OPTIMIZATION TECHNIQUES AND OPERATIONS RESEARCH

Dr. Nibaran Das
Department of Computer Science and engineering

COURSE OUTCOME:

- Able to understand the theory of optimization methods and algorithms
- Able to develop optimization problem using mathematical analysis and numerical method
- Able to solve various linear and nonlinear optimization problems
- Able to solve optimization using software tools.
- Able to develop and promote research interest in optimization techniques in the problems of Engineering and Technology especially in Machine learning

Optimization Techniques

OPTIMIZATION LITERATURE

Textbooks:

1. Natarajan, Balasubramanie, Tamilarasi, Operation Research
2. H. A. Taha: Operations research
3. Rao S.S., *Engineering Optimization - Theory and Practice,* John Wiley & Sons, New York, 903 pp, 1996.
4. K. Deb: Optimization for Engineering Design – AlgExamples. orithms and
5. Chong E.K.P. and Zak S.H., *An Introduction to Optimization, Second Edition,* John Wiley & Sons, New York, 476 pp, 2001.
6. Gill P.E., Murray W. and Wright M.H., *Practical Optimization,* Elsevier, 401 pp., 2004.
7. Goldberg D.E., *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison Wesley, Reading, Mass., 1989.
8. S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.(available at http://www.stanford.edu/~boyd/cvxbook/)
9. S. Fang et al: Linear optimizations and Extensions
10. G. Hadley: Linear programming, Narosa Publishing House, New Delhi, 1990.

Optimization Techniques

COURSE SCHEDULE:

- Introduction [4L]
  - Historical development, Engineering application of optimization, Formulation of design problems as mathematical programming problems, classification of optimization problems.
- Linear Programming I [10L]
  - Graphical method, Simplex method, Revised simplex method, Duality in linear programming, Sensitivity analysis, other algorithms for solving LP problems.
- Linear Programming II [6L]
  - Transportation Problem, Assignment Problem and other applications, Integer Programming.
- Non Linear Programming [8L]
  - Unconstrained optimization techniques, Direct search methods, Descent methods, Constrained optimization, Direct and indirect methods, Optimization with calculus, Khun-Tucker conditions.
- Dynamic Programming [6L]
  - Introduction, Sequential optimization, computational procedure, curse of dimensionality.
- Queuing Theory [6L]
  - Kendall's notation, M/M/1 queue, M/G/1 queue, bulk arrival

1. INTRODUCTİON

- Optimization is the act of obtaining the best result under given circumstances.
- Optimization can be defined as the process of finding the conditions that give the maximum or minimum of a function.
- The optimum seeking methods are also known as *mathematical programming techniques* and are generally studied as a part of operations research.
- *Operations research* is a branch of mathematics concerned with the application of scientific methods and techniques to decision making problems and with establishing the best or optimal solutions.

1. INTRODUCTİON

- **Operations research** (in the UK) or **operational research (OR)** (in the US) or **yöneylem araştırması** (in Turkish) is an interdisciplinary branch of mathematics which uses methods like:
  - mathematical modeling
  - statistics
  - algorithms to arrive at optimal or good decisions in complex problems which are concerned with optimizing the maxima (profit, faster assembly line, greater crop yield, higher bandwidth, etc) or minima (cost loss, lowering of risk, etc) of some objective function.
- The eventual intention behind using operations research is to elicit a best possible solution to a problem mathematically, which improves or optimizes the performance of the system.

1. INTRODUCTION

TABLE 1.1 Methods of Operations Research

Mathematical Programming Techniques	Stochastic Process Techniques	Statistical Methods
Calculus methods	Statistical decision theory	Regression analysis
Calculus of variations	Markov processes	Cluster analysis, pattern recognition
Nonlinear programming	Queueing theory	Design of experiments
Geometric programming	Renewal theory	Discriminate analysis (factor analysis)
Quadratic programming	Simulation methods	
Linear programming	Reliability theory	
Dynamic programming		
Integer programming		
Stochastic programming		
Separable programming		
Multiobjective programming		
Network methods: CPM and PERT		
Game theory		
Simulated annealing		
Genetic algorithms		
Neural networks		

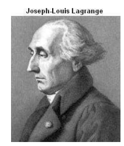
ENGINEERING APPLICATIONS OF OPTIMIZATION.

- Design of structural units in construction, machinery, and in space vehicles.
- Maximizing benefit/minimizing product costs in various manufacturing and construction processes.
- Optimal path finding in road networks/freight handling processes.
- Optimal production planning, controlling and scheduling.
- Optimal Allocation of resources or services among several activities to maximize the benefit.
- Optimize loss function in machine learning

Historical development

- Isaac Newton (1642-1727)
  (The development of *differential calculus* methods of optimization)

- Joseph-Louis Lagrange (1736-1813)
  (Calculus of variations, minimization of functionals, method of optimization for constrained problems)



- Augustin-Louis Cauchy (1789-1857)
  (Solution by direct substitution, steepest descent method for unconstrained optimization)

Optimization Techniques

1. INTRODUCTION

Historical development

- Leonhard Euler (1707-1783)
  (Calculus of variations, minimization of functionals)

- Gottfried Leibniz (1646-1716)
  (Differential calculus methods of optimization)

isim: Gottfired Wilhelm von Leibniz

Historical development

- George Bernard Dantzig (1914-2005)
  (Linear programming and Simplex method (1947))

- Richard Bellman (1920-1984)
  (Principle of optimality in dynamic programming problems)

- Harold William Kuhn (1925-)
  (Necessary and sufficient conditions for the optimal solution of programming problems, game theory)

Historical development

- Albert William Tucker (1905-1995)
  (Necessary and sufficient conditions for the optimal solution of programming problems, nonlinear programming, game theory: his PhD student was John Nash)

- Von Neumann (1903-1957)
  (game theory)

John von Neumann

1. INTRODUCTION

- **Mathematical optimization problem:**

$$\begin{aligned} &\text{minimize} \quad f_0(x) \\ &\text{subject to} \ \ g_i(x) \le b_i, \quad i = 1,\ldots.,m \end{aligned}$$

- $f_0 : \mathbf{R}^n \rightarrow \mathbf{R}$: objective function
- $x=(x_1,\ldots..,x_n)$: design variables (unknowns of the problem, they must be linearly independent)
- $g_i : \mathbf{R}^n \rightarrow \mathbf{R}$: $(i=1,\ldots,m)$: inequality constraints

- The problem is a constrained optimization problem

ART OF MODELING : MODEL BUILDING

- Development of an optimization model can be divided into five major phases.
  - Collection of data
  - Problem definition and formulation
  - Model development
  - Model validation and evaluation or performance
  - Model application and interpretation of results

DATA COLLECTION

- **Data collection**
  - may be time consuming but is the fundamental basis of the model-building process
  - extremely important phase of the model-building process
  - the availability and accuracy of data can have considerable effect on the accuracy of the model and on the ability to evaluate the model.

PROBLEM DEFINITION

- Problem definition and formulation, steps involved:
  - identification of the decision variables;
  - formulation of the model objective(s);
  - the formulation of the model constraints.
- In performing these steps one must consider the following.
  - Identify the important elements that the problem consists of.
  - Determine the number of independent variables, the number of equations required to describe the system, and the number of unknown parameters.
  - Evaluate the structure and complexity of the model
  - Select the degree of accuracy required of the model

MODEL DEVELOPMENT

- **Model development** includes:
  - the mathematical description,
  - parameter estimation,
  - input development, and
  - software development
- The model development phase is an iterative process that may require returning to the model definition and formulation phase.

MODEL VALIDATION AND EVALUATION

- This phase is checking the model as a whole.
- **Model validation** consists of validation of the assumptions and parameters of the model.
- The performance of the model is to be evaluated using standard performance measures such as Root mean squared error and R^2 value.
- Sensitivity analysis to test the model inputs and parameters.
- This phase also is an iterative process and may require returning to the model definition and formulation phase.
- One important aspect of this process is that in most cases data used in the formulation process should be different from that used in validation.

MODELING TECHNIQUES

- Different modeling techniques are developed to meet the requirement of different type of optimization problems. Major categories of modeling approaches are:
  - classical optimization techniques,
  - linear programming,
  - nonlinear programming,
  - geometric programming,
  - dynamic programming,
  - integer programming,
  - stochastic programming,
  - evolutionary algorithms, etc.
- These approaches will be discussed in the subsequent modules.

INTRODUCTION AND BASIC CONCEPTS

Optimization Problem and Model Formulation

OBJECTIVES

- To study the basic components of an optimization problem.
- Formulation of design problems as mathematical programming problems.

Optimization Techniques

INTRODUCTION - PRELIMINARIES

- Basic components of an optimization problem :
  - An **objective function** expresses the main aim of the model which is either to be minimized or maximized.
  - A set of **unknowns** or **variables** which control the value of the objective function.
  - A set of **constraints** that allow the unknowns to take on certain values but exclude others.

INTRODUCTION (CONTD.)

- The optimization problem is then to:
  - find values of the *variables* that minimize or maximize the *objective function* while satisfying the *constraints*.

OBJECTIVE FUNCTION

- As already defined the objective function is the mathematical function one wants to maximize or minimize, subject to certain constraints. Many optimization problems have a single objective function (When they don't they can often be reformulated so that they do). The two interesting exceptions are:
  - ***No objective function.*** The user does not particularly want to optimize anything so there is no reason to define an objective function. Usually called a *feasibility problem.*
  - ***Multiple objective functions.*** In practice, problems with multiple objectives are reformulated as single-objective problems by either forming a weighted combination of the different objectives or by treating some of the objectives by constraints.

STATEMENT OF AN OPTIMIZATION PROBLEM

To find $\mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{pmatrix}$ *which maximizes* $f(\mathbf{X})$

Subject to the constraints

$$g_i(\mathbf{X}) <= 0, \quad i = 1, 2, \ldots, m$$
$$l_j(\mathbf{X}) = 0, \quad j = 1, 2, \ldots, p$$

where

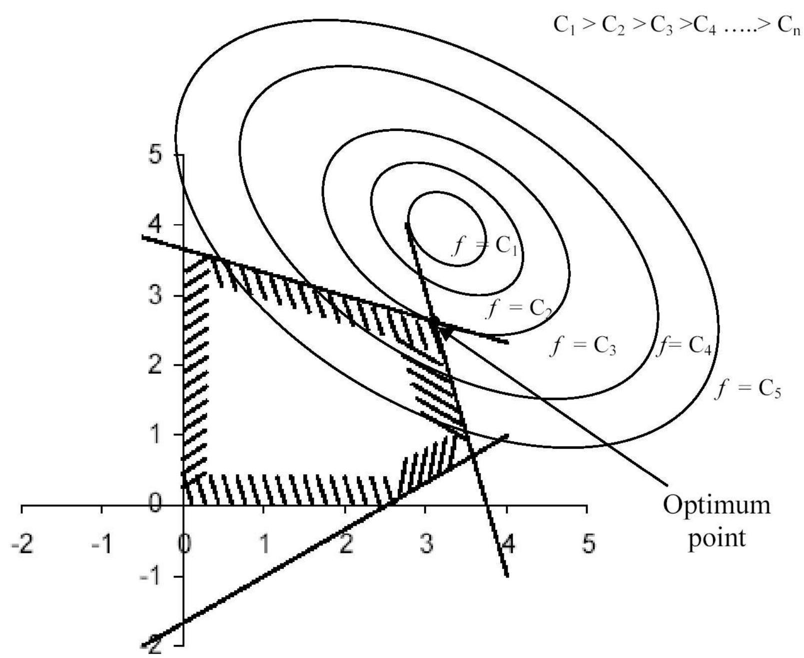
- **X** is an *n*-dimensional vector called the design vector
- *f*(**X**) is called the *objective function*, and
- $g_i(\mathbf{X})$ and $l_j(\mathbf{X})$ are known as inequality and equality constraints, respectively.

- This type of problem is called a *constrained optimization problem*.
- Optimization problems can be defined without any constraints as well. Such problems are called *unconstrained optimization problems.*

OBJECTIVE FUNCTION SURFACE

- If the locus of all points satisfying ***f***(**X**) = a constant **c** is considered, it can form a family of surfaces in the design space called the ***objective function surfaces***.
- When drawn with the constraint surfaces as shown in the figure we can identify the optimum point (maxima).
- This is possible graphically only when the number of design variable is two.
- When we have three or more design variables because of complexity in the objective function surface we have to solve the problem as a mathematical problem and this visualization is not possible.

OBJECTIVE FUNCTION SURFACES TO FIND THE OPTIMUM POINT (MAXIMA)



VARIABLES AND CONSTRAINTS

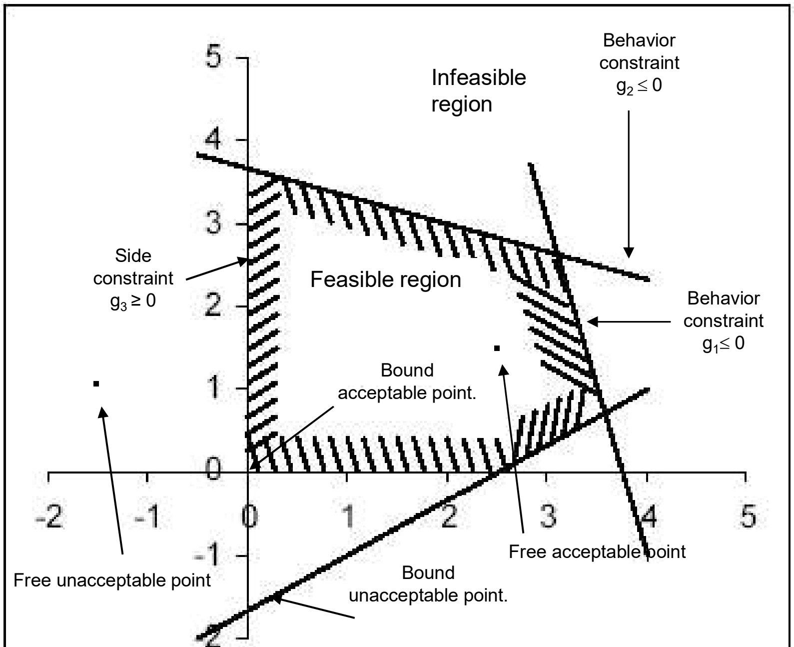
- Variables
  - These are essential. If there are no variables, we cannot define the objective function and the problem constraints.
- Constraints
  - Even though Constraints are not essential, it has been argued that almost all problems really do have constraints.
  - In many practical problems, one cannot choose the design variable arbitrarily. *Design constraints* are restrictions that must be satisfied to produce an acceptable design.

CONSTRAINTS (CONTD.)

- Constraints can be broadly classified as :
  - Behavioral or Functional constraints : These represent limitations on the behavior and performance of the system.
  - Geometric or Side constraints : These represent physical limitations on design variables such as availability, fabricability, and transportability.

CONSTRAINT SURFACES

- Consider the optimization problem presented earlier with only inequality constraints $g_i(\mathbf{X})$. The set of values of $\mathbf{X}$ that satisfy the equation $g_i(\mathbf{X})$ forms a boundary surface in the design space called a *constraint surface.*
- The constraint surface divides the design space into two regions: one with $g_i(\mathbf{X}) < 0$ (feasible region) and the other in which $g_i(\mathbf{X}) > 0$ (infeasible region). The points lying on the hyper surface will satisfy $g_i(\mathbf{X}) = 0$.



A HYPOTHETICAL TWO-DIMENSIONAL DESIGN SPACE WHERE THE FEASIBLE REGION IS DENOTED BY HATCHED LINES.

FORMULATION OF DESIGN PROBLEMS

- **The following steps summarize the procedure used to formulate and solve mathematical programming problems.**

1. Analyze the process to identify the process variables and specific characteristics of interest i.e. make a list of all variables.
2. Determine the criterion for optimization and specify the objective function in terms of the above variables together with coefficients.

Develop via mathematical expressions a valid process model that relates the input-output variables of the process and associated coefficients.

a) Include both equality and inequality constraints
b) Use well known physical principles
c) Identify the independent and dependent variables to get the number of degrees of freedom

3. If the problem formulation is too large in scope:
   a) break it up into manageable parts/ or
   b) simplify the objective function and the model
4. Apply a suitable optimization technique for mathematical statement of the problem.
5. Examine the sensitivity of the result to changes in the coefficients in the problem and the assumptions.

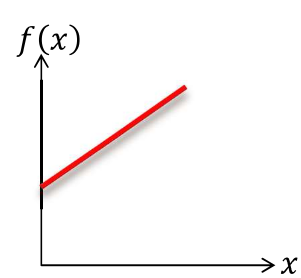
IDEA OF LINEAR ROGRAMMING

WHAT IS MATHEMATICAL PROGRAMMING?

- Mathematical programming is selecting the best option(s) from a set of alternatives mathematically
  - Minimizing/maximizing a function where your alternatives are defined by functions
- Many industries use mathematical programming
  - Supply chain, logistics, and transportation
  - Health industry, energy industry, finance, airlines
  - Manufacturing industry, agriculture industry
  - Education, Military
    - Operations Research raised

WHAT IS LINEAR PROGRAMMING?

- Linear Programming problem (LP) is a mathematical programming problem where all of your functions are linear
  - A function is linear when
    - The variables have power of 1 and the variables are not multiplied with each other
      - $f(x) = 3x + 5$ → linear
      - $f(x) = 3x^2 + 5$ → not linear
      - $f(x_1, x_2) = 3x_1 + 5x_2 + 7$ → linear
      - $f(x_1, x_2) = 3x_1x_2 + 7$ → not linear



  - It is assumed that our variables are continuous (for now)

Optimization Techniques

OUTLINE

- Definition of Linear Programming
- Formulation of Linear Programming
  - Surviving in an island
  - Extending a problem formulation
  - Wyndor Glass Co. Product Mix Problem
- Linear Programming Terminology
- Graphical Solution to Linear Programming
- Properties of Linear Programming

Optimization Techniques

HOW TO FORMULATE?

- What do we need to formulate a linear programming problem?
  - Have a problem!!
  - Know your problem
    - Gather the relevant data
    - Know what each number means
  - Pay attention to the class ☺
  - Best way to learn is to do some examples….

EXAMPLE 1: SURVIVING IN AN ISLAND

- Linear programming will save you!
  - Suppose that you will be left in a deserted island
  - You want to live as many days as you can to increase chance of rescue
  - Fortunately (!), you can take a bag with you to the island
  - What would you put into the bag?

EXAMPLE 1: SURVIVING IN AN ISLAND

- The bag can carry at most 50 lbs
- There are limited set of items you can carry
  - Bread: You can survive for 2 days with 1 lb of bread
  - Steak: You can survive for 5 days with 1 lb of steak (Memphis style grilled!)
  - Chicken: You can survive for 3 days with 1 lb of chicken (southern style deep fried!)
  - Chocolate: You can survive for 6 days with 1 lb of chocolate
- How much of each item to take with you?

EXAMPLE 1: FORMULATION STEPS

- STEP 0: Know the problem and gather your data
  - Your problem is to increase your chance of rescue by maximizing the number of days you survive
  - You can have 1 bag which can carry 50 lbs at most
  - You can only put bread, steak, chicken and chocolate into you bag
  - Each item enables you survive for a specific number of days for each pound you take
    - →Bread: 2 days/lb →Steak: 5 days/lb
    - →Chicken: 3days/lb →Chocolate: 6days/lb

EXAMPLE 1: FORMULATION STEPS

- STEP 1: Identify your decision variables
  - Decision variables are the things you control
  - The amount of each item you will take with you
    - x_1: *the amount of bread (lbs)*
    - x_2: *the amount of steak (lbs)*
    - x_3: *the amount of chicken (lbs)*
    - x_4: *the amount of chocolate (lbs)*

Warning: Always be careful with the metrics (try to use the same metrics)

EXAMPLE 1: FORMULATION STEPS

- STEP 2: Define your objective function and objective
  - Your objective function is the measure of performance as a result of your decisions
  - Your objective is what you want to do with your objective function

    Objective

  - Recall that you want to maximize the number of days you will survive in the island

    Performance measure
    Not a function though

EXAMPLE 1: FORMULATION STEPS

- STEP 2: We know our objective (maximize) and our performance of measure (number of days)
  - Express the performance of measure as a function of your decision variables to get the objective function
    - If you take x_1 lbs of bread, you will survive for $2x_1$ days
    - If you take x_2 lbs of steak, you will survive for $5x_2$ days
    - If you take x_3 lbs of chicken, you will survive for $3x_3$ days
    - If you take x_4 lbs of chocolate, you will survive for $6x_4$ days
  - Then objective function in terms of decision variables:

$$Maximize \quad 2x_1 + 5x_2 + 3x_3 + 6x_4$$

Optimization Techniques

EXAMPLE 1: FORMULATION STEPS

- STEP 3: Define your restrictions (constraints)
  - There may be some restrictions which limit what you can do (hence, they define set of your alternatives)
  - You can carry 1 bag and it can carry 50 lbs at most

$$\underbrace{x_1 + x_2 + x_3 + x_4}_{\text{Total amount you decide to carry}} \leq 50$$

Limit on how much you can carry

  - You cannot get negative amounts!!!

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

Optimization Techniques

EXAMPLE 1: LP FORMULATION

- Combine your objective, objective function, and constraints

$$\begin{array}{ll} \textit{Maximize} & 2x_1 + 5x_2 + 3x_3 + 6x_4 \\ \textit{subject to} & x_1 + x_2 + x_3 + x_4 \leq 50 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{array}$$

- This is your LP model
- Note that everything is linear!

EXAMPLE 1: EXTENDING THE LP

- Suppose that you can also take cheese
  - Cheese: You can survive for 4 days with 1 lb of cheese
    - x_5: *the amounf of cheese* (*lbs*)
  - We have a new decision variable: Update LP

$$
\begin{array}{ll}
\textit{Maximize} & 2x_1+5x_2+3x_3+6x_4+4x_5 \\
\textit{subject to} & x_1+x_2+x_3+x_4+x_5 \le 50 \\
 & x_1 \ge 0, x_2 \ge 0, x_3 \ge 0, x_4 \ge 0,\ x_5 \ge 0
\end{array}
$$

EXAMPLE 1: EXTENDING THE LP

- Furthermore, suppose you have a budgetary limit
    - 1 lb of bread costs you $3
    - 1 lb of steak costs you $6
    - 1 lb of chicken costs you $7
    - 1 lb of chocolate costs you $15
    - 1 lb of cheese costs you $8
  - You can spend at most $150
  - Write the new restriction:

EXAMPLE 1: EXTENDING THE LP

$$\begin{aligned} \textit{Maximize} \quad & 2x_1+5x_2+3x_3+6x_4+4x_5 \\ \textit{subject to} \quad & x_1+x_2+x_3+x_4+x_5 \leq 50 \\ & 3x_1+6x_2+7x_3+15x_4+8x_5 \leq 150 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0,\ x_5 \geq 0 \end{aligned}$$

- You cannot get more meat (chicken+steak) than bread

$$x_2 + x_3 \leq x_1$$

- For each lb of cheese, you need at least 2 lbs of bread

$$2x_5 \leq x_1$$

SUMMARY OF FORMULATION STEPS

- Gather all of your data, know what they mean
- Then
  1. Identify your decision variables
  2. Identify your objective and objective function
  3. Identify your constraints
     - Express your objective function and constraints in terms of your decision variables
     - Steps 2 and 3 can change order

CASE 1: WYNDOR GLASS CO. PRODUCT

- Wyndor Glass Co. produces high-quality glass products
  - The company decides to produce two new products
    - A glass door with aluminum framing
    - A wood-framed glass window

  - The company has 3 plants

	Production Time Used for Each Unit Produced		
Plant	Doors	Windows	Availability/week
1	1 hour	0	4 hours
2	0	2 hours	12 hours
3	3 hours	2 hours	18 hours

Optimization Techniques

CASE 1: WYNDOR GLASS CO. PRODUCT

- Unit profit for doors is $300
- Unit profit for windows is $500

- What should be the product mix to maximize profits?
  - How many of each item to produce weekly?
    - This is production rate, so it can be continuous, i.e., you can choose to produce 2.5 windows per week (this would be 10 windows per month assuming 4 weeks in a month)

Optimization Techniques

CASE 1: FORMULATING THE LP

- Step 0: Gather the data (we have it all!)
- Step 1: Decision variables
  - *D:* the number of doors produced weekly
  - *W:* the number of windows produced weekly
- Step 2: Define the objective & objective functions
  - Objective: Maximization
  - Objective function: *Profit=300D+500W*
    - *It is the weekly profit*

Optimization Techniques

CASE 1: FORMULATING THE LP

- Step 3: Define the constraints
  - Plant 1 capacity: $D \leq 4$
  - Plant 2 capacity: $2W \leq 12$
  - Plant 3 capacity: $3D+2W \leq 18$
- Combine what you have!

$$
\begin{aligned}
\text{Maximize } & 300D+500W \\
\text{subject to } & D \leq 4 \\
& 2W \leq 12 \\
& 3D+2W \leq 18 \\
& D \geq 0,\ W \geq 0
\end{aligned}
$$

Do not forget the non-negativity constraints

LP TERMINOLOGY

- Decision Variables: things we control
- Objective function: measure of performance
- Nonnegativity constraints
- Functional constraints: restrictions we have
- Parameters: constants we use in the objective function and constraint definitions
- Solution: any choice of values for the decision variables
  - Feasible solution is one that satisfies the constraints
  - Optimal solution is the best feasible solution

INTRODUCTION AND BASIC CONCEPTS

Classification of Optimization Problems

INTRODUCTION

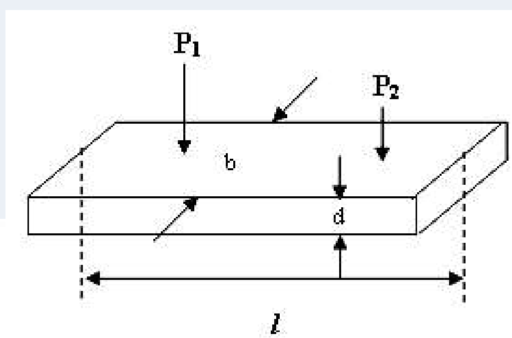
- Optimization problems can be classified based on the type of constraints, nature of design variables, physical structure of the problem, nature of the equations involved, deterministic nature of the variables, permissible value of the design variables, separability of the functions and number of objective functions. These classifications are briefly discussed in this lecture.

CLASSIFICATION BASED ON EXISTENCE OF CONSTRAINTS.

- **Constrained optimization problems:** which are subject to one or more constraints.
- **Unconstrained optimization problems**: in which no constraints exist.

CLASSIFICATION BASED ON THE NATURE OF THE DESIGN VARIABLES

- There are two broad categories of classification within this classification
- First category : the objective is to find a set of design parameters that make a prescribed function of these parameters minimum or maximum subject to certain constraints.
  - For example to find the minimum weight design of a strip footing with two loads shown in the figure, subject to a limitation on the maximum settlement of the structure.



The problem can be defined as follows

$$\text{Find } \mathbf{X} = \begin{Bmatrix} b \\ d \end{Bmatrix} \text{ which minimizes}$$

$$f(\mathbf{X}) = \boldsymbol{h}(b,d)$$

subject to the constraints

$$\delta_s(\mathbf{X}) \le \delta_{\max}$$

$$b \ge 0$$

$$d \ge 0$$

The length of the footing (l) the loads P1 and P2 , the distance between the loads are assumed to be constant and the required optimization is achieved by varying b and d. Such problems are called *parameter* or *static* optimization problems.

CLASSIFICATION BASED ON THE NATURE OF THE DESIGN VARIABLES (CONTD.)

- Second category: the objective is to find a set of design parameters, which are all continuous functions of some other parameter, that minimizes an objective function subject to a set of constraints.

CLASSIFICATION BASED ON THE PHYSICAL STRUCTURE OF THE PROBLEM

- **Based on the physical structure, we can classify optimization problems are classified as optimal control and non-optimal control problems.**

  (i) An ***optimal control*** **(OC)** problem is a mathematical programming problem involving a number of stages, where each stage evolves from the preceding stage in a prescribed manner.

  - It is defined by two types of variables: the control or design variables and state variables.

- The problem is to find a set of control or design variables such that the total objective function (also known as the performance index, PI) over all stages is minimized subject to a set of constraints on the control and state variables. An OC problem can be stated as follows:

$$\text{Find } \mathbf{X} \text{ which minimizes } f(\mathbf{X}) = \sum_{i=1}^{l} f_i(x_i, y_i)$$

subject to the constraints:

$$q_i(x_i, y_i) + y_i = y_{i+1} \qquad i = 1, 2, \ldots, l$$

$$g_j(x_j) \le 0, \qquad j = 1, 2, \ldots, l$$

$$h_k(y_k) \le 0, \qquad k = 1, 2, \ldots, l$$

- Where x_i is the i^{th} control variable, y_i is the i^{th} state variable, and f_i is the contribution of the i^{th} stage to the total objective function. **g_j, h_k**, and q_i are the functions of xj, yj ; x_k, y_k and x_i and y_i, respectively, and l is the total number of states. The control and state variables x_i and y_i can be vectors in some cases.

(ii) The problems which are not *optimal control problems* are called *non-optimal control problems.*

Optimization Techniques

CLASSIFICATION BASED ON THE NATURE OF THE EQUATIONS INVOLVED

- **Based on the nature of expressions for the objective function and the constraints, optimization problems can be classified as linear, nonlinear, geometric and quadratic programming problems.**

CLASSIFICATION BASED ON THE NATURE OF THE EQUATIONS INVOLVED (CONTD.)

(i) *Linear programming problem*

- If the objective function and all the constraints are linear functions of the design variables, the mathematical programming problem is called a linear programming (LP) problem.

often stated in the standard form :

$$\text{Find } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$$

$$\text{Which maximizes } f(\mathbf{X}) = \sum_{i=1}^{n} c_i x_i$$

subject to

$$\sum_{i=1}^{n} a_{ij} x_i = b_j, \qquad j = 1, 2, \ldots, m$$

$$x_i \geq 0, \qquad j = 1, 2, \ldots, m$$

where c_i, a_{ij}, and b_j are constants.

CLASSIFICATION BASED ON THE NATURE OF THE EQUATIONS INVOLVED (CONTD.)

(ii) *Nonlinear programming problem*

- If any of the functions among the objectives and constraint functions is nonlinear, the problem is called a nonlinear programming (NLP) problem this is the most general form of a programming problem.

CLASSIFICATION BASED ON THE NATURE OF THE EQUATIONS INVOLVED (CONTD.)

(iii) *Geometric programming problem*

- A geometric programming (GMP) problem is one in which the objective function and constraints are expressed as polynomials in X.

  A polynomial with N terms can be expressed as

$$h(\mathbf{X}) = c_i x_1^{ai1} x_2^{ai2} \ldots x_n^{ain} + \ldots + c_N x_1^{aN1} x_2^{aN2} \ldots x_n^{aNn}$$

- Thus GMP problems can be expressed as follows: Find **X** which minimizes :

subje

$$f(\mathbf{X}) = \sum_{i=1}^{N_0} c_i \left(\prod_{j=1}^{n} x_j^{p_{ij}} \right), \qquad c_i > 0, \quad x_j > 0$$

$$g_k(\mathbf{X}) = \sum_{i=1}^{N_k} a_{ik} \left(\prod_{j=1}^{n} x_j^{q_{ijk}} \right) > 0, \qquad a_{ik} > 0, \quad x_j > 0, k = 1,2,\ldots,m$$

CLASSIFICATION BASED ON THE NATURE OF THE EQUATIONS INVOLVED (CONTD.)

- where N_0 and N_k denote the number of terms in the objective and k^{th} constraint function, respectively.

(iv) *Quadratic programming problem*

- A quadratic programming problem is the best behaved nonlinear programming problem with a quadratic objective function and linear constraints and is concave (for maximization problems). It is usually formulated as follows:

$$F(\mathbf{X}) = c + \sum_{i=1}^{n} q_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} x_i x_j$$

Subject to:

$$\sum_{i=1}^{n} a_{ij} x_i = b_j, \qquad j = 1,2,\ldots,m$$

$$x_i \geq 0, \qquad i = 1,2,\ldots,n$$

where c, q_i, Q_{ij}, a_{ij}, and b_j are constants.

CLASSIFICATION BASED ON THE PERMISSIBLE VALUES OF THE DECISION VARIABLES

- **Under this classification problems can be classified as integer and real-valued programming problems**

(i) ***Integer programming problem***

- If some or all of the design variables of an optimization problem are restricted to take only integer (or discrete) values, the problem is called an integer programming problem.

(ii) ***Real-valued programming problem***

- A real-valued problem is that in which it is sought to minimize or maximize a real function by systematically choosing the values of real variables from within an allowed set. When the allowed set contains only real values, it is called a real-valued programming problem.

CLASSIFICATION BASED ON DETERMINISTIC NATURE OF THE VARIABLES

- **Under this classification, optimization problems can be classified as deterministic and stochastic programming problems**

(i) ***Deterministic programming problem***

- In this type of problems all the design variables are deterministic.

(ii) ***Stochastic programming problem***

- In this type of an optimization problem some or all the parameters (design variables and/or pre-assigned parameters) are probabilistic (non deterministic or stochastic). For example estimates of life span of structures which have probabilistic inputs of the concrete strength and load capacity. A deterministic value of the life-span is non-attainable.

CLASSIFICATION BASED ON SEPARABILITY OF THE FUNCTIONS

- **Based on the separability of the objective and constraint functions optimization problems can be classified as separable and non-separable programming problems**

  (i) ***Separable programming problems***

  - In this type of a problem the objective function and the constraints are separable. A function is said to be separable if it can be expressed as the sum of n single-va rogramming problem (as :

$$\text{Find } \mathbf{X} \text{ which minimizes } f(X) = \sum_{i=1}^{n} f_i(x_i)$$

subject to : $g_j(X) = \sum_{i=1}^{n} g_{ij}(x_i) \le b_j\ , \quad j = 1, 2, \ldots, m$

where b_j is a constant.

CLASSIFICATION BASED ON THE NUMBER OF OBJECTIVE FUNCTIONS

- **Under this classification objective functions can be classified as single and multiobjective programming problems.**

  (i) ***Single-objective programming problem*** in which there is only a single objective.

  (ii) ***Multi-objective programming problem***

  - A multiobjective programming problem can be stated as follows:

$$\text{Find } \mathbf{X} \text{ which minimizes } f_1(X), f_2(X), \ldots f_k(X)$$

subject to : $g_j(X) \le 0, \quad j = 1, 2, \ldots, m$

  - where $f_1, f_2, \ldots f_k$ denote the objective functions to be minimized simultaneously.

INTRODUCTION AND BASIC CONCEPTS

Classical and Advanced Techniques for Optimization

CLASSICAL OPTIMIZATION TECHNIQUES

- The classical optimization techniques are useful in finding the optimum solution or unconstrained maxima or minima of continuous and differentiable functions.
- These are analytical methods and make use of differential calculus in locating the optimum solution.
- The classical methods have limited scope in practical applications as some of them involve objective functions which are not continuous and/or differentiable.
- Yet, the study of these classical techniques of optimization form a basis for developing most of the numerical techniques that have evolved into advanced techniques more suitable to today's practical problems

CLASSICAL OPTIMIZATION TECHNIQUES (CONTD.)

- These methods assume that the function is differentiable twice with respect to the design variables and the derivatives are continuous.
- Three main types of problems can be handled by the classical optimization techniques:
  - single variable functions
  - multivariable functions with no constraints,
  - multivariable functions with both equality and inequality constraints. In problems with equality constraints the Lagrange multiplier method can be used. If the problem has inequality constraints, the Kuhn-Tucker conditions can be used to identify the optimum solution.
- These methods lead to a set of nonlinear simultaneous equations that may be difficult to solve.

NUMERICAL METHODS OF OPTIMIZATION

- **Linear programming**: studies the case in which the objective function *f* is linear and the set A is specified using only linear equalities and inequalities. (A is the design variable space)
- **Integer programming**: studies linear programs in which some or all variables are constrained to take on integer values.
- **Quadratic programming**: allows the objective function to have quadratic terms, while the set A must be specified with linear equalities and inequalities
- **Nonlinear programming**: studies the general case in which the objective function or the constraints or both contain nonlinear parts.

NUMERICAL METHODS OF OPTIMIZATION (CONTD.)

- **Stochastic programming**: studies the case in which some of the constraints depend on random variables.
- **Dynamic programming**: studies the case in which the optimization strategy is based on splitting the problem into smaller sub-problems.
- **Combinatorial optimization**: is concerned with problems where the set of feasible solutions is discrete or can be reduced to a discrete one.
- **Infinite-dimensional optimization**: studies the case when the set of feasible solutions is a subset of an infinite-dimensional space, such as a space of functions.
- **Constraint satisfaction**: studies the case in which the objective function *f* is constant (this is used in artificial intelligence, particularly in automated reasoning).

ADVANCED OPTIMIZATION TECHNIQUES

- **Hill climbing**: it is a graph search algorithm where the current path is extended with a successor node which is closer to the solution than the end of the current path.
- In **simple hill climbing**, the first closer node is chosen whereas in **steepest ascent hill climbing** all successors are compared and the closest to the solution is chosen. Both forms fail if there is no closer node. This may happen if there are local maxima in the search space which are not solutions.
- Hill climbing is used widely in artificial intelligence fields, for reaching a goal state from a starting node. Choice of next node/ starting node can be varied to give a number of related algorithms.

SIMULATED ANNEALING

- The name and inspiration come from annealing process in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects.

  - *The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy;*
  - the slow cooling gives them more chances of finding configurations with *lower internal energy* than the initial one.

- In the **simulated annealing** method, each point of the search space is compared to a state of some physical system, and the function to be minimized is interpreted as the internal energy of the system in that state. Therefore the goal is to bring the system, from an arbitrary initial state, to a state with the *minimum possible energy*.

GENETIC ALGORITHMS

- A **genetic algorithm (GA)** is a local search technique used to find approximate solutions to optimization and search problems.

- Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as *inheritance, mutation, selection, and crossover* (also called recombination).

Optimization Techniques

GENETIC ALGORITHMS (CONTD.)

- Genetic algorithms are typically implemented as a computer simulation, in which a population of abstract representations (called *chromosomes*) of candidate solutions (called individuals) to an optimization problem, evolves toward better solutions.

- The evolution starts from a population of completely random individuals and occurs in generations.

- In each generation, the fitness of the whole population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (mutated or recombined) to form a new population.

- The new population is then used in the next iteration of the algorithm.

Optimization Techniques

ANT COLONY OPTIMIZATION

- In the real world, ants (initially) wander randomly, and upon finding food return to their colony while laying down pheromone trails. If other ants find such a path, they are likely not to keep traveling at random, but instead follow the trail laid by earlier ants, returning and reinforcing it if they eventually find food

- Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate.

- A short path, by comparison, gets marched over faster, and thus the pheromone density remains high

- Pheromone evaporation has also the advantage of avoiding the convergence to a locally optimal solution. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the solution space would be constrained.

ANT COLONY OPTIMIZATION (CONTD.)

- Thus, when one ant finds a good (short) path from the colony to a food source, other ants are more likely to follow that path, and such positive feedback eventually leaves all the ants following a single path.
- The idea of the ant colony algorithm is to mimic this behavior with "simulated ants" walking around the search space representing the problem to be solved.
- Ant colony optimization algorithms have been used to produce near-optimal solutions to the traveling salesman problem.
- They have an advantage over simulated annealing and genetic algorithm approaches when the graph may change dynamically. The ant colony algorithm can be run continuously and can adapt to changes in real time.
- This is of interest in network routing and urban transportation systems.