computed. This is the table.

Jobs/Machines	J1	J2	J3	J4	J5	J6	J7
M1	(6, 3)	(1, 0)	(2, 0)	(5, 7)	0, 0)	(1, 0)	(2, 5)
M2	(1, 2)	(2, 5)	(3, 5)	(0, 6)	(4, 8)	(1, 4)	(2, 9)
M3	(6, 0)	(5, 1)	(7, 2)	(1, 0)	(7, 4)	(6, 2)	(0, 0)

Due to their shared vector zero, jobs J5 in machine 1 and J7 in machine 3 are allocated to the following job sequence.

J5						J7

The following is the time duration chart for the occupations other than J5 and J7.

Jobs/Machines	J1	J2	J3	J4	J6
M1	3	8	7	4	8
M2	4	3	2	5	4
M3	6	7	5	11	6

The following is the time deviation table for the occupations mentioned above.

Jobs/Machines	J1	J2	J3	J4	J6
M1	(5, 3)	(0, 0)	(1, 0)	(4, 7)	(0, 0)
M2	(1, 2)	(2, 4)	(3, 5)	(0, 6)	(1, 4)
M3	(5, 0)	(4, 1)	(6, 2)	(0, 0)	(5, 2)

Because both of the tasks' vectors are zero in machines J2, J6, and J4, they are ordered as follows.

J5	J6	J2			J4	J7

The following is the time duration chart for the remaining jobs.

Jobs/Machines	J1	J3
M1	3	7
M2	4	2
M3	6	5

The following is the time deviation table for the two jobs mentioned above.

Jobs/Machines	J1	J3
M1	(4, 3)	(0, 0)
M2	(0, 2)	(2, 5)
M3	(0, 0)	(1, 2)

Next, the jobs J3 in machine 1 and J1 in machine 3 are allocated to the following order:

J5	J6	J2	J3	J1	J4	J7

Then, this sequence is put in reverse order, starting with the machine 1 and machine 2 jobs and working backwards to the final job acquired. This is how it is depicted:

J1	J3	J2	J6	J5	J7	J4

For the obtained sequence, the minimal total elapsed time is computed as follows.

Job sequence	Machine 1		Machine 2		Machine 3	
	Time-in	Time-out	Time-in	Time-out	Time-in	Time - out
1	0	3	3	7	7	13
3	3	10	7	9	13	18
2	10	18	9	12	18	25
6	18	26	12	16	25	31
5	26	35	16	17	31	36
7	35	42	17	20	36	48
4	42	46	20	25	48	59

Therefore, for the provided sequence, the minimal total elapsed time is 59 hours.

12.4 jobs through m machines

The scenario you've described is a job sequencing problem where two jobs need to be processed through multiple machines. This type of problem is often referred to as a "flow shop scheduling problem." In a flow shop scheduling problem, each job must follow the same sequence of processing on the available machines. Here's an overview of the problem and potential approaches to solving it:

Problem Description:

- There are two jobs, labeled Job 1 and Job 2.
- There are mm machines, labeled Machine 1, Machine 2, ..., Machine mm.
- Each job must be processed through all machines in the same order (i.e., the sequence of machines is the same for both jobs).
- The processing times of each job on each machine are known.

Approaches to Solving the Problem:

1. **Brute Force Enumeration**:
   - Enumerate all possible sequences of machines for processing the two jobs.
   - Calculate the total processing time for each sequence and select the sequence with the minimum total processing time.
   - This approach becomes impractical as the number of machines increases due to the combinatorial explosion of possible sequences.
2. **Heuristic Algorithms**:
   - Develop heuristic algorithms to find near-optimal solutions efficiently.
   - Eg.s include the Johnson's Rule algorithm, which is effective for two-machine flow shop scheduling problems, and metaheuristic algorithms like genetic algorithms or simulated annealing.
3. **Dynamic Programming**:
   - Apply dynamic programming techniques to solve smaller instances of the problem optimally.
   - For larger instances, dynamic programming may become computationally infeasible due to the large number of possible sequences.
4. **Mathematical Optimization**:
   - Formulate the problem as a mathematical optimization model, such as a mixed-integer linear program (MILP) or a constraint programming (CP) model.
   - Use optimization solvers to find the optimal sequence of machines that minimizes the total processing time for both jobs.

The choice of approach depends on factors such as problem size, complexity, and available computational resources. For small instances, brute force enumeration or dynamic programming may be feasible, while heuristic algorithms or mathematical optimization may be more suitable for larger or more complex instances.

12.5 n jobs through m machines

The following procedures can be used to sequence the N jobs in M machines.

- One of the following circumstances initially transforms the M machine problem into the two machine problem.

$$Minimum\ of\ M1\ \geq\ Maximum\ of\ (M2, M3 \ldots Mm - 1) \quad (3)$$

$$Minimum\ of\ Mm\ >\ Maximum\ of\ (M2, M3 \ldots Mm - 1) \quad (4)$$

- The time deviation approach, which was used in the previous two machine problems, is then applied to produce the necessary sequence.

Eg. 12.5.1

Considering of N jobs in M machines can be specified by the next problem

Jobs/Machines	J1	J2	J3	J4
M1	7	6	5	8
M2	5	6	4	3
M3	2	4	5	3
M4	3	5	6	2
M5	9	10	8	6

The criteria (3) and (4) transform the M machine issue into a two machine problem. The following is the converted two machine issue.

Jobs/Machines	J1	J2	J3	J4
G	17	21	20	16
H	19	25	23	14

The process that is often employed for the two machine problem, as described by the preceding Example, is then used to acquire the necessary work sequence. In summary as a result, the necessary work sequence is obtained using the newly updated heuristic strategy known as the time deviation method, and the minimal total elapsed time for this job sequence is likewise determined using the standard process as stated in the above

Examples. The EGs also provided an explanation for dealing out N jobs in two machines, dealing out N jobs in three machines, and dealing out N jobs in M machines.

12.6 Summary

- In operations research, job sequencing problems deal with the scheduling of jobs or tasks on machinery or in a production line in order to maximize specific goals. These goals could be maximizing resource use, minimizing lateness, or minimizing completion time.
- Understanding how to formulate problems for various circumstances, including flow shop scheduling, job shop scheduling, parallel machine scheduling, and single-machine sequencing, is essential to analyzing job sequencing difficulties. To address these issues, many optimization strategies can be used, including heuristic approaches, dynamic programming, greedy algorithms, and integer linear programming.
- Sensitivity analysis is used to evaluate how resilient scheduling solutions are to modifications in input parameters. Making well-informed decisions to increase productivity, cut lead times, and boost competitiveness across a range of industries is made easier with the use of decision support systems built on mathematical models and optimization approaches.

12.7 Keywords

- Job sequencing
- Sequencing problems
- N jobs through 2, 3, machines
- 2, N jobs through m machines
- Operations research

12.8 Self Assessment Question

1. What are job sequencing problems in operations research?
2. What are the key objectives in job sequencing problems?
3. Name some common types of job sequencing problems.
4. What are the performance metrics used to evaluate scheduling solutions?
5. What optimization techniques can be applied to solve job sequencing problems?

6. Describe the difference between single-machine sequencing and parallel-machine scheduling.
7. How does complexity analysis help in understanding job sequencing problems?
8. What are some real-world applications of job sequencing problems?
9. How does sensitivity analysis contribute to scheduling decisions?
10. How do decision support tools aid in solving job sequencing problems?

12.9 Case Study

ABC Manufacturing is a corporation that utilizes numerous machinery to make an array of products. Each product takes a varied amount of processing time on a separate equipment during the multiple steps of the production process. The corporation wants to decrease completion times and increase resource efficiency, so it optimizes production scheduling. The aim of this study is to create the best possible production schedule that minimizes the overall completion time, taking into account the machine availability and processing durations for individual products.

12.10 References

1. Chong, E.K.P., Żak, S. H. (2013). An Introduction to Optimization. Germany: Wiley.
2. Gupta, C. B. (2008). Optimization Techniques in Operation Research. India: I.K. International Publishing House Pvt. Limited.