SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO & CHEMICAL ENGINEERING

DEPARTMENT OF CHEMICAL ENGINEERING

UNIT – I – Optimization of Chemical Processes – SCH1402

UNIT 1 OBJECTIVE AND FORMULATION OF OPTIMIZATION

1.0 Introduction to Optimization

Optimization is the mathematical discipline which is concerned with finding the maxima and minima of functions, possibly subject to constraints.

Optimization is an important tool in making decisions and in analyzing physical systems. In mathematical terms, an **optimization problem** is the problem of finding the best solution from among the set of all feasible solutions.

Constructing a Model

The first step in the optimization process is constructing an appropriate model; ***modeling*** is the process of identifying and expressing in mathematical terms the **objective**, the **variables**, and the **constraints** of the problem.

An ***objective*** is a quantitative measure of the performance of the system that we want to minimize or maximize. In manufacturing, we may want to maximize the profits or minimize the cost of production, whereas in fitting experimental data to a model, we may want to minimize the total deviation of the observed data from the predicted data.

The ***variables*** or the *unknowns* are the components of the system for which we want to find values. In manufacturing, the variables may be the amount of each resource consumed or the time spent on each activity, whereas in data fitting, the variables would be the parameters of the model.

The ***constraints*** are the functions that describe the relationships among the variables and that define the allowable values for the variables. In manufacturing, the amount of a resource consumed cannot exceed the available amount.

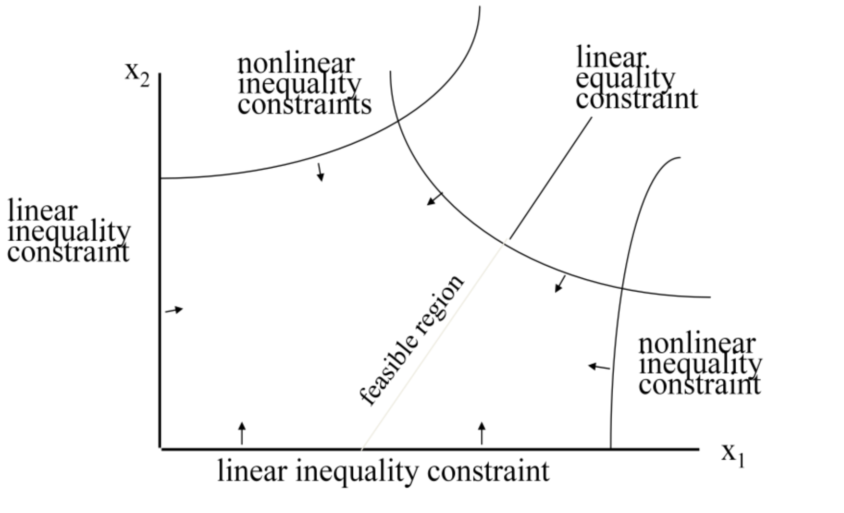
Essential Features

Every optimisation problem contains three essential categories:

At least one objective function to be

optimised Equality constraints

Inequality constraints. By a feasible solution we mean a set of variables which satisfy categories 2 and 3. The region of feasible solutions is called the *feasible region*.



An optimal solution is a set of values of the variables that are contained in the feasible region and also provide the best value of the objective function in category 1.

For a meaningful optimisation problem the model needs to be *underdetermined*.

1.1 Mathematical Description

Steps Used To Solve Optimisation Problems

Analyse the process in order to make a list of all the variables.

Determine the optimisation criterion and specify the objective

function.

Develop the mathematical model of the process to define the equality and inequality constraints.

Identify the independent and dependent variables to obtain the number of degrees of freedom.

If the problem formulation is too large or complex simplify it if

possible. Apply a suitable optimisation technique.

Check the result and examine it’s sensitivity to changes in model parameters and assumptions.

Classification of Optimisation Problems

Properties of *f*(x)

- single variable or multivariable
- linear or nonlinear
- sum of squares
- quadratic
- smooth or non-smooth
- sparsity

Properties of h(x) and g(x)

- simple bounds
- smooth or non-smooth
- sparsity
- linear or nonlinear

no constraints Properties of optimization variables x time variant or invariant continuous or discrete take only integer values mixed

Obstacles and Difficulties

Objective function and/or the constraint functions may have finite *discontinuities* in the continuous parameter values.

Objective function and/or the constraint functions may be *non-linear functions* of the variables.

Objective function and/or the constraint functions may be defined in terms of *complicated interactions* of the variables. This may prevent calculation of unique values of the variables at the optimum.

Objective function and/or the constraint functions may exhibit nearly "flat" behaviour for some ranges of variables or exponential behaviour for other ranges. This causes the problem to be insensitive, or too sensitive.

The problem may exhibit many local optima whereas the global optimum is sought. A solution may be obtained that is less satisfactory than another solution elsewhere.

Absence of a feasible region. Model-reality differences.

Typical Examples of Application static optimisation

Plant design (sizing and layout).

Operation (best steady-state operating condition). Parameter estimation (model fitting).

Allocation of resources.

Choice of controller parameters (e.g. gains, time constants) to minimise a given performance index (e.g. overshoot, settling time, integral of error squared).

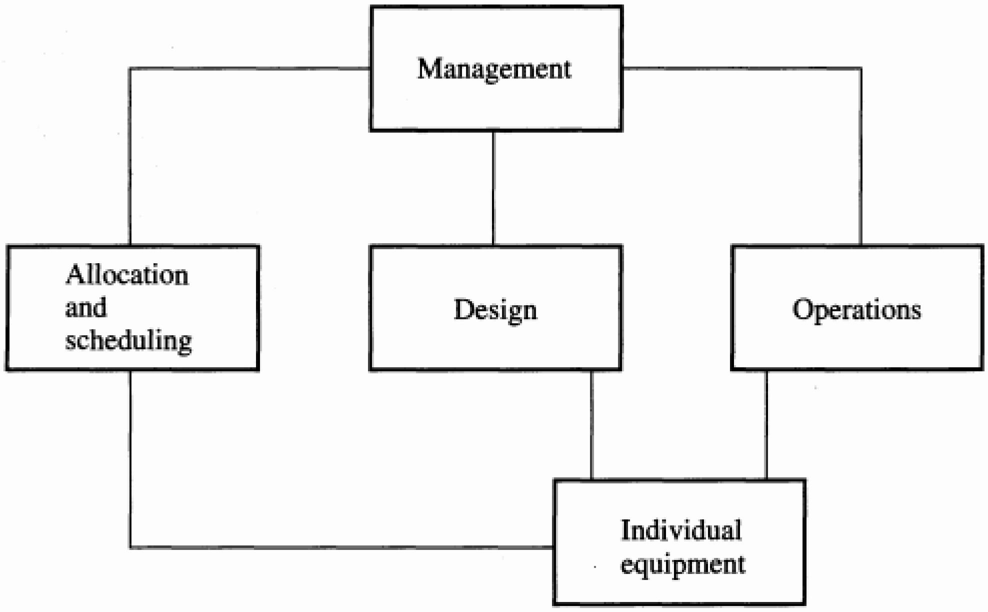


FIGURE 1.1
Hierarchy of levels of optimization.

1.2 Six steps to do optimization problems

1. Analyze the process itself so that the process variables and specific characteristics of interest are defined; that is, make a list of all of the variables. 2. Determine the criterion for optimization, and specify the objective function in terms of the variables defined in step 1 together with coefficients. This step provides the performance model (sometimes called the economic model when appropriate). 3. Using mathematical expressions, develop a valid process or equipment model that relates the input-output variables of the process and associated coefficients. Include both equality and inequality constraints. Use well-known physical principles (mass balances, energy balances), empirical relations, implicit concepts, and external restrictions. Identify the independent and dependent variables to get the number of degrees of freedom. 4. If the problem formulation is too large in scope: (a) break it up into manageable parts or (b) simplify the objective function and model 5. Apply a suitable optimization technique to the mathematical statement of the problem. 6. Check the answers, and examine the sensitivity of the result to changes in the coefficients in the problem and the assumptions.

Dynamic optimisation

Determination of a control signal $u(t)$ to transfer a dynamic system from an initial state to a desired final state to satisfy a given performance index.

Optimal plant start-up and/or shut down. Minimum time problems

1.3 BASIC PRINCIPLES OF STATIC OPTIMISATION THEORY

Continuity of Functions

Functions containing discontinuities can cause difficulty in solving optimisation problems.

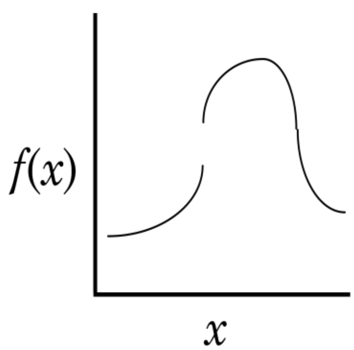
Definition: A function of a single variable x is continuous at a point x_o if:

(a) $f(x_o)$ exists

(b) $\lim_{x \to x_o} f(x)$ exists

(c) $\lim_{x \to x_o} f(x) = f(x_o)$

If $f(x)$ is continuous at every point in a region R, then $f(x)$ is said to be continuous throughout R. $f(x)$ is discontinuous



$f(x)$ is continuous, but $f'(x) = \frac{df}{dx}(x)$ is not

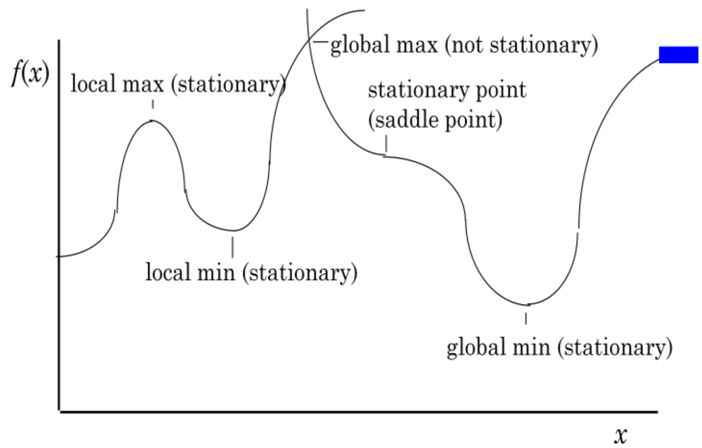
Unimodal and Multimodal Functions

A *unimodal function* $f(x)$ (in the range specified for x) has a single extremum (minimum or maximum). A *multimodal function* $f(x)$ has two or more extrema.

If $f'(x) = 0$ at the extremum, the point is called a *stationary point*.

There is a distinction between the *global extremum* (the biggest or smallest between a set of extrema) and *local extrema* (any extremum). *Note*: many numerical procedures terminate at a local extremum.

A multimodal function



Multivariate Functions - Surface and Contour Plots

We shall be concerned with basic properties of a scalar function $f(\mathbf{x})$ of n variables

$(x_1,...,x_n)$. If n = 1, f(x) is a univariate function

If n > 1, f($\mathbf{x}$) is a multivariate function.

For any multivariate function, the equation

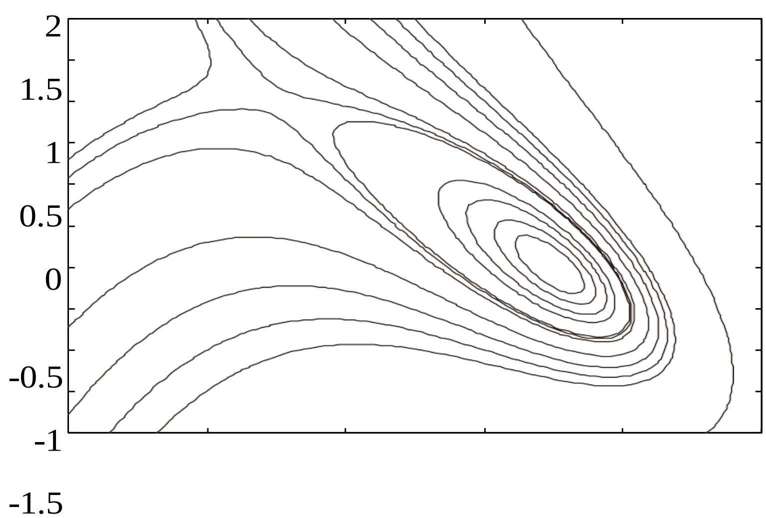
$z = f(\mathbf{x})$ defines a *surface* in n+1 dimensional space.

In the case $n = 2$, the points $z = f(x_1,x_2)$ represent a three dimensional surface.

Let c be a particular value of $f(x_1,x_2)$. Then $f(x_1,x_2) = c$ defines a curve in x_1 and x_2 on the plane $z = c$. If we consider a selection of different values of c, we obtain a family of curves which provide a *contour map*

of the function $z = f(x_1,x_2)$.

Contourmap



Optimization Problem and Model Formulation

Introduction In the previous lecture we studied the history of evolution of optimization methods and their engineering applications. A brief introduction was also given to the art of modeling. In this lecture we will study the Optimization problem, its various components and its formulation as a mathematical programming problem.

1.4 Basic components of an optimization problem:

An objective function expresses the main aim of the model which is either to be minimized or maximized. For example, in a manufacturing process, the aim may be to maximize the profit or minimize the cost. In comparing the data prescribed by a user-defined model with the observed data, the aim is minimizing the total deviation of the predictions based on the model from the observed data. In designing a bridge pier, the goal is to maximize the strength and minimize size.

A set of unknowns or variables control the value of the objective function. In the manufacturing problem, the variables may include the amounts of different resources used or the time spent on each activity. In fitting- the-data problem, the unknowns are the parameters of the model. In the pier design problem, the variables are the shape and dimensions of the pier.

A set of constraints are those which allow the unknowns to take on certain values but exclude others. In the manufacturing problem, one cannot spend negative amount of time on any activity, so one constraint is that the "time" variables are to be non-negative. In the pier design problem, one would probably want to limit the breadth of the base and to constrain its size.

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 stated, the objective function is the mathematical function one wants to maximize or minimize, subject to certain constraints. Many all optimization problems have a single objective function. (When they don't they can often be reformulated so that they do) The two exceptions are:

• **No objective function.** In some cases (for example, design of integrated circuit layouts), the goal is to find a set of variables that satisfies the constraints of the model. The user does not particularly want to optimize anything and so there is no reason to define an objective function. This type of problems is usually called a feasibility problem.

• **Multiple objective functions.** In some cases, the user may like to optimize a number of different objectives concurrently. For instance, in the panel design problem, it would be nice to minimize weight and maximize strength simultaneously. Usually, the different objectives are not compatible; the variables that optimize one objective may be far from optimal for the others. 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 as constraints.

Statement of an optimization problem An optimization or a mathematical programming problem can be stated as follows:

To find X

f(X) (1.1) Subject to the

constraints gi(X) ≤ 0 , i = 1,

2,….,m

lj(X) = 0 , j = 1, 2,….,p

where X is an n-dimensional vector called the design vector, f(X) is called the objective function, and gi(X) and lj(X) are known as inequality and equality constraints, respectively. The number of variables n and the number of constraints m and/or p need not be related in any way. This type problem is called a constrained optimization problem.

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 Fig 1 we can identify the optimum point (maxima). This is possible graphically only when the number of design variables 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.

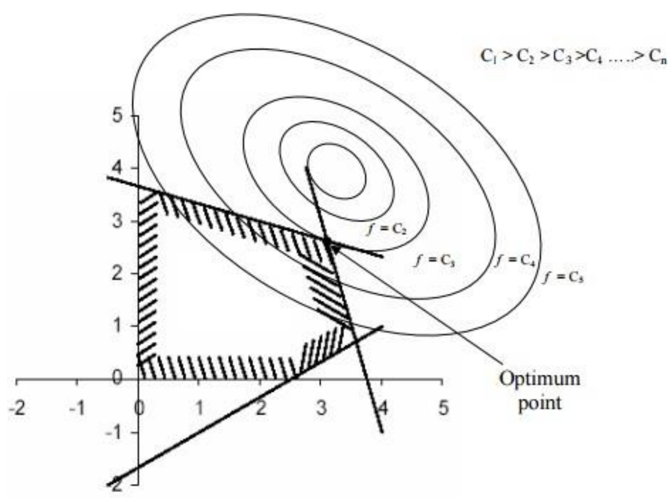
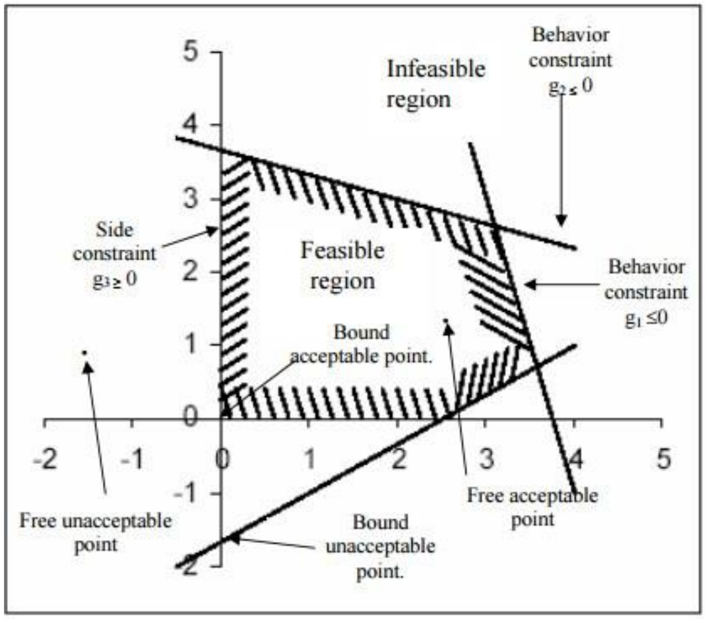


Fig 1



1.5 Linear Programming

Optimization is an important and fascinating area of management science and operations research. It helps to do less work, but gain more.

Linear programming (LP) is a central topic in optimization. It provides a powerful tool in modeling many applications. LP has attracted most of its attention in optimization during the last six decades for two main reasons:

Applicability: There are many real world applications that can be modeled as linear programming; Solvability: There are theoretically and practically efficient techniques for solving large-scale problems.

Basic Components of an LP:

Each optimization problem consists of three elements: decision variables: describe our choices that are under our control; objective function: describes a criterion that we wish to minimize (e.g., cost) or maximize (e.g., profit); constraints: describe the limitations that restrict our choices for decision variables. **Problem Statement:** A company makes two products (say, P and Q) using two machines (say, A and B). Each unit of P that is produced requires 50 minutes processing time on machine A and 30 minutes processing time on machine B. Each unit of Q that is produced requires 24 minutes processing time on machine A and 33 minutes processing time on machine B.Machine A is going to be available for 40 hours and machine B is available for 35 hours. The profit per unit of P is $25 and the profit per unit of Q is $30. Company policy is to determine the production quantity of each product in such a way as to maximize the total profit given that the available resources should not be exceeded

Task: The aim is to formulate the problem of deciding how much of each product to make in the current week as an LP.

Step 1: Defining the Decision Variables

We often start with identifying decision variables (i.e., what we want to determine among those things which are under our control). Tom! Can you identify the decision variables for our example?

The company wants to determine the optimal product to make in the current week. So there are two decision variables:

x: the number of units

of P y: the number of

units of Q

Step 2: Choosing an Objective Function

We usually seek a criterion (or a measure) to compare alternative solutions. This yields the objective function. Tom! It is now your turn to identify the objective function.

We want to maximize the total profit. The profit per each unit of product P is $25 and profit per each unit of product Q is $30. Therefore, the total profit is 25x+30y if we produce x units of P and y units of Q. This leads to the following objective function:

max 40x+35y

Note that: 1: The objective function is linear in terms of decision variables x and y (i.e., it is of the form ax + by, where a and b are constant). 2: We typically use the variable z to denote the value of the objective. So the objective function can be stated as:

max z=25x+30y

Step 3: Identifying the Constraints

In many practical problems, there are limitations (such as resource / physical / strategic / economical) that restrict our decisions. We describe these limitations using mathematical constraints. Tom!What are the constraints in our example?

The amount of time that machine A is available restricts the quantities to be manufactured. If we produce x units of P and y units of Q, machine A should be used for 50x+24y minutes since each unit of P requires 50 minutes processing time on machine A and each unit of Q requires 24 minutes processing time on machine

A. On the other hand, machine A is available for 40 hours or equivalently for 2400 minutes. This imposes the following constraint:

$50x + 24y \leq 2400$.

Similarly, the amount of time that machine B is available imposes the following constraint: $30x + 33y \leq 2100$.

These constrains are linear inequalities since in each constraint the left-hand side of the inequality sign is a linear function in terms of the decision variables x and y and the right hand side is constant.

Step 3: Identifying the Constraints

Note: In most problems, the decision variables are required to be nonnegative, and this should be explicitly included in the formulation. This is the case here. So you need to include the following two no negativity constraints as well:

$x \geq 0$ and $y \geq 0$

I see your point. So the constraints we are subject to (s.t.) are :

$50x + 24y \leq 2400$, (machine A time) $30x + 33y \leq 2100$, (machine B time) $x \geq 0$, $y \geq 0$.

Here is the LP:

max $z= 25x + 30y$

s.t. $50x + 24y \leq 2400$, $30x + 33y \leq 2100$,

$x \geq 0$, $y \geq 0$.

A Manufacturing Example

Problem Statement: An operations manager is trying to determine a production plan for the next week. There are three products (say, P, Q, and Q) to produce using four machines (say, A and B, C, and D). Each of the four machines performs a unique process. There is one machine of each type, and each machine is available for 2400 minutes per week. The unit processing times for each machine is given in Table 1.

Table 1: Machine Data

Unit Processing Time (min)				
Machine	Product P	Product Q	Product R	Availability (min)
A	20	10	10	2400
B	12	28	16	2400
C	15	6	16	2400
D	10	15	0	2400
Total processing time	57	59	42	9600

The unit revenues and maximum sales for the week are indicated in Table 2. Storage from one week to the next is not permitted. The operating expenses associated with the plant are $6000 per week, regardless of how many components and products are made. The $6000 includes all expenses except for material costs.

Table 2: Product Data

Item	Product P	Product Q	Product R
Revenue per unit	$90	$100	$70
Material cost per unit	$45	$40	$20
Profit per unit	$45	$60	$50
Maximum sales	100	40	60

Task: Here we seek the "optimal" product mix-- that is, the amount of each product that should be manufactured during the present week in order to maximize profits. Formulate this as an LP.

Step 1: Defining the Decision Variables

We are trying to select the optimal product mix, so we define three decision variables as follows: p: number of units of product P to produce,

q: number of units of product Q to

produce, r: number of units of product

R to produce. Step 2: Choosing an

Objective Function Our objective is to

maximize profit:

Profit = (90-45)p + (100-40)q + (70-20r) – 6000

= 45p + 60q + 50r – 6000

Note: The operating costs are not a function of the variables in the problem. If we were to drop the \$6000 term from the profit function, we would still obtain the same optimal mix of products. Thus, the objective function is

$z = 45p + 60q + 50r$

Step 3: Identifying the Constraints

The amount of time a machine is available and the maximum sales potential for each product restrict the quantities to be manufactured. Since we know the unit processing times for each machine, the constraints can be written as linear inequalities as follows:

$20p+10q+10r \leq 2400$ (Machine

A) 1 $2p+28q+16r \leq 2400$

(Machine B) $15p+6q+16r \leq 2400$

(Machine C) $10p+15q+0r \leq 2400$

(Machine D)

Observe that the unit for these constraints is minutes per week. Both sides of an inequality must be in the same unit. The market limitations are written as simple upper bounds. Market Constraints: $P \leq 100$, $Q \leq 40$, $R \leq 60$.

Logic indicates that we should also include no negativity restrictions on the variables. No negativity constraints: $P \geq 0$, $Q \geq 0$, $R \geq 0$.

By combining the objective function and the constraints, we obtain the LP model as follows: max $z=45p+60q+50r$

s.t. $20p+10q+10r \leq$

2400 $12p+28q+16r \leq$

2400

$15p+6q+16r \leq 2400$

$10p+15q+0r \leq 2400$

$0 \leq p \leq 100$

$0 \leq q \leq 40$

$0 \leq r \leq 60$

Introduction to Factorial

Designs Basic Definitions

and Principles

Study the effects of two or more

factors. Factorial designs

Crossed: factors are arranged in a factorial design

Main effect: the change in response produced by a change in the level of the factor

1.6 Factorial experiments

Response variable(s) in any experiment can be found to be affected by a number of factors in the overall system some of which are controlled or maintained at desired levels in the experiment. An experiment in which the treatments consist of all possible combinations of the selected levels in two or more factors is referred as a factorial experiment. For example, an experiment on rooting of cuttings involving two factors, each at two levels, such as two hormones at two doses, is referred to as a 2 x 2 or a 2^2 factorial experiment. Its treatments consist of the following four possible combinations of the two levels in each of the two factors.

	Treatment combination	
Treatment number	Hormone	Dose (ppm)
1	NAA	10
2	NAA	20
3	IBA	10
4	IBA	20

The term *complete factorial experiment* is sometimes used when the treatments include all combinations of the selected levels of the factors. In contrast, the term fractional factorial experiment is used when only a fraction of all the combinations is tested. Throughout this manual, however, complete factorial experiments are referred simply as factorial experiments. Note that the term *factorial* describes a specific way in which the treatments are formed and does not, in any way, refer to the design used for laying out the experiment. For example, if the foregoing 2^2 factorial experiment is in a randomized complete block design, then the correct description of the experiment would be 2^2 *factorial experiment in randomized complete block design.*

The total number of treatments in a factorial experiment is the product of the number of levels of each factor; in the 2^2 factorial example, the number of treatments is 2 x 2 = 4, in the 2^3 factorial, the number of treatments is 2 x 2 x 2 = 8.

The number of treatments increases rapidly with an increase in the number of factors or an increase in the levels in each factor. For a factorial experiment involving 5 clones, 4 espacements, and 3 weed-control methods, the total number of treatments would be 5 x 4 x 3 = 60. Thus, indiscriminate use of factorial experiments has to be avoided because of their large size, complexity, and cost.

Furthermore, it is not wise to commit oneself to a large experiment at the beginning of the investigation when several small preliminary experiments may offer promising results. For example, a tree breeder has collected 30 new clones from a neighbouring country and wants to assess their reaction to the local environment. Because the environment is expected to vary in terms of soil fertility, moisture levels, and so on, the ideal experiment would be one that tests the 30 clones in a factorial experiment involving such other variable factors as fertilizer, moisture level, and population density. Such an experiment, however, becomes extremely large as factors other than clones are added.

Even if only one factor, say nitrogen or fertilizer with three levels were included, the number of treatments would increase from 30 to 90. Such a large experiment would mean difficulties in financing, in obtaining an adequate experimental area, in controlling soil heterogeneity, and so on. Thus, the more practical approach would be to test the 30 clones first in a single-factor experiment, and then use the results to select a few clones for further studies in more detail.

For example, the initial single-factor experiment may show that only five clones are outstanding enough to warrant further testing. These five clones could then be put into a factorial experiment with three levels of nitrogen, resulting in an experiment with 15 treatments rather than the 90 treatments needed with a factorial experiment with 30 clones.

The effect of a factor is defined to be the average change in response produced by a change in the level of that factor. This is frequently called the main effect. For example, consider the data in Table 4.12.

Table 4.12. Data from a 2x2 factorial experiment

		Factor B	
	Level	b1	b2
	a1	20	30
Factor A			
	a2	40	52

The main effect of factor A could be thought of as the difference between the average response at the first level of A and the average response at the second level of A. Numerically, this is

$$A = \frac{40+52}{2} - \frac{20+30}{2} = 21$$

That is, increasing factor A from level 1 to level to 2 causes an average increase in the response by 21 units. Similarly, the main effect of B is

$$B = \frac{30+52}{2} - \frac{20+40}{2} = 11$$

If the factors appear at more than two levels, the above procedure must be modified since there are many ways to express the differences between the average responses.

The major advantage of conducting a factorial experiment is the gain in information on interaction between factors. In some experiments, we may find that the difference in response between the levels of one factor is not the same at all levels of the other factors. When this occurs, there is an interaction between the factors. For example, consider the data in Table 4.13.

Table 4.13. Data from a 2x2 factorial experiment

		Factor B

	Levels	b1	b2
	a1	20	40
Factor A			
	a2	50	12

At the first level of factor B, the factor A effect is

A = 50-20 = 30

and at the second level of factor B, the factor A effect is

A = 12-40 = -28

Since the effect of A depends on the level chosen for factor B, we see that there is interaction between A and B.

These ideas may be illustrated graphically. Figure 4.5 plots the response data in Table 4.12. against factor A for both levels of factor B.

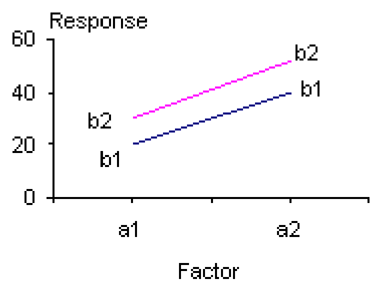


Figure 4.5. Graphical representation of lack of interaction between factors.

Note that the b1 and b2 lines are approximately parallel, indicating a lack of interaction between factors A and B.

Similarly, Figure 4.6 plots the response data in Table 4.13. Here we see that the b1 and b2 lines are not parallel. This indicates an interaction between factors A and B. Graphs such as these are frequently very useful in interpreting significant interactions and in reporting the results to nonstatistically trained management. However, they should not be utilized as the sole technique of data analysis because their interpretation is subjective and their appearance is often misleading.

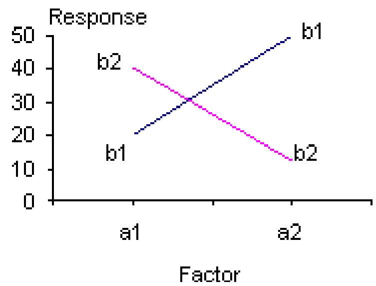


Figure 4.6. Graphical representation of interaction between factors.

Note that when an interaction is large, the corresponding main effects have little practical meaning. For the data of Table 4.13, we would estimate the main effect of A to be

$$A = \frac{50+12}{2} - \frac{20+40}{2} = 1$$

which is very small, and we are tempted to conclude that there is no effect due to A. However, when we examine the effects of A at different levels of factor B, we see that this is not the case. Factor A has an effect, but it depends on the level of factor B *i.e.*, a significant interaction will often mask the significance of main effects. In the presence of significant interaction, the experimenter must usually examine the levels of one factor, say A, with level of the other factors fixed to draw conclusions about the main effect of A.

For most factorial experiments, the number of treatments is usually too large for an efficient use of a complete block design. There are, however, special types of designs developed specifically for large factorial experiments such as confounded designs. Descriptions on the use of such designs can be found in Das and Giri (1980).

1.7 Analysis of variance

Any of the complete block designs discussed in sections 4.2 and 4.3 for single-factor experiments is applicable to a factorial experiment. The procedures for randomization and layout of the individual designs are directly applicable by simply ignoring the factor composition of the factorial treatments and considering all the treatments as if they were unrelated. For the analysis of variance, the computations discussed for individual designs are also directly applicable. However, additional computational steps are required to partition the treatment sum of squares into factorial components corresponding to the main effects of individual factors and to their interactions. The procedure for such partitioning is the same for all complete block designs and is, therefore, illustrated for only one case, namely, that of RCBD.

The step-by-step procedure for the analysis of variance of a two-factor experiment on bamboo involving two levels of spacing (Factor A) and three levels of age at planting (Factor A) laid out in RCBD with three replications is illustrated here. The list of the six factorial treatment combinations is shown in Table 4.14, the experimental layout in Figure 4.7, and the data in Table 4.15.

Table 4.14. The 2 x 3 factorial treatment combinations of two levels of spacing and three levels of age.

Age at planting	Spacing (m)	
(month)	10 m x 10 m	12 m x 12m
	(a1)	(a2)
6 (b1)	a1b1	a2b1
12 (b2)	a1b2	a2b2
24 (b3)	a1b3	a2b3

Replication I Replication II Replication III

a2b3	a2b3	a1b2
a1b3	a1b2	a1b1
a1b2	a1b3	a2b2
a2b1	a2b1	a1b3
a1b1	a2b2	a2b1
a2b2	a1b1	a2b3

Figure 4.7. A sample layout of 2 x 3 factorial experiment involving two levels of spacing and three levels of age in a RCBD with 3 replications.

Table 4.15. Mean maximum culm height of *Bambusa arundinacea* tested with three age levels and two levels of spacing in a RCBD.

Treatment	Maximum culm height of a clump (cm)			Treatment
combination	Rep. I	Rep. II	Rep. III	total (*Tij*)
a1b1	46.50	55.90	78.70	181.10
a1b2	49.50	59.50	78.70	187.70
a1b3	127.70	134.10	137.10	398.90
a2b1	49.30	53.20	65.30	167.80
a2b2	65.50	65.00	74.00	204.50
a2b3	67.90	112.70	129.00	309.60
Replication total (*Rk*)	406.40	480.40	562.80	*G*=1449.60

Step 1. Denote the number of replication by *r*, the number of levels of factor A (*i.e*., spacing) by *a*, and that of factor B (*i.e*., age) by *b*. Construct the outline of the analysis of variance as follows:

Table 4.16. Schematic representation of ANOVA of a factorial experiment with two levels of factor A, three levels of factor B and with three replications in RCBD.

Source of variation	Degrees of freedom (*df*)	Sum of squares (*SS*)	Mean square $\left(MS = \frac{SS}{df}\right)$	Computed *f*
Replication	*r*-1	SSR	MSR	
Treatment	*ab*- 1	SST	MST	$\frac{MST}{MSE}$

A	a- 1	SSA	MSA	$\frac{MSA}{MSE}$
B	b- 1	SSB	MSB	$\frac{MSB}{MSE}$
AB	(a-1)(b-1)	SSAB	MSAB	$\frac{MSAB}{MSE}$
Error	(r-1)(ab-1)	SSE	MSE	
Total	rab -1	SSTO		

Step 2.Compute treatment totals (T_{ij}), replication totals (R_k), and the grand total (G), as shown in Table 4.15 and compute the *SSTO*, *SSR*, *SST* and *SSE* following the procedure described in Section 4.3.3. Let y_{ijk} refer to the observation corresponding to the ith level of factor A and jth level factor B in the kth replication.

$$C.F. = \frac{G^2}{rab} \quad (4.22)$$

$$= \frac{(1449.60)^2}{(3)(2)(3)} = 116741.12$$

$$SSTO = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{r} y_{ijk}^2 - C.F. \quad (4.23)$$

$$= \left[(46.50)^2 + (55.90)^2 + \ . \ . \ . \ + (129.00)^2\right] - 116741.12$$

= 17479.10

$$SSR = \frac{\sum_{k=1}^{r} R_k^2}{ab} - C.F. \quad (4.24)$$

$$= \frac{(406.40)^2 + \ . \ . \ . \ + (562.80)^2}{(2)(3)} - 116741.12$$

= 2040.37

$$SST = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b} T_{ij}^2}{r} - C.F. \quad (4.25)$$

$$= \frac{(181.10)^2 + \ . \ . \ . \ + (309.60)^2}{3} - 116741.12$$

= 14251.87

SSE = SSTO - SSR - SST (4.26)

= 17479.10 - 2040.37 - 14251.87

= 1186.86

The preliminary analysis of variance is shown in Table 4.17.

Table 4.17. Preliminary analysis of variance for data in Table 4.15.

Source of variation	Degree of freedom	Sum of squares	Mean square	Computed *F*	Tabular *F*5%
Replication	2	2040.37	1020.187	8.59567*	4.10
Treatment	5	14251.87	2850.373	24.01609*	3.33
Error	10	1186.86	118.686		
Total	17	17479.10			

*Significant at 5% level.

1.8 Degrees of Freedom

To determine the degrees of freedom (the number of variables whose values may be independently specified) in our model we could simply count the number of independent variables (the number of variables which remain on the right- hand side) in our modified equations. This suggests a possible definition:

degrees of freedom = # variables - # equations

Definition:

The degrees of freedom for a given problem are the number of independent problem variables which must be specified to uniquely determine a solution. In our distillation example, there are: 16 equations 16 variables (recall that F and XF are fixed by upstream processes). This seems to indicate that there are no degrees of freedom.

Consider the three equations relating QC, QR, and qvapour:

QR - QC = 0

QR - DHvap qvapour

= 0 QC - DHvap

qvapour = 0

Notice that if we subtract the last from the second equation:

QR - DHvap qvapour = 0

- QC - DHvap qvapour = 0

QR - QC = 0 the result is the first equation.

It seems that we have three different equations, which contain no more information than two of the equations. In fact any of the equations is a linear combination of the other two equations. We require a clearer, more precise definition for degrees of freedom.

Measures that ignore time value of money

net profit

payout

time

return on investment, ROI

Measures that recognize the time value of money

Internal rate of

return, IRR Net

present value,

NPV Discounted return on investment,

DROI A measure of the total

profitability

• Alternative, Benefits-to-cost ratio

• Strengths

– Recognizes a profit in relation to investment size and simple design.

– Can also use discounted cashflows, DROI

• Weakness

– Continuing investments

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO & CHEMICAL ENGINEERING

DEPARTMENT OF CHEMICAL ENGINEERING

UNIT – II – Optimization of Chemical Processes – SCH1402

Unit II

2.0 CONTINUITY OF FUNCTIONS

Definition

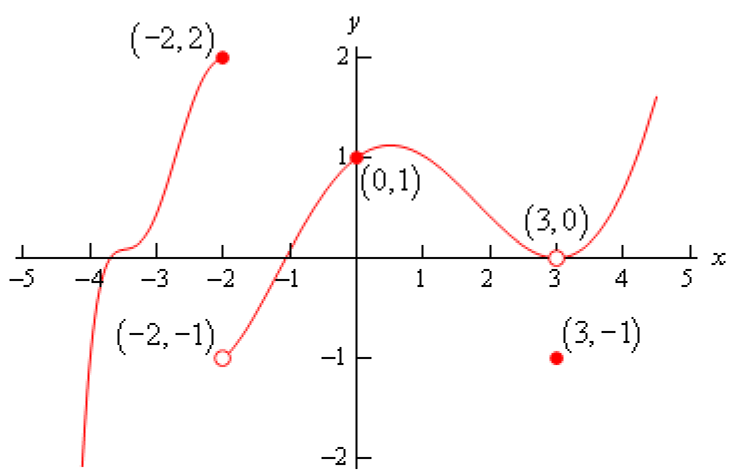
A $f(x)$ $f(x)$ is said to be **continuous** $x = a$ $x = a$ i

$$\lim_{x \to a} f(x) = f(a)$$

$\lim_{x \to a} f(x) = f(a)$

A function is said to be continuous on the interval [*a, b*] if it is continuous at each point in

Example 1 Given the graph of *f(x)*, shown below, determine if *f(x)* is $x = -2$ $x = -2$, $x = 0$ $x = 0$, $x = 3$ $x = 3$.



Solution

To answer the question for each point we'll need to get both the limit at that point and the function value at

that point. If they are equal the function is continuous at that point and if they aren't equal the

Firs $x = -2$ $x = -2$.

$$f(-2) = 2 \qquad \lim_{x \to -2} f(x) \text{ doesn't exist}$$

$f(-2) = 2 \qquad \lim_{x \to -2} f(x) \text{ doesn't exist}$

The function value and the limit aren't the same and so the function is not continuous at this kind of discontinuity in a graph is called a **jump discontinuity**. Jump discontinuities occur where the graph

No $x = 0$ $x = 0$.

$$f(0) = 1 \qquad \lim_{x \to 0} f(x) = 1$$

$$f(0) = 1 \qquad \lim_{x \to 0} f(x) = 1$$

The function is continuous at this point since the function and limit have the

Finall $x = 3$ $x = 3$.

$$f(3) = -1 \qquad \lim_{x \to 3} f(x) = 0$$

$$f(3) = -1 \qquad \lim_{x \to 3} f(x) = 0$$

The function is not continuous at this point. This kind of discontinuity is called a **removable discontinuity**.

From this example we can get a quick -working‖ definition of continuity. A function is continuous on **a**interval if we can draw the graph from start to finish without ever once picking up our pencil. The graph in the last example has only two discontinuities since there are only two places where we would have to pick up our pencil in sketching it.
In other words, a function is continuous if its graph has no holes or breaks in it.

For many functions it‘s easy to determine where it won‘t be continuous. Functions won‘t be continuous where we have things like division by zero or logarithms of zero. Let‘s take a quick look at an example of determining where a function is not continuous.

Example 2 Determine where the function below is not continuous.

$$h(t) = \frac{4t + 10}{t^2 - 2t - 15}$$

$$h(t) = \frac{4t + 10}{t^2 - 2t - 15}$$

Rational functions are continuous everywhere except where we have division by zero. So all that we need determining where the denominator is zero. That's easy enough to determine by setting the denominator equal to zero and solving.

$$t^2 - 2t - 15 = (t-5)(t+3) = 0$$

$$t^2 - 2t - 15 = (t-5)(t+3) = 0$$

A nice consequence of continuity is the following fact.

Fact 2

I $f(x)$ $f(x)$ is continuo a $x = b$ $x = b$ an $\lim_{x \to a} g(x) = b$ $\lim_{x \to a} g(x) = b$ then

$$\lim_{x \to a} f\left(g(x)\right) = f\left(\lim_{x \to a} g(x)\right)$$

$$\lim_{x \to a} f\left(g(x)\right) = f\left(\lim_{x \to a} g(x)\right)$$

To see a proof of this fact see the **Proof of Various Limit Properties** section in the Extras chapter. With this fact we can now do limits like the following example.

Example 3 Evaluate the following limit.

$$\lim_{x \to 0} \mathbf{e}^{\sin x}$$ $$\lim_{x \to 0} \mathbf{e}^{\sin x}$$

Solution

Since we know that exponentials are continuous everywhere we can use the fact above.

$$\lim_{x \to 0} \mathbf{e}^{\sin x} = \mathbf{e}^{\lim_{x \to 0} \sin x} = \mathbf{e}^0 = 1$$

$$\lim_{x\to 0} \mathbf{e}^{\sin x} = \mathbf{e}^{\lim_{x\to 0}\sin x} = \mathbf{e}^{0} = 1$$

Another very nice consequence of continuity is the Intermediate Value Theorem.

2.1 Intermediate Value Theorem

Suppose that *f(x)* is continuous on [*a, b*] and let *M* be any number between *f(a)* and *f(b)*. Then there exists a number *c* such that,

1. $a < c < b$ $a < c < b$
2. $f(c) = M$ $f(c) = M$

All the Intermediate Value Theorem is really saying is that a continuous function will take on all values between *f(a)* and *f(b)*. Below is a graph of a continuous function that illustrates the Intermediate Value Theorem.

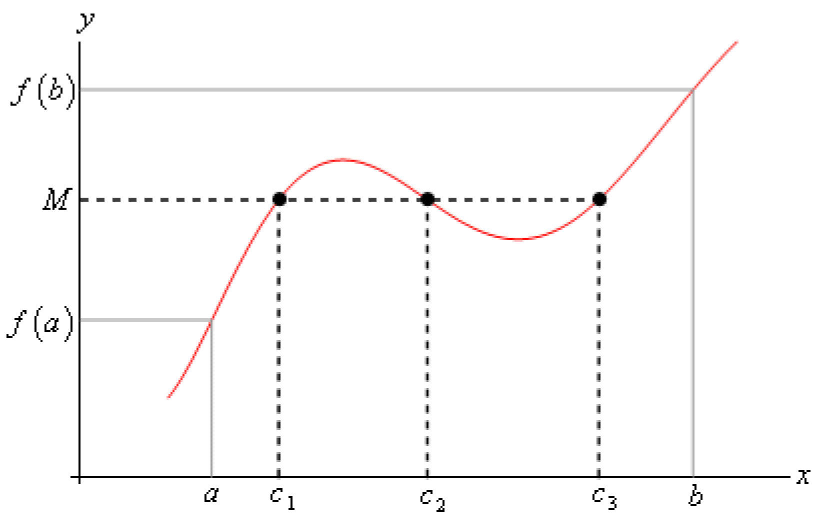


Fig. 4

As we can see from this image if we pick any value, *M*, that is between the value of *f(a)* and the value of *f(b)* and draw a line straight out from this point the line will hit the graph in at least one point. In other words somewhere between *a* and *b* the function will take on the value of *M*. Also, as the figure shows the function may take on the value at more than one place.

It‘s also important to note that the Intermediate Value Theorem only says that the function will take on the value of *M* somewhere between *a* and *b*. It doesn‘t say just what that value will be. It only says that it exists.

So, the Intermediate Value Theorem tells us that a function will take the value of *M* somewhere between *a* and *b* but it doesn‘t tell us where it will take the value nor does it tell us how many times it will take the value. There are important ideas to remember about the Intermediate Value Theorem.

A nice use of the Intermediate Value Theorem is to prove the existence of roots of equations as the following example shows.

Exampl 4 Show that $p(x) = 2x^3 - 5x^2 - 10x + 5$

$p(x) = 2x^3 - 5x^2 - 10x + 5$

[-

Solution

What we‘re really asking here is whether or not the function will take

$$p(x) = 0 \qquad p(x) = 0$$

somewhere between -1 and 2. In other words, we want to show that there is a number *c* such that $-1 < c < 2$ $-1 < c < 2$ and $p(c) = 0$ $p(c) = 0$. However if defin $M = 0$ $M = 0$ and acknowledge $a = -1$ $a = -1$ $b = 2$ we $b = 2$ can see that these two condition on *c* are exactly the conclusions of the Intermediate

So, this problem is set up to use the Intermediate Value Theorem and in fact, all we need to do that the function is continuous and that $M = 0$ $M = 0$ is $p(-1)$ $p(-1)$ an $p(2)$ $p(2)$ (*i.e* $p(-1) < 0 < p(2)$ $p(-1) < 0 < p(2)$ o $p(2) < 0 < p(-1)$ $p(2) < 0 < p(-1)$ and we‘ll be

To do this all we need to do is

$$p(-1)=8 \qquad\qquad p(2)=-19$$

$p(-1)=8$ $p(2)=-19$

So we

$$-19=p(2)<0<p(-1)=8$$

$-19=p(2)<0<p(-1)=8$

Therefor $M=0$ $M=0$ is betwee $p(-1)$ $p(-1)$ an $p(2)$ $p(2)$ an since $p(x)$ $p(x)$ is a polynomial it's continuous everywhere and so in particular it's continuous on the interval [-1,2]. So by the Intermediate Value Theorem there must be a $-1<c<2$ number $-1<c<2$

$$p(c)=0$$

$p(c)=0$

Therefore the polynomial does have a root between -1

For the sake of completeness here is a graph showing the root that we just proved existed. Note that we

used a computer program to actually find the root and that the Intermediate Value Theorem

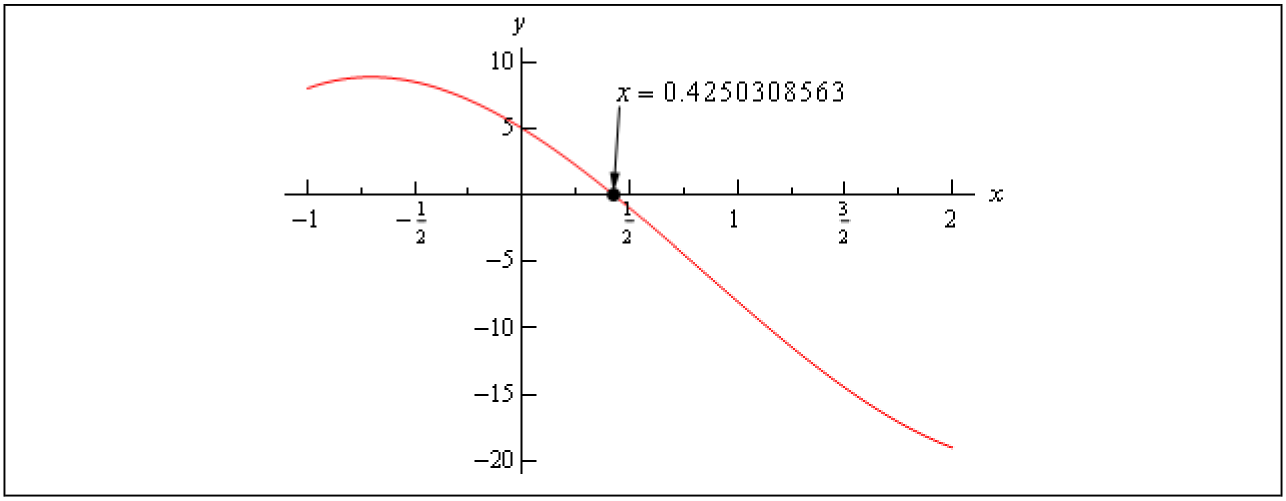


Fig. 5

Let's take a look at another example of the Intermediate Value Theorem.

$$f(x) = 20\sin(x+3)\cos\left(\frac{x^2}{2}\right)$$

Exampl 5 I possible determi i

$f(x) = 20\sin(x+3)\cos\left(\frac{x^2}{2}\right)$ takes the following values in the

[0,5]

(a) $f(x)=10$ $f(x)=10$? [**Solution**]

(b) $f(x)=-10$ $f(x)=-10$?

Solution

Okay, so much as the previous example we're being asked to determine, if possible, if the function takes on either of the two values above in the interval [0,5]. First, let's notice that this is a continuous function and

Now, for each part we will let *M* be the given value for that part and then we'll need to show betwee $f(0)$ $f(0)$ an $f(5)$ $f(5)$. If it does then we can use the Intermediate Theorem to prove that the function will take the

So, since we'll need the two function evaluations for each part let's give them here,

$$f(0) = 2.8224 \qquad f(5) = 19.7436$$

Now, let's take a look at each part.

(a) Okay, in this case we'll define $M = 10$ and we can see that,

$$f(0) = 2.8224 < 10 < 19.7436 = f(5)$$

So, by the Intermediate Value Theorem there must be a number $0 \le c \le 5$ such that

$$f(c) = 10$$

(b) In this part we'll define $M = -10$. We now have a problem. In this part M does not live between $f(0)$ and $f(5)$. So, what does this mean for us? Does this mean that $f(x) \ne -10$ in [0,5]?

Unfortunately for us, this doesn't mean anything. It is possible that $f(x) \ne -10$ in [0,5], but is it also possible that $f(x) = -10$ in [0,5]. The Intermediate Value Theorem will only tell us that c's will exist. The theorem will NOT tell us that c's don't exist.

In this case it is not possible to determine if $f(x) = -10$ in [0,5] using the Intermediate Value Theorem.

Okay, as the previous example has shown, the Intermediate Value Theorem will not always be able to tell us what we want to know. Sometimes we can use it to verify that a function will take some value in a given interval and in other cases we won't be able to use it.

For completeness sake here is the graph of $f(x) = 20\sin(x+3)\cos\left(\frac{x^2}{2}\right)$

$f(x) = 20\sin(x+3)\cos\left(\frac{x^2}{2}\right)$ in the interval [0,5].

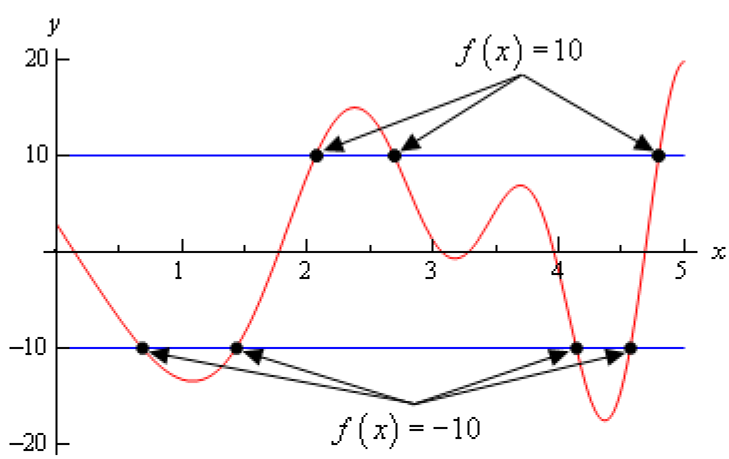


Fig. 6 $f(x) = -10$ $f(x) = -10$

From this graph we can see that not only does an interval with a range value from in [0,5] it does so a total of 4 times! Also note that as we verified in the first part of the previous example $f(x) = 10$ $f(x) = 10$ in [0,5] and in fact it does so a total of 3 times.

So, remember that the Intermediate Value Theorem will only verify that a function will take on a given value. It will never exclude a value from being taken by the function. Also, if we can use the Intermediate Value Theorem to verify that a function will take on a value it never tells us how many times the function will take on the value, it only tells us that it does take the value.

2.3 Nonlinear Programming

Numerous mathematical-programming applications, including many introduced in previous chapters, are cast naturally as linear programs. Linear programming assumptions or approximations may also lead to appropriate problem representations over the range of decision variables being considered. At other times, though, nonlinearities in the form of either nonlinear objective functions or nonlinear constraints are crucial for representing an application properly as a mathematical program. This chapter provides an initial step toward coping with such nonlinearities, first by introducing several characteristics of nonlinear programs and then by treating problems that can be solved using simplex-like pivoting procedures. As a consequence, the techniques to be discussed are primarily algebra-based. The final two sections comment on some techniques that do not involve pivoting. As our discussion of nonlinear programming unfolds, the reader is urged to reflect upon the linear- programming theory that we have developed previously, contrasting the two theories to understand why the nonlinear problems are intrinsically more difficult to solve. At the same time, we should try to understand the similarities between the two theories, particularly since the nonlinear results often are motivated by, and are direct extensions of, their linear analogs. The similarities will be particularly visible for the material of this chapter where simplex-like techniques predominate.

2.4 NONLINEAR PROGRAMMING PROBLEMS

A general optimization problem is to select n decision variables $x_1, x_2, \ldots, x_n$ from a given feasible region in such a way as to optimize (minimize or maximize) a given objective function

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

of the decision variables. The problem is called a *nonlinear programming problem* (NLP) if the objective function is nonlinear and/or thefeasible region is determined by nonlinear constraints. Thus, in maximization form, the general nonlinear program is stated as:

Maximize $f(x_1, x_2, \ldots,$

$x_n)$, subject to:

$g_1(x_1, x_2, \ldots, x_n) \le b_1,$

$g_m(x_1, x_2, \ldots, x_n) \le b_m,$

where each of the constraint functions g_1 through g_m is given. A special case is the linear program that has been treated previously. The obvious association for this case is

n

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

$c_j x_j$,

$j = 1$

Nonlinear Programming Problems

n

$g_i(x_1, x_2, \ldots, x_n) =$

$a_{ij} x_j$

$(i = 1, 2, \ldots, m)$.

$j = 1$

Note that nonnegativity restrictions on variables can be included simply by appending the additional con- straints:

$g_{m+i}(x_1, x_2, \ldots, x_n) = -x_i \leq 0$

$(i = 1, 2, \ldots, n)$.

Sometimes these constraints will be treated explicitly, just like any other problem constraints. At other times, it will be convenient to consider them implicitly in the same way that nonnegativity constraints are handled implicitly in the simplex method.

For notational convenience, we usually let x denote the vector of n decision variables $x_1, x_2, \ldots, x_n$ —

that is, $x = (x_1, x_2, \ldots, x_n)$ — and write the problem more

concisely as Maximize $f(x)$,

subject to:

$g_i(x) \leq b_i$

$(i = 1, 2, \ldots, m)$.

As in linear programming, we are not restricted to this formulation. To minimize $f(x)$, we course

can of maximize $-f(x)$. Equality constraints $h(x) = b$ can be written as two inequality

constraints $h(x) \leq b$ and

$-h(x) \leq -b$. In addition, if we introduce a slack variable, each inequality constraint is transformed

to an equality constraint. Thus sometimes we will consider an alternative equality

form: Maximize $f(x)$, subject to:

$h_i(x) = b_i$

$(i = 1, 2, \ldots, m)$

$x_j \geq 0$

$(j = 1, 2, \ldots, n)$.

Usually the problem context suggests either an equality or inequality formulation (or a formulation with both types of constraints), and we will not wish to force the problem into either form. The following three simplified examples illustrate how nonlinear programs can arise in practice.

LOCAL vs. GLOBAL OPTIMUM

Geometrically, nonlinear programs can behave much differently from linear programs, even for problems with linear constraints. In Fig. 13.1, the portfolio-selection example from the last section has been plotted for several values of the tradeoff parameter θ. For each fixed value of θ, contours of constant objective values are concentric ellipses. As Fig. 13.1 shows, the optimal solution can occur:

a) at an interior point of the feasible region;

b) on the boundary of the feasible region, which is not an extreme point; or

c) at an extreme point of the feasible region.

As a consequence, procedures, such as the simplex method, that search only extreme points may not determine an optimal solution.

Figure 5 illustrates another feature of nonlinear-programming problems. Suppose that we are to minimize $f(x)$ in this example, with $0 \leq x \leq 10$. The point $x = 7$ is optimal. Note, however, that in the indicated dashed interval, the point $x = 0$ is the best feasible point; i.e., it is an optimal feasible point in the local vicinity of $x = 0$ specified by the dashed interval.

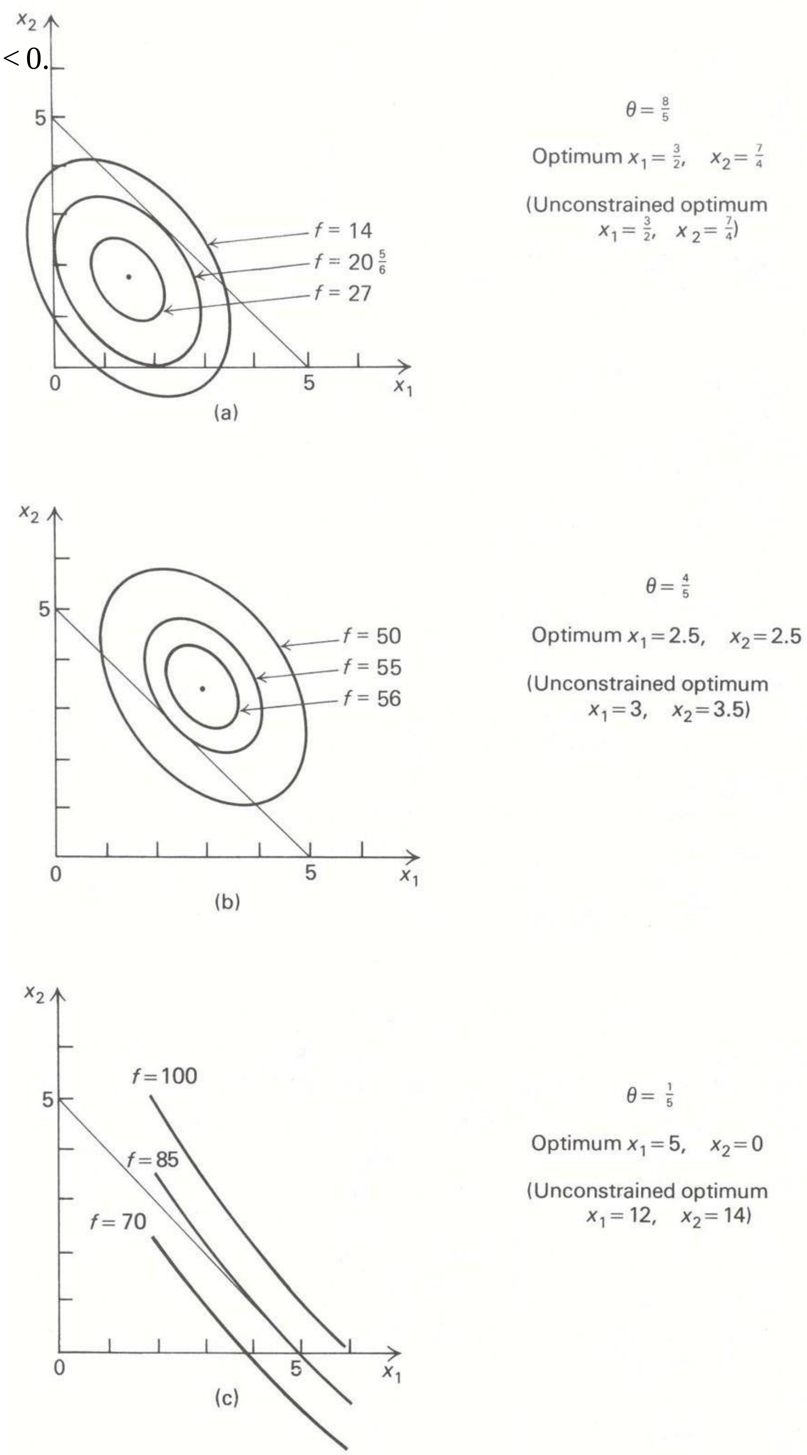
The latter example illustrates that a solution optimal in a local sense need not be optimal for the overall problem. Two types of solution must be distinguished. A global optimum is a solution to the overall optimization problem. Its objective value is as good as any other point in the feasible region. A local optimum, on the other hand, is optimal only with respect to feasible solutions close to that point. Points far removed from a local optimum play no role in its definition and may actually be preferred to the local optimum. Stated more formally,

Definition

Let $x = (x_1, x_2, \ldots, x_n)$ be a feasiblesolution to a maximization problem with objective function $f($

$x)$. We call x

1. A *global maximum* if $f(x) \geq f(y)$ for every feasible point $y = (y_1, y_2, \ldots, y_n)$; denotes absolute value; that is, $|x| = x$ if $x \geq 0$ and $|x| = -x$ if $x < 0$.



Portfolio-selection example for various values of θ. (Lines are contours of constant objective
values
)

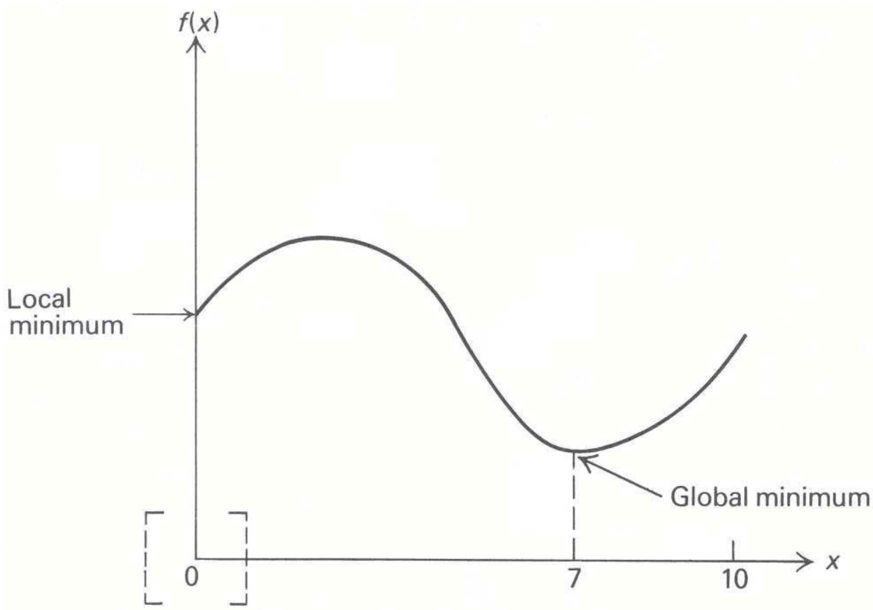


Fig. 7

2.5 Convex and Concave Functions

Local and global minima.

2. A *local maximum* if $f(x) \geq f(y)$ for every feasible point $y = (y_1, y_2, \ldots, y_n)$ sufficiently close to *x*.

That is, if there is a number

> 0 (possibly quite small) so that, whenever each variable y_j is

within of x_j — that is, $x_j - \leq y_j \leq x_j +$ —and *y* is feasible,

then $f(x) \geq f(y)$.

Global and local minima are defined analogously. The definition of local maximum simply says that if we place an *n*-dimensional box (e.g., a cube in three dimensions) about *x*, whose side has length 2 , then $f(x)$ is as small as $f(y)$ for every feasible point *y* lying within the box. (Equivalently, we can use *n*- dimensional spheres in this definition.) For instance, if

= 1 in the above example, the one-dimensional box, or interval, is pictured about the local minimum

$x = 0$ in Fig.7 The concept of a local maximum is extremely important. As we shall see, most general-purpose nonlinear- programming procedures are near-sighted and can do no better than determine local maxima.

We should point out that, since every global maximum is also a local maximum, the overall optimization problem can be viewed as seeking the best local maxima.

Under certain circumstances, local maxima and minima are known to be global. Whenever a function ‗curves upward‘‘ as in Fig. 13.3(a), a local minimum will be global. These functions are called *convex*. Whenever a function ‗curves downward‘‘ as in Fig. 13.3(b) a local maximum will be a global maximum.

These functions are called *concave*.† For this reason we usually wish to minimize convex functions and maximize concave functions. These observations are formalized below.

CONVEX AND CONCAVE FUNCTIONS

Because of both their pivotal role in model formulation and their convenient mathematical properties, certain functional forms predominate in mathematical programming. Linear functions are by far the most important. Next in importance are functions which are convex or concave. These functions are so central to the theory that we take some time here to introduce a few of their basic properties.

An essential assumption in a linear-programming model for profit maximization is constant returns to scale for each activity. This assumption implies that if the level of one activity doubles, then that activity‘s profit contribution also doubles; if the first activity level changes from x 1 to 2 x 1, then profit increases proportionally from say $20 to $40 [i.e., from c 1 x 1 to c 1(2 x 1)]. In many instances, it is realistic to assume constant returns to scale over the range of the data. At other times, though, due to economies of scale, profit might increase disproportionately, to say $45; or, due to diseconomies of scale (saturation effects), profit may be only $35.

In the former case, marginal returns are increasing with the activity level, and we say that the profit functions. As a mnemonic, the ‗‗A‘‘ in concave reflects the shape of these functions.

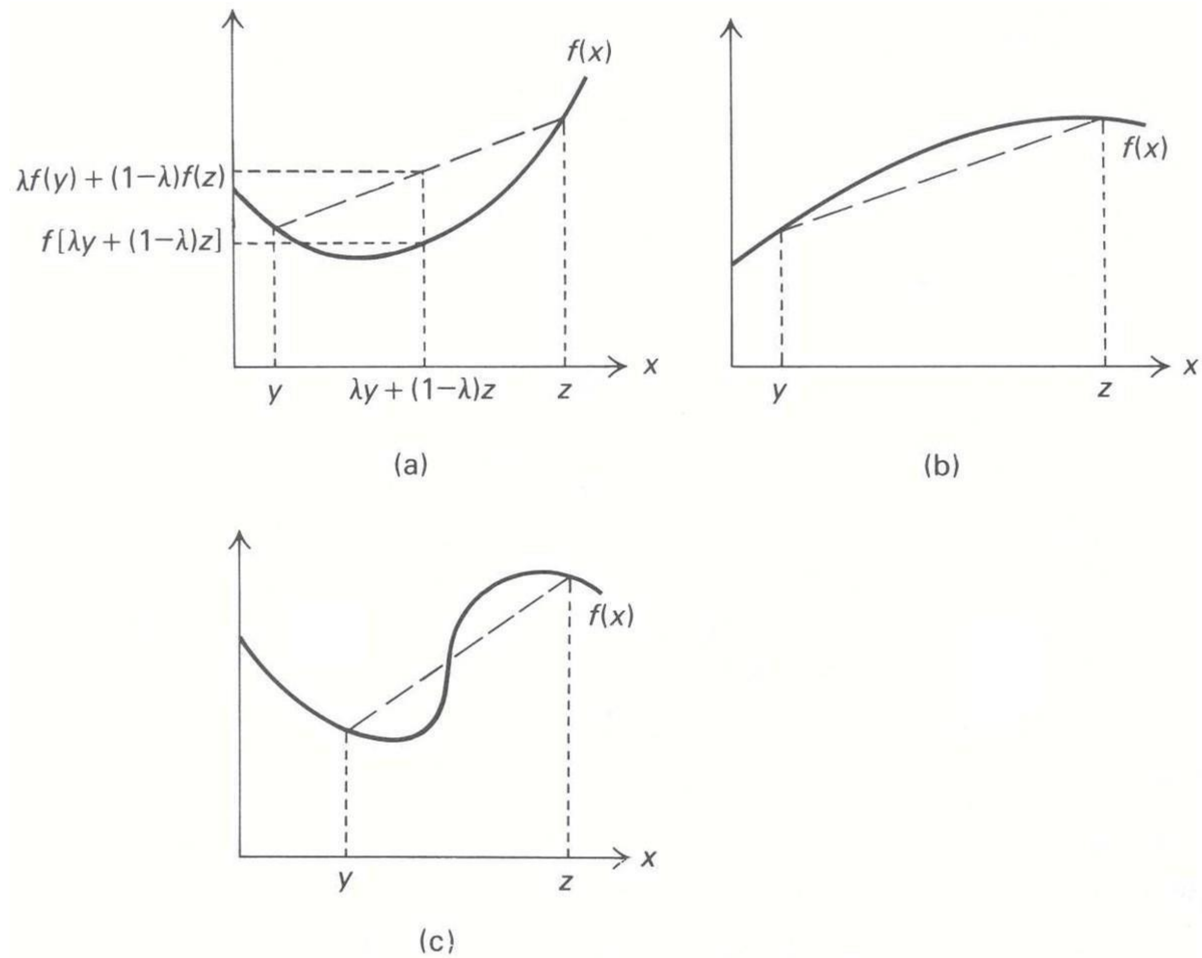


Fig. 8

Nonlinear Programming

a) Convex function b) concave function (c) nonconvex, nonconcave function.

is *convex* (Fig. 13.3(a)). In the second case, marginal returns are decreasing with the activity level and we say that the profit function is *concave* (Fig.b). Of course, marginal returns may increase over parts of the data range and decrease elsewhere, giving functions that are neither convex nor concave (Fig. (c)).

An alternative way to view a convex function is to note that linear interpolation overestimates its
values

That is, for any points y and z, the line segment joining $f(y)$ and $f(z)$ lies above the
. Fig).

function (see More intuitively, convex functions are __bathtub like‘‘ and hold water.

Algebraically,

Definition

A function $f(x)$ is called *convex* if,for every y and z and every $0 \leq \lambda \leq 1$,

$f[\lambda y + (1 - \lambda) z] \leq \lambda f(y) + (1 - \lambda) f(z)$.

It is called *strictly convex* if, for every two distinct points y and z and every $0 < \lambda < 1$,

$f[\lambda y + (1 - \lambda) z] < \lambda f(y) + (1 - \lambda) f(z)$.

The lefthand side in this definition is the function evaluation on the line joining x and y; the righthand side is the linear interpolation. Strict convexity corresponds to profit functions whose marginal returns are strictly increasing.

Note that although we have pictured f above to be a function of one decision variable, this is not a restric-tion. If $y = (y1, y2, \ldots, yn)$ and $z = (z1, z2, \ldots, zn)$, we must interpret $\lambda y + (1 - \lambda) z$ only as weighting the decision variables one at a time, i.e., as the decision vector $(\lambda y1 + (1 - \lambda) z1, \ldots, \lambda yn + (1 - \lambda) zn)$.

Concave functions are simply the negative of convex functions. In this case, linear interpolation under-estimates the function. The definition above is altered by reversing the direction of the inequality. Strict concavity is defined analogously. Formally,

Definition

A function $f(x)$ is called *concave* if, for every y and z and every $0 \leq \lambda \leq 1$,

$f[\lambda y + (1 - \lambda) z] \geq \lambda f(y) + (1 - \lambda) f(z)$.

It is called *strictly concave* if, for every y and z and every $0 < \lambda < 1$,

$f[\lambda y + (1 - \lambda) z] > \lambda f(y) + (1 - \lambda) f(z)$.

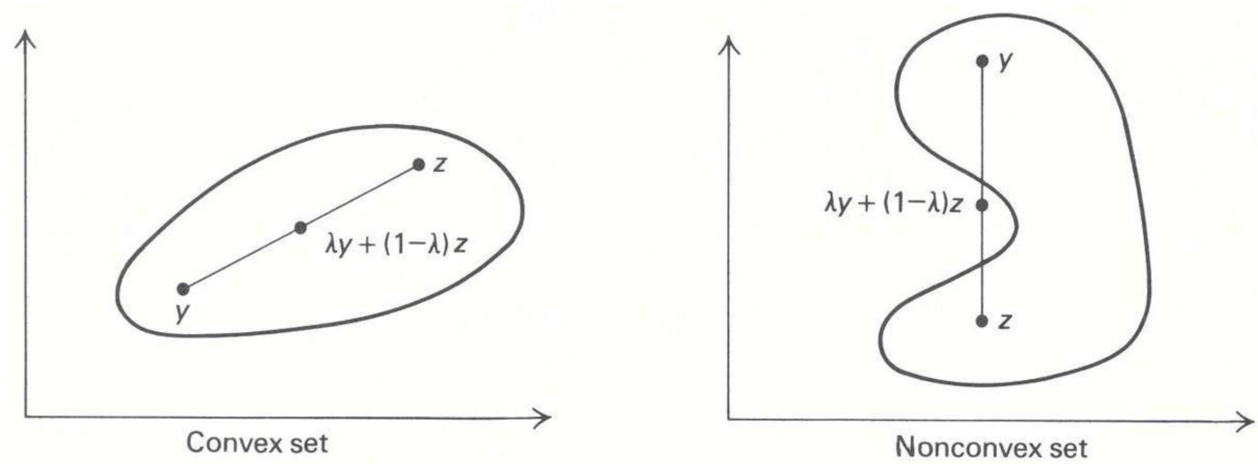


Fig. 9

2.6 Quadratic Approximation

The Formula for Quadratic Approximation

Quadratic approximation is an extension of linear approximation – we‘re adding one more term, which is related to the second derivative. The formula for the quadratic approximation of a function f(x) for values of x near x0 is:

f(x) ≈ f(x0) + f‘ (x0)(x − x0) + f‖(x0) (x − x0) 2 (x ≈ x0) 2 Compare this to our old formula for the linear approximation of f: f(x) ≈ f(x0) + f‘ (x0)(x − x0) (x ≈ x0).

We got from the linear approximation to the quadratic one by adding one more term that is related to the second derivative: f(x) ≈ f(x0) + f‘ (x0)(x − x0) + f‖(x0) (x − x0) 2 (x ≈ x0) 2 Linear Part Quadratic Part These are more complicated and so are only used when higher accuracy is needed. We‘d like to develop a catalog of quadratic approximations similar to our catalog of linear approximations.

Let‘s start by looking at the quadratic version of our estimate of ln(1.1). The formula for the quadratic approximation turns out to be: 2 x ln(1 + x) ≈ x − , 2 1 1 1 (1)2 and so ln(1.1) = ln(1 + 10) ≈ 10 − 2 10 = 0.095. This is not the value 0.1 that we got from the linear approximation, but it‘s pretty close (and slightly more accurate).

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO & CHEMICAL ENGINEERING

DEPARTMENT OF CHEMICAL ENGINEERING

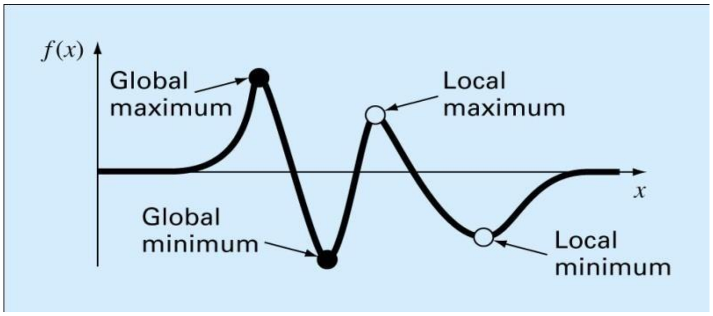
UNIT – III –Optimization of Chemical Processes – SCH1402

UNIT III One-Dimensional Unconstrained optimization

3.0 One dimensional minimization methods

1. **Analytical methods** (differential calculus methods)
2. **Numerical methods**
    a. **Elimination methods**
        i. Unrestricted search
        ii. Exhaustive search
        iii. Dichotomous search
        iv. Fibonacci method
        v. Golden section method
    b. **Interpolation methods**
        i. Requiring no derivatives (quadratic)
        ii. Requiring derivatives
            1. Cubic
            2. Direct root
                a. Newton
                b. Quasi-Newton
3. **Secant**

In *multimodal* functions, both local and global optima can occur. In almost all cases, we are interested in finding the absolute highest or lowest value of a function.



Direct search methods

The direct search methods use only the objective function values to locate the minimum point. The typical direct search methods include uniform search, uniform dichotomous search, sequential dichotomous search, Fibonacci search and golden section search methods.

Uniform search In the uniform search method, the trial points are spaced equally over the allowable range of values. Each point is evaluated in turn in an exhaustive search. For example, the designer wants to optimize the yield of a chemical reaction by varying the concentration of a catalyst, x and x lies over the range 0 to 10. Four experiments are available, and the same are distributed at equivalent spacing over the range =10L.

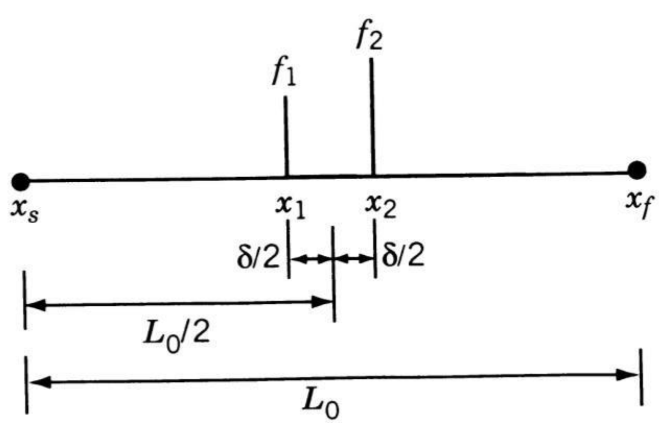
This divides L into intervals each of width L/n+1, where n is the number of experiments. From inspection of the results at the experimental points, we can conclude that the optimum will not lie in the ranges < 2x or > 6x . Therefore, we know the optimum will lie in between the range << 6x2 . So, the range of values that require further search is reduced to 40% of the total range with only four experiments.

3.1 Uniform dichotomous search Dichotomous search

1. The **dichotomous search method**, as well as the **Fibonacci** and the **golden section methods** discussed in subsequent sections, are **sequential search methods** in which the result of any experiment influences the location of the subsequent experiment.
2. In the dichotomous search, two experiments are placed as close as possible at the center of the interval of uncertainty.
3. Based on the relative values of the objective function at the two points, almost half of the interval of uncertainty is eliminated.

   Let the positions of the two experiments be given by:

where is a small positive number chosen such that the two experiments give significantly different results.



1. Then the new interval of uncertainty is given by (L0/2+/2).

2. The building block of dichotomous search consists of conducting a pair of experiments at the center of the current interval of uncertainty.
3. The next pair of experiments is, therefore, conducted at the center of the remaining interval of uncertainty.
4. This results in the reduction of the interval of uncertainty by nearly a factor of two.
5. The intervals of uncertainty at the ends of different pairs of experiments are given in the following table.

Number of experiments	2	4	6
Final interval of uncertainty	$(L_0+\delta)/2$	$\frac{1}{2}\left(\frac{L_0+\delta}{2}\right)+\frac{\delta}{2}$	$\frac{1}{2}\left(\frac{L_0+\delta}{4}+\frac{\delta}{2}\right)+\frac{\delta}{2}$

6. In general, the final interval of uncertainty after conducting n experiments (n even) is given by:

$$L_n = \frac{L_0}{2^{n/2}} + \delta\left(1 - \frac{1}{2^{n/2}}\right)$$

3.2 Fibonacci method

This method makes use of the sequence of Fibonacci numbers, $\{F_n\}$, for placing the experiments.

These numbers are defined as:

$$F_0 = F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2}, \quad n = 2,3,4,\ldots$$

which yield the sequence 1,1,2,3,5,8,13,21,34,55,89,...

Procedure:

Let L_0 be the initial interval of uncertainty defined by $a \le x \le b$ and n be the total number of experiments to be conducted.

n

define and place the first two experiments at points x_1 and x_2, which are located at a distance of L_2^* from each end of L_0.

This gives

$$x_1 = a + L_2^* = a + \frac{F_{n-2}}{F_n} L_0$$

$$x_2 = b - L_2^* = b - \frac{F_{n-2}}{F_n} L_0 = a + \frac{F_{n-1}}{F_n} L_0$$

Discard part of the interval by using the unimodality assumption. Then there remains a smaller interval of uncertainty L_2 given by:

$$L_2 = L_0 - L_2^* = L_0\left(1 - \frac{F_{n-2}}{F_n}\right) = \frac{F_{n-1}}{F_n} L_0$$

The only experiment left in will be at a distance of

$$L_2^* = \frac{F_{n-2}}{F_n} L_0 = \frac{F_{n-2}}{F_{n-1}} L_2$$

from one end and

$$L_2 - L_2^* = \frac{F_{n-3}}{F_n} L_0 = \frac{F_{n-3}}{F_{n-1}} L_2$$

from the other end. Now place the third experiment in the interval L_2 so that the current two experiments are located at a distance of:

$$L_3^* = \frac{F_{n-3}}{F_n} L_0 = \frac{F_{n-3}}{F_{n-1}} L_2$$

Value of n	Fibonacci Number, F_n
0	1
1	1
2	2
3	3
4	5
5	8
6	13
7	21
8	34
9	55
10	89
11	144
12	233
13	377
14	610
15	987
16	1,597
17	2,584
18	4,181
19	6,765
20	10,946

3.3 Newton's Method

A similar approach to Newton- Raphson method can be used to find an optimum of f(x) by defining a new function g(x)=f'(x). Thus because the same optimal value x* satisfies both

f'(x)=g(x*)=0*

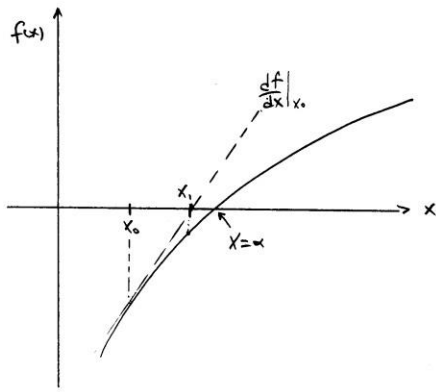
We can use the following as a technique to the extremum of *f(x)*.

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}$$

Newton's Method for Solving a Nonlinear Equation—an example

Let's say we want to evaluate the cube root of 467. That is, we want to find a value of *x* such that $x^3 = 467$. Put another way, we want to find a *root* of the following equation:

$$f(x) = x^3 - 467 = 0 .$$



If $f(x)$ were a straight line, then $f(x_1) = f(x_o) + \frac{df}{dx}(x_1 - x_o) = 0$.

In fact, $f(x_1) \neq 0$, but let's say that $f(x_1) \approx 0$ and solve for x_1.

$$x_1 = x_o + \frac{f(x_1) - f(x_o)}{\frac{df(x_o)}{dx}} = x_o - \frac{f(x_o)}{f'(x_o)}.$$

Note that we are using $f'(x_o) = \frac{df(x_o)}{dx}$.

Having now obtained a new estimate for the root, we repeat the process to obtain a sequence of estimated roots which we hope converges on the exact or correct root.

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$

etc.

In our example, $f(x) = x^3 - 467$ and $f'(x) = 3x^2$. If we take our *initial guess* to be $x_o = 6$, then by *iterating* the formula above, we generate the following table:

i	x_i	$f(x_i)$	$f'(x_i)$
0	6	-251	108

1	8.32 4	109.771 8	207.870 6
2	7.79 6	6.8172	182.331 6
3	7.75 9	0.108	0.0350

Fitting models by least squares

This section describes the basic idea of least squares estimation, which is used to calculate the values of the coefficients in a model from experimental data. In estimating the values of coefficients for either an empirical or theoretically based model, keep in mind that the number of data sets must be equal to or greater than the number of coefficients in the model. For example, with three data points of y versus x, you can estimate at most the values of three coefficients. Examine Figure 2.7. A straight line might represent the three points adequately, but the data can be fitted exactly using a quadratic model

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 \tag{2.2}$$

By introducing the values of a data point (Y_1, x_1) into Equation 2.2, you obtain one equation of Y_1 as a function of three unknown coefficients. The set of three data points therefore yields three linear equations in three unknowns (the coefficients) that can be solved easily.

To compensate for the errors involved in experimental data, the number of data sets should be greater than the number of coefficients p in the model. Least squares is just the application of optimization to obtain the "best" solution of the equations, meaning that the sum of the squares of the errors between the predicted and the experimental values of the dependent variable y for each data point x is minimized. Consider a general algebraic model that is linear in the coefficients.

3.4 Evaluation of Derivatives: Issues and Problems

All major NLP algorithms require estimation of first derivatives of the problem functions to obtain a solution and to evaluate the optimality conditions. If the val-ues of the derivatives are computed inaccurately, the algorithm may progress very slowly, choose poor directions for movement, and terminate due to lack of progress or reaching the iteration limits at points far from the actual optimum, or, in extreme cases, actually declare optimality at nonoptimal points.

3.5 Finite difference substitutes for derivatives

When the user, whether working on stand-alone software or through a spread-sheet, supplies only the values of the problem functions at a proposed point, the NLP code computes the first partial derivatives by finite differences. Each function is evaluated at a base point and then at a perturbed point. The difference between the function values is then divided by the perturbation distance to obtain an approx-imation of the first derivative at the base point. If the perturbation is in the positive direction from the base point, we call the resulting approximation a forward differ-ence approximation. For highly nonlinear functions, accuracy in the values of derivatives may be improved by using central differences; here, the base point is perturbed both forward and backward, and the derivative approximation is formed from the difference of the function values at thosk points. The price for this increased accuracy is that central differences require twice as many function eval-uations of forward differences. If the functions are inexpensive to evaluate, the additional effort may be modest, but for large problems with complex functions, the use of central differences may dramatically increase solution times. Most NLP codes possess options that enable the user to specify the use of central differences. Some codes attempt to assess derivative accuracy as the solution progresses and switch to central differences automatically if the switch seems warranted.

A critical factor in the accuracy of finite difference approximations for deriva-tives is the value of the perturbation step. The default values employed by all NLP codes (generally 1.E-6 to 1.E-7 times the value of the variable) yield good accuracy when the problem functions can be evaluated to full machine precision. When prob-lem functions cannot be evaluated to this accuracy (perhaps due to functions that are the result of iterative computations), the default step is often too small. The resulting derivative approximations then contain significant error. If the function(s) are highly nonlinear in the neighborhood of the base point, the default perturbation step may be too large to accurately approximate the tangent to the function at that point. Special care must be taken in derivative computation if the problem functions are not closed-form functions in compiled code or a modeling language (or, equiv-alently, a sequence of simple computations in a spreadsheet). If each function eval-uation involves convergence of a simulation, solution of simultaneous equations, or convergence of an empirical model, the interaction between the derivative pertur-bation step and the convergence criteria of the functions strongly affects the deriv-ative accuracy, solution progress, and reliability. In such cases, increasing the per-turbation step by two or three orders of magnitude may aid the solution process.

3.6 Analytic derivatives

Algebraic modeling systems, such as those described in Section 8.9.3, accept user-provided expressions for the objective and constraint functions and process them to produce additional expressions for the analytic first partial deriv-atives of these functions with respect to all decision variables. These expressions are exact, so the derivatives are evaluated to full machine precision (about 15 correct decimal digits using double precision arithmetic), and they are used by any derivative-based nonlinear code that is interfaced to the system. Finite-difference approximations to first derivatives have at most seven or eight signif-icant digits. Hence, an NLP code used within an algebraic modeling system can be expected to produce more accurate results in fewer iterations than the same solver using finite-difference derivatives. Chemical process simulators like Aspen also compute analytic derivatives and provide these to their nonlinear optimizers. Spreadsheet solvers currently use finite-difference approximations to derivatives.

Of course, many models in chemical and other engineering disciplines are difficult to express in a modeling language, because these are usually coded in FORTRAN or C (referred to as "general purpose" programming languages), as are many existing "legacy" models, which were developed before modeling systems became widely used. General-purpose languages offer great flexibility, and mod-els coded in these languages generally execute about ten times faster than those in an algebraic modeling system because FORTRAN and C are compiled, whereas statements in algebraic modeling systems are interpreted. This additional speed is ' especially important in on-line control applications derivatives in FORTRAN or C models may be approximated by differencing, or expressions for the derivatives can be derived by hand and coded in subroutines used by a solver. Anyone who has tried to write expressions for frrst derivatives of many complex functions of many variables knows how error-prone and tedious this process is. These shortcomings motivated the development of computer programs for ***automatic diflerentiation (AD).*** Given FORTRAN or C source code which eval-uates the functions, plus the user's specification of which variables in the program are independent, AD software augments the given program with additional state-ments that compute partial derivatives of all functions with respect to all indepen-dent variables. In other words, using AD along with FORTRAN or C produces a program that computes the functions and their first derivatives.

Currently, the most widely used AD codes are ADIFOR (automatic differentia-tion of FORTRAN) and ADIC (automatic differentiation of C). These are available at no charge from the Mathematics and Computer Science division of Argonne National Laboratories-see www.mcs.anl.gov for information on downloading the software and further information on AD. This software has been successfully applied to several difficult problems in aeronautical and structural design as well as chemi-cal process modeling.

What to Do When an NLP Algorithm Is Not "Working"

Probably the most common mode of failure of NLP algorithms is termination due to "fractional change" (i.e., when the difference in successive objective function values is a small fraction of the value itself over a set of consecutive iterations) at a point where the Kuhn-Tucker optimality conditions are far from satisfied. Some-times this criterion is not considered, so the algorithm terminates due to an itera-tion limit. Termination at a significantly nonoptimal point is an indication that the algorithm is unable to make any further progress. Such lack of progress is often associated with poor derivative accuracy, which can lead to search directions that do not improve the objective function. In such cases, the user should analyze the problem functions and perhaps experiment with different derivative steps or differ-ent starting points.

Parameter adjustment

Most **NLP** solvers use a set of default tolerances and parameters that control the algorithm's determination of which values are "nonzero," when constraints are satisfied, when optimality conditions are met, and other tuning factors.

Feasibility and optimality tolerances

Most **NLP** solvers evaluate the first-order optimality conditions and declare optimality when a feasible solution meets these conditions to within a specified tol-erance. Problems that reach what appear to be optimal solutions in a practical sense but require many additional iterations to actually declare optimality may be sped up by increasing the optimality or feasibility tolerances. See Equations (8.3 la) and (8.3 1b) for definitions of these tolerances. Conversely, problems that terminate at points near optimality may often reach improved solutions by decreasing the opti-mality or feasibility tolerances if derivative accuracy is high enough.

Other "tuning" issues

The feasibility tolerance is a critical parameter for GRG algorithms because it represents the convergence tolerance for the Newton iterations (see Section 8.7 for details of the GRG algorithm). Increasing this tolerance from its default value may speed convergence of slow problems, whereas decreasing it may yield a more accu-rate solution (at some sacrifice of speed) or "unstick" a sequence of iterations that are going nowhere. MINOS requires specification of a parameter that penalizes con-straint violations. Penalty parameter values affect the balance between seeking feasi-bility and improving of the objective function.

Scaling

The performance of most NLP algorithms (particularly on large problems) is greatly influenced by the relative scale of the variables, function 'values, and Jaco-bian elements. In general, NLP problems in which the absolute values of these quantities lie within a few orders of magnitude of each other (say in the range 0-100) tend to solve (if solutions exist) faster and with fewer numerical difficulties. Most codes either scale problems by default or allow the user to specify that the problem be scaled. Users can take advantage of these scaling procedures by build-ing models that are reasonably scaled in the beginning.

3.7 Model formulation

Users can enhance the reliability of any NLP solver by considering the follow-ing simple model formulation issues:

Avoid constructs that may result in discontinuities or undefined function argu-ments. Use exponential functions rather than logs. Avoid denominator terms that may tend toward zero (i.e., llx or ll(x- 1), etc.), multiplying out these denominators where possible.

Be sensitive to possible "domain violations," that is, the potential for the optimizer to move variables to values for which the functions are not defined (negative log arguments, negative square roots, negative bases for fractional exponents) or for which the functions that make up the model are not valid expressions of the sys-tems being modeled.

Starting points

The performance of NLP solvers is strongly influenced by the point from which the solution process is started. Points such as the origin (0,0, . . .) should be avoided because there may be a number of zero derivatives at that point (as well as problems with infinite values). In general, any point where a substantial number of zero derivatives are possible is undesirable, as is any point where tiny denominator values are possible. Finally, for models of physical processes, the user should avoid starting points that do not represent realistic operating conditions. Such points may cause the solver to move toward points that are stationary points but unacceptable configurations of the physical system.

Local and global optima

is was discussed in Section, a global optimum is a feasible solution that has the best objective value. A local optimum has an objective value that is better than that of any "nearby" feasible solution. All NLP algorithms and solvers here are only capable of finding local optima. For convex programs, any local optimum is also global. Unfortunately, many NLPs are not convex or cannot be guaranteed to be convex, hence we must consider any solution returned by an NLP solver to be local. The user should examine the solution for reasonableness, perhaps re-solving the problem from several starting points to investigate what local optima exist and how these solutions differ from one another. He/she can also try a global optimizer;

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO & CHEMICAL ENGINEERING
DEPARTMENT OF CHEMCAL ENGINEERING

UNIT IV Constrained optimization

4.0 The Linear Programming Model

Let: $X_1, X_2, X_3, \ldots\ldots\ldots, X_n$ = decision variables Z = Objective function or linear function <u>Requirement:</u> Maximization of the linear function Z.

$$Z = c_1X_1 + c_2X_2 + c_3X_3 + \ldots\ldots\ldots + c_nX_n$$ subject to the following

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\
\vdots \qquad\qquad\qquad & \quad \vdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_n \\
\text{all } x_j &\geq 0
\end{aligned}$$

<u>constraints</u>:

where a_{ij}, b_i, and c_j are given constants.

The linear programming model can be written in more efficient notation as:

Maximize

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

subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i$$

where

$$i = 1, 2, \ldots, m$$

and

$$x_j \geq 0$$

where

$$j = 1, 2, \ldots, n$$

The decision variables, x_1, x_2, ..., x_n, represent levels of *n* competing activities. The linear programming model for this example can be summarized as:

Maximize

$$Z = 13x_1 + 11x_2$$

subject to:

$$\begin{aligned}
4x_1 + 5x_2 &\leq 1500 \\
5x_1 + 3x_2 &\leq 1575 \\
x_1 + 2x_2 &\leq 420 \\
x_1 &\geq 0 \\
x_2 &\geq 0
\end{aligned}$$

Evaluation Of Unidimensional Search Methods

In this chapter we described and illustrated only a few unidimensional search methods. Refer to Luenberger (1984), Bazarra et al. (1993), or Nash and Sofer (1996) for many others. Naturally, you can ask which unidimensional search method is best to use, most robust, most efficient, and so on. Unfortunately, the various algorithms are problem-dependent even if used alone, and if used as subroutines in optimization codes, also depend on how well they mesh with the particular code. Most codes simply take one or a few steps in the search direction, or in more than one direction, with no requirement for accuracy that f(x) be reduced by a sufficient amount.

From a given starting point, a search direction is determined, and f (x) is minimized in that direction. The search stops based on some criteria, and then a new search direction is determined, followed by another line search. The line search can be carried out to various degrees of precision. For example, we could use a simple successive doubling of the step size as a screening method until we detect the optimum has been bracketed. At this point the screening search can be terminated and a more sophisticated method employed to yield a higher degree of accuracy. In any event, refer to the techniques discussed in Chapter 5 for ways to carry out the line search

4.1 Graphical Solution to LP Problems

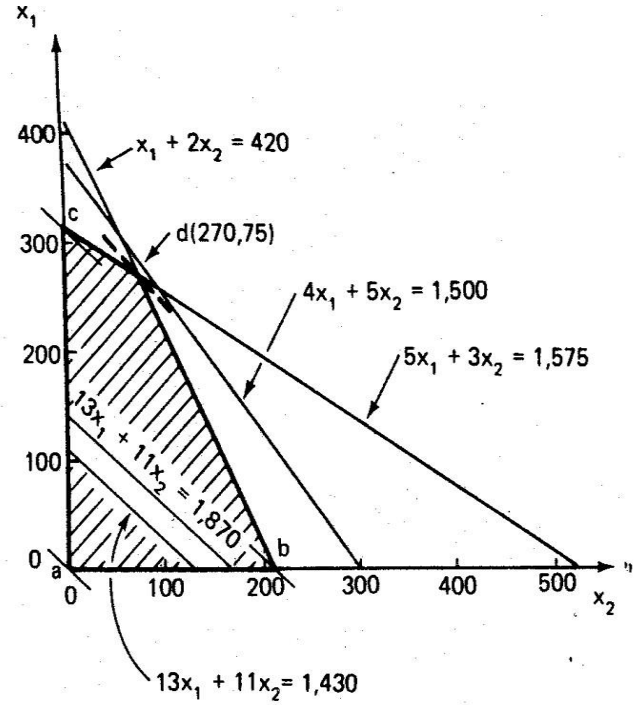


Figure 9.3 Graphical solution for linear programming problem.

- An equation of the form $4x1 + 5x2 = 1500$ defines a straight line in the $x1$-$x2$ plane. An inequality defines an area bounded by a straight line. Therefore, the region below and including the line $4x1 + 5x2 = 1500$ in the Figure represents the region defined by $4x1 + 5x2$ □ 1500.
- Same thing applies to other equations as well.
- The shaded area of the figure comprises the area common to all the regions defined by the constraints and contains all pairs of xI and $x2$ that are feasible solutions to the problem.
- This area is known as the *feasible region* or *feasible solution space.* The optimal solution must lie within this region.
- There are various pairs of $x1$ and $x2$ that satisfy the constraints such as:

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Trying different solutions, the optimal solution will be:

X1 = 270

X2 = 75 This indicates that maximum income of $4335 is obtained by producing 270 units of product I and 75 units of product II.

- In this solution, all the raw material and available time are used, because the optimal point lies on the two constraint lines for these resources.
- However, 1500- [4(270) + 5(75)], or 45 ft^2 of storage space, is not used. Thus the storage space is not a constraint on the optimal solution; that is, more products could be produced before the company ran out of storage space. Thus this constraint is said to be *slack.*

4.2 The Simplex Method

- When decision variables are more than 2, it is always advisable to use Simplex Method to avoid lengthy graphical procedure.
- The simplex method is not used to examine all the feasible solutions.
- It deals only with a small and unique set of feasible solutions, the set of vertex points (i.e., extreme points) of the convex feasible space that contains the optimal solution.
- **Steps involved:**
  - Locate an extreme point of the feasible region.
  - Examine each boundary edge intersecting at this point to see whether movement along any edge increases the value of the objective function.
  - If the value of the objective function increases along any edge, move along this edge to the adjacent extreme point. If several edges indicate improvement, the edge providing the greatest rate of increase is selected.
  - Repeat steps 2 and 3 until movement along any edge no longer increases the value of the objective function.

Example: Product Mix Problem

The N. Dustrious Company produces two products: I and II. The raw material requirements, space needed for storage, production rates, and selling prices for these products are given

	Product I	Product II
Storage space (ft^2/unit)	4	5
Raw material (lb/unit)	5	3
Production rate (units/hr)	60	30
Selling price ($/unit)	13	11

below:

The total amount of raw material available per day for both products is 15751b. The total storage space for all products is 1500 ft^2, and a maximum of 7 hours per day can be used for production.

The company wants to determine how many units of each product to produce per day to maximize its total income.

Solution

Step 1: Convert all the inequality constraints into equalities by the use of slack variables. Let:

S_1 = unused storage space
S_2 = unused raw materials
S_3 = unused production time

$$Z - 13x_1 - 11x_2 = 0 \quad (A1)$$
$$4x_1 + 5x_2 + S_1 = 1500 \quad (B1)$$
$$5x_1 + 3x_2 + S_2 = 1575 \quad (C1)$$
$$x_1 + 2x_2 + S_3 = 420 \quad (D1)$$
$$x_i \geq 0, \quad i = 1, 2$$

From the equations above, it is obvious that one feasible solution that satisfies all the constraints is: $x_1 = 0$, $x_2 = 0$, $S_1 = 1500$, $S_2 = 1575$, $S_3 = 420$, and $Z = 0$.

$$x_1 = -\frac{3}{5}x_2 - \frac{1}{5}S_2 + 315$$

Step 2: From Equation CI, which limits the maximum value of x1.

Substituting this equation into Eq. (5) yields the following new formulation of the model.

$$Z - \frac{16}{5}x_2 + \frac{13}{5}S_2 = 4095 \quad (A2)$$
$$+\frac{13}{5}x_2 + S_1 - \frac{4}{5}S_2 = 240 \quad (B2)$$
$$x_1 + \frac{3}{5}x_2 + \frac{1}{5}S_2 = 315 \quad (C2)$$
$$\frac{7}{5}x_2 - \frac{1}{5}S_2 + S_3 = 105 \quad (D2)$$

- It is now obvious from these equations that the new feasible solution is: x1 = 315, x2 = 0, S1 = 240, S2 = 0, S3 = 105, and Z = 4095
- It is also obvious from Eq.(A2) that it is also not the optimum solution. The coefficient of x1 in the objective function represented by A2 is negative (-16/5), which means that the value of Z can be further increased by giving x2 some

positive value.

Step 3: From Equation D2:

At each iteration, to minimize f(x), f(x) is evaluated at each of three vertices of the triangle. The direction of search is oriented away from the point with the highest value for the function through the centroid of the simplex. By making the search direction bisect the line between the other two points of the triangle, the direction goes through the centroid. A new point is selected in this reflected direction (as shown in Figure 6.3), preserving the geometric shape. The objective function is then evaluated at the new point, and a new search direction is determined. The method proceeds, rejecting one vertex at a time until the simplex straddles the optimum.

Various rules are used to prevent excessive repetition of the same cycle or simplexes. As the optimum is approached, the last equilateral triangle straddles the optimum point or is within a distance of the order of its own size from the optimum (examine Figure 6.4). The procedure cannot therefore get closer to the optimum and repeats itself so that the simplex size must be reduced, such as halving the length of all the sides of the simplex containing the vertex where the oscillation started. A new simplex composed of the midpoints of the ending simplex is constructed. When the simplex size is smaller than a prescribed tolerance, the routine is stopped. Thus, the optimum position is determined to within a tolerance influenced by the size of the simplex.

Nonlinear objective functions are sometimes nonsmooth due to the presence of functions like abs, min, max, or if-then-else statements, which can cause derivatives, or the function itself, to be discontinuous at some points. Unconstrained optimization methods that do not use derivatives are often able to solve nonsmooth NLP problems, whereas methods that use derivatives can fail. Methods employing derivatives can get "stuck" at a point of discontinuity, but -the functionvalue-only methods are less affected. For smooth functions, however, methods that use derivatives are both more accurate and faster, and their advantage grows as the number of decision variables increases. Hence, we now turn our attention to unconstrained optimization methods that use only first partial derivatives of the objective function.

$$x_2 = \frac{1}{7}S_2 - \frac{5}{7}S_3 + 75$$

Substituting this equation into Eq. (7) yield:

$$
\begin{aligned}
Z \quad & + \frac{15}{7}S_2 + \frac{16}{7}S_3 = 4335 \quad (A3) \\
S_1 & - \frac{3}{7}S_2 - \frac{13}{7}S_3 = 45 \quad (B3) \\
x_1 \quad & + \frac{2}{7}S_2 - \frac{3}{7}S_3 = 270 \quad (C3) \\
x_2 \quad & - \frac{1}{7}S_2 + \frac{5}{7}S_3 = 75 \quad (D3)
\end{aligned}
$$

From these equations, the new feasible solution is readily found to be: x1 = 270, x2 = 75, S1 = 45, S2 = 0, S3

= 0, Z = 4335.

Simplex Tableau for Maximization

Step I: Set up the initial tableau using Eq. (5).

$$
\begin{aligned}
Z - 13x_1 - 11x_2 \quad & = 0 \quad (A1) \\
4x_1 + 5x_2 + S_1 \quad & = 1500 \quad (B1) \\
5x_1 + 3x_2 \quad + S_2 \quad & = 1575 \quad (C1) \\
x_1 + 2x_2 \quad + S_3 & = 420 \quad (D1) \\
x_i \geq 0, \quad i = 1, 2 &
\end{aligned}
$$

Row Number	Basic Variable	Coefficients of: Z	x_1	x_2	S_1	S_2	S_3	Right-Hand Side	Upper Bound on Entering Variable
Initial tableau									
A1	Z	1	−13	−11	0	0	0	0	
B1	S_1	0	4	5	1	0	0	1500	375
C1	S_2	0	5	3	0	1	0	1575	315
D1	S_3	0	1	2	0	0	1	420	420

Step II: . Identify the variable that will be assigned a nonzero value in the next iteration so as to increase the value of the objective function. This variable is called the *entering variable.*

- It is that nonbasic variable which is associated with the smallest negative coefficient in the objective function.

- If two or more nonbasic variables are tied with the smallest coefficients, select one of these arbitrarily and continue.

Step III: Identify the variable, called the *leaving variable,* which will be changed from a nonzero to a zero value in the next solution.

Step IV: . Enter the basic variables for the second tableau. The row sequence of the previous tableau should be maintained, with the leaving variable being replaced by the entering variable.

Row Number	Basic Variable	Coefficients of: Z	x_1	x_2	S_1	S_2	S_3	Right-Hand Side	Upper Bound on Entering Variable
Initial tableau									
A1	Z	1	−13	−11	0	0	0	0	
B1	S_1	0	4	5	1	0	0	1500	375
C1	S_2	0	5	3	0	1	0	1575	315
D1	S_3	0	1	2	0	0	1	420	420
Second tableau at end of first iteration									
A2	Z	1	0	$-\frac{16}{5}$	0	$+\frac{13}{5}$	0	4095	
B2	S_1	0	0	$\frac{13}{5}$	1	$-\frac{4}{5}$	0	240	92.3
C2	x_1	0	1	$\frac{3}{5}$	0	$\frac{1}{5}$	0	315	525
D2	S_3	0	0	$\frac{7}{5}$	0	$-\frac{1}{5}$	1	105	75

Step V: Compute the coefficients for the second tableau. A sequence of operations will be performed so that at the end the x1 column in the second tableau will have the following coefficients:

	x_1
Z	0
S_1	0
x_1	1
S_3	0

The second tableau yields the following feasible solution:

x1 = 315, x2 = 0, SI = 240, S2 = 0, S3 = 105, and Z = 4095

Step VI: Check for optimality. The second feasible solution is also not optimal, because the objective function (row A2) contains a negative coefficient. Another iteration beginning with step 2 is necessary.

- In the third tableau (next slide), all the coefficients in the objective function (row A3) are positive. Thus an optimal solution has been reached and it is as follows: x1 = 270, x2 = 75, SI = 45, S2 = 0, S3 = 0, and Z = 4335

Row Number	Basic Variable	Coefficients of: Z	x_1	x_2	S_1	S_2	S_3	Right-Hand Side	Upper Bound on Entering Variable
Initial tableau									
A1	Z	1	−13	−11	0	0	0	0	
B1	S_1	0	4	5	1	0	0	1500	375
C1	S_2	0	5	3	0	1	0	1575	315
D1	S_3	0	1	2	0	0	1	420	420
Second tableau at end of first iteration									
A2	Z	1	0	$-\frac{16}{5}$	0	$+\frac{13}{5}$	0	4095	
B2	S_1	0	0	$\frac{13}{5}$	1	$-\frac{4}{5}$	0	240	92.3
C2	x_1	0	1	$\frac{3}{5}$	0	$\frac{1}{5}$	0	315	525
D2	S_3	0	0	$\frac{7}{5}$	0	$-\frac{1}{5}$	1	105	75
Third tableau at end of second and final iteration									
A3	Z	1	0	0	0	$+\frac{15}{7}$	$+\frac{16}{7}$	4335	
B3	S_1	0	0	0	1	$-\frac{3}{7}$	$-\frac{13}{7}$	45	
C3	x_1	0	1	0	0	$\frac{2}{7}$	$-\frac{3}{7}$	270	
D3	x_2	0	0	1	0	$-\frac{1}{7}$	$\frac{5}{7}$	75	

Answers to these questions can be obtained from the objective function in the last tableau of the simplex solution:

$$Z + \frac{15}{7}S_2 + \frac{16}{7}S_3 = \$4335$$

that is,

$$Z = \$4335 - \frac{15}{7}S_2 - \frac{16}{7}S_3$$

Barrier Methods

Barrier method

The barrier method solves a sequence of problems

$$\min_x \quad tf(x) + \phi(x)$$
$$\text{subject to} \quad Ax = b$$

for increasing values of $t > 0$, until $m/t \leq \epsilon$. We start at a value $t = t^{(0)} > 0$, and solve the above problem using Newton's method to produce $x^{(0)} = x^\star(t)$. Then for a barrier parameter $\mu > 1$, we repeat, for $k = 1, 2, 3, \ldots$

- Solve the barrier problem at $t = t^{(k)}$, using Newton's method initialized at $x^{(k-1)}$, to produce $x^{(k)} = x^\star(t)$
- Stop if $m/t \leq \epsilon$
- Else update $t^{(k+1)} = \mu t$

The first step above is called a centering step (since it brings $x^{(k)}$ onto the central path)

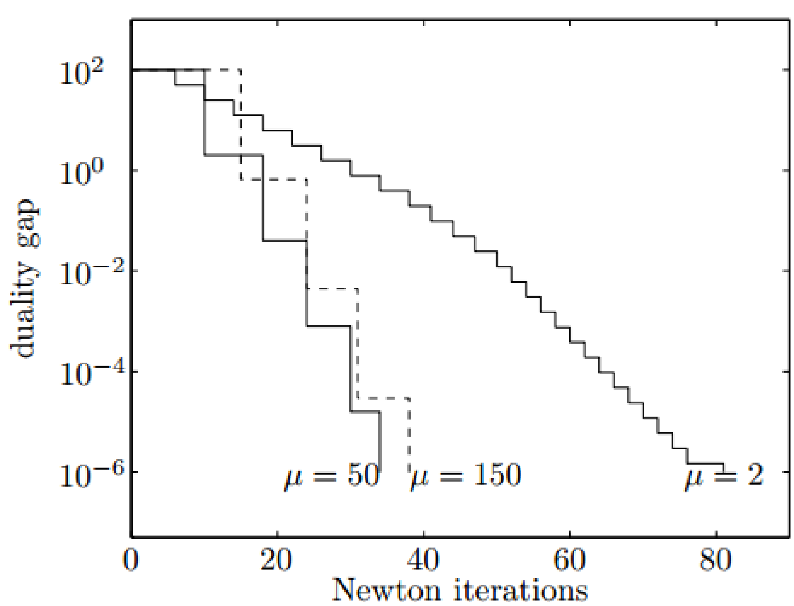
Considerations:

- Choice of μ: if μ is too small, then many outer iterations might be needed; if μ is too big, then Newton's method (each centering step) might take many iterations to converge
- Choice of $t^{(0)}$: if $t^{(0)}$ is too small, then many outer iterations might be needed; if $t^{(0)}$ is too big, then the first Newton's solve (first centering step) might require many iterations to compute $x^{(0)}$

Fortunately, the performance of the barrier method is often quite robust to the choice of μ and $t^{(0)}$ in practice

(However, note that the appropriate range for these parameters is scale dependent)

Example of a small LP in $n = 50$ dimensions, $m = 100$ inequality constraints (from B & V page 571):



Convergence analysis

Assume that we solve the centering steps exactly. The following result is immediate

Theorem: The barrier method after k centering steps satisfies

$$f(x^{(k)}) - f^\star \le \frac{m}{\mu^k t^{(0)}}$$

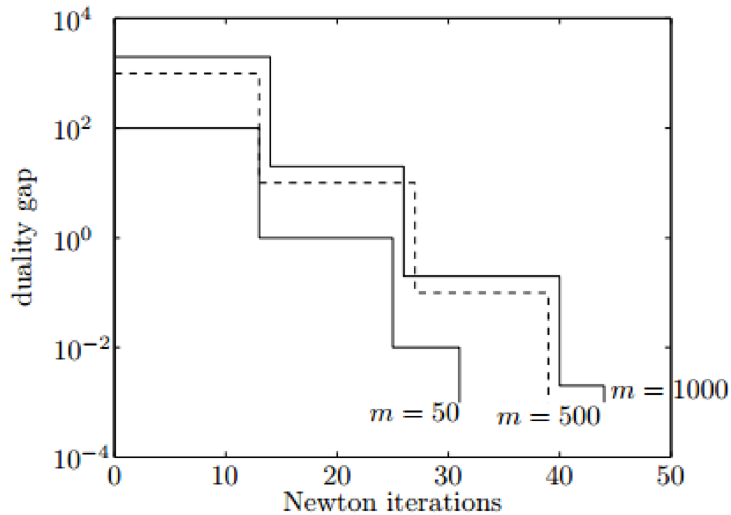
In other words, to reach a desired accuracy level of ϵ, we require

$$\frac{\log(m/(t^{(0)}\epsilon))}{\log \mu} + 1$$

centering steps with the barrier method (plus initial centering step)

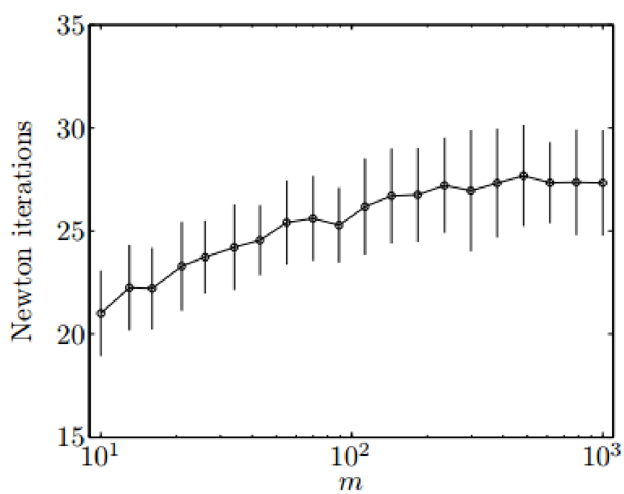
Is it reasonable to assume exact centering? Under mild conditions, Newton's method solves each centering problem to sufficiently high accuracy in nearly a constant number of iterations. (More precise statements can be made under self-concordance)

Example of barrier method progress for an LP with m constraints (from B & V page 575):



Can see roughly linear convergence in each case, and logarithmic scaling with m

Seen differently, the number of Newton steps needed (to decrease initial duality gap by factor of 10^4) grows very slowly with m:



Note that the cost of a single Newton step does depends on m (and moreso, on the problem dimension n)

4.3 Theory of linear programming

The linear programming approach to optimization problems includes the assumption that input data is known and is not subject to changes. In real life this assumption may be found inaccurate. For example, cost estimates are sometimes subject to errors, and to changes over time due to dynamic behavior of the environment; Demand reflects market behavior, which in itself is unpredictable to some degree; Resource availability may change when management changes its preferences. So, a question about the sensitivity of the optimal solution to changes in input parameters seems to be valid, and important for the sake of making informed decisions. This is the topic dealt with in these notes.

Sensitivity analysis allows for only one parameter change at a time. Since in reality several changes may occur simultaneously, we"ll extend the discussion to the multiple changes case later. For now, two types of changes are considered within the framework of a linear programming model.

(i) Changes in one objective-function coefficient.

(ii) Changes in one constraint right-hand-side.

First let us present a decision problem to be solved using linear programming. This problem will then serve as the vehicle with which we demonstrate the sensitivity analysis concepts.

Example

CPI manufactures a standard dining chair used in restaurants. The demand forecasts for chairs for quarter 1 and quarter 2 are 3700 and 4200, respectively. The chair contains an upholstered seat that can be produced by CPI or purchased from DAP. DAP currently charges $12.25 per seat, but has announced a new prices of $13.75 effective the second quarter. CPI can produce at most 3800 seats per quarter at a cost of $10.25 per seat. Seats produced or purchased in quarter 1 can be stored in order to satisfy demand in quarter 2. A seat cost CPI $1.50 each to hold in inventory, and maximum inventory cannot exceed 300 seats. Find the optimal make-or-buy plan for CPI.

The problem is formulated as follows:

X_1 = Number of seats produced by CPI in quarter 1.

X_2 = Number of seats purchased from DAP in quarter 1.

X_3 = Number of seats carried in inventory from quarter 1 to 2. X_4 = Number of seats produced by CPI in quarter 2. X_5 = Number of seats purchased from DAP in quarter 2.

The linear programming model is provided next:

Minimize $10.25X_1+12.5X_2+1.5X_3+10.25X_4+13.75X_5$

Subject to: $X_1+ X_2 = 3700+ X_3$

X_3+

$X_4+X_5=4200$

X_1□3800

X_4□380

0

X_3□300

X_1, X_2, X_3, X_4, and X_5 are non-negative

The linear programming model was run using SOLVER and the output results are given in the attached printout:

a. What is the optimal solution including the optimal value of the objective function?

X_1	X_2	X_3	X_4	X_5
3800	0	100	3800	300

The total cost (objective function) = \$82,175.

Management is interested in the analysis of a few changes that might be needed for various reasons. For example, the per-unit inventory cost may change from \$1.50 to \$2.50 due to an expected increase in the interest rate and the insurance costs. How will this change affect the optimal production plan? In addition, if CPI is considering increasing storage space such that 100 more seats can be stored, what is the maximum it should be willing to pay for this additional space? Questions like these can be answered by performing sensitivity analysis. Let us discuss the relevant concepts and then return to this problem to answer a few interesting questions.

Changing the value of one objective - function coefficient

Changing the value of one coefficient in the objective function, makes the variable associated with this coefficient more attractive or less attractive for the optimization mechanism. For example, if we look for the solution that maximizes the objective $5X_1 + 4X_2$, when the coefficient 4 becomes 6 (max $5X_1 + \mathbf{6}X_2$), the variable X_2 becomes more attractive. Therefore, one would expect the maximization mechanism to increase the value of X_2 in the optimal solution. It turns out that it is not necessarily so.

Statement 1: The Range of Optimality

The optimal solution of a linear programming model does not change if a single coefficient of some variable in the objective function changes within a certain range. This range is called the range of optimality. *Note, that only one coefficient is allowed to change for the range of optimality to apply.*

We can find the range of optimality for each objective coefficient in the SOLVER output. Adjustable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
B2	X1	3800	0	10.25	2	1E+30
C2	X2	0	0.25	12.5	1E+30	0.25
D2	X3	100	0	1.5	2	0.25
E2	X4	3800	0	10.25	3.5	1E+30
F2	X5	300	0	13.75	0.25	2

Range of optimality:
Upper bound = 10.25+2 = 12.25

Lower bound = 10.25 – infinity
= -infinity

Let us return to our example and answer a few sensitivity questions related to the range of optimality. **Question 1:** If the per-unit inventory cost increased from $1.50 to $2.50, would the optimal solution change?

Answer: First look for the changing parameter in the model. The coefficient 1.50 of the variable X_3 in the objective function is changing to 2.5: ($10.25X_1+12.5X_2+\mathbf{1.5}X_3+10.25X_4+13.75X_5$). We need to look for the range of optimality of the coefficient 1.5. From the output (see below) the range of optimality is: Lower bound =1.5 – 0.25 = 1.25

Upper bound = 1.5 + 2 = 3.5

Objective Coefficient	Allowable Increase	Allowable Decrease
1.5	2	0.25

<u>Interpretation:</u> As long as the coefficient of X_3 in the objective function (currently equals to 1.5) falls in the interval [1.25, 3.5] the current optimal solution does not change. Since the value 2.5 does fall in this

range, there will be no change in the optimal solution (in terms of the variable values!). However, the objective value changes!

New objective value = Current objective value + (Change in coefficient value)(the variable X3) = 82,175 + (2.5 – 1.5)(100)= 82,275. So, in spite of the increasing cost of holding inventory, it remains optimal to store 100 chairs at the end of quarter 1.

Question 2: If DAP reduced the selling price per seat in quarter 1 from $12.50 to $12.20, should CPI consider the purchase of seats in quarter 1 (note that currently no seat is purchased in quarter 1)? Answer: The parameter changing is the coefficient of X2 (12.5) in the objective function. It is changing to

12.25. The range of optimality is:

Lower bound = 12.5 – 0.25 =

12.25 Upper bound = 12.5 +

infinity = Infinity

Objective Coefficient	Allowable Increase	Allowable Decrease
12.5	1E+30	0.25

Interpretation: Since $12.20 falls below the lower bound of the range of optimality, there will be a change in the optimal solution, and seats will be purchased at this price (to rephrase, X2 becomes sufficiently attractive, so the minimization mechanism will make it a part of the optimal plan). Notice, that the objective value is likely to change, because the variables are optimized at different values. However, we cannot calculate the new objective value without re-running the model.

Comment: If the changing coefficient falls exactly on the boundary of the range of optimality, there will be more than one optimal solution with the same objective function value (called the multiple optimal or the alternate optimal solution case). For example, assume the coefficient 12.50 just discussed becomes 12.25. The two optimal solutions are:

Solution 1: the current solution;

Solution 2: a new solution, shown next:

X1	X2	X3	X4	X5
3800	200	300	3800	100

For both solutions the objective value is 82,175

The right hand side of a constraint (when the linear programming model is written in a standard form) is a constant that represents resource availability, minimum requirement of some property, activity level, etc. Changes in the right hand side value may occur, and sometimes affect the optimal solution. The effects that such changes may cause depend on whether the constraint is relaxed or is restricted more. Let us look at the following small example:

$$\text{Max } 2X_1 + 3X_2$$

Subject to

$X_1 + 2X_2$ □ 15□

$2X_1 + X_2$ □ 5□

Changing the right hand side of constraint 1 by +1 unit (making it $X_1 + 2X_2$ □ **16**□) makes it more relaxed, because we allow more values of X_1 and X_2 participate in the search for the optimal solution. Thus, the objective value cannot suffer from this change; either it remain the same or becomes better.

Changing the right hand side of constraint 1 by -1 unit (making it $X_1 + 2X_2$ □ **14**□) makes it more restrictive, because values included before in the feasible region are not feasible anymore. Thus, the objective function value cannot improve; either it remains the same (because the missing values did not constitute the previous optimal solution), or it suffers since the previous optimal solution is now infeasible.

The same observations (only of opposite directions) can be made for constraint 2. Since this constraint is of the „□" type, reducing its right hand side (making it $2X1 + X_2$ □ **4**)□ relaxes the constraint (check for yourself that more values of the decision variables are now feasible); while increasing its right hand side makes it more restrictive. To summarize we can state:

Statement 2:

- Increasing the right hand side of a „□" type constraint, or decreasing the right hand side of a „□" type constraint relaxes the constraint, thus the new objective value at the new optimal solution is either the same (no change) or better.

  For a maximization problem "better" means higher (a positive change in the objective value), and for a minimization problem "better" means lower (negative change in the objective value).
- Decreasing the right hand side of a „□" type constraint, or increasing the right hand side of a „□" type constraint restricts the constraint, thus the new objective value at the new optimal solution is either the same (no change) or worse.

Statement 3: The Shadow Price and the range of feasibility.

(i) **The shadow price for a constraint is defined as the change in the objective value when the right hand side of that constraint is increased by one unit.**

(ii) **The shadow price value remains unchanged as long as the right hand side of the constraint in question remains within a certain range called "Range of Feasibility".**

The shadow price and the range of feasibility for all the constraints appear in the computer output.

Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
Demand Quart 1	3700	12.25	3700	100	200
Demand Quart 2	4200	13.75	4200	1E+30	300
Capacity Quart 1	3800	-2	3800	200	100
Capacity Quart 2	3800	-3.5	3800	300	3800
Storage	100	0	300	1E+30	200

Range of Feasibility:
Upper bound = 3700+ 100
= 3800

To illustrate, for constraint 1 the range of feasibility is [3500, 3800]. That is, if the right hand side of the constraint changes within this range the shadow price remains 12.25. What is the significance of this result? What economical implication does it have? The following two questions deal with this topic.

Example – continued

Question 3: Management at CPI would like to understand the effects of different demand levels on the optimal solution and the total cost. Specifically, you are asked to find the total cost of meeting the demand, if in the first quarter we"ll experience an increase of 50 units in the demand for chairs. Answer: The parameter changing is the right hand side of constraint 1. The new right hand side value is 3750, still in the range of feasibility. The shadow price remains increase in the demand of quarter 1 the total cost increases by $12.25. The new total cost = Current total cost + (Shadow price)(50) = 82,175 + 12.25(50) = $82,787.5

Question 4: How much is it worth for CPI to increase production capacity in quarter 2?

Answer: The parameter changing is the right hand side of constraint 4. Observing the shadow price of constraint 4, for each unit increase in the production capacity in quarter 2 the objective reduces by $3.5. So each additional unit of production capacity saves CPI $3.5, and therefore, the worth of each additional unit of production capacity for CPI in quarter 2 is $3.5. Note that this is the maximum management at CPI should be willing to pay for one additional unit of production capacity in quarter 2 (to illustrate, suppose management considers the use of overtime, which results in production capacity increase. Then, every unit produced in overtime should not cost more than additional $3.5 – that is 10.25 + 3.5 = $13.75 at most).

Question 5: How much is it worth for CPI to increase its inventory capacity from 300 to 400 chairs? Answer: The parameter changing is the right hand side of constraint 5 (X_3 □ **40□0**). Its shadow price is [illegible].

Thus, the total cost (the objective function) does not change when the space allocated to inventory increases (this should not surprise you because currently only 100 seats are stored at the end of quarter 1, while 300 more could be stored). It turns out that no saving is obtained by adding storage space, thus CPI should not be willing to pay anything for this additional storage.

Multiple changes

All the changes considered above occurred one at a time (i.e. one objective coefficient changed while the others remained unchanged; one constraint right hand side changed while the other right hand sides remained unchanged); however, in many real world applications two or more changes need to be considered simultaneously.

For example, in our example, DAP might announce purchase price changes in both quarters.

To use the above results (that assume a single parameter change at a time) we turn to an empirical rule called "The 100% percent rule". It deals with multiple changes in different objective coefficients and determine when would the optimal solution remains unchanged, as well as with multiple changes in constraints right hand sides and determine when would the shadow prices not change.

The 100% Rule for objective-function coefficients: calculate ratios

a. **Define the objective function by $C_1X_1+C_2X_2+\ldots+C_nX_n$, and let more then one objective function coefficient change. Define the changes by □1, □2, ...k**

- **If □i is positive (that is the coefficient Ci increases) calculate the ratio {□i/"Max increase"}. ("Max increase" appears in the SOLVER output for the range of optimality).**
- **If □i is negative (that is the coefficient Ci decreases) calculate the ratio {|□i|/"Max decrease"}. ("Max decrease" appears in the SOLVER output for the range of optimality).**

b. **Add all the ratios calculated in part „b". If the sum of ratios is less than „1" the optimal solution remains unchanged. If the sum of ratios is „1" or more it is unclear whether or not the optimal solution changes.**

To understand this rule let us return to our example.

Question 6: If in quarter 2 the production cost per seat at CPI increases by $1.25; and DAP is changing its mind about the announced price increase leaving it at $12.50 per seat, would the optimal solution change? What would be the optimal total cost?

Answer: Two parameters are changing simultaneously: (i) the production unit cost in quarter 2 increases by DProd. = +1.25; (ii) the unit purchase price in quarter 2 decreases by DPurch. = – 1.25 (note: we first used a unit purchase price of 13.75, but now we need to use12.5, so the parameter change is 12.5 – 13.75 = - 1.25). To answer the question whether or not the solution changes, we must turn to the 100% rule since two parameters are changing simultaneously. By this rule we need to calculate two ratios and add them:

{DProd./Max increase}+ {DPurch./Max decrease} = 1.25/3.5 +|(-1.25)|/2 = .982 Interpretation: Since the sum is less than „1", the optimal solution won't change.

In spite of the changes that occur in favor of increasing the amount purchased from DAP while reducing the amount self-produced; the make-or-buy plan does not change.

A new value of the objective function can now be calculated, in accordance with the changes in the unit cost. New Total Cost = Current Total Cost + (1.25)(X_4) + (-1.25)(X_5)

= 82,175+1.25(3800)-1.25(300) = $86,550.

The 100% Rule for constraints right hand side:

c. **Let the constraints' right hand sides be called B_1, B_2, … , B_m, and let more then one constraint B_i change. Define the changes by □1, □2, ... , □k.**

d. **Calculate ratios as explained next:**

- **If □i is positive (that is the coefficient Bi increases) calculate the ratio {□i/"Max increase"}. ("Max increase" appears in the SOLVER output for the range of feasibility).**
- **If □i is negative (that is the coefficient Bi decreases) calculate the ratio {|□i|/"Max decrease"}. ("Max decrease" appears in the SOLVER output for the range of feasibility).**

e. **Add all the ratios calculated in part „b". If the sum of ratios is less than „1" the shadow prices remains unchanged. If the sum of ratios is „1" or more it is unclear whether or not the optimal solution changes.**

To understand this rule let us return again to our example.

Question 7: If CPI increases its production capacity by 100 seats in both quarter 1 and 2, will there be any savings or total cost increase?

Answer: Two constraints" right hand sides are changing simultaneously. The production capacity of 3800 in quarters 1 and 2 increase by DQuart1 = DQuart2 = 100. By the 100% rule we have: {DQuart1/Max increase}+{DQuart2/Max increase} = {100/200 + 100/300) = .833.

Since the sum is less than „1" the shadow prices remain unchanged, and thus can be used to find whether or not there going to be some savings. We need to calculate the change in the total cost. Change in total cost = (Shadow price Quarter 1)(100) + Shadow price in quarter 2)(100) = (-2)(100)+(-3.5)(100) = -$550. There will be a saving of $550 due to the production capacity increase (management should not pay more than $550 for the capacity increase in the two quarters combined).

In our next topic, "Parametric Analysis", we deal with multiple changes when the 100% rule is violated (possibly, when changes are greater than their maximum allowed).

Linear Mixed Integer Programs

IP is the name given to LP problems which have the additional constraint that some or all the variables have to be *integer*.

1. CLASSICAL INTEGER PROGRAMMING PROBLEMS

EXAMPLE 1: CAPITAL BUDGETING

A firm has n projects that it would like to undertake but because of budget limitations not all can be selected. In particular project j is expected to produce a revenue of c_j but requires an investment of a_{ij} in the time period i for i = 1,...m. The capital available in time period i is b_i. The problem of maximising revenue subject to the budget constraints can be formulated as follows: let $x_j = 0$ or 1 correspond to not proceeding or respectively proceeding with project j then we have to

$$\max \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i; \; i = 1, \ldots, m$$

$$0 \leq x_j \leq 1 \quad x_j \text{ integer} \quad j = 1, \ldots, n$$

Excel's Solver is a numerical optimization add-in (an additional file that extends the capabilities of Excel). It can be fast, easy, and accurate. It is not, however, a 100 percent guaranteed silver bullet. This document shows how to load and use Solver. It concludes with two important caveats concerning Solver.

Organization:

Accessing Excel’s Solver

Reviewing the Solver Parameters Dialog

Box Using Excel’s Solver: General

Description

Using Excel’s Solver: An Example

(**Solver.xls**) Two Dangers of Numerical

Optimization

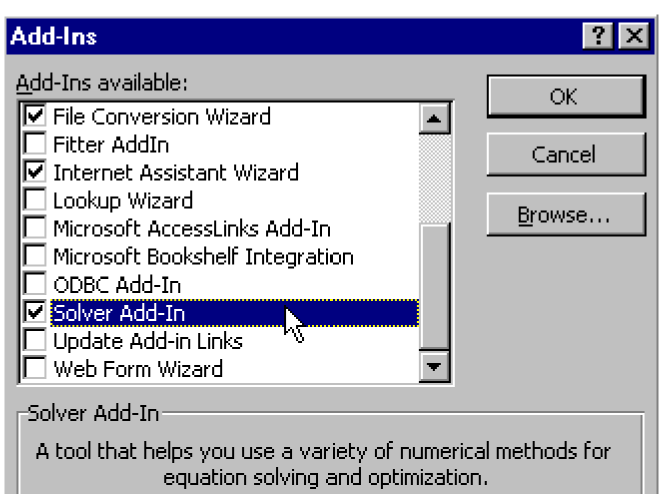
Accessing Excel‟s Solver

To use the Solver, click on the Tools heading on the menu bar and select the Solver . . . item.

Solver is not listed at left. Execute Tools: Add-Ins as described below in order to install Solver.

Upon successful installation, execute Tools: Solver as indicated at right.

If Solver is not listed (as shown above on the left), you must manually include it in the algorithms that Excel has available. To do this, select Tools from the menu bar and choose the "Add-Ins . . ." item. In the Add- Ins dialog box, scroll down and click on the Solver Add-In so that the box is checked as shown by the picture below:



After selecting the Solver Add- In and clicking on the OK button, Excel takes a moment to call in the Solver file and adds it to the Tools menu.

When you click on the Tools menu, it should be listed somewhere as shown above on the right.

If the Solver add-in is not listed in the Add-Ins dialog box, click on the Select or Browse button and navigate to the Solver add-in (called solver.xla in Windows and Solver on the MacOS) and open it. It should be in the Library directory in the folders where Microsoft Office is installed.

If you cannot find the Solver Add-In, try using the Mac"s Find File or Find in Windows to locate the file. Search for “solver.” Note the location of the file, return to the Add-Ins dialog box (by executing Tools: Add- Ins…), click on Select or Browse, and open the Solver Add-In file.

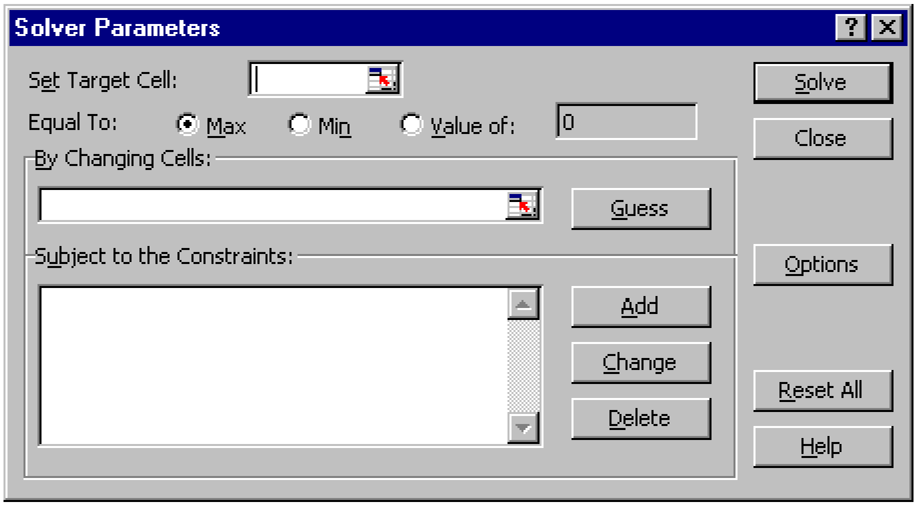
What if you still cannot find it? Then it is likely your installation of Excel failed to include the Solver Add-In. Run your Excel or Office Setup again from the original CD-ROM and install the Solver Add-In. You should now be able to use the Solver by clicking on the Tools heading on the menu bar and selecting the Solver item.

Although Solver is proprietary, you can download a trial version from Frontline Systems, the makers of Solver, at www.frontsys.com.

It is imperative that you successfully load and install the Solver add-in because without it, neither Solver nor the Dummy Dependent Variable Analysis add-in will be available.

Reviewing the Solver Parameters Dialog Box

After executing Tools: Solver . . . , you will be presented with the Solver Parameters dialog box below:



Let us review each part of this dialog box, one at a time.

Set Target Cell is where you indicate the objective function (or goal) to be optimized. This cell must contain a formula that depends on one or more other cells (including at least one "changing cell"). You can either type in the cell address or click on the desired cell.

(NOTE: If you click on the Collapse Dialog button, , the dialog box disappears and it will be easier to select a cell.)

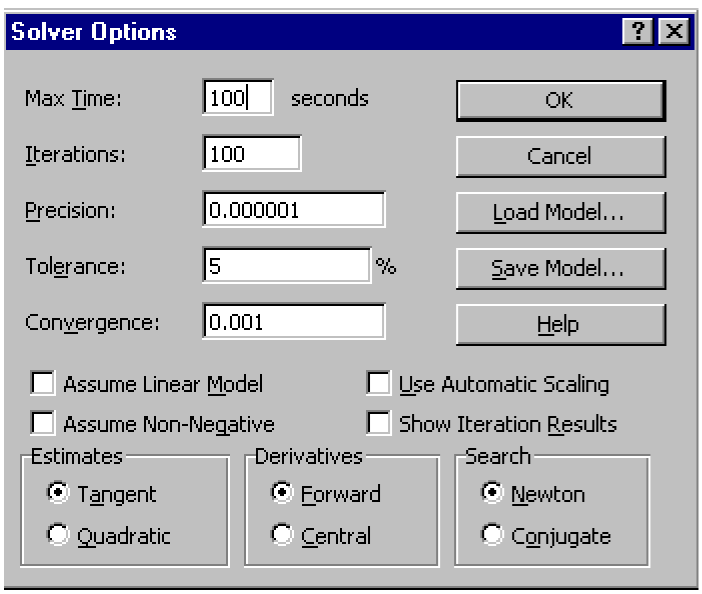
Equal to: gives you the option of treating the Target Cell in three alternative ways. **Max** (the default) tells Excel to maximize the Target Cell and **Min,** to minimize it, whereas **Value** is used if you want to reach a certain particular value of the Target Cell by choosing a particular value of the endogenous variable. If you choose Value, you must enter the particular value you want to attain in the box to the immediate right unless you want the value to be 0 (which is the default). Making the value equal to 0 enables Solver to find equilibrium solutions or roots to first-order conditions.

By Changing Cells permits you to indicate which cells are the adjustable cells (i.e., endogenous variables). As in the Set Target Cell box, you may either type in a cell address or click on a cell in the spreadsheet. Excel handles multivariable optimization problems by allowing you to include additional cells in the By Changing Cells box. Each noncontiguous choice variable is separated by a comma. If you use the mouse technique (clicking on the cells), the comma separation is automatic.

Guess controls the initial position of the changing cells. Excel uses the current values of the cells as the default. Solver is sensitive to the initial values. If a solution cannot be found, try different starting values.

Subject to the Constraints is used to impose constraints on the endogenous variables. We will rely on this important part of Solver when we do Constrained Optimization problems.

You can also use the Constraints part of Solver to help it find a solution. For example, in a profit maximization problem, you could tell Solver that Quantity must be greater than or equal to 0 (i.e., that negative values of *Q* are not allowed). If Excel has trouble finding a solution to a problem, limiting the possible values of the choice variables will help it find a solution.
Solver allows equality (Lagrange) or inequality (Kuhn-Tucker) constraints.
Add..., Change..., Delete buttons are used to create and alter the constraints you set. These buttons lead to dialog boxes on which you indicate your choices; then hit OK.
Returning to the top right-hand side of the Solver Parameters dialog box, we have the following:
Solve is obviously the button you click to get Excel's Solver to find a solution. This is the last thing you do in the Solver Parameters dialog box.
Close is just like cancel; it closes the Solver dialog box, and no changes are made.
Options...allows you to adjust the way in which Solver approaches the solution..

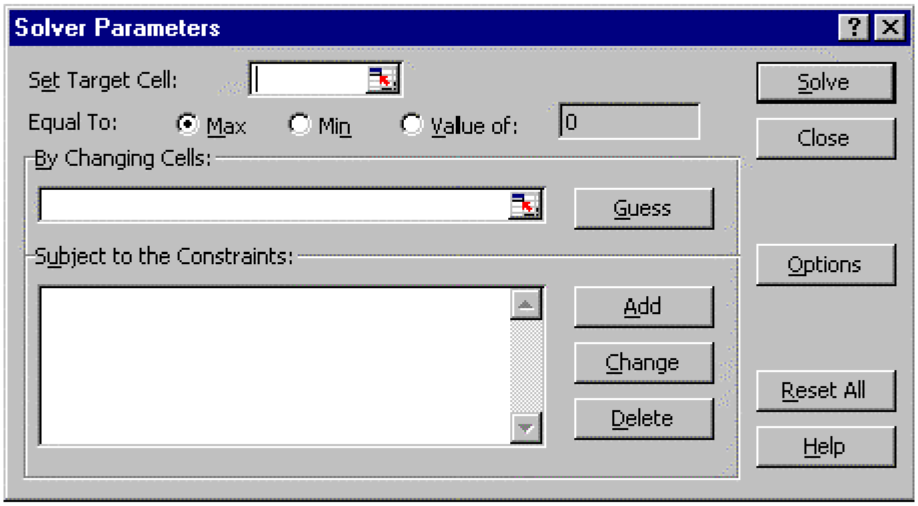


As you can see, a series of choices are included in the Solver Options dialog box that direct Solver's search for the optimum solution and for how long it will search. These options may be changed if Solver is having difficulty finding the optimal solution. Lowering the Precision, Tolerance, and Convergence values slows down the algorithm but may enable Solver to find a solution.

The Load and Save Model buttons enable you to recall and keep a complicated set of constraints or choices so that you do not have to reenter them every time.

Clicking OK or Cancel returns you to the Solver Parameters dialog box.

We continue our review of Solver options by going over the remaining buttons in the Solver Parameters dialog box (which we display again below):

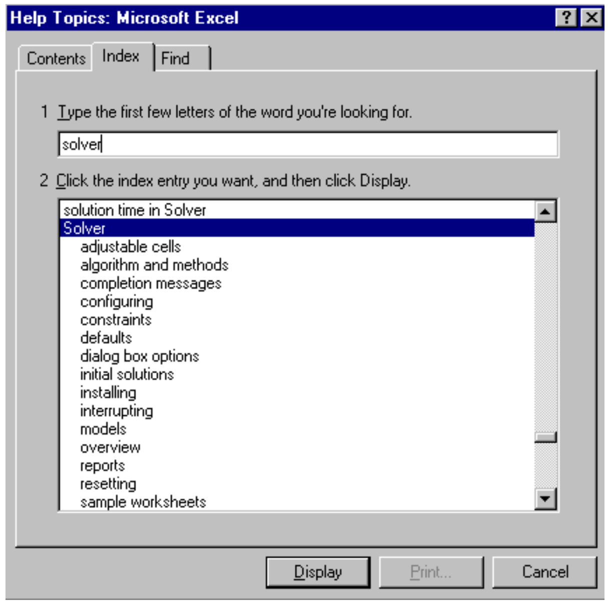


Reset All. This button changes everything back to the original, default choices, blanking out the Set Cell, By Changing Cells, and Subject to the Constraints options.

It is important to understand that a saved Excel workbook will remember the information included in the last Solver run.

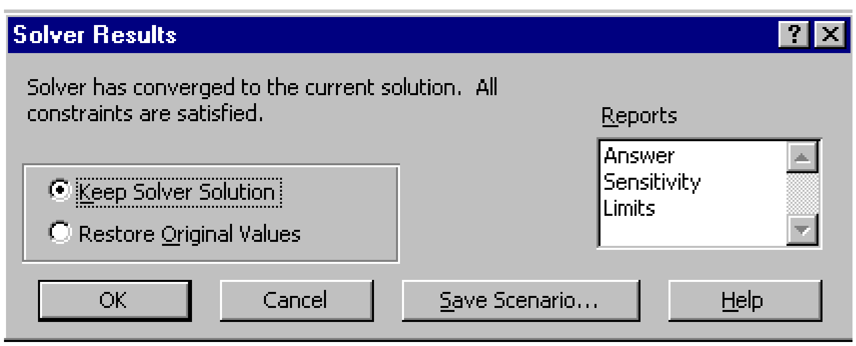
If you wish to explore a different problem and want to begin with a “clean” Solver quickly, then click on the Reset All button. If you wish to keep a particular Solver model, then use the Options and Save Model . . . Buttons.

Help brings up limited documentation on Solver. Better help is available by typing “solver” in the Index of Help (which can be accessed by executing Help: Contents and Index), as shown below



Using Excel"s Solver: General Description

When you run Excel"s Solver, it executes a series of macros and routines that constitute the Solver add-in. Upon completion of the various algorithms, Excel presents the user with a Solver Results dialog box:



A message appears on the top left-hand side of the box. In this case, Excel reports that "Solver has converged to the current solution. All constraints are satisfied." This is good news!

Bad news is a message like, "Solver could not find a solution." If this happens, you must diagnose, debug, and otherwise think about what went wrong and how it could be fixed. The two quickest fixes are to try different initial values and to add constraints to the problem.

From the Solver Results dialog box, you elect whether to have Excel write the solution it has found into the Changing Cells (i.e., Keep Solver Solution) or whether to leave the spreadsheet alone and NOT write the value of the solution into the Changing Cells (i.e., Restore Original Values). When Excel reports a successful run, you would usually want it to Keep the Solver Solution.

On the right-hand side of the Solver Results dialog box, Excel presents a series of reports. The Answer, Sensitivity, and Limits reports are additional sheets inserted into the current workbook. They contain diagnostic and other information and should be selected if Solver is having trouble finding a solution.

Along the bottom of the Solver Results dialog box are four buttons:

OK is obviously the button you click after reading and choosing various options you want to keep. This is the last thing you do in the Solver Results dialog box.

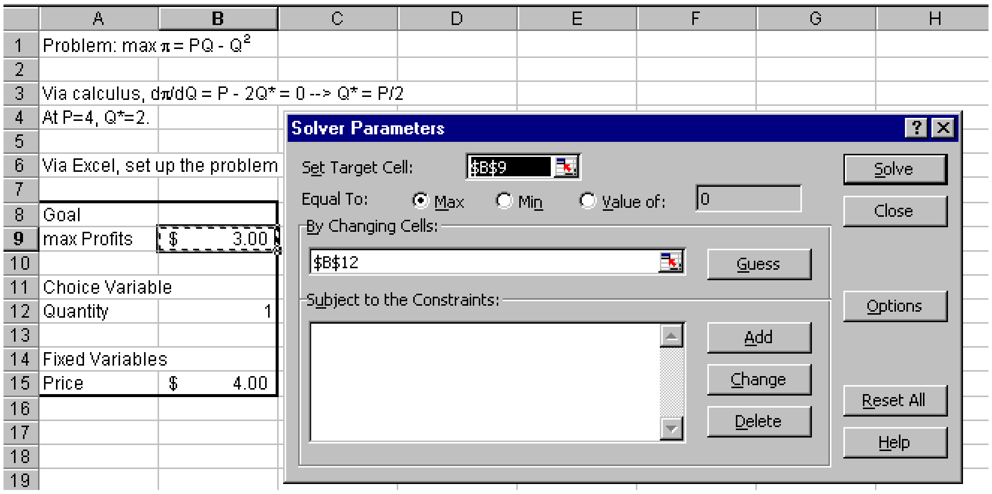
Cancel closes the Solver Results dialog box and no changes are made.

Save Scenario... enables the user to save particular solutions for given configurations.

Help brings up information from Excel"s Help application.

Using Excel‟s Solver: An Example

You can see Solver in action by opening the file Solver.xls. The *OptimalSolution* sheet has a simple profit- maximization problem set up, and Solver is ready to run. Execute Tools: Solver to access the Solver Parameters dialog box shown below.



converges on the optimal solution.

Another type of widely used modeling system is the spreadsheet solver. Microsoft Excel contains a module called the Excel Solver, which allows the user to enter the decision variables, constraints, and objective of an optimization problem into the cells of a spreadsheet and then invoke an LP, MILP, or NLP solver.

The power of linear programming solvers Modern LP solvers can solve very large LPs very quickly and reliably on a PC or workstation. LP size is measured by several parameters: (1) the number of variables n, (2) the number of constraints m, and (3) the number of nonzero entries nz in the constraint matrix A. The best measure is the number of nonzero elements nz because it directly determines the required storage and has a greater effect on computation time than n or m. For almost all LPs encountered in practice, nz is much less than mn, because each constraint involves only a few of the variables x. The problem density 100(ndmn) is usually less than 1%, and it almost always decreases as m and n increase. Problems with small densities are called sparse, and real world LPs are always sparse. Roughly speaking, a problem with under 1000 nonzeros is small, between 1000 and 50,000 is medium-size, and over 50,000 is large. A small problem probably has m and n in the hundreds, a medium-size problem in the low to mid thousands, and a large problem above 10,000. Currently, a good LP solver running on a fast (> 500 rnI-Iz) PC with substantial memory, solves a small LP in less than a second, a medium-size LP in minutes to tens of minutes, and a large LP in an hour or so. These codes hardly ever fail, even if the LP is badly formulated or scaled. They include preprocessing procedures that detect and remove redundant constraints, fixed variables, variables that must be at bounds in any optimal solution, and so on. Preprocessors produce an equivalent LP, usually of reduced size. A postprocessor then determines values of any removed variables and Lagrange multipliers for removed constraints. Automatic scaling of variables and constraints is also an option. Armed with such tools, an analyst can solve virtually any LP that can be formulated. Solving MILPs is much harder. Focusing on MILPs with only binary variables, problems with under 20 binary variables are small, 20 to 100 is medium-size, and over 100 is large. Large MILPs may require many hours to solve, but the time depends greatly. on the problem structure and the availability of a good starting point.

In addition to their use as stand-alone systems, LPs are often included within larger systems intended for decision support. In this role, the LP solver is usually hidden from the user, who sees only a set of critical problem input parameters and a set of suitably formatted solution reports. Many such systems are available for supply chain management-for example, planning raw material acquisitions and deliveries, production and inventories, and product distribution. In fact, the process industries--oil, chemicals, pharmaceuticals-have been among the earliest users. Almost every refinery in the developed world plans production using linear programming.

Available Linear Programming software

Many LP software vendors advertise in the monthly journal *OR/MS Today*, published by INFORMS. For a survey of LP software, see Fourer (1997, 1999) in that journal. All vendors now have Websites, and the following table provides a list of LP software packages along with their Web addresses.

Company name	Solver name	Web addresses/E-mail address
CPLEX Division of ILOG	CPLEX	www.cplex.com
IBM	Optimization Software Library (OSL)	www.research.ibm.com/osl/
LINDO Systems Inc.	LINDO	www.lindo.com
Dash Associates	XPRESS-MP	www.dashopt.com
Sunset Software Technology	AXA	Sunsetw@ix.netcom.com
Advanced Mathematical Software	LAMPS	info@amsoft.demon.co.uk

Solver parameters dialog to define this problem for the Excel Solver, the cells containing the decision variables, the constraints, and the objective must be specified. This is done by choosing the Solver command from the Tools menu, which causes the Solver parameters dialog shown to appear. The "Target Cell"' is the cell containing the objective function. Clicking the "Help'3ut~on explains all the steps needed to enter the "changing" (i.e., decision) variables and the constraints. We encourage you to "Reset all," and fill in this .dialog from scratch. Solver options dialog. Selecting the "Options" button in the Solver Parameters dialog brings up the Solver Options dialog box. The current Solver version does not determine automatically if the problem is linear or nonlinear.

Consequently, in this chapter we will discuss five major approaches for solving nonlinear programming problems with constraints: 1. Analytic solution by solving the first-order necessary conditions for optimality 2. Penalty and barrier methods 3. Successive linear programming 4. Successive quadratic programming 5. Generalized reduced gradient

Constrained Nonlinear Optimization

- Previously in this chapter, we solved NLP problems that only had objective functions, with no constraints.
- Now we will look at methods on how to solve problems that include constraints.

NLP with Equality Constraints

- First, we will look at problems that only contain equality

constraints: Minimize $f(\mathbf{x})$ $\mathbf{x} = [x_1\ x_2\ \dots\ x_n]$

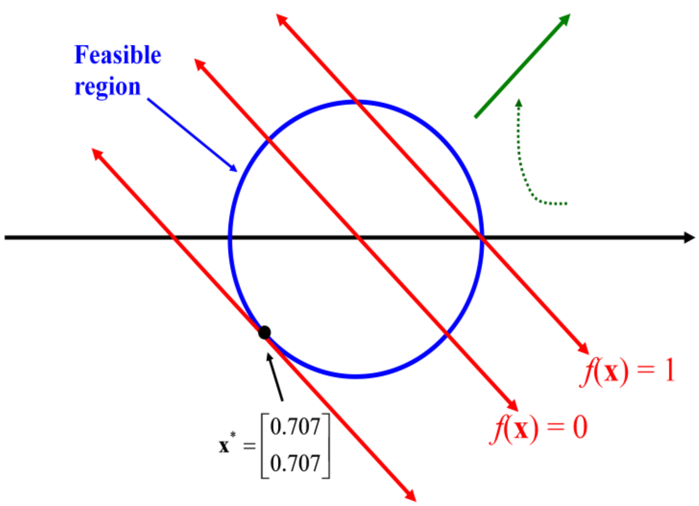
Subject to: $h_i(\mathbf{x}) = b_i$ $i = 1, 2, \dots, m$

Consider the problem:

Minimize $x_1 + x_2$

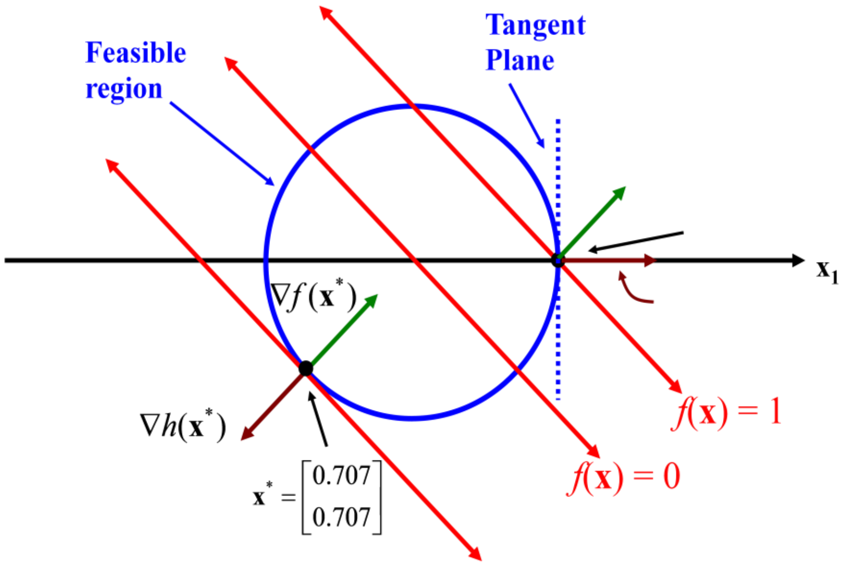
Subject to: $(x_1)^2 + (x_2)^2 - 1 = 0$

The feasible region is a circle with a radius of one. The possible objective function curves are lines with a slope of -1. The minimum will be the point where the lowest line still touches the



circle.

- Since the objective function lines are straight parallel lines, the gradient of f is a straight line pointing toward the direction of increasing f, which is to the upper right
- The gradient of h will be pointing out from the circle and so its direction will depend on the point at which the gradient is evaluated.



- At the optimum point, □f(**x**) is perpendicular to □h(**x**)
- As we can see at point $\mathbf{x}^1$, □f(**x**) is not perpendicular to □h(**x**) and we can move (down) to improve the objective function
- We can say that at a max or min, □f(**x**) must be perpendicular to □h(**x**)
  - Otherwise, we could improve the objective function by changing position

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO & CHEMICAL ENGINEERING
DEPARTMENT OF CHEMICAL ENGINEERING

UNIT V
APPLICATION OF OPTIMIZATION IN CHEMICAL ENGINEERING

5.0 Pipeline Problem

variables	parameters
V	ρ
Δp	μ
f	L
Re	$\dot{m}$
D	pipe cost
	electricity cost
	#operating days/yr
	pump efficiency

Equality Constraints

$$\frac{\pi}{4} D^2 v \rho = \dot{m}$$

$$Re = Dv\rho/\mu$$

$$\Delta p = 2\rho v^2 \frac{L}{D} f$$

$$f = .046\, Re^{-0.2}$$

min (Coper + Cinv.)

subject to equality constraints

$$\Delta p = 2 f \rho \frac{L}{D} v^2$$

need analytical formula for f

$f = .046\, Re^{-0.2}$ *smooth tubes*

pump power cost $= C_o \dot{m} \frac{\Delta p}{\rho}$

$\dot{m}$ = mass flow rate $= \frac{\pi D^2}{4} v \rho$

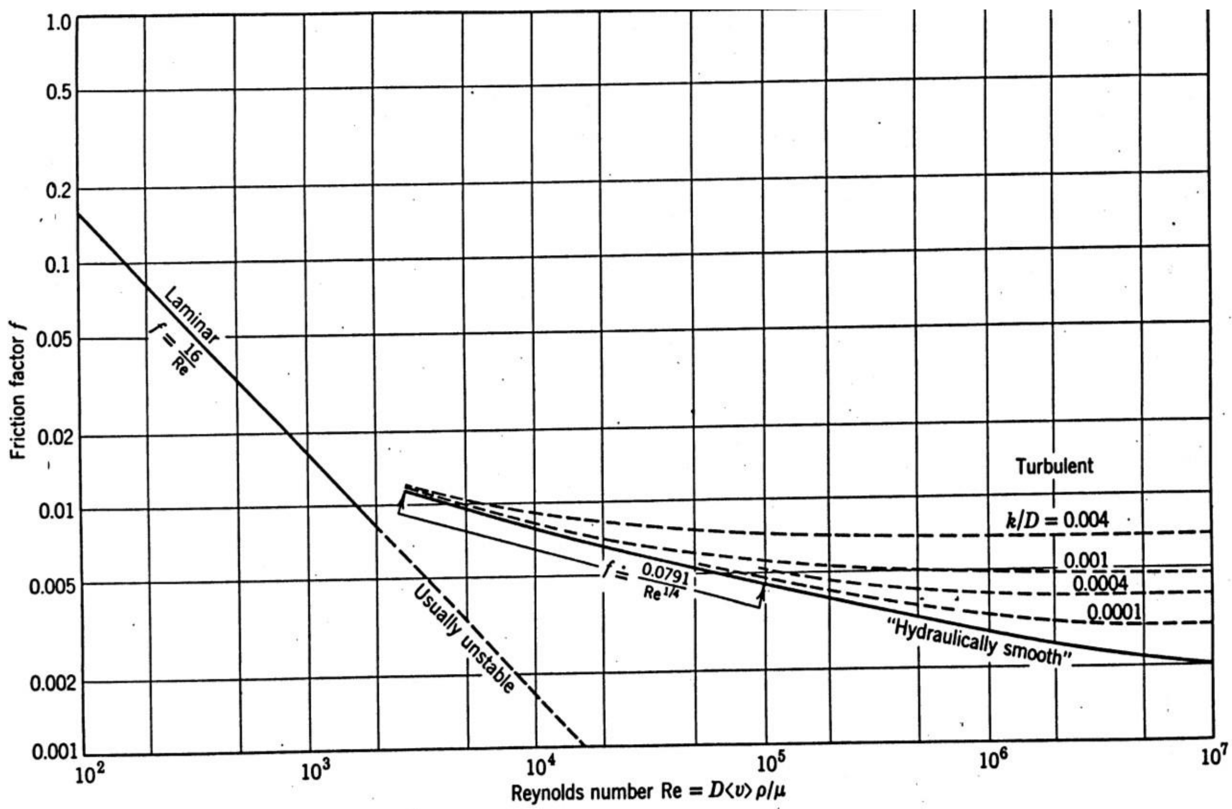


Fig. 6.2-2. Friction factors for tube flow (see definition of *f* in Eqs. 6.1-2 and 6.1-3. [Curves of L. F. Moody, *Trans. ASME*, **66**, 671 (1944) as presented in W. L. McCabe and J. C. Smith, *Unit Operations of Chemical Engineering*, McGraw-Hill, New York (1954).]

$$C_{oper} = C_o D^{-4.8} \mu^{0.2} \dot{m}^{2.8} \rho^{-2.0}$$

$$C_{inv} = C_1 D^{1.5} \quad (annualized)$$

$$\text{Total cost} = TC = C_o \mu^{0.2} \rho^{-2} \dot{m}^{2.8} D^{-4.8} + C_1 D^{1.5}$$

$$\frac{d(TC)}{dD} = 0 \qquad \text{necessary condition for a minimum}$$

solving,

$$(D_{opt})^{6.3} = \frac{C_o}{C_1} \mu^{0.2} \rho^{-2} \dot{m}^{2.8}$$

$$(D_{opt}) = \left(\frac{C_o}{C_1}\right)^{0.16} \rho^{-.32} \dot{m}^{.45} \mu^{.03}$$

$$\text{opt velocity } V_{opt} = \frac{\dot{m}}{\rho \frac{\pi}{4} D_{opt}^2}$$

(sensitivity analysis)

Heat Exchanger Variables

1. heat transfer area
2. heat duty
3. flow rates (shell, tube)
4. no. passes (shell, tube)
5. baffle spacing
6. length
7. diam. of shell, tubes
8. approach temperature
9. fluid A (shell or tube, co-current or countercurrent)
10. tube pitch, no. tubes
11. velocity (shell, tube)
12. Δp (shell, tube)
13. heat transfer coeffs (shell, tube)
14. exchanger type (fins?)
15. material of construction

The heat transfer rate under fouled conditions, Q_f, can be expressed as:

$$Q_f = U_f A_f \Delta T_{mf} \quad (1)$$

where the subscript f refers to the fouled conditions. Mean temperature difference by:

$$\Delta T_m = \frac{\Delta T_1 - \Delta T_2}{\ln \frac{\Delta T_1}{\Delta T_2}} \quad (2)$$

4. Cost of Fouling

Fouling of heat-transfer equipment reduces the thermal efficiency of the equipment. Economic aspects of fouling have design and operational stage. Cost at design stage is increased capital expenditure. The heat transfer area of a heat exchanger is increased to compensate for fouling at design stage. Costs at operational stage are operation and maintenance and loss of production and cleaning and utilization of energy use of antifoulants.

5.3 Cleaning cost

If C'_C is the hourly cleaning cost during the shut down period, then the total cleaning cost per cycle can be expressed as [4].

$$C_C = C'_C . t_d \tag{13}$$

5.4 Miscellaneous costs

Other costs related indirectly to fouling for each cycle are included here as C_M. These include the cleaning program, the shutdown and start up of the process unit, the anti-foulant injection system maintenance, etc.

5.5 The Total Fouling Cost

Operating periyod of heat exchanger,t_c, is written as: [4].

$$t_c = t_p + t_d \tag{14}$$

The total fouling cost (in $) through an operation cycle can be written as [4].

$$C_T(t) = C_H(t) + C_{AF}(t) + C_C + C_M \tag{15}$$

Making appropriate substitutions and calculating for hours costs, we can express the total cost per unit cycle time as:[4].

$$C_T(t_p) = \frac{C_{per}(t_p).H}{(t_p + t_d)} \tag{16}$$

Design Integrated Distillation Columns for Binary Systems

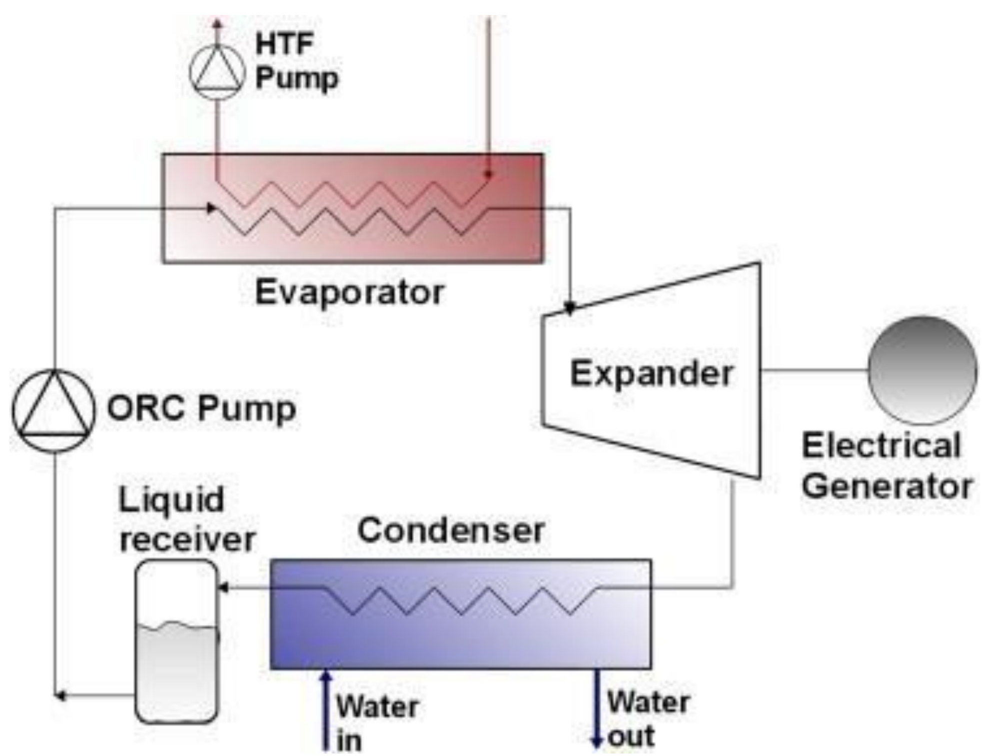
Classical methods: Methods or deterministic local search are studied, specifically the optimization technique based on gradient search. Heuristics and Stochastic methods: Genetic Algorithms, being itself a global optimization method, the probabilistic method is used based on population. The primary objective pursued by the design of the process under study is to determine the optimal physical dimensions (diameter, number of trays andfeed location) of binary distillation column for the system described and intonation parameters associated controller control compositions and products bottom column distillate to be carried out effectively the separation of the components of the binary mixture. The proposed design must be optimal gain both economic criteria and the controllability of the process ensuring a feasible operation. Being integrated, design optimization problem, the objectives are translated into: y Minimize the total cost, which is the aim of designing processes and simultaneously;

y Minimize the ISE (integral square error), which is the goal of dynamic controllability.

To resolve this multiobjective problem becomes one of the objectives restriction (mo) ≤ ε j obj , in this case the integral square error becomes part of the nonlinear constraints of the problem, which involves solving the problem MINLP-DAE (mixed integer no linear programming-differencial algebraic equations) to different values ranging from optimal controllable to the economic optimum, at this point the multiobjective problem becomes a problem of nonlinear programming with algebraic differential equations. According to Schweiger, C., the following terms and restrictions to the process established first objective function of total cost (cost, expressed in $) is related to design costs are assumed and utility through the following mathematical expression, where the cost or capital cost design (Cc, expressed in $) is related to the dimensions of the column and the cost of utility (Cu, expressed in $) with flows steam and reflux. Expression for the total cost:

$$\mathbf{Cost=7756.V_{SS}+3.075.\ (615 + 324DC^{2} + 486\ (6+0.76N_{t})\ DC) + 61.25\ N_{t}(0.7+1.5\ DC^{2})}$$

Optimization of waste heat recovery



Waste heat recovery

This section describes the modeling of each component of the waste heat recovery ORC. All the models are implemented under the EES environment.

Heat exchangers model

The plate heat exchangers are modeled by means of the Logarithmic Mean Temperature Difference (LMTD) method for counter-flow heat exchangers. They are subdivided into 3 moving-boundaries zones, each of them being characterized by a heat transfer area *A*and a heat transfer coefficient *U*.

The heat transfer coefficient *U* is calculated by considering two convective heat transfer resistances in series (secondary fluid and refrigerant sides).

$$\frac{1}{U} = \frac{1}{h_r} + \frac{1}{h_{sf}}$$

The total heat transfer area of the heat exchanger is given by:

$$A_{tot} = A_l + A_{tp} + A_v = (N_p - 2) \cdot L \cdot W$$

N_p being the number of plates, L the plate length and W the plate width.

Single-phase

Forced convection heat transfer coefficients are evaluated by means of the non-dimensional relationship:

$$Nu = C Re^m Pr^n$$

where the influence of temperature-dependent viscosity is neglected.

The parameters *C*, *m* and *n* are set according to Thonon's correlation for corrugated plate heat exchangers. The pressure drops are computed with the following relation:

$$\Delta p = \frac{2 \cdot f \cdot G^2}{\rho \cdot D_h} \cdot L$$

Where *f* is the friction factor, calculated with the Thonon correlation, *G* is the mass velocity (kg/s m^2),ρ is the mean fluid density,D_h is the hydraulic diameter and L is the plate length.

Boiling heat transfer coefficient

The overall boiling heat transfer coefficient is estimated by the Hsieh correlation, established for the boiling of refrigerant R410a in a vertical plate heat exchanger. This heat exchange coefficient is considered as constant during the whole evaporation process and is calculated by:

$$h_{tp} = C \cdot h_l \cdot Bo^{0.5}$$

Where Bo is the boiling number and h_l is the all-liquid non-boiling heat transfer coefficient.

The pressure drops are calculated in the same manner, using the Hsieh correlation for the calculation of the friction factor.

Condensation heat transfer coefficient

The condensation heat transfer coefficient is estimated by the Kuo correlation, established in the case of a vertical plate heat exchanger fed with R410A. It is given by:

$$h_{tp} = C \cdot h_l \cdot \left(0.25 \cdot Co^{-0.45} \cdot Fr_l^{0.25} + 75 \cdot Bo^{0.75}\right)$$

Where Fr_l is the Froude Number in saturated liquid state, Bo the boiling number and Co the convection number.

The pressure drops are calculated in the same manner, using the Kuo correlation for the calculation of the friction factor.

Heat exchanger sizing

For a given corrugation pattern (amplitude, chevron angle, and enlargement factor), two degrees of freedom are available when sizing a plate heat exchanger: the length and the total flow width. The total flow width is given by the plate width multiplied by the number of channels:

$$W_{tot} = W_{hx} \cdot \frac{N_p - 1}{2}$$

The two degrees of freedom are fixed by the heat exchange area requirement and the limitation on the pressure drop on the working fluid side:

Increasing the total width decreases the Reynolds number. This leads to a lower pressure drop and to a higher required heat transfer area, since the heat transfer coefficient is also decreased. Increasing the plate length leads to a higher pressure drop.

Optimum design of Ammonia reactor

This example based on the reactor described by Murase et al. (1970) shows one way to mesh the numerical solution of the differential equations in the process model with an optimization code. The reactor, illustrated in Figure E14.2a, is based on the Haber process.

$$N_2 + 3H_2 \rightleftharpoons 2NH_3$$

Figure E14.2b illustrates the suboptimal concentration and temperature profiles experienced. The temperature at which the reaction rate is a maximum decreases as the conversion increases.

Assumptions made in developing the model are

1. The rate expression is valid.
2. Longitudinal heat and mass transfer can be ignored.
3. The gas temperature in the catalytic zone is also the catalyst particle temperature.
4. The heat capacities of the reacting gas and feed gas are constant.
5. The catalytic activity is uniform along the reactor and equal to unity.
6. The pressure drop across the reactor is negligible compared with the total pressure in the system.

Objective function. The objective function for the reactor optimization is based on the difference between the value of the product gas (heating value and ammonia value) and the value of the feed gas (as a source of heat only) less the amortization of reactor capital costs. Other operating costs are omitted. As shown in Murase et al., the final consolidation of the objective function terms (corrected here) is

$$f(x,N_{N_2},T_f,T_g) = 11.9877 \times 10^6 - 1.710 \times 10^4 N_{N_2} + 704.04 T_g$$
$$-699.3 T_f - [3.4566 \times 10^7 + 2.101 \times 10^9 x]^{1/2} \qquad (a)$$

Equality constraints. Only 1 degree of freedom exists in the problem because there are three constraints; x is designated to be the independent variable.

Energy Balance, Feed Gas

$$\frac{dT_f}{dx} = -\frac{US_1}{WC_{pf}}(T_g - T_f) \qquad (b)$$

Energy Balance, Reacting Gas

$$\frac{dT_g}{dx} = \frac{US_1}{WC_{pg}}(T_g - T_f) + \frac{(-\Delta H)S_2}{WC_{pg}}(f)\left[K_1 \frac{(1.5)p_{N_2}p_{H_2}}{p_{NH_3}} - K_2 \frac{p_{NH_3}}{(1.5)p_{H_2}}\right] \qquad (c)$$

where $K_1 = 1.78954 \times 10^4 \exp(-20{,}800/RT_g)$
$K_2 = 2.5714 \times 10^{16} \exp(-47{,}400/RT_g)$

Mass Balance, N_2

$$\frac{dN_{N_2}}{dx} = -f\left[K_1\frac{(1.5)p_{N_2}p_{H_2}}{p_{NH_3}} - K_2\frac{p_{NH_3}}{(1.5)p_{H_2}}\right] \qquad (d)$$

The boundary conditions are

$$T_f(x = L) = 421°\text{C (694 K)} \qquad (e)$$

$$T_g(x = 0) = 421°\text{C (694 K)} \qquad (f)$$

$$N_{N_2}(x = 0) = 701.2 \text{ kg mol/(h)(m}^2) \qquad (g)$$

For the reaction, in terms of N_{N_2}, the partial pressures are

$$p_{N_2} = 286\left[\frac{N_{N_2}}{1 - 2(N^0_{N_2} - N_{N_2})}\right]$$

$$p_{N_2} = 286\left[\frac{3N_{N_2}}{1 - 2(N^0_{N_2} - N_{N_2})}\right]$$

$$p_{NH_3} = 286\left[\frac{2(N^0_{N_2} - N_{N_2})}{1 - 2(N^0_{N_2} - N_{N_2})}\right]$$

Inequality constraints.

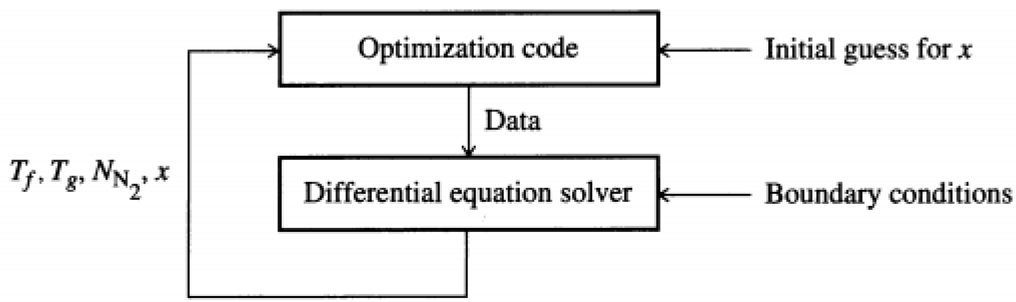
$$0 \le N_{N_2} \le 3220$$

$$400 \le T_f \le 800$$

$$x \ge 0$$

Solution procedure. Because the differential equations must be solved numerically, a two-stage flow of information is needed in the computer program used to solve the problem. Examine Figure E14.2c. The code GRG2 (refer to Chapter 8) was coupled with the differential equation solver LSODE, resulting in the following exit conditions:

	Initial guesses	**Optimal solution**
N_{N_2}	646 kg mol/(m²)(h)	625 kg mol/(m²)(h)
Mole fraction N_2	20.06%	19.4%
T_g	710 K	563 K
T_f	650 K	478 K
x	10.0 m	2.58 m
$f(x)$	8.451×10^5 \$/year	1.288×10^6 \$/year



Optimization of a reactor

The nonlinear ordinary differential equations and boundary conditions in the model can be put in dimensionless form and converted to algebraic equations using orthogonal collocation (Finlayson, 1980). Setalvad and coworkers (1989) used these algebraic equations as constraints in formulating a nonlinear programming problem to study the effects of temperature, flow parameters, reactor geometry, and wafer size on the LPCVD process, particularly the uniformity of silicon deposition. Strategies were devised to determine the potential improvements in the system performance by using optimum temperature staging and reactant injection schemes. It shows the inputs and performance measures for the reactor that can be optimized to maximize the film growth rate (production rate), subject to constraints on radial film uniformity (on each wafer), as well as axial uniformity (wafer-to-wafer). The growth rate is quite sensitive to the axial temperature profile. An axial temperature profile that increases along the reactor because it improves the deposition uniformity is commonly used in industry. The temperature of each successive zone in the furnace (defined by the furnace elements in Figure E14.5a) can be adjusted by voltage applied to variac heaters. The zone temperatures are assumed constant within each zone, T,, j = 1, . . . , n,, where n, is the number of temperature zones to be used.