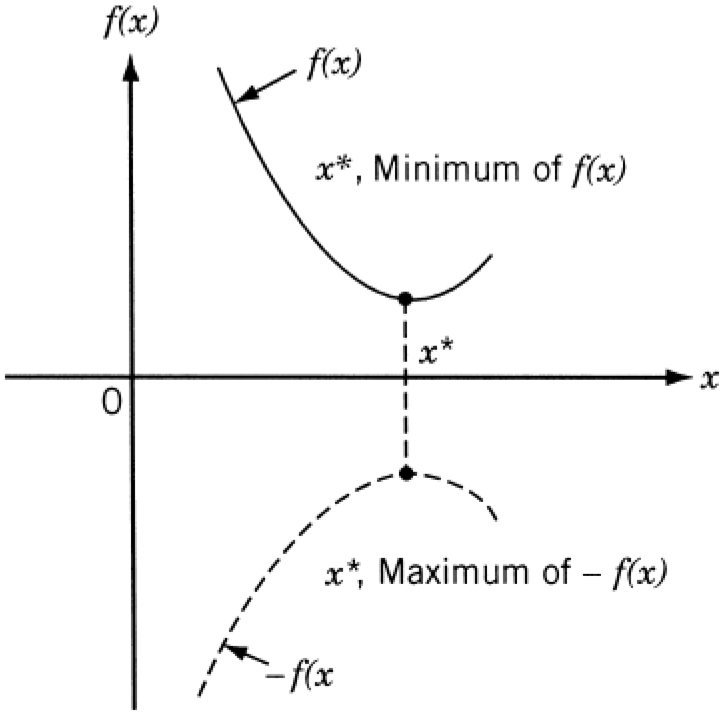
Advanced Optimization Techniques

Optimization

- Optimization can be defined as the process of finding the conditions that give the maximum or minimum value of a function.



Operations research

- **Branch of mathematics concerned with the application of scientific methods and techniques to decision making problems and with establishing the best or optimal solutions.**

- **The optimum seeking methods are also known as mathematical programming techniques and are generally studied as a part of operations research.**

Operations research

- Early period of World War II. During the war, the British military faced the problem of allocating very scarce and limited resources (such as fighter airplanes, radars, and submarines) to several activities (deployment to numerous targets and destinations).
- Because there were no systematic methods available to solve resource allocation problems, the military called upon a team of mathematicians to develop methods for solving the problem in a scientific manner.
- The methods developed by the team were instrumental in the winning of the Air Battle by Britain. These methods, such as linear programming, which were developed as a result of research on (military) operations, subsequently became known as the methods of operations research.

Methods of Operations Research

Mathematical programming or optimization 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		
Modern or nontraditional optimization techniques		
Genetic algorithms		
Simulated annealing		
Ant colony optimization		
Particle swarm optimization		
Neural networks		
Fuzzy optimization		

Applications

- Knapsack problem.
- Travelling sales man problem.
- Job assignment problem.
- Weapon target assignment problem.
- Vehicle routing problem

OPTIMIZATION PROBLEM

An optimization or a mathematical programming problem can be stated as follows.

$$\text{Find } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ which minimizes } f(\mathbf{X})$$

subject to the constraints

$$g_j(\mathbf{X}) \le 0, \qquad j = 1, 2, \ldots, m$$

$$l_j(\mathbf{X}) = 0, \qquad j = 1, 2, \ldots, p$$

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

constrained optimization problem.

OPTIMIZATION PROBLEM

$$\text{Find } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ which minimizes } f(\mathbf{X})$$

Such problems are called *unconstrained optimization problems*.

Design Vector

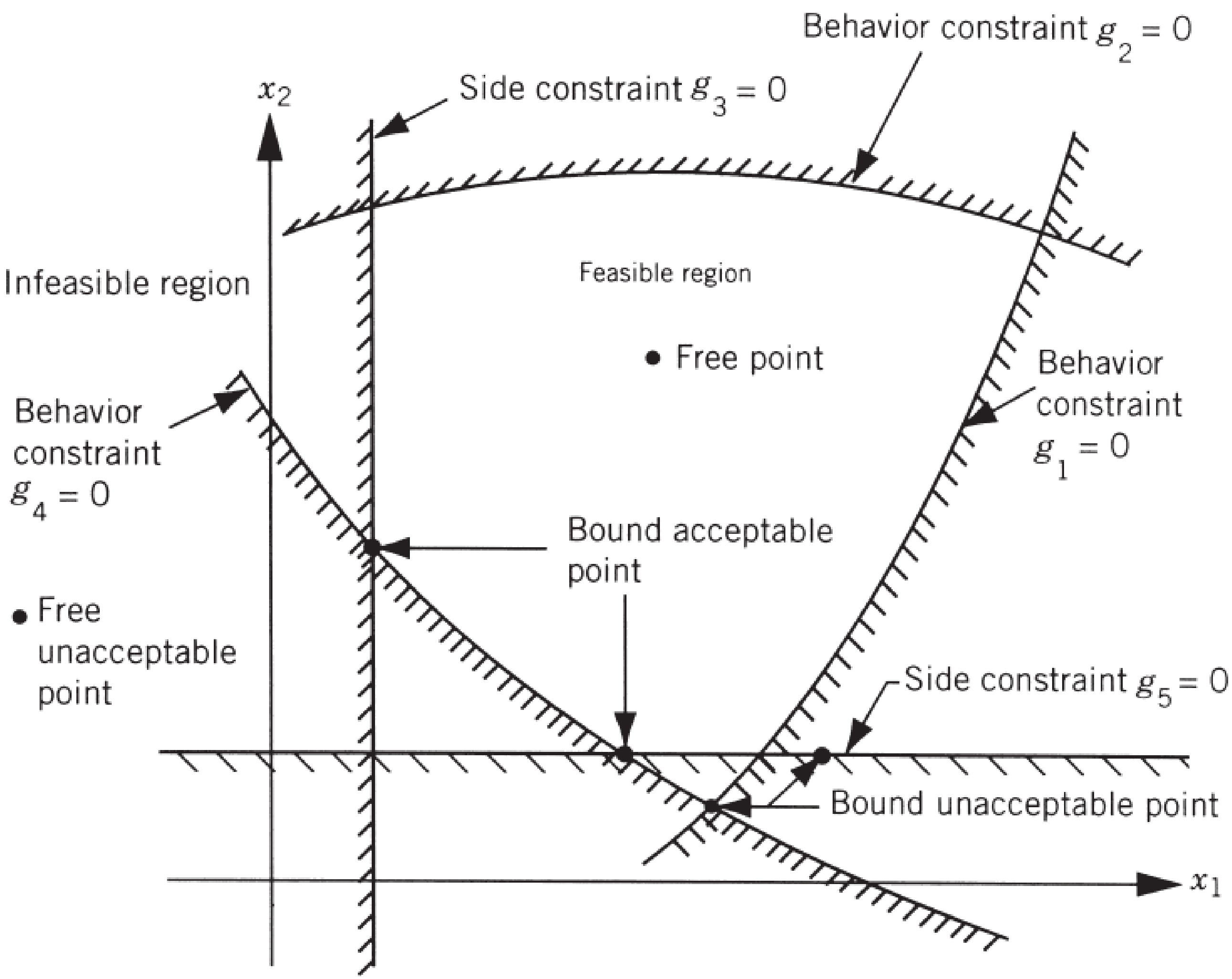
- Any engineering system or component is defined by **a set of quantities** some of which are viewed as variables during the design process.
- In general, certain quantities are usually fixed at the outset and these are called **preassigned parameters**.
- All the other quantities are treated as variables in the design process and are called **design or decision variables**

$$\mathbf{X} = \{x_1, x_2, \ldots, x_n\}^{\mathrm{T}}$$

Design Constraints

- The **restrictions** that must be satisfied to produce an **acceptable design** are collectively called ***design constraints***.
- **Constraint Surface**
  - Consider an optimization problem with only inequality constraints. $g_j(X) \geq 0$
  - The set of values of X that satisfy the equation $g_j(X) = 0$ forms a hyper-surface in the design space and is called a **constraint surface**.

Constraint surfaces in a hypothetical two-dimensional design space.



1. Free and acceptable point
2. Free and unacceptable point
3. Bound and acceptable point
4. Bound and unacceptable point

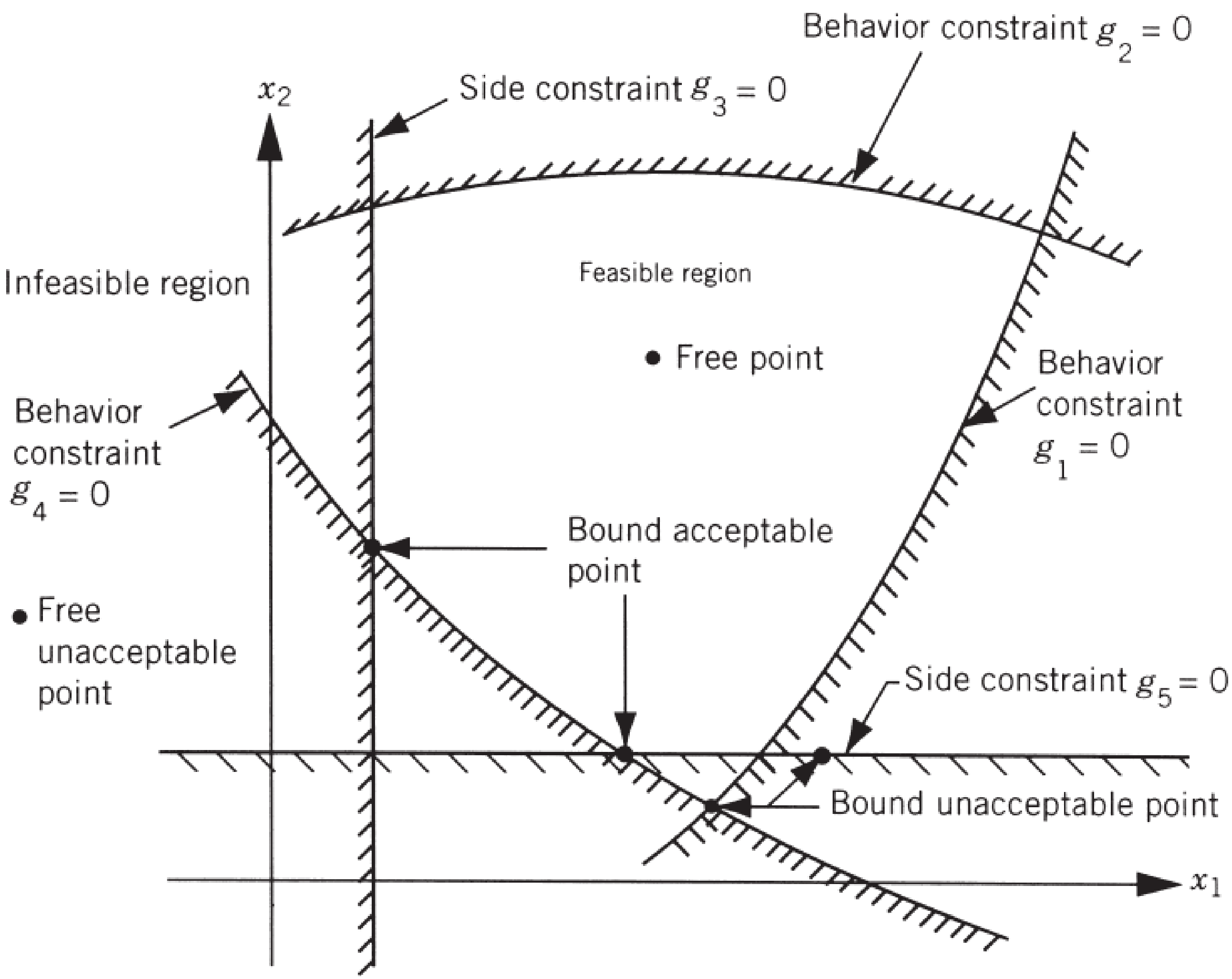
Design Constraints

- **Constraint Surface**
  - The constraint surface divides the design space into two regions

$g_j(X) > 0$ **Infeasible or unacceptable**

$g_j(X) < 0$ **feasible or acceptable**

Constraint surfaces in a hypothetical two-dimensional design space.

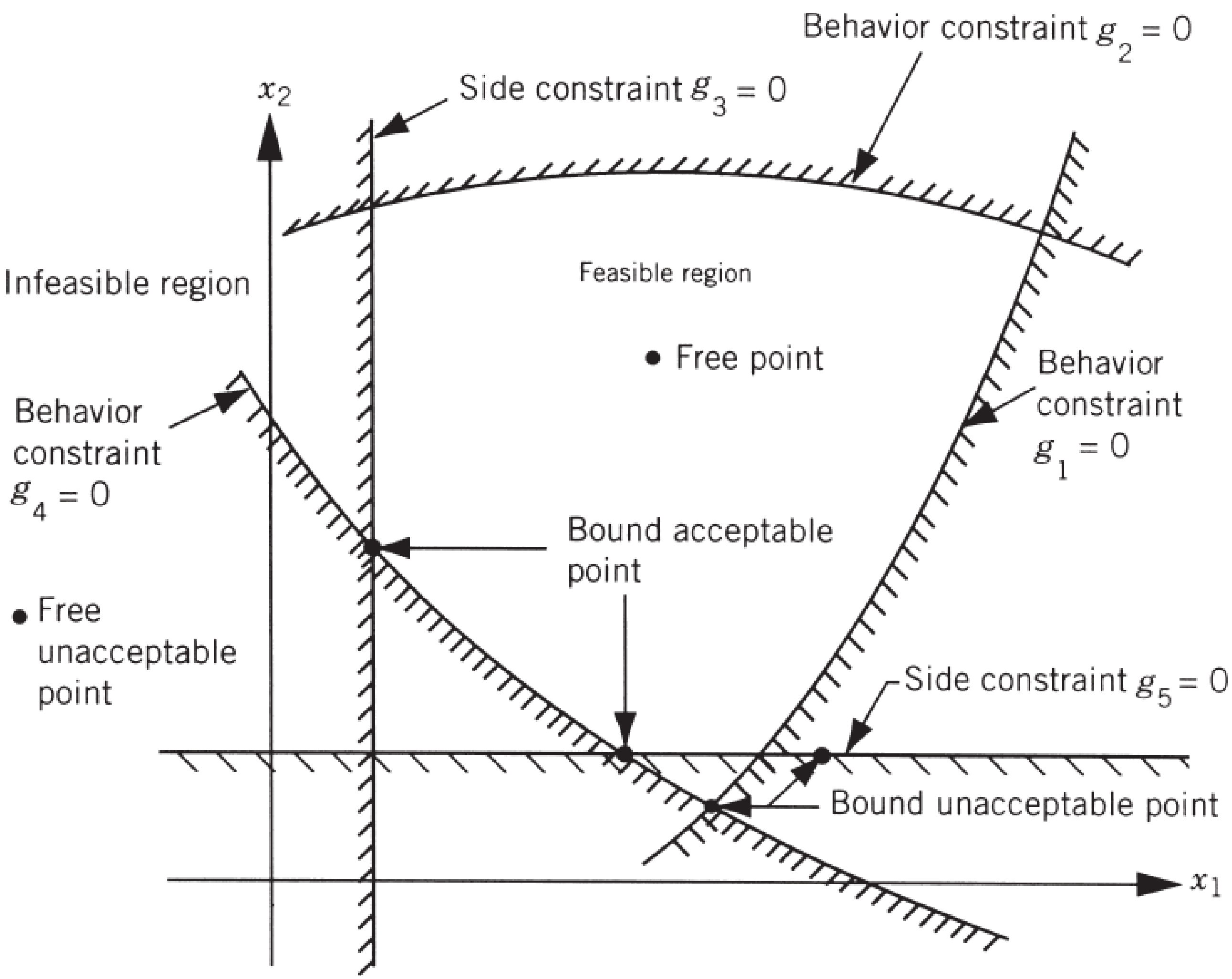


1. Free and acceptable point
2. Free and unacceptable point
3. Bound and acceptable point
4. Bound and unacceptable point

Design Constraints

- **Composite Constraint Surface**
  - The **collection of all the constraint surfaces** $g_j(X) = 0$, j = 1, 2, . . . , m, which **separates the acceptable region** is called the **composite constraint surface**.

Constraint surfaces in a hypothetical two-dimensional design space.

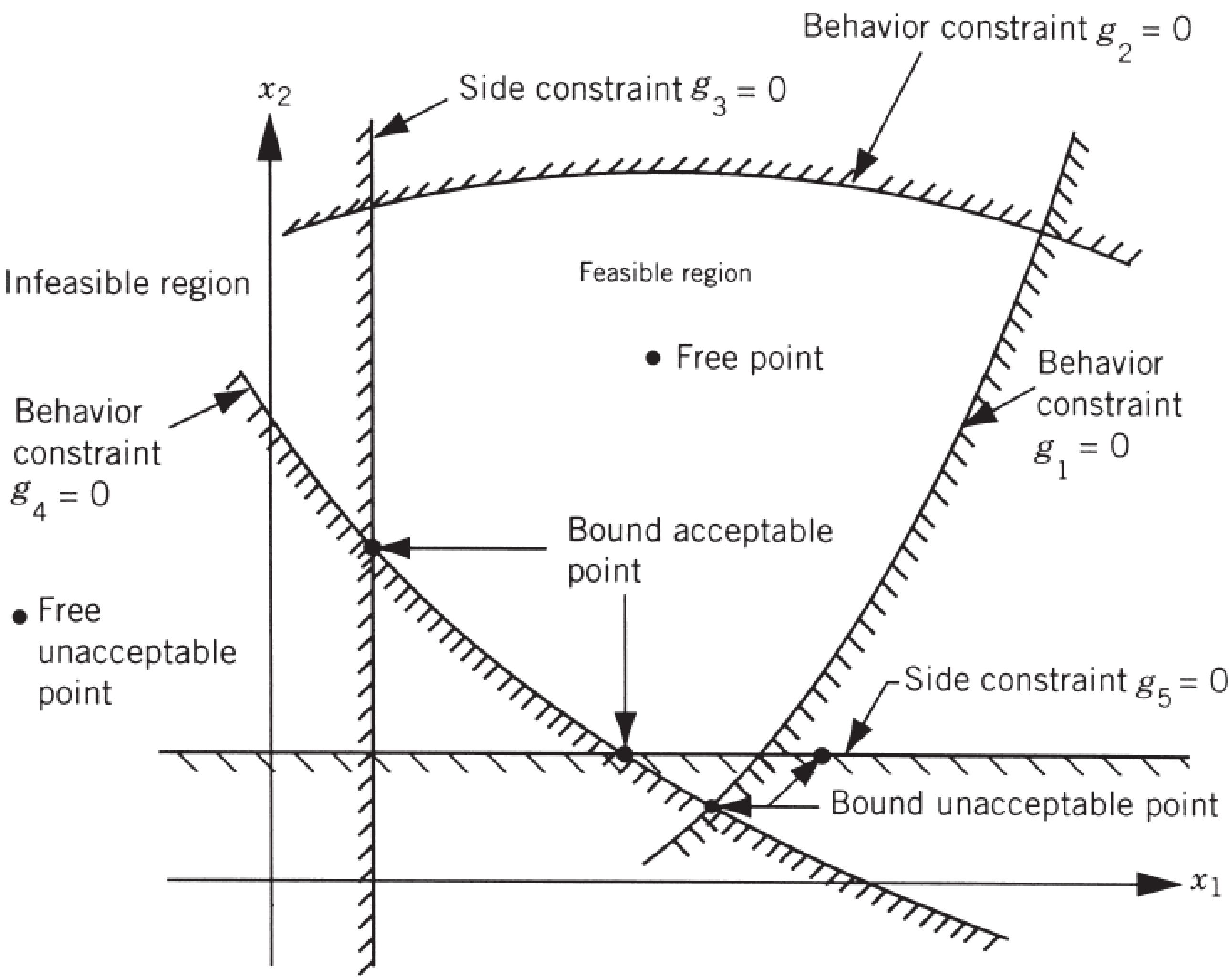


1. Free and acceptable point
2. Free and unacceptable point
3. Bound and acceptable point
4. Bound and unacceptable point

Design Constraints

- **Bound point and Active constraint**
  - A design point that lies on **one or more than** one constraint surface is called a **bound point,** and the **associated constraint** is called an **active constraint**.
- **free points**
  - Design points that do not lie on any constraint surface are known as **free points**.

Constraint surfaces in a hypothetical two-dimensional design space.



1. Free and acceptable point
2. Free and unacceptable point
3. Bound and acceptable point
4. Bound and unacceptable point

Objective Function

- The criterion with respect to which the **design is optimized**, when expressed as a **function of the design variables**, is known as the **criterion or merit or objective function**.
- The choice of objective function is governed by the nature of problem.

Objective Function

- An optimization problem involving multiple objective functions is known as a **multi-objective programming problem**.
- With multiple objectives there arises a **possibility of conflict**, and one simple way to handle the problem is to construct an overall objective function as a **linear combination** of the conflicting multiple objective functions.

$$f(\mathbf{X}) = \alpha_1 f_1(\mathbf{X}) + \alpha_2 f_2(\mathbf{X})$$

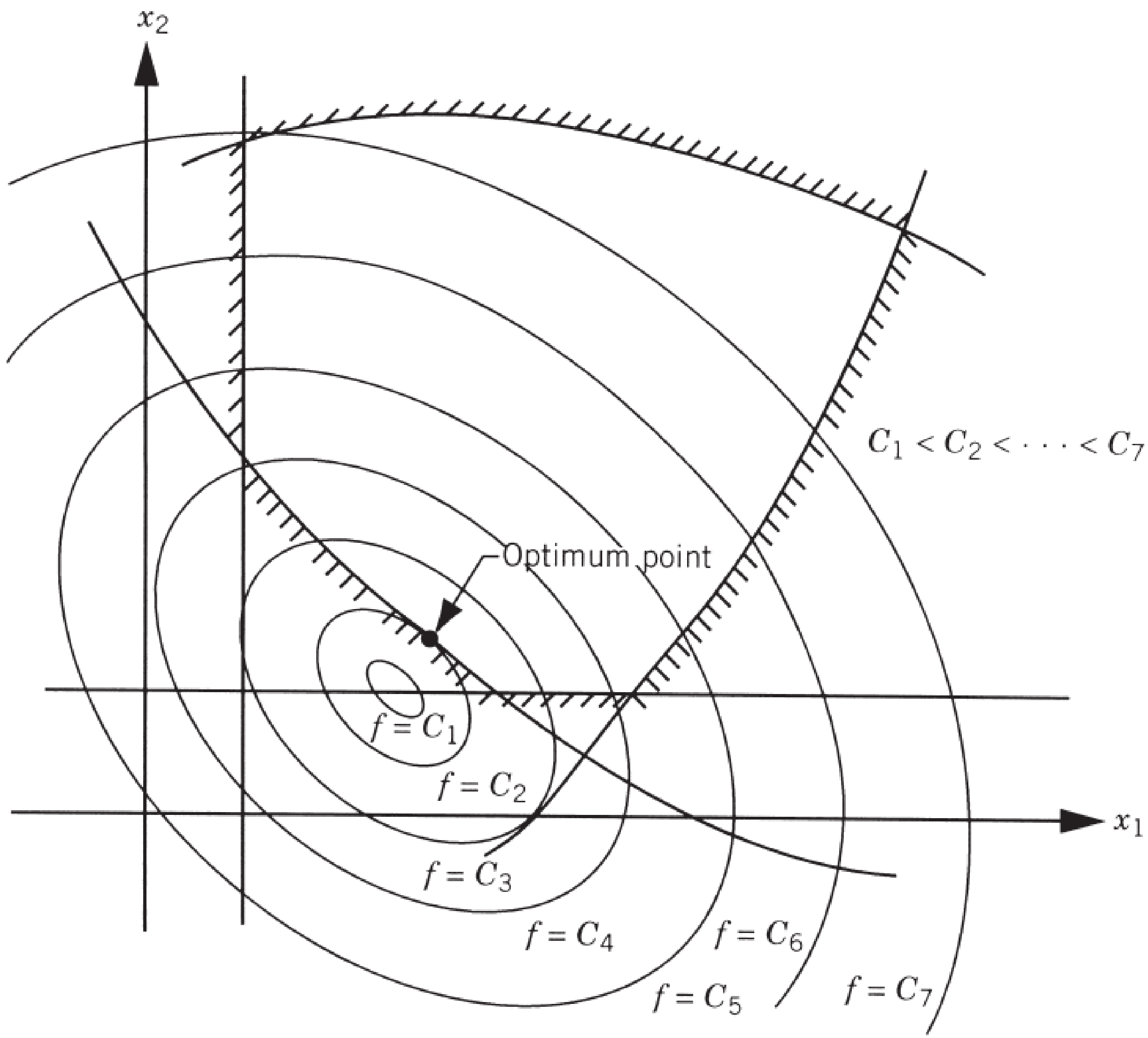
Objective Function Surfaces

- The locus of all points satisfying

  $f(X) = C = \text{constant}$

  forms a **hyper surface in the design space**, and each value of C corresponds to a different member of a family of surfaces. These surfaces, called **objective function surfaces**

Objective Function Surfaces



Traveling salesman problems

- Traveling salesman problems
  - Given a **set of cities** and a **cost** to travel from one city to another, seeks to identify the tour that will allow a salesman to **visit each city only once, starting and ending in the same city**, at the **minimum cost**.

$$x_{ij} = \begin{cases} 1 & \text{the path goes from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases}$$

$$\min \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij}:$$

$$0 \leq x_{ij} \leq 1 \qquad i, j = 1, \ldots, n;$$

$$i = 1, \ldots, n;$$

$$\sum_{i=1, i \neq j}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n;$$

$$\sum_{i,j \in S,\, i \neq j} x_{ij} \leq |S| - 1, \quad \forall S \subset V, S \neq \emptyset$$

$$\sum_{j=1, j \neq i}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n;$$

Traveling salesman problems

- Traveling salesman problems
  - Given a **set of cities** and a **cost** to travel from one city to another, seeks to identify the tour that will allow a salesman to **visit each city only once, starting and ending in the same city**, at the **minimum cost**.

$$x_{ij} = \begin{cases} 1 & \text{the path goes from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases}$$

$$\min \sum_{i=1}^{n} \sum_{j\neq i, j=1}^{n} c_{ij} x_{ij}:$$

$$0 \leq x_{ij} \leq 1 \qquad i, j = 1, \ldots, n;$$

$$i = 1, \ldots, n;$$

$$\sum_{i=1, i\neq j}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n;$$

Each city be arrived at from exactly one other city

$$\sum_{j=1, j\neq i}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n;$$

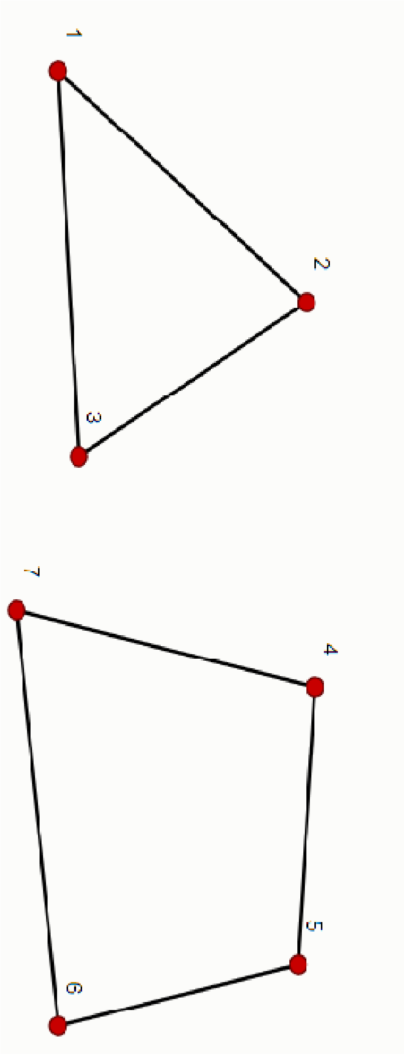
each city there is a departure to exactly one other city.

$$\sum_{i,j\in S,\, i\neq j} x_{ij} \leq |S| - 1, \quad \forall S \subset V, S \neq \emptyset$$

There is only a single tour covering all cities

Traveling salesman problems

$$x_{ij} = \begin{cases} 1 & \text{the path goes from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases}$$

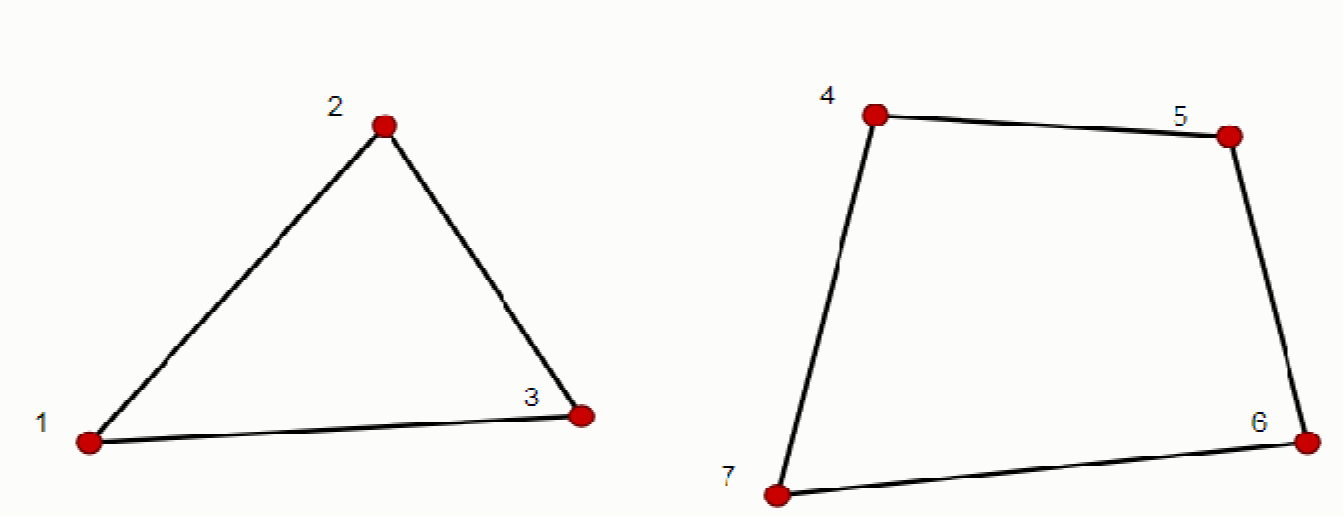


$$\begin{aligned} \min \sum_{i=1}^{n} \sum_{j\neq i, j=1}^{n} c_{ij} x_{ij} &\colon \\ 0 \leq x_{ij} \leq 1 & \qquad i, j = 1, \ldots, n; \\ u_i \in \mathbf{Z} & \qquad i = 1, \ldots, n; \\ \sum_{i=1, i\neq j}^{n} x_{ij} = 1 & \qquad j = 1, \ldots, n; \\ \sum_{j=1, j\neq i}^{n} x_{ij} = 1 & \qquad i = 1, \ldots, n; \end{aligned}$$

$$\sum_{i,j \in S,\, i\neq j} x_{ij} \leq |S| - 1, \quad \forall S \subset V, S \neq \emptyset$$

Traveling salesman problems

$$\sum_{i,j \in S,\, i \neq j} x_{ij} \leq |S| - 1, \quad \forall S \subset V, S \neq \emptyset$$



$$\sum_{i,j \in \{1,2,3\}, i \neq j} x_{ij} = 3 > 2 = |\{1,2,3\}| - 1$$

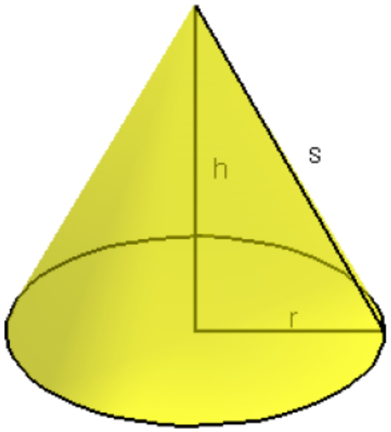
Thus, the subtour elimination constraint above is violated.

multi-objective programming problem

Right circular cone:

r = base radius
h = height
s = slant height

V = volume
B = base area
S = lateral surface area
T = total area



$$s = \sqrt{r^2 + h^2}$$

$$V = \frac{\pi}{3} r^2 h$$

$$B = \pi r^2$$

$$S = \pi r s$$

$$T = B + S = \pi r (r + s)$$

Cones problem

- two input variables: r, h

$$r \in [0, 10] \text{ cm} \;, \quad h \in [0, 20] \text{ cm}$$

- two objectives:

$$\min S$$
$$\min T$$

- one constraint:

$$V > 200 \text{ cm}^3$$

The cone shape (i.e. the design) is defined univocally when both *r* and *h* are given.

We want to minimize both the lateral surface area and the total surface area

A constraint for the cone volume is given, in order to guarantee a minimum volume.