Introduction and Basic Concepts

(iii) 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.

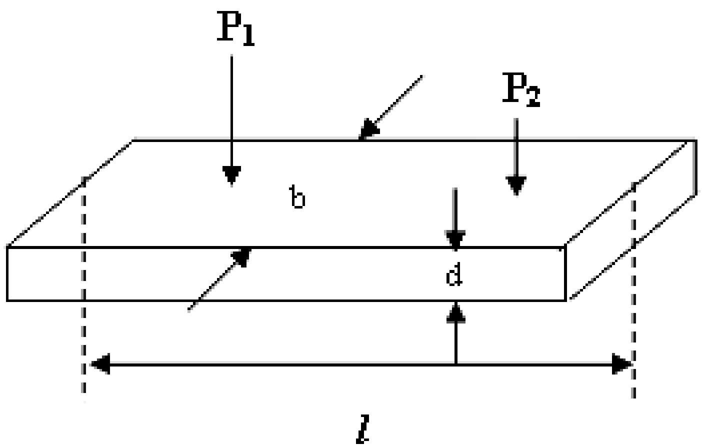
Classification based on existence of constraints.

- **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

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

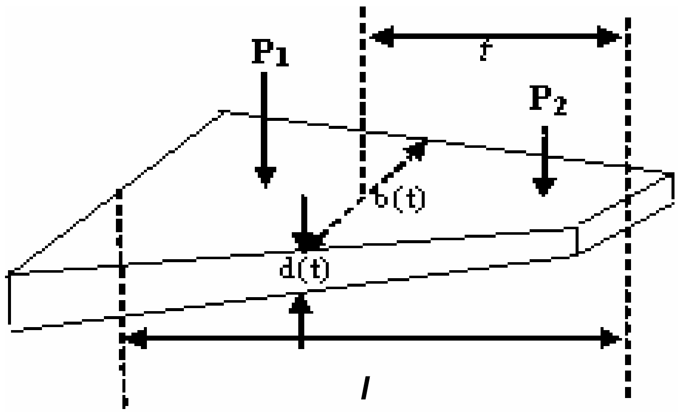
$$b \geq 0$$

$$d \geq 0$$

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.
  - For example, if the cross sectional dimensions of the rectangular footing is allowed to vary along its length as shown in the following figure.



The problem can be defined as follows

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

$$f(\mathbf{X}) = \mathbf{g}(\, b(t), d(t) \,)$$

subject to the constraints

$$\delta_s(\,\mathbf{X}(t)\,) \leq \delta_{max} \qquad 0 \leq t \leq l$$

$$b(t) \geq 0 \qquad 0 \leq t \leq l$$

$$d(t) \geq 0 \qquad 0 \leq t \leq l$$

The length of the footing (l) the loads P_1 and P_2 , 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 *trajectory* or *dynamic* optimization problems.

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. $\mathbf{g_j}$, $\mathbf{h_k}$, 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.*

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 :

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

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

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

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

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

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

subject to:

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

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:

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

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-variable functions and separable programming problem can be expressed in standard form as :

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

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

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 : $g_j(X) \leq 0, \qquad j = 1, 2, \ldots, m$

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

Thank You