1

Preliminary Concepts

1.1 Introduction

Optimization is the process of maximizing or minimizing a desired objective function while satisfying the prevailing constraints. Nature has an abundance of examples where an optimum system status is sought. In metals and alloys, the atoms take positions of least energy to form unit cells. These unit cells define the crystalline structure of materials. A liquid droplet in zero gravity is a perfect sphere, which is the geometric form of least surface area for a given volume. Tall trees form ribs near the base to strengthen them in bending. The honeycomb structure is one of the most compact packaging arrangements. Genetic mutation for survival is another example of nature's optimization process. Like nature, organizations and businesses have also strived toward excellence. Solutions to their problems have been based mostly on judgment and experience. However, increased competition and consumer demands often require that the solutions be optimum and not just feasible solutions. A small savings in a mass-produced part will result in substantial savings for the corporation. In vehicles, weight minimization can impact fuel efficiency, increased payloads, or performance. Limited material or labor resources must be utilized to maximize profit. Often, optimization of a design process saves money for a company by simply reducing the developmental time.

In order for engineers to apply optimization at their work place, they must have an understanding of both the theory and the algorithms and techniques. This is because there is considerable effort needed to apply optimization techniques on practical problems to achieve an improvement. This effort invariably requires tuning algorithmic parameters, scaling, and even modifying the techniques for the specific application. Moreover, the user may have to try several optimization methods to find one that can be successfully applied. To date, optimization has been used more as a design or decision aid, rather than for concept generation or detailed design. In this sense, optimization is an engineering tool similar to, say, finite element analysis (FEA).

This book aims at providing the reader with basic theory combined with development and use of numerical techniques. A CD-ROM containing computer programs that parallel the discussion in the text is provided. The computer programs give the reader the opportunity to gain hands-on experience. These programs should be valuable to both students and professionals. Importantly, the optimization programs with source code can be integrated with the user's simulation software. The development of the software has also helped to explain the optimization procedures in the written text with greater insight. Several examples are worked out in the text; many of these involve the programs provided. User subroutines to solve some of these examples are also provided on the CD-ROM.

1.2 Historical Sketch

The use of a gradient method (requiring derivatives of the functions) for minimization was first presented by Cauchy in 1847. Modern optimization methods were pioneered by Courant's [1943] paper on penalty functions, Dantzig's paper on the simplex method for linear programming [1951]; and Karush, Kuhn, and Tucker who derived the "KKT" optimality conditions for constrained problems [1939, 1951]. Thereafter, particularly in the 1960s, several numerical methods to solve nonlinear optimization problems were developed. Mixed integer programming received impetus by the branch and bound technique originally developed by Land and Doig [1960] and the cutting plane method by Gomory [1960]. Methods for unconstrained minimization include conjugate gradient methods of Fletcher and Reeves [1964] and the variable metric methods of Davidon–Fletcher–Powell (DFP) in [1959]. Constrained optimization methods were pioneered by Rosen's gradient projection method [1960], Zoutendijk's method of feasible directions [1960], the generalized reduced gradient method by Abadie and Carpenter [1969] and Fiacco and McCormick's SUMT techniques [1968]. Multivariable optimization needed efficient methods for single variable search. The traditional interval search methods using Fibonacci numbers and Golden Section ratio were followed by efficient hybrid polynomial-interval methods of Brent [1971] and others. Sequential quadratic programming (SQP) methods for constrained minimization were developed in the 1970s. Development of interior methods for linear programming started with the work of Karmarkar in 1984. His paper and the related US patent (4744028) renewed interest in interior methods (see the IBM Web site for patent search: http://patent.womplex.ibm.com/).

Also in the 1960s, alongside developments in gradient-based methods, there were developments in nongradient or "direct" methods, principally Rosenbrock's method of orthogonal directions [1960], the pattern search method of Hooke and

Jeeves [1961], Powell's method of conjugate directions [1964], the simplex method of Nelder and Meade [1965], and the method of Box [1965]. Special methods that exploit some particular structure of a problem were also developed. Dynamic programming originated from the work of Bellman who stated the principle of optimal policy for system optimization [1952]. Geometric programming originated from the work of Duffin, Peterson, Zener [1967]. Lasdon [1970] drew attention to large-scale systems. Pareto optimality was developed in the context of multiobjective optimization. More recently, there has been focus on stochastic methods, which are better able to determine global minima. Most notable among these are genetic algorithms [Holland 1975, Goldberg 1989], simulated annealing algorithms that originated from Metropolis [1953], and differential evolution methods [Price and Storn, http://www.icsi.berkeley.edu/~storn/code.html].

In operations research and industrial engineering, use of optimization techniques in manufacturing, production, inventory control, transportation, scheduling, networks, and finance has resulted in considerable savings for a wide range of businesses and industries. Several operations research textbooks are available to the reader. For instance, optimization of airline schedules is an integer program that can be solved using the branch and bound technique [Nemhauser 1997]. Shortest path routines have been used to reroute traffic due to road blocks. The routines may also be applied to route messages on the Internet.

The use of nonlinear optimization techniques in structural design was pioneered by Schmit [1960]. Early literature on engineering optimization are Johnson [1961], Wilde [1967], Fox [1971], Siddall [1972], Haug and Arora [1979], Morris [1982], Reklaitis, Ravindran and Ragsdell [1983], Vanderplaats [1984], Papalambros and Wilde [1988], Banichuk [1990], Haftka and Gurdal [1991]. Several authors have added to this collection including books on specialized topics such as structural topology optimization [Bendsoe and Sigmund 2004], design sensitivity analysis [Haug, Choi and Komkov 1986], optimization using evolutionary algorithms [Deb 2001] and books specifically targeting chemical, electrical, industrial, computer science, and other engineering systems. We refer the reader to the bibliography at end of this book. These, along with several others that have appeared in the last decade, have made an impact in educating engineers to apply optimization techniques. Today, applications are everywhere, from identifying structures of protein molecules to tracing of electromagnetic rays. Optimization has been used for decades in sizing airplane wings. The challenge is to increase its utilization in bringing out the final product.

Widely available and relatively easy to use optimization software packages, popular in universities, include the MATLAB optimization toolbox and the EXCEL SOLVER. Also available are GAMS modeling packages (http://gams.nist.gov/) and CPLEX software (http://www.ilog.com/). Other resources include Web sites maintained by Argonne national labs (http://www-fp.mcs.anl.gov/OTC/Guide/

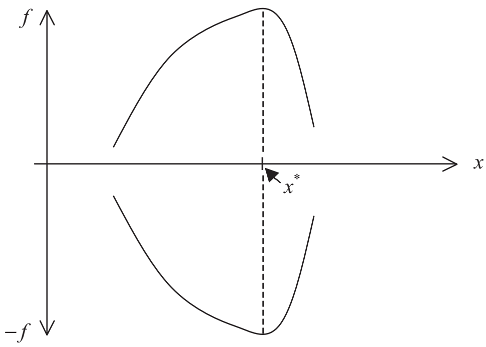


Figure 1.1. Maximization of f is equivalent to minimization of $-f$.

SoftwareGuide/) and by SIAM (http://www.siam.org/). GAMS is tied to a host of optimizers.

Structural and simulation-based optimization software packages that can be procured from companies include ALTAIR (http://www.altair.com/), GENESIS (http://www.vrand.com/), iSIGHT (http://www.engineous.com/), modeFRONTIER (http://www.esteco.com/), and FE-Design (http://www.fe-design.de/en/home.html). Optimization capability is offered in analysis commercial packages such as ANSYS and NASTRAN.

1.3 The Nonlinear Programming Problem

Most engineering optimization problems may be expressed as minimizing (or maximizing) a function subject to inequality and equality constraints, which is referred to as a nonlinear programming (NLP) problem. The word "programming" means "planning." The general form is

$$\begin{array}{lll} \text{minimize} & f(\mathbf{x}) & \\ \text{subject to} & g_i(\mathbf{x}) \le 0 & i = 1, \ldots, m \\ \text{and} & h_j(\mathbf{x}) = 0 & j = 1, \ldots, \ell \\ \text{and} & \mathbf{x}^{\mathrm{L}} \le \mathbf{x} \le \mathbf{x}^{\mathrm{U}} & \end{array} \tag{1.1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ is a column vector of n real-valued *design variables*. In Eq. (1.1), f is the *objective* or *cost* function, g's are *inequality constraints*, and h's are *equality constraints*. The notation $\mathbf{x}^0$ for the starting point, $\mathbf{x}^*$ for optimum, and $\mathbf{x}^k$ for the (current) point at the kth iteration will be generally used.

Maximization versus Minimization

Note that maximization of f is equivalent to minimization of $-f$ (Fig. 1.1).

Problems may be manipulated so as to be in the form (1.1). Vectors $\mathbf{x}^{\mathrm{L}}$, $\mathbf{x}^{\mathrm{U}}$ represent explicit lower and upper bounds on the design variables, respectively, and

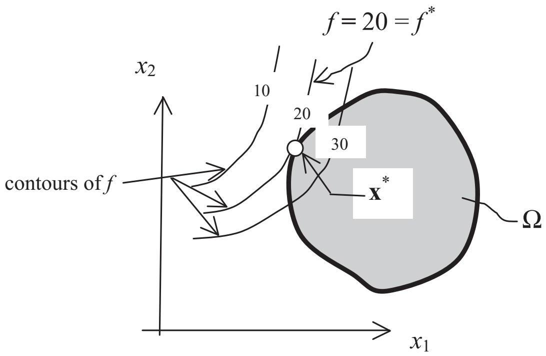


Figure 1.2. Graphical representation of NLP in **x**-space.

are also inequality constraints like the g's. Importantly, we can express (1.1) in the form

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{x} \in \Omega \end{aligned} \tag{1.2}$$

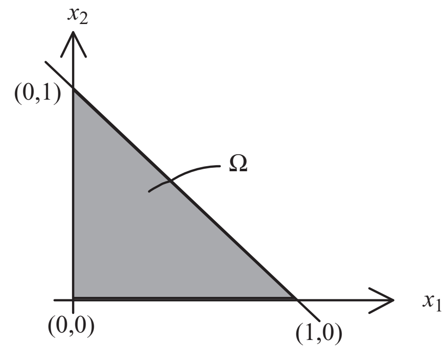
where $\Omega = \{\mathbf{x} : \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{x}^{\mathrm{L}} \leq \mathbf{x} \leq \mathbf{x}^{\mathrm{U}}\}$. Ω, a subset of R^n, is called the *feasible region*.

In unconstrained problems, the constraints are not present – thus, the feasible region is the entire space R^n. Graphical representation in design-space (or **x**-space) for $n = 2$ variables is given in Fig. 1.2. Curves of constant f value or *objective function contours* are drawn, and the optimum is defined by the highest contour curve passing through Ω, which usually, but not always, is a point on the boundary Ω.

Example 1.1

Consider the constraints $\{g_1 \equiv x_1 \geq 0, g_2 \equiv x_2 \geq 0, g_3 \equiv x_1 + x_2 \leq 1\}$. The associated feasible set Ω is shown in Fig. E1.1.

Figure E1.1. Illustration of feasible set Ω.



Upper Bound

It is important to understand the following inequality, which states that any feasible design provides an upper bound to the optimum objective function value:

$$f(\mathbf{x}^*) \leq f(\hat{\mathbf{x}}) \qquad \text{for } any \ \hat{\mathbf{x}} \in \Omega$$

Minimization over a Superset

Given sets (i.e., feasible regions) S_1 and S_2 with $S_1 \subseteq S_2$; that is, S_1 is a subset of S_2 (or contained within S_2). If f_1^*, f_2^* represent the minimum values of a function f over S_1 and S_2, respectively, then

$$f_2^* \leq f_1^*$$

To illustrate this, consider the following example. Let us consider monthly wages earned among a group of 100 workers. Among these workers, assume that Mr. Smith has the minimum earnings of $800. Now, assume a new worker joins the group. Thus, there are now 101 workers. Evidently, the minimum wages among the 101 workers will be less than or equal to $800 depending on the wages of the newcomer.

Types of Variables and Problems

Additions restrictions may be imposed on a variables x_j as follows:

x_j is continuous (default).
x_j is binary (equals 0 or 1).
x_j is integer (equals 1 or 2 or 3, . . . , or N)
x_j is discrete (takes values 10 mm, 20 mm, or 30 mm, etc.)

Specialized names are given to the NLP problem in (1.1) as follows:

Linear Programming (LP): when all functions (objective and constraints) are linear (in $\mathbf{x}$).
Integer Programming (IP): an LP when all variables are required to be integers.
0–1 Programming: special case of an IP where variables are required to be 0 or 1.
Mixed Integer Programming (MIP): an IP where some variables are required to be integers, others are continuous.
MINLP: an MIP with nonlinear functions.
Quadratic Programming (QP): when an objective function is a quadratic function in $\mathbf{x}$ and all constraints are linear.

Convex Programming: when the objective function is convex (for minimization) or concave (for maximization) and the feasible region Ω is a convex set. Here, any local minimum is also a global minimum. Powerful solution techniques that can handle a large number of variables exist for this category. Convexity of Ω is guaranteed when all inequality constraints g_i are convex functions and all equality constraints h_j are linear.

Combinatorial Problems: These generally involve determining an optimum permutation of a set of integers, or equivalently, an optimum choice among a set of discrete choices. Some combinatorial problems can be posed as LP problems (which are much easier to solve). Heuristic algorithms (containing *thumb rules*) play a crucial role in solving large-scale combinatorial problems where the aim is to obtain near-optimal solutions rather than the exact optimum.

1.4 Optimization Problem Modeling

Modeling refers to the translation of a physical problem into mathematical form. While modeling is discussed throughout the text, a few examples are presented below, with the aim of giving an immediate idea to the student as to how variables, objectives, and constraints are defined in different situations. Detailed problem descriptions and exercises and solution techniques are given throughout the text.

Example 1.2 (Shortest Distance from a Point to a Line)

Determine the shortest distance d between a given point $\mathbf{x}^0 = (x_1^0, x_2^0)$ and a given line $a_0 + a_1x_1 + a_2x_2 = 0$ (Fig. E1.2). If $\mathbf{x}$ is a point on the line, we may pose the optimization problem:

$$\begin{aligned} &\text{minimize} \quad f = \left(x_1 - x_1^0\right)^2 + \left(x_2 - x_2^0\right)^2 \\ &\text{subject to} \quad h(\mathbf{x}) \equiv a_0 + a_1x_1 + a_2x_2 = 0 \end{aligned}$$

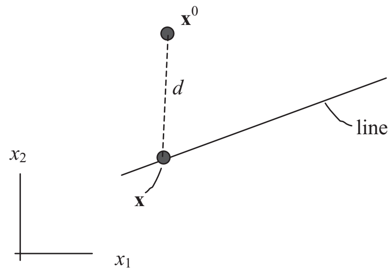


Figure E1.2. Shortest distance problem posed as an optimization problem.

where f is the objective function denoting the square of the distance (d^2), $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$ are two variables in the problem, and h represents a linear *equality* constraint. The reader is encouraged to understand the objective function. At optimum, we will find that the solution $\mathbf{x}^*$ will lie at the foot of the perpendicular drawn from $\mathbf{x}^0$ to the line.

In fact, the preceding problem can be written in matrix form as

$$\begin{aligned} &\text{minimize} \quad f = (\mathbf{x} - \mathbf{x}^0)^{\mathrm{T}}(\mathbf{x} - \mathbf{x}^0) \\ &\text{subject to} \quad h(\mathbf{x}) \equiv \mathbf{a}^{\mathrm{T}}\mathbf{x} - b = 0 \end{aligned}$$

where $\mathbf{a} = [a_1 \;\; a_2]$, $b = -a_0$. Using the method of *Lagrange multipliers* (Chapter 5), we obtain a closed-form solution for the point $\mathbf{x}^*$:

$$\mathbf{x}^* = \mathbf{x}^0 - \frac{(\mathbf{a}^{\mathrm{T}}\mathbf{x}^0 - b)}{(\mathbf{a}^{\mathrm{T}}\mathbf{a})}\mathbf{a}$$

Note that $\mathbf{a}^{\mathrm{T}}\mathbf{a}$ is a scalar. The shortest distance d is

$$d = \frac{\left|\mathbf{a}^{\mathrm{T}}\,\mathbf{x}^0 - b\right|}{\sqrt{\mathbf{a}^{\mathrm{T}}\,\mathbf{a}}}$$

Extensions

The problem can be readily generalized to finding the shortest distance from a point to a plane, in two or three dimensions.

Example 1.3 (Beam on Two Supports)

First, consider a uniformly loaded beam on two supports as shown in Fig. E1.3a. The beam length is $2L$ units, and the spacing between supports is $2a$ units. We wish to determine the half-spacing a/L so as to minimize the maximum deflection that occurs in the beam. One can assume $L = 1$.

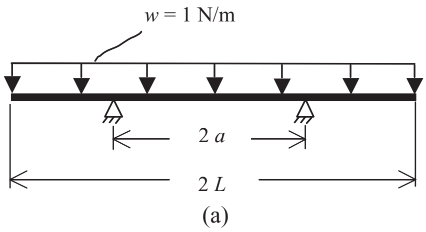


Figure E1.3a. Uniformly loaded beam on two supports.

This simple problem takes a little thought to formulate and solve using an available optimization routine. To provide insight, consider the deflected shapes when the support spacing is too large (Fig. E1.3a), wherein the maximum deflection occurs at the center, and when the spacing is too small (Fig. E1.3b), wherein the maximum deflection occurs at the ends. Thus, the graph of maximum deflection versus spacing is convex or cup-shaped with a well-defined minimum.

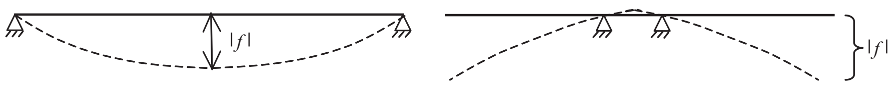


Figure E1.3b–c. Effect of support spacing on maximum deflection in beam.

Thus, we may state that the maximum deflection δ at any location in the beam, can be reduced to checking the maximum at just two locations. With this insight, the objective function f, which is to be minimized, is given by

$$f(a) \equiv \max_{0 \le x \le 1} \delta(x, a) = \max\{\delta(0, a), \delta(1, a)\} = \max(\delta_{\text{center}}, \delta_{\text{end}})$$

Absolute values for δ are to be used. Beam theory provides the relationship between a and $\delta\,(x, a)$. We now have to solve the optimization problem

$$\begin{aligned} &\text{minimize} \quad f(a) \\ &\text{subject to} \quad 0 \le a \le 1 \end{aligned}$$

We may use the Golden Section Search or other techniques discussed in Chapter 2 to solve this unconstrained one-dimensional problem.

Extension – I (Minimize Peak Stress in Beam)

The objective function in the beam support problem may be changed as follows: determine a to minimize the maximum bending stress. Also, the problem may be modified by considering multiple equally spaced supports.

Further, the problem of supporting above-ground, long, and continuous pipelines such as portions of the famous Alaskan oil pipeline is considerably more complicated owing to supports with foundations, code specifications, wind and seismic loads, etc.

Extension – II (Plate on Supports)

The following (more difficult) problem involves supporting a plate rather than a beam as in the preceding example. Given a fixed number of supports N_s, where $N_s \ge 3$ and an integer, determine the optimum support locations to minimize the maximum displacement due to the plate's self-weight. This problem occurs, for example, when a roof panel or tile has to be attached to a ceiling with a discrete number of pins. Care must be taken that all supports do not line up in a straight line to avoid instability. Generalizing this problem still further will lead to optimum *fixturing* of three-dimensional objects to withstand loads in service, handling, or during manufacturing.

Example 1.4 (Designing with Customer Feedback)

In this example, we show one way in which customer feedback is used to develop an objective function f for subsequent optimization. We present a rather simple example to focus on concepts. A fancy outdoor cafe is interested in designing a unique beer mug. Two criteria or *attributes* are to be chosen. The first attribute is the volume, V in oz, and the second attribute is the aspect ratio, H/D, where $H =$ height and $D =$ diameter. To manufacture a mug, we need to know the *design variables* H and D. Lower and upper limits have been identified on each of the attributes, and within these limits, the attributes can take on continuous values. Let us choose

$$8 \le V \le 16, \qquad 0.6 \le H/D \le 3.0$$

To obtain customer feedback in an economical fashion, three discrete levels, LMH or low/medium/high, are set for each attribute, leading to only 9 different types of mugs. For further economy, prototypes of only a subset of these 9 mugs may be made for customer feedback. However, in this example, all 9 mugs are made, and ratings of these from a customer (or a group) is obtained as shown in Table E1.4. For example, an ML mug corresponds to a mug of volume 12 oz and aspect ratio 0.6, and is rated at 35 units. The linearly scaled ratings in the last column correspond to a range of 0–100.

Table E1.4. *Sample customer ratings of a set of mugs.*

Sample mugs (V, H/D)	Customer rating	Scaled rating
L L	50.00	27.27
L M	60.00	45.45
L H	75.00	72.73
M L	35.00	0.00
M M	90.00	100.00
M H	70.00	63.64
H L	35.00	0.00
H M	85.00	90.91
H H	50.00	27.27