BUSINESS OPTIMIZATION

A MATHEMATICAL OPTIMIZATION APPROACH

ASHOK AGARWAL

Business Optimization: A Mathematical Optimization Approach

Business Optimization: A Mathematical Optimization Approach

Ashok Agarwal

THE INFO LIBRARY

Published by The InfoLibrary,
4/21B, First Floor, E-Block,
Model Town-II,
New Delhi-110009, India

Business Optimization: A Mathematical Optimization Approach
Ashok Agarwal
ISBN: 978-93-5496-572-2

Table of Contents

Chapter 1	LINEAR PROGRAMMING	**1**
Chapter 2	DUALITY AND NETWORKS	**38**
Chapter 3	INTEGER PROGRAMMING	**60**
Chapter 4	CLASSICAL OPTIMISATION THEORY	**103**
Chapter 5	OBJECT SCHEDULING	**127**

1

LINEAR PROGRAMMING

Principal components of decision problem – Modeling phases – LP Formulation and graphic solution –Resource allocation problems – Simplex method – Sensitivity analysis.

1.1 Principal components of decision problem

Assumptions in Linear Programming

The following four assumptions are made in the linear programming problems.

Linearity: The amount of resource required for a given activity level is directly proportional to the level of that activity. For example, if the number of hours required on a particular machine (for a given activity level) is 5 hours per unit of that activity, then the total number of hours required on that machine to produce 10 units of that activity is 50 hours.

Divisibility: This means that fractional values of the decision variables are permitted.

Non-negativity: This means that the decision variables are permitted to have only the values which are greater than or equal to zero.

Additivity: This means that the total output for a given combination of activity levels is the algebraic sum of the output of each individual process.

Properties of Linear Programming Solution

Feasible solution: If all the constraints of the given linear programming model are satisfied by the solution of the model, then that solution is known as a feasible solution. Several such solutions are possible for a given linear programming model.

Optimal solution: If there is no other superior solution obtained for a given linear programming model, then the solution obtained is treated as the optimal solution.

Alternate optimum solution: For some linear programming model, there may be more than one combination of values of the decision variables yielding the best objective function value. Such combinations of the values of the decision variables are known as alternate optimum solutions.

Unbounded solution: For some linear programming model, the objective function value can be increased/decreased infinitely without any limitation. Such solution is known as unbounded solution.

Infeasible solution: If there is no combination of the values of the decision variables satisfying all the constraints of the linear programming model, then that model is said to have infeasible solution. This means that there is no solution for the given model which can be implemented.

Degenerate solution: In linear programming problems, intersection of two constraints will define a corner point of the feasible region. But if more than two constraints pass through any one of the corner points of the feasible region, excess constraints will not serve any purpose, and therefore they act as redundant constraints. Under such situation,degeneracy will occur. This means that some iterations will be carried out in simplex method without any improvement in the objective function.

1.2 Modeling phases

CONCEPT OF LINEAR PROGRAMMING MODEL

A model, which is used for optimum allocation of scarce or limited resources to competing products or activities under such assumptions as certainty, linearity, fixed technology, and constant profit per unit, is linear programming.

Linear programming is a most versatile, powerful and useful techniques for making managerial decisions. Linear programming technique may be used for solving broad range of problems arising in business, government, hospitals, Industry Libraries and etc . Whenever we want to allocate the available limited resources for various competing activities for achieving our desired objective, it has technique that helps us is LINEAR PROGRAMMING. As a decision making tool, it has demonstrated its value in various fields such as production, finance, marketing, research and demonstrated its value in various fields such as production, finance, marketing, research and development and personnel management. Determination of optimal product mix, transportation schedules, assignment problem and many more. In this chapter let us discuss about various types of linear programming models.

Properties of Linear programming model

Any linear programming model must have the following properties:

- The model must have an objective function.

- The relationship between variables and constraints must be linear

- The model must have structural constraints.

- The model must have non-negativity constraint.

The model of any linear programming problem will contain: objective function, set of constraints and non-negativity restrictions. Each of the components may consists of one or more of the following:

• Decision variables

• Objective function coefficients

• Technological coefficients

• Availability of resources.

The components and other terminologies of the linear programming model are explained with the help of a product-mix problem as described here.

Modelling is an art. One can develop this expertise only by seeing more and more models.

Problems:

A company manufactures two types of products, P_1 and P_2.Each product uses lathe and milling machine. The processing time per unit of P_1 on the lathe is 5 hours and on the milling machine is 4 hours. The processing time per unit of P_2 on the lathe is 10 hours and on the milling machine, 4 hours. The maximum number of hours available per week on the lathe and the milling machine are 60 hours and 40 hours, respectively. Also the profit per unit of selling P_1, and P_2 are Rs. 6.00 and Rs. 8.00, respectively. Let us formulate a linear programming model to determine the production volume of each of the products such that the total profit is maximized.

Solution:

Given:

P_1, P_2.

Profit per unit:Rs. 6.00, Rs. 8.00.

Processing time per unit:

5 hours and on the milling machine is 4 hours.

10 hours and on the milling machine, 4 hours.

The data of the problem are summarized in Table 1.

Table 1 Details of Products(in hour)

Machine	**Machine hours/unit**		**Limit on machine hours**
	Product P_1	**Product P_2**	
Lathe	5	10	60
Milling machine	4	4	40
Profit/unit (Rs.)	6	8	

Let X_1 and X_2 be the production volumes of the products P_1 and P_2respectively. The corresponding linear programming model to determine the production volume of each of the products, such that the total profit is maximized, is presented below.

Maximize $Z = 6X_1 + 8X_2$

subject to

$5X_1 + 10X_2 \leq 60$

$4X_1, + 4X_2 \leq 40$

X_1 and $X_2 \geq 0$

This is Cargo loading problem. Let us consider the cargo loading problem, where five items are to be loaded on a vessel. The weight (w_i,) and volume (v_i) of each unit of the different items as well as their corresponding returns per unit (r_i) are tabulated in Table.

Table :

Item-*i*	w_i	v_i	r_i
1	**5**	**1**	***4***
2	**8**	**8**	***1***
3	**3**	**6**	**6**
4	**2**	**5**	**5**
5	**7**	**4**	**4**

The maximum cargo weight (W) and volume (V) are given as 112 and 109 respectively. It is required to determine the optimal cargo load in discrete units of each item such that the total return is maximized. Formulate the problem as an integer programming model.

Solution:

Let, X_i be the number of units of the i^{th} item to be loaded in the cargo, where i varies from 1 to 5. A model to maximize the return is as follows:

Maximize $Z = 4X_1, + 7X_2 + 6X_3 + 5X_4 + 4X_5$

subject to

$5X_1 + 8X_2 + \mathbf{3}X_3 + 2X_4 + 7X_5 \leq 112$

$X_1 + 8X_2 + 6X_3 + 5X_4 + 4X_5 \leq 112$

$X_1 + X_2 + X_3 + X_4$ and $X_5 \geq 0$ and integers

A company wants to engage casual labours to assemble its product daily. The company works for only one shift which consists of 8hours and 6 days a week. The casual labours consist of two categories, viz. skilled and semi-skilled. The daily production per skilled labour is 80 assemblies and that of the semi-skilled labour is 60 assemblies. The rejection rate of the assemblies produced by the skilled labours is 5% and that of the semi-skilled labours is 10%. The loss to the company for rejecting an assembly is Rs. 25. The daily wage per labour of the skilled and semi-skilled labours are Rs. 240 and Rs. 160 respectively. The required weekly production is 1,86,000

assemblies. The company wants to limit the number of semi-skilled labours per day to utmost 400. Let us Develop a linear programming model to determine the optimal mix of the casual labours to be employed so that the total cost (total wage + total cost of rejections) is minimized.

Solution:

Daily wage per skilled labour = Rs. 240

Daily wage per semi-skilled labour = Rs. 160

Weekly required production = 1,86,000 assemblies

Number of working days per week = 6 days

Therefore, daily required production = 31,000 assemblies

Number of assemblies produced per skilled labour in a day is 80. Rejection rate of assemblies produced by skilled labours is 5% and hence, his number of rejected assemblies in a day is 4. Therefore, the acceptable number of assemblies produced per skilled labour in a day is 76.

Number of assemblies produced per semi-skilled labour in a day is 60. Rejection rate of assemblies produced by a semi-skilled labour is 10% and hence, his number of rejected assemblies in a day is 6. Therefore, the acceptable number of assemblies produced per semi-skilled labour in a day is 54.

The loss per rejected assembly= Rs. 25

The loss due to rejections per skilled labour in a day= 4 × Rs. 25 = Rs. 100

The loss due to rejections per semi-skilled labour in a day= 6 × Rs. 25 = Rs. 150

Let, X_1 be the number of skilled labours to be employed per day. X_2 be the number of semiskilled labours to be employed per day.

A linear programming model to determine the number of labours to be employed per day under each category of casual labours to minimize the sum of the total wages and penalty of rejections in a day is presented below.

Minimize Z = $240X_1, + 160X_2 + (100X_1, + 150X_2)$

$340X_1, + 310X_2$

subject to

$76X_1 + 54X_2 \geq 31000$

$X_2, \leq 400$

X_1, and $X_2 \geq 0$.

1.3 LP Formulation and graphic solution

Methods for the solution of a linear programming problem

This is a method of solving the Various types of problems in which two or more candidates or activities are competing to utilize the available limited resources, with a view to optimize the objective function of the problem. The objective may be maximize the returns or to minimize the costs.

The various methods are:

- The Systematic trial and error method, where we go on giving various values to variables untile we get optimal solution. This method takes too much of time and laborious, hence this method is not discussed here.

- The Graphical method when we have two decision variables in the problem. To deal with more decision variables by graphical method will become complicated, because we have to deal with planes instead of straight lines. Hence in graphical method let us limit ourselves to two variable problems.

- The vector method. In this method each decision variable is considered as a vector & principles of vector algebra is used to get the optimal solution. This method is also time consuming.

- This simplex method. When the problem is having more then two decision variables, simplex method is the most powerful method to solve the problem.

One problem with two variable is solved by using both graphical & simplex method, so as to enable the reader to understand the relationship between the two.

Graphical Method

In graphical method, the inequalities are considered to be equations. This is because; one cannot draw straight lines in two-dimensional plane. Moreover as we have non-negativity constraint in the problem that is all the decision variables must have positive values always the solution to the problem lies in first quadrant of the graph. Sometimes the value of variables may fall in quadrants other then the first quadrant. In such cases, the line joining the values of the variables must be extended into the first quadrant. The procedure of the method will be explained in details while solving a numerical problem.

The characteries of graphical method are,

- Generally the method is used to solve the problem, when it involves two decision variables.

- For three or more decision variable, the graph deals with planes and requires high imagination to identify the solution area.

- This method provides a basis for understanding the other method of solution

LP Model Formulation

Decision variables : Mathematical symbols represents the levels of activity of an operation.

Objective function :

• A linear relationship reflecting the objective of an operation.

•Most frequent objective of business firms is to maximize profit.

•Most frequent objective of individual operational units (such as a production or packaging department) is to minimize cost.

Constraint :

•A linear relationship representing a restriction on decision making.

Max/min $Z=c_1x_1+c_2x_2+....+c_n x_n$

Subject to:

$a_{11}x_1+a_{12}x_2+.....+a_{1n}x_n(\leq,=,\geq)b_1$

$a_{21}+x_1+a_{22}x_2+.....+a_{2n}x_n(\leq,=,\geq)b_2$

$A_{m1}x_1+a_{m2}x_2+....+a_{mn}x_n(\leq,=,\geq)b_m$

X_j = Decision variables

b_i = Constraint levels

C_j = Objective function coefficients

A_{ij} = Constraint coefficients

LP Terminology :

Solution (decision, point):

Any specification of values for all decision variables, regardless of whether it is a desirable or even allowable choice.

Feasible solution:

Its a solution for which all the constraints are satisfied.

Feasible region (constraint set, feasible set): The collection of all feasible solution.

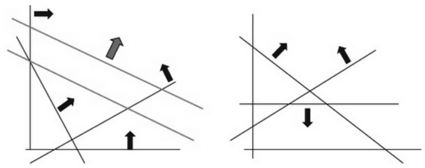
Optimal solution (optimum): A feasible solution that has the most favorable value of the objective function.

Optimal (objective) value: The value of the objective function evaluated at an optimal solution.

Unbounded or Infeasible Case

On the left, the objective function is unbounded.

On the right, the feasible set is empty.



LP Formulation Example:

RESOURCE REQUIREMENTS			
PRODUCT	Labor (hr/unit)	Clay (lb/unit)	Revenue($/unit)
Bowl	1	4	40
Mug	2	3	50

There are 40 hours of labor and 120 pounds of clay available each day

Decision variables

X_1 = Number of bowls to produce

x_2 = Number of mugs to produce

Maximize $Z=\$40\, x_1 + 50\, x_2$

Subject to

$x_1 + 2x_2 < 40$ hr (labor constraint)

$4x_1+3x_2 < 120$ lb (clay constraint)

$X_1, x_2 \geq 0$

Solution is $x_1 = 24$ bowls $x_2 = 8$ mugs

Revenue = $1,360

Graphical Solution Method:

1.Plot model constraint on a set of coordinates in a plane.

2.Identify the feasible solution space on the graph where all constraints are satisfied simultaneously.

3.Plot objective function to find the point on boundary of this space that maximizes (or minimizes) value of objective function.

Graphical Method

As stated earlier, if the number of variables in any linear programming problem is only two, one can use graphical method to solve it. In this section, the graphical method is demonstrated with three example problems.

Problems:

Solve the following LP problem using graphical method.

Maximize Z = 6X_1, + 8X_2

subject to

5X_1 + 10X_2 ≤ 60

4X_1 + 4X_2 ≤ 40

X_1 and X_2 ≥ 0

Solution:

In graphical method, the introduction of the non-negative constraints (X_1≥ 0 and X_2 ≥ 0) will eliminate the second, third and fourth quadrants of the X_1,X_2 plane, as shown in Figure.

Now, we compute the coordinates on the X_1,X_2 plane. From the first constraint

5X_1, + 10X_2 = 60

we get X_2 = 6, when X_1, = 0; and X_1, = 12, when X_2 = 0. Now, plot the first constraint as shown in Figure .

From the second constraint

4X_1, + 4X_2 = 40

we get X_2 = 10, when X_1 = 0; and X_1, = 10, when X_2 = 0. Now, plot the second constraint as shown in Figure.

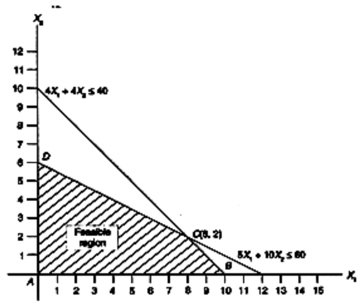
The closed polygon A-B-C-D is the feasible region. The objective function value at each of the corner points of the closed polygon is computed by substituting its coordinates in the objective function as:

Z(A) = 6 × 0 × 8 × 0 = 0

Z(B) = 6 × 10 × 8 × 0 = 60

Z(C) = 6 × 8 + 8 × 2 = 48 × 16 = 64

Z(D) = 6 × 0 + 8 × 6 = 48



Feasible region of Example.

Since the type of the objective function here is maximization, the solution corresponding to the maximum Z value is to be selected as the optimum solution. The Z value is maximum for the corner point C. Hence, the corresponding solution is presented below.

X_1*=8,

X_2*=2 Z(optimum) = 64

Let us solve the following LP problem using graphical method:

Minimize Z = $2X_l + 3X_2$

subject to

$X_1 + X_2 \ge 6$

$7X_1 + X_2 \ge 14$

X_1 and $X_2 \ge 0$

Solution:

The introduction of the non-negative constraints ($X_1 \ge 0$ and $X_2 \ge 0$) will eliminate the second, third and fourth quadrants of the X_1, X_2 plane as shown in Figure.

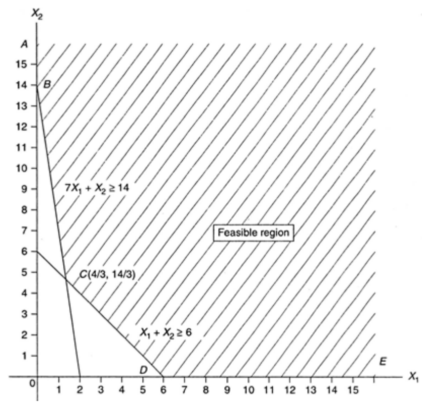
Now, we compute the coordinates to plot on the X_1X_2 plane relating to different constraints. From the first constraint

$X_1 + X_2 = 6$

we get $X_2 = 6$, when $X_1 = 0$; and $X_1 = 6$, when $X_2 = 0$. Now, plot the constraint 1 as shown in Figure.

Second constraint is given as

$7X_1 + X_2 = 14$



Feasible region of Example.

we get X_2 = 14, when X_1 = 0; and X_1 = 2, when X_2 = 0. Now, plot the constraint 2 as shown in Figure.

In Figure, A-B-C-D-E is the feasible region. The optimum solution will be in any one of the comer points, B, C and D. The objective function value at each of these corner points is computed as follows by substituting its coordinates in the objective function.

$Z(B) = 2 \times 0 \div 3 \times 14 = 42$

$$Z(C) = 2 \times \frac{4}{3} + 3 \times \frac{14}{3} = \frac{50}{3} = 16.67$$

$$Z(D) = 2 \times 6 + 3 \times 0 = 12$$

Since the type of the objective function is minimization, the solution corresponding to the minimum Z value is to be selected as the optimum solution. The Z value is minimum for the comer point D. Hence, the corresponding optimum solution is:

$X_1^* = 6$,

$X_2^* = 0$, Z(optimum) = 12

1.4 Resource allocation problems

This type of problems are often identified with the application of linear program. It is the problem of distributing scarce resources among alternative activities. Here the Product Mix problem is a special case.

In this example, let us consider a manufacturing facility, that produces five different products using four machines. The scarce resources are the times available on the machines and the alternative activities are the individual production volumes.

The machine requirements in one hour per unit are shown for every product in the table. With the exception of product 4 that does not require machine 1, each product must pass through all four machines. The unit profits are also shown in the table.

The facility has 4 machines of type 1, 5 of type 2, 3 of type 3 and 7 of type 4. Each machine operates 40 hours per week. The problem is to determine the optimum weekly production quantities for the products.

The objective is to increase the total profit. In constructing a model, the initial step is to define the decision variables, the next is to write the constraints and objective function in terms of these variables of the problem data.

In the problem of sentence phrases like "at least," "no greater than," "equal to," and "less than or equal to" imply one or more constraints.

Machine data and processing requirements (hrs/unit)

Machine	**Quantity**	**Product 1**	**Product 2**	**Product 3**	**Product 4**	**Product 5**
M1	4	1.2	1.3	0.7	0.0	0.5
M2	5	0.7	2.2	1.6	0.5	1.0
M3	3	0.9	0.7	1.3	1.0	0.8
M4	7	1.4	2.8	0.5	1.2	0.6
Unit profit, $	--	1.8	25	10	12	15

Variable Definitions

P_j : Quantity of product j produced, j = 1,...,5

Machine Availability Constraints

The number of hours available on each machine type is 40 times the number of machines. All the constraints are dimensioned in hours. For machine 1, for example, we have 40 hrs/machine ∞ 4 machines = 160 hrs.

M1	$1.2P_1+1.3P_2+0.7P_3+0.0P_4+0.5P_5<160$
M2	$0.7P_1+2.2P_2+1.6P_3+0.5P_4+1.0P_5<200$
M3	$0.9P_1+0.7P_2+1.3P_3+1.0P_4+0.8P_5<120$
M4	$1.4P_1+2.8P_2+0.5P_3+1.2P_4+0.6P_5<280$

Non negativity

$P_j > 0$ for j = 1,...,5

Objective Function

The unit profit coefficients are given in below the table. We can assuming proportionality, the profit maximization criterion can be written as:

Maximize $Z = 18P_1 + 25P_2 + 10P_3 + 12P_4 + 15P_5$

Solution:

The model constructed with the Math Programming add-in is shown below. The model has been solved with the Jensen LP add-in. The following observations were made.

The solution is not an integer. Although the experimental considerations may demand that only integer quantities of the products be manufactured, the solution to a linear programming model is not, in general, integer. To get an optimum integer solution, one should specify in the model that the variables are to be integer.

The output model is called an integer programming model and is highly complicated to solve for larger models. The analyst should report the optimal solution as shown and then if necessary, round the solution to integer values.

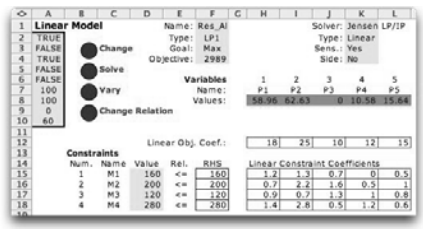
For this problem, rounding down the solution to: $P_1 = 59$, $P_2 = 62$, $P_3 = 0$, $P_4 = 10$ and $P_5 = 15$ will result in a feasible solution, but the solution may not be optimal.

The solution is basic one. The simplex solution procedure used by the Jensen LP add-in will always return a basic solution. It will have as more basic variables as there are constraints. As described elsewhere in this site, basic variables are allowed to assume the values that are not at their upper or lower bounds.

There are four constraints in this problem, and four basic variables, P_1, P_2, P_4 and P_5. Variable P_3 and the slack variables for the constraints are the nonbasic variables.

All the machine resources are bottlenecks for the optimum solution with the hours used exactly equal to the hours available. This is implied by the fact that the slack variables for the constraints are all zero.

This model does not have lower or upper bounds specified for the variables. This is an option allowed with the Math Programming add-in. When not specified, lower bounds on variables are zero and upper bounds are unlimited.



Sensitivity analysis

The sensitivity analysis amplifies the solution. The analysis shows the results of changing one parameter at a time. While a single parameter is changing, all other problem parameters are held constant. For changes in the limits of tight constraints, the values of the basic variables must also change so that the equations defining the solution remains satisfied.

Variable Analysis

The "reduced cost" column indicates the increase in the objective function per unit change in the value of the associated variable. The reduced costs for the basic variables are all zero because the values of these variables are uniquely determined by the problem parameters and cannot be changed.

The decreased cost of P_3 notifies that if this variable were increased from 0 to 1 the objective value (or profit) will decrease by $13.53. It is not good that the decreased cost is negative since the optimum value of P_3 is 0.

When a nonbasic variable changes, the basic variables change so that the equations defining the solution remain satisfied. There is no information from the sensitivity analysis on how the basic variables change or P_3 can change before the current basis becomes infeasible.

Note that the reduced costs are really derivatives that indicate the rate of change. For degenerate solutions in the amount a nonbasic variable may change before a basis change is required may actually be is zero.

The ranges at the right of the display indicate how far the associated objective coefficient may change before the resent solution values P_1 through P_5 must change to maintain optimality. For example, the unit profit on P_1 may assume any value between 13.26 and 24.81. lower limit of P_3 indicates an lower bound. Since P_3 is zero at the optimum, reducing its unit profit by any amount will make it even less appropriate to produce this product.

Constraint Analysis

The shadow price indicates the increase in the objective value per unit increase of the associated constraint limit. The status of all the constraints are "Upper" and indicating that the upper limits are tight. From given the table An increase in the hour limit of 120 for M3 increases the objective value by the more then ($8.96), while increasing the limit for M4 increases the objective value by the least ($0.36). Then again, these quantities are rates of change. When the solution is degenerate, no change may actually be possible.

The ranges at the right of the display indicate how far the limiting value may change while keeping the same optimum basis. The shadow prices remain valid within this range. As an example consider M1. For the solution, there are 160 hours of capacity for this machine.

A capacity may range from 99.35 hours to 173 hours while keeping the same basis optimal. The Changes above 120 cause an increase in profit of $4.82 per unit, while changes below 120 cause a reduction in profit by $4.82 per unit. As the value of one parameter changes, the other parameters remain constant and the basic variables change to keep the equations defining the solution satisfied.

Sensitivity Analysis for Worksheet Res_AI

Variable Analysis — **Objective Value: 2,988.73**

Num.	Name	Value	Status	Reduced Cost	Objective Coefficient	Range Lower Limit	Range Upper Limit
1	P1	58.9614	Basic	0.	18.	13.2552	24.8066
2	P2	62.6346	Basic	0.	25.	23.8437	41.8158
3	P3	0.	Lower	-13.5303	10.	---	23.5303
4	P4	10.5763	Basic	0.	12.	11.29	17.9222
5	P5	15.6428	Basic	0.	15.	9.9091	15.3631

Constraint Analysis

Num.	Name	Value	Status	Shadow Price	Constraint Limit	Range Lower Limit	Range Upper Limit
1	M1	160.	Upper	4.8195	160.	99.3485	172.9961
2	M2	200.	Upper	5.2016	200.	184.6901	230.3636
3	M3	120.	Upper	8.9635	120.	101.1299	237.505
4	M4	280.	Upper	0.3631	280.	259.3189	295.6412

Sensitivity analysis worksheet solution

General Resource Allocation Model

It is common to describe a problem class with a general algebraic model where numeric values are represented by lower case letters usually drawn from the early part of the alphabet. Variables are given alphabetical representations commonly drawn from the later in the alphabet. Terms are combined with summation signs.

The general resource allocation model is given below. When the parameters are given specific numerical values the result is an instance of the general model.

Parameters

n: Number of activities. Activities are indexed by j=1...n.

n: Number of resources. Resources are indexed by j=1...n.

P_j: Profit for activity j.

b_i: Amount available for resource i.

a_{ij}: Amount of resource i used by a unit of activity j.

Variables

x_j: amount of activity i is selected.

Model

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

Subject to:

$\sum_{j=1}^{n} a_{ij} x_j \le b_i$ for i=1.....m

$x_j \geq 0$ for j=1...n

1.5 Simplex method

Simplex method is the basic building block for all other methods. This method is devised based on the concept of solving simultaneous equations. It is demonstrated using a suitable numerical problem.

Let us consider the linear programming model of Example (as reproduced below) and solve it using the simplex method.

Maximize $Z = 6X_1 + 8X_2$

subject to

$5X_1 + 10X_2 \leq 60$

$4X_1 + 4X_2 \leq 40$

X_1 ***and*** $X_2 \leq 0$

Solution:

The standard form of the above LP problem is shown below:

Maximize $Z = 6X_1 + 8X_2 + 0S_1 + 0S_2$

subject to

$5X_1 + 10X_1 + S_1 = 60$

$4X_1 + 4X_2 + S_2 = 40$

X_1, X_2 S_1 and $S_2 \geq 0$

where S_1 and S_2 are slack variables, which are introduced to balance the constraints.

Canonical form is the form in which each constraint has a basic variable.

Definition of basic variable: A variable is said to be a basic variable if it has unit coefficient in one of the constraints and zero coefficient in the remaining constraints. If all the constraints are '≤' type, then the standard form is to be treated as the canonical form. The canonical form is generally used to prepare the initial simplex table. The initial simplex table of the above problem is shown in Table1.

Table 1: Initial Simplex Table (Example)

	Cj	**6**	**8**	**0**	**0**		
CB_i	**Basic variable**	**X_1**	**X_2**	**S_1**	**S_2**	**Solution**	**Ratio**
0	S_1	5	10	1	0	60	60/10 = 6**
0	S_2	4	4	0	1	40	40/4 = 10
	Zj	0	0	0	0	0	
	Cj-Zj	6	8 *	0	0		

*Key column. **Key row.

Here, C_j is the coefficient of the y^{th} term of the objective function and CB_i is the coefficient of the *i*th basic variable. The value at the intersection of the key row and the key column is called the key element. The value of $Z_{j;}$ is computed using the following formula.

$$Zj = \sum_{i=1}^{2}(CB_i)(a_{ij})$$

where a_{ij} is the technological coefficient for the j^{th} row and y^{th} column of the table. C_j - Z_j is the relative contribution. In this term, C_j is the objective function coefficient for the j^{th} variable. The value of Z_j against the solution column is the value of the objective function and in this iteration, it is zero.

Optimality condition: For maximization problem, if all C_j - Z_j are less than or equal to zero, then optimality is reached; otherwise select the variable with the maximum C_j - Z_j value as the entering variable. (For minimization problem, if all C_j - Z_j are greater than or equal to zero, the optimality is reached; otherwise select the variable with the most negative value as the entering variable.)

In Table, all the values for C_j - Z_j are either equal to or greater than zero. Hence, the solution can be improved further. C_j - Z_j is the maximum for the variable X_2. So, X_2 enters the basis. This is known as entering variable, and the corresponding column is called key column.

Feasibility condition: To maintain the feasibility of the solution in each iteration, the following steps need to be followed:

1. In each row, find the ratio between the solution column value and the value in the key column.

2. Then, select the variable from the present set of basic variables with respect to the minimum ratio (break tie randomly). Such variable is the leaving variable and the corresponding row is called the key row. The value at the intersection of the key row and key column is called key element or pivot element.

In Table1, the leaving variable is S_1 and the row 1 is the key row. Key element is 10. The next iteration is shown in Table 2. In this table, the basic variable S_1 of the previous table is replaced by X_2. The formula to compute the new values of Table 2 is as shown below:

Table 2: Iteration 1

	c_j	**6**	**8**	**0**	**0**		
CB_i	**Basic variable**	**X_1**	**X_2**	**S_1**	**S_2**	**Solution**	**Ratio**
8	X_2	1/2	1	1/10	0	6	6/(l/2) = 12
0	S_2	2	0	-2/5	1	16	16/2 = 8**
	Zj	4	8	4/5	0	48	
	C_j - Z_j	2*	0	-4/5	0		

Here

$$\text{New value} = \text{Old value} - \frac{\text{Key column value} \times \text{Key row value}}{\text{Key value}}$$

As a sample calculation, the computation of the new value of row 2 and column X_1 is shown below:

$$\text{New value} = 4 - \frac{4 \times 5}{10} = 4 - \frac{20}{10} = 4 - 2 = 2$$

Computation of the cell values of different tables using this formula is a cumbersome process. So, a different procedure can be used as explained below.

Let the first and second rows in Table 1. be L_1 and L_2, respectively; and the first and second rows in Table 2. be L_3 and L_4, respectively. The coefficient of the first row of Table 2. can now be obtained by using the following formula.

$$L_3 = \frac{L_1}{\text{Pivot element}} = \frac{L_1}{10}$$

This operation makes the value of the cell with respect to the first row and the second column in Table 2 as unity. Since the new basic variable is becoming X_2, the cell value with respect to the second row and the second column in Table 2 should be made equal to 0.

This can be achieved by multiplying/dividing the value of the first row and the second column in Table 2 by a suitable constant and then by adding/subtracting the resultant value to/ from the value of the second row and second column in Table 1 such that the net value is zero.

The necessary formula to achieve this result is shown below.

$L_4 = L_2 - 4L_3$

The entries of the second row in Table 2 are obtained by using the above formula.

The solution in Table 2 is not optimal. The criterion row value for the variable X_1 is the maximum positive value. Hence, the variable X_1 is selected as the entering variable and after computing the ratios, S_2 is selected as the leaving variable. The next iteration is shown in Table 3.

Table 3: Iteration 2

CB_i	**C_j**	**6**	**8**	**0**	**0**	**Solution**
	Basic variable	X_1	X_2	S_1	S_2	
8	X_2	0	1	1/5	-1/4	2
6	X_1	1	0	-1/5	1/2	8
	Z_j	6	8	2/5	1	64

	C_j-Z_j	0	0	-2/5	-1	

In Table 3, all the values for C_j - Z_j are either 0 or negative. Hence, the optimality is reached. The corresponding optimal solution is as follows:

X_1 (production volume of P_1)= 8 units

X_2 (production volume of P_2)= 2 units

and the optimal objective function value, Z (total profit) is Rs. 64.

Let us solve the following LP problem using simplex method.

Maximize $Z = 10X_1, + 15X_2 + 20X_3$

subject to

$2X_1, + 4X_2 + 6X_3 \leq 24$

$3X_1, + 9X_2 + 6X_3 \leq 30$

X_1, X_2 and $X_3 \geq 0$

Solution:

The standard form of this problem is

Maximize Z= 10X1, + 15X_2 + 20X_3

subject to

$2X_1 + 4X_2 + 6X_3 + S_1 = 24$

$3X_1 + 9X_2 + 6X_3 + S_2 = 30$

$X_1, X_2, X_3, + S_1$, and $S_2 \geq 0$

where S_1 and S_2 are slack variables. Here, all the constraints are '≤' type, so the canonical form of the given LP problem is same as the standard form represented below:

Maximize $Z = 10X_1, + 15X_2 + 20X_3 + 0S_1 + 0S_2$

subject to

$2X_1 + 4X_2 + 6X_3 + S_1 = 24$

$3X_1 + 9X_2 + 6X_3 + S_2 = 30$

$X_1, X_2, X_3, + S_1$, and $S_2 \geq 0$

The initial simplex table of the above problem is shown in Table 4.

In Table 4, all the values of Cj - Zj are not less than or equal to zero. Hence, the initial solution is not optimum.

Table 4: Iteration 2

CBi	***Cj***	**10**	**15**	**20**	**0**	**0**	**Solution**	**Ratio**
	Basic variable	X_1	X_2	X_3	S_1	S_2		
0	S_1	2	4	0	1	0	24	4**
0	S_2	3	9	6	0	1	30	5
	Zj	0	0	0	0	0	0	
	C_j-Z_j	10	15	20X	0	0		

The variable X_3 is the entering variable because the column with respect to this variable has the highest C_j - Z_j value. The variable S_1 is the leaving variable since the ratio with respect to this row is the least ratio. Hence, the key column is the column corresponding to the variable X_3 and the key row is row 1. The corresponding key element is 6.

The next iteration is shown in Table 5. In this table, the basic variable S_1 is replaced by X_3. In Table 5, the solution is not optimal. The variable X_1 is selected as the entering variable since the column with respect to this variable has the highest C_j - Z_j value. Row 2 is selected as the key row and the corresponding variable S_2 is treated as the leaving variable, because it has the least ratio. The next iteration is shown in Table 6.

Table 5: Iteration 1

	C_j	**10**	**15**	**20**	**0**	**0**		

CBi	**Basic variable**	X_1	X_2	X_3	S_1	S_2	**Solution**	**Ratio**
20	X_3	1/3	2/3	1	1/6	0	4	12
10	X_1	1	5	0	-1	1	6	6**
	Z_j	20/3	40/3	20	10/3	0	80	
	$C_{-}Z_j$	10/3*	5/3	0	-10/3	0		

Table 6: Iteration 2

	C_j	10	15	20	0	0		
CB_i	Basic variable	X_1	X_2	X_3	S_1	S_2	Solution	
20	X_3	0	-1	1	1/2	-1/3	2	
0	X_1	1	5	0	-1	1	6	
	Z_j	10	30	20	0	10/3	100	
	$C_{-}Z_j$	0	-15	0	0	-10/3		

In Table 6, since all the values of C_j - Z_j are less than or equal to zero, the optimality is reached and the corresponding optimal solution is presented as:

X_1 = 6, X_2 = 0, X_3 = 2 and Z(optimum) = 100.

1.6 Sensitivity analysis

In many situations, the parameters and characteristics of a linear programming model may change over a period of time. Also, the analyst may be interested to know the effect of changing the parameters and characteristics of the model on the optimality. This kind of sensitivity analysis can be carried out in the following ways:

1.Making changes in the right-hand side constants of the constraints

2.Making changes in the objective function coefficients

3.Adding a new constraint

4.Adding a new variable.

These are discussed in the following sections.

1. Changes in the Right-hand Side Constants of Constraints

The right-hand side constant (resource availability) of one or more constraints of a linear programming model may change over a period of time. So, the analyst may be interested in knowing the revised optimum solution based on the optimum table of the original problem after incorporating the new changes in the right-hand side constants. The changes bring in the following results.

(a) Same set of basic variables with modified right-hand side constants in the optimal table.

(b) Different set of basic variables in the optimal table.

The following example is considered to demonstrate the above two cases.

Problems:

Let us Maximize $Z = 6X_1 + 8X_2$

subject to

$5X_1 + 10X_2 \leq 60$

$4X_1 + 4X_2 \leq 40$

$X_1 + X_2 \geq 0$

The optimum solution of this problem is shown in Table1.

Table 1: Optimal Table of Example.

CBi	**C_j**	**6**	**8**	**0**	**0**	
	Basic variable	**X_1**	**X_2**	**S_1**	**S_2**	**Solution**
8	X_2	0	1	1/5	-1/4	2
6	X_1	1	0	-1/5	1/2	8
	Z_j	6	8	2/5	1	64
	C_j-Z_j	0	0	-2/5	-1	

(a)if the right-hand side constants of constraint 1 and constraint 2 are changed from 60 and 40 to 40 and 20, respectively.

(b)if the right-hand side constants of the constraints are changed from 60 and 40 to 20 and 40 respectively.

Solution:

(a) The revised right-hand side constants after incorporating the changes in the constraints are obtained by using the following formula.

$$\begin{bmatrix}\text{Basic variables in}\\ \text{the optimal table}\end{bmatrix} = \begin{bmatrix}\text{Technological coefficient}\\ \text{columns in the optimal}\\ \text{table with respect to the basic}\\ \text{variables in initial table}\end{bmatrix}\begin{bmatrix}\text{New R.H.S.}\\ \text{constants}\end{bmatrix}$$

Applying the formula, we have

$$\begin{bmatrix}X_2\\ X_1\end{bmatrix} = \begin{bmatrix}\frac{1}{5} & -\frac{1}{4}\\ -\frac{1}{5} & \frac{1}{2}\end{bmatrix}\begin{bmatrix}40\\ 20\end{bmatrix} = \begin{bmatrix}3\\ 2\end{bmatrix}$$

Solving them, we get X_1 is 2 and that of X_2 is 3. Since, these values are non-negative, the revised solution is feasible and optimal. The corresponding optimal objective function value is 36.

(b)The revised solution of the basic variables in Table 1 after incorporating the changes in the right-hand side values of the constraints are obtained as shown below.

$$\begin{bmatrix}X_2\\ X_1\end{bmatrix} = \begin{bmatrix}\frac{1}{5} & -\frac{1}{4}\\ -\frac{1}{5} & \frac{1}{2}\end{bmatrix}\begin{bmatrix}20\\ 40\end{bmatrix} = \begin{bmatrix}-6\\ 16\end{bmatrix}$$

which gives the values of X_1 and X_2 as 16 and -6, respectively. Since, the value of the basic variable X_2 is negative, the solution is infeasible. This infeasibility can be removed using the dual simplex method. Based on Table 1, the table for the dual simplex method is shown in Table 2.

Table 2: Table for Dual Simplex Method

CB_i	C_j	**6**	**8**	**0**	**0**	
	Basic variable	X_1	X_2	S_1	S_2	**Solution**
8	X_2	0	1	1/5	-1/4	-6
6	X_1	1	0	-1/5	1/2	16
	Z_j	6	8	2/5	1	48
	C_j-Z_j	0	0	-2/5	-1	

Application of the dual simplex method to Table 2. yields the following optimum result in the next iteration itself:

$X_1 = 4$, $X_2 = 0, 5$, $S_1 = 0$, $S_2 = 24$, Z(optimum) = 24

2. Changes in the Objective Function Coefficients

In reality, the profit coefficients or the cost coefficients of the objective function undergo changes over a period of time. Under such a situation, one can obtain the revised optimum solution from the optimal table of the original problem by following certain steps.

Also, one will be interested to know the range of the coefficient of a variable in the objective function over which the optimality is unaffected. These are illustrated using the following example.

Let us Maximize $Z = 10X_1 + 15X_2 + 20X_3$

subject to

$2X_1 + 4X_2 + 6X_3 \leq 24$

$3X_1 + 9X_2 + 6X_3 \leq 30$

$X1$, X_2 and $X_3 \geq 0$

The optimum table of the above problem is given as in Table 3.

Table 3: Optimal Table for Example.

CB_i	C_j	10	15	20	0	0	Solution
	Basic variable	X_1	X_2	X_3	S_1	S_2	
20	X_3	0	-1	1	1/2	-1/3	2
10	X_1	1	5	0	-1	1	6
	z_j	10	30	20	0	10/3	100
	C_j-Z_j	0	-15	0	0	-10/3	

(a)Find the range of the objective function coefficient C_1 of the variableX_1 such that the optimality is unaffected.

(b)Find the range of the objective function coefficient C_2 of the variableX_2 such that the optimality is unaffected.

(c)Check whether the optimality is affected, if the profit coefficients are changed from (10, 15, 20) to (7, 14, 15). If so, find the revised optimum solution.

Solution:

(a) ***Determination of the range of C_1 of the basic variable X_1.*** After making some changes in the objective function coefficients, if the optimality is not affected, then the present set of basic variables will continue, and in that case the C_j - Z_j values of the basic variables will be equal to 0; but the C_j - Z_j values of the non-basic variables will change. Hence, care should be taken in establishing the range for each of C_j values such that the corresponding C_j - Z_j value of that non-basic variable is limited to at most 0.

Since, the variable X_1 with respect to the coefficient C_1 is a basic variable in the optimal table of the original problem, the C_j - Z_j value will change for the non-basic variables: X_2, S_1, and S_2. The values can be computed in terms of C_1. Then, these expressions can be restricted to at most 0 to maintain optimality. By solving the above inequalities for C_1 its range can be determined.

The expressions of C_2 - Z_2, C_4 - Z_4 and C_5 - Z_5 for the non-basic variables,X_2, S_1 and S_2, respectively are:

$$C_2 - Z_2 = 15 - [20\ C_1]\begin{bmatrix}-1\\5\end{bmatrix} = 35 - 5C_1$$

$$C_4 - Z_4 = 0 - [20\ C_1]\begin{bmatrix}\frac{1}{2}\\-1\end{bmatrix} = -10 + C_1$$

$$C_5 - Z_5 = 0 - [20\ C_1]\begin{bmatrix}-\frac{1}{3}\\1\end{bmatrix} = \frac{20}{3} - C_1$$

The above relative contributions are restricted to at most 0 to maintain the optimality as shown below.

$35 - 5C_1 \leq 0$ or $C_1 \geq 7$

$-10 + C_1 \leq 0$ or $C_1 \leq 10$

$$\frac{20}{3} - C_1 \leq 0 \quad \text{or} \quad C_1 \geq \frac{20}{3}$$

Here the value of C_1 ranges from 7 to 10 (i.e. $7 \leq C_1 \leq 10$). In this interval of C_1 the optimality is unaffected.

(b) ***Determination of the range of C_2 of the non-basic variable X_2.*** Since, C_2 corresponds to one of the non-basic variables X_2, the range of C_2 can be obtained by just restricting the C_2 - Z_2 to at most 0. Therefore,

$$C_2 - Z_2 = C_2 - [20\ \ 10]\begin{bmatrix} -1 \\ 5 \end{bmatrix} = C_2 - 30$$

This relative contribution is restricted to at most 0 to maintain the optimality as shown below.

$C_2 - 30 \leq 0$ or $C_2 \leq 30$

Hence, it is clear that the optimality will remain the same as long as the value of C_2 is less than or equal to 30.

(c) ***Checking the optimality.*** The new values of the objective function coefficients, C_1, C_2 and C_3 of the variables, X_1 X_2 and X_3 are 7, 14 and 15, respectively. The corresponding modified relative contributions of all the variables are computed as follows:

$$C_1 - Z_1 = 7 - [15\ 7]\begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$$

$$C_2 - Z_2 = 14 - [15\ 7]\begin{bmatrix} -1 \\ 5 \end{bmatrix} = -6$$

$$C_3 - Z_3 = 15 - [15\ 7]\begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0$$

$$C_4 - Z_4 = 0 - [15\ 7]\begin{bmatrix} \frac{1}{2} \\ -1 \end{bmatrix} = -\frac{1}{2}$$

$$C_5 - Z_5 = 0 - [15\ 7]\begin{bmatrix} -\frac{1}{3} \\ 1 \end{bmatrix} = -2$$

Since all the C_j - Z_j values are less than or equal to 0, the optimality is unaffected. The solution in the optimal table of the original problem, which is shown in Table 3, is the optimal solution for the problem with the modified objective function coefficients of this section.

3. Adding a New Constraint

Sometimes, a new constraint may be added to an existing linear programming model as per changing realities.

Under such situation, each of the basic variables in the new constraint is substituted with the corresponding expression based on the current optimal table. This will yield a modified version of the new constraint in terms of only the current non-basic variables.

If the new constraint is satisfied by the values of the current basic variables, the constraint is said to be a redundant one. So, the optimality of the original problem will not be affected even after including the new constraint into the existing model.

If the new constraint is not satisfied by the values of the current basic variables, the optimality of the original problem will be affected. So, the modified version of the new constraint is to be augmented to the optimal table of the original problem and iterated till the optimality is reached.

Consider an example to demonstrate the impact of adding a new constraint into an existing model:

Maximize $Z = 6X_1 + 8X_2$

subject to

$5X_1 + 10X_2 \leq 60$

$4X_1 + 4X_2 \leq 40$

$X_1 + X_2 \geq 0$

The optimal table of the aforementioned example is reproduced as Table 4.

(a) Let us Check whether the addition of the constraint $7X_1 + 2X_2 \leq 65$ affects the optimality. If it does, find the new optimum solution.

(b) Let us also Check whether the addition of the constraint $6X_1 + 3X_2 \leq 48$ affects the optimality. If it does, find the new optimum solution.

Solution

(a) The new constraint is:

$7X_1 + 2X_2 \leq 65$

Table 4: Optimal Table for Example.

CB_i	**C_j**	**6**	**8**	**0**	**0**	**Solution**
	Basic variable	**X_1**	**X_2**	**S_1**	**S_2**	
8	X_2	0	1	1/5	—1/4	2

6	X_1	1	0	-1/5	1/2	8
	Z_j	6	8	2/5	1	64
	C_j-Z_j	0	0	-2/5	-1	

This is satisfied by the values of the current basic variables (X_1 = 8 and X_2= 2). Hence, the new constraint is said to be a redundant constraint and the optimality will not be affected even after including the new constraint into the existing model.

(b) The new constraint is:

$6X_1 + 3X_2 \leq 48$

This is not satisfied by the values of the current basic variables (X_1 = 8 and X_2 = 2). So, the modified form of the new constraint in terms of only non-basic variables is obtained.

The standard form of the new constraint after including a slack variable $S_3$is as follows:

$6X_1, + 3X_2 + S_3 = 48$

From Table 4, the expressions with respect to X_1 and X_2 rows can be written as:

$$X_2 + \frac{1}{5}S_1 - \frac{1}{4}S_2 = 2$$

$$X_1 - \frac{1}{5}S_1 + \frac{1}{2}S_2 = 8$$

Substitution of the expressions for X_1 and X_2 in the standard form of the new constraint, yields the following.

$$\frac{3}{5}S_1 - \frac{9}{4}S_2 + S_3 = -6$$

Now, the above constraint is included in the optimal Table 4 and the result is shown in Table 5.

In Table 5, S_3 row contains a negative right-hand side constant. Hence, the solution is infeasible. This infeasibility can be removed by using dual simplex method. Application of the dual simplex method to Table 5 yields the following results:

$$X_1 = \frac{20}{3},\ X_2 = \frac{8}{3},\ S_1 = 0,\ S_2 = \frac{8}{3},\ S_3 = 0,\ Z(\text{optimum}) = \frac{184}{3} = 61.33$$

Table 5: Augmented Version of Table 4.

***CB*$_j$**	***C*$_j$**	**6**	**8**	**0**	**0**	**0**	**Solution**
	Basic variable	**X_1**	**X_2**	**S_1**	**S_2**	**S_3**	
8	X_2	0	1	1/5	-1/4	0	2
6	X_1	1	0	-1/5	1/2	0	8
0	S_3	0	0	3/5	-9/4	1	-6
	Z_j	6	8	2/5	1	0	64
	C_j - Z_j	0	0	-2/5	-1	0	

4. Adding a New Variable

In a problem like the product mix problem, over a period of time, a new product may be added to the existing product mix. Under such a situation, one will be interested in finding the revised optimal solution from the optimal table of the original problem.

In this analysis, the following items are to be determined after incorporating the data of the new variable (new product).

The C_j - Z_j value

$$C_j - Z_j = C_j - [CB]_{1\times m}\begin{bmatrix}\text{Technological coefficients of}\\ \text{optimal table with respect to the basic}\\ \text{variables of the initial table}\end{bmatrix}_{m\times m} \times \begin{bmatrix}\text{Constraint coefficients}\\ \text{of new variable}\end{bmatrix}_{m\times 1}$$

where, m is the number of constraints in the problem. If the C_j - Z_j value of the new variable indicates the optimality as per the nature of optimization (maximization or minimization), the optimality of the problem after including the new variable is not affected. Otherwise, the constraint coefficients (technological coefficients) of the new variable are to be computed.

The constraint coefficients (technological coefficients) of the column corresponding to the new variable

$$\begin{bmatrix}\text{Revised constraint} \\ \text{coefficients of the} \\ \text{new variable}\end{bmatrix}_{m\times 1} = \begin{bmatrix}\text{Technological coefficients of} \\ \text{optimal table with respect to the} \\ \text{basic variables of the initial table}\end{bmatrix}_{m\times m} \times \begin{bmatrix}\text{Constraints coefficients} \\ \text{of new variable}\end{bmatrix}_{m\times 1}$$

where, *m* is the number of constraints in the problem. These coefficients are incorporated in the current optimal table and the necessary number of iterations is to be carried out from the current table till the optimality is reached.

Example : Let us Maximize $Z = 6X_1 + 8X_2$

subject to

$5X_1 + 10X_2 \le 60$

$X_1 + 4X_2 \le 40$

$X_1 + X_2 \ge 0$

The optimal table of Example is reproduced in Table 6.

Table 6: Optimal Table of Example.

CB_i	C_j	6	8	0	0	Solution
	Basic variable	X_1	X_2	S_1	S_2	
8	X_2	0	1	1/5	-1/4	2
6	X_1	1	0	-1/5	1/2	8
	Z_j	6	8	2/5	1	64
	C_j - Z_j	0	0	-2/5	-1	

A new product P_3 is included in the existing product mix. The profit per unit of the new product is Rs. 20. The processing requirements of the new product on lathe and milling machines are 6 hours per unit and 5 hours per unit, respectively.

(a) Let us Check whether die inclusion of the product P_3 changes the optimality.

(b)If it changes the optimality, Let us find the revised optimal solution.

Solution

The LP problem after incorporating the data of the new product P_3 is shown below.

Maximize $Z = 6X_1 + 8X_2 + 20X_3$

subject to

$5X_1 + 10X_2 + 8X_2 \leq 60$

$4X_1 + 4X_2 + 5X_3 \leq 40$

$X_1, X_2, \geq 0$

(a)Determination of C_3 - Z_3. The relative contribution of the new product, P_3 is computed using the following formula.

$$C_j - Z_j = C_j - [CB] \begin{bmatrix} \text{Technological coefficients of} \\ \text{optimal table with respect to the} \\ \text{basic variables of the initial table} \end{bmatrix} \times \begin{bmatrix} \text{Constraint coefficients} \\ \text{of new variable} \end{bmatrix}$$

or

$$C_3 - Z_3 = 20 - [8\ 6] \begin{bmatrix} \frac{1}{5} & -\frac{1}{4} \\ -\frac{1}{5} & \frac{1}{2} \end{bmatrix} \times \begin{bmatrix} 6 \\ 5 \end{bmatrix} = \frac{63}{5}$$

Since, the C_3 - Z_3 value is greater than zero, the solution of the modified problem is not optimal. This means that the inclusion of the new product (new variable) in the original problem changes the optimality.

(b)Optimization of the modified problem. The constraint coefficients of the new variable X_3 are determined using the following formula:

$$\begin{bmatrix} \text{Revised constraint} \\ \text{coefficients of the} \\ \text{new variable} \end{bmatrix} = \begin{bmatrix} \text{Technological coefficients of} \\ \text{optimal table with respect to the} \\ \text{basic variables of the initial table} \end{bmatrix} \times \begin{bmatrix} \text{Constraint coefficients} \\ \text{of new variable} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{5} & -\frac{1}{4} \\ -\frac{1}{5} & \frac{1}{2} \end{bmatrix} \times \begin{bmatrix} 6 \\ 5 \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{1}{20} \\ \frac{13}{10} \end{bmatrix}$$

Table 6 of the original problem is modified as per the above details and shown in Table 7.

Table 7: Modified Table for Table 6.

CB_i	C_j	6	8	20	0	0	Solution
	Basic variable	X_1	X_2	X_3	S_1	S_2	
8	X_2	0	1	-1/20	1/5	-1/4	2
6	X_1	1	0	13/10	-1/5	1/2	8
	Z_j	6	8	37/5	2/5	1	64
	$C_j - Z_j$	0	0	63/5	-2/5	-1	

The following optimal result is obtained after carrying out two more iterations from Table 7.

$X_1 = 0$, $X_2 = 0$, $X_3 = 8$, $S_1 = 12$, $S_2 = 0$, Z(optimum) = 160 .

2

DUALITY AND NETWORKS

Definition of dual problem – Primal – Dual relation ships – Dual simplex methods – Post optimality analysis – Transportation and assignment model - Shortest route problem.

2.1 Definition of dual problem

Dual problem refers to the Lagrangian dual problem but other dual problems are used, for example, the Wolfe dual problem and the Fenagle dual problem.

The Lagrangian dual problem is obtained by forming the Lagrangian, using nonnegative Lagrange multipliers to add the constraints to the objective function, and then solving for some primal variable values that minimize the Lagrangian.

This solution gives the primal variables as functions of the Lagrange multipliers, which are called dual variables, so that the new problem is to maximize the objective function with respect to the dual variables under the derived constraints on the dual variables (including at least the non negativity).

In general given two dual pairs of separated locally convex spaces (X,X*)and (Y,Y*)and the function $f : X \to \mathbb{R} \cup \{+\infty\}$,f:X $\to$ we can define the primal problem as finding $\hat{x}$ such that $f(\hat{x}) = \inf_{x \in X} f(x)$. In other words, $f(\hat{x})$ is the infimum (greatest lower bound) of the function f.

If there are constraint conditions, these can be built into the function f by letting $\tilde{f} = f + I$ where I is the indicator function. Then let $F : X \times Y \to \mathbb{R} \cup \{+\infty\}$ be a perturbation function such that $F(x,0) = \tilde{f}(x)$
.

$$\sup_{y^* \in Y^*} -F^*(0,y^*) \leq \inf_{x \in X} F(x,0),$$

The duality gap is the difference of the right and left hand sides of the inequality.

where F^* is the convex conjugate in both variables and **Sup** denotes the supremium (least upper bound).

2.2 Primal Dual relationships

A generalized format of the linear programming problem is represented here.

Maximize or minimize $Z = C_1X_1 + C_2X_2 + ... + Cn\ X_n$

subject to

$$
\begin{array}{llll}
a_{11}X_1 & + a_{12}X_2 & + \cdots + & a_{1n}X_n \le, = \text{or} \ge b_1 \\
a_{21}X_1 & + a_{22}X_2 & + \cdots + & a_{2n}X_n \le, = \text{or} \ge b_2 \\
\vdots & \vdots & & \vdots \qquad\qquad\quad \vdots \\
a_{i1}X_1 & + a_{i2}X_2 & + \cdots + & a_{in}X_n \le, = \text{or} \ge b_i \\
\vdots & \vdots & & \vdots \qquad\qquad\quad \vdots \\
a_{m1}X_1 & + a_{m2}X_2 & + \cdots + & a_{mn}X_n \le, = \text{or} \ge b_m
\end{array}
$$

where, X_1 X_2, $X_{3,}..., X_n \ge 0$.

Let this problem be called as a *primal linear programming* problem. If the constraints in the primal problem are too many, then the time taken to solve the problem is expected to be higher Under such situation, the primal linear programming problem can be converted into its dual linear programming problem which requires relatively lesser time to solve.

Then the solution of the primal problem can be obtained from the optimal table of its dual problem by following certain rules.

Formulation of Dual Problem

The primal problem is again reproduced below:

Maximize or Minimize $Z = C_1X_1 + C_2X_2 + ...+ C_n\ X_n$

subject to

$$
\begin{array}{lcr}
a_{11}X_1 + a_{12}X_2 + \cdots + a_{1n}X_n \lesseqgtr b_1 & \longleftarrow & Y_1 \\
a_{21}X_1 + a_{22}X_2 + \cdots + a_{2n}X_n \lesseqgtr b_2 & \longleftarrow & Y_2 \\
\vdots & & \vdots \\
a_{i1}X_1 + a_{i2}X_2 + \cdots + a_{in}X_n \lesseqgtr b_i & \longleftarrow & Y_i \\
\vdots & & \vdots \\
a_{m1}X_1 + a_{m2}X_2 + \cdots + a_{mn}X_n \lesseqgtr b_m & \longleftarrow & Y_m
\end{array}
$$

where, $X_1, X_2, X_3, \ldots, X_n \geq 0$.

In the above model, the variable Y_i is called as the *dual variable* associated with the constraint *i.*

Objective function: The number of variables in the dual problem is equal to the number of constraints in the primal problem. The objective function of the dual problem is constructed by adding the multiples of the right-hand side constants of the constraints of the primal problem with the respective dual variables.

Constraints: The number of constraints in the dual problem is equal to the number of variables in the primal problem. Each dual constraint corresponds to each primal variable.

The left-hand side of the dual constraint corresponding to the j^{th} primal variable is the sum of the multiples of the left- hand side constraint coefficients of the j^{th} primal variable with the corresponding dual variables.

The right-hand side constant of the dual constraint corresponding to the j^{th} primal variable is the objective function coefficient of the j^{th} primal variable.

Some more guidelines for forming the dual problem are presented in Table 1.

Table 1. Guidelines for Dual Formation

Type of problem	**Objective function**	**Constraints type**	**Nature of variables**
Primal	Maximize	≤	Restricted in sign
Dual	Minimize	≥	Restricted in sign
Primal	Minimize	≥	Restricted in sign
Dual	Maximize	≤	Restricted in sign
Primal	Maximize	=	Restricted in sign

Dual	Minimize	≥	Unrestricted in sign
Primal	Minimize	=	Restricted in sign
Dual	Maximize	≤	Unrestricted in sign
Primal	Maximize	≤	Unrestricted in sign
Dual	Minimize	=	Restricted in sign
Primal	Minimize	≥	Unrestricted in sign
Dual	Maximize	=	Restricted in sign

Problems:

Let us form the dual of the following primal problem.

Maximize $Z = 4X_1 + 10X_2 + 25X_3$

subject to

$2X_1 + 4X_2 + 8X_3 \leq 25$

$4X_1 + 9X_2 + 8X_3 \leq 30$

$6X_1 + 8X_2 + 2X_3 \leq 40$

$X_1 + X_2 + X_3 \geq 0$

Solution:

The given problem is termed as a primal problem which is as shown below. Let Y_i be the dual variable associated with the i^{th} constraint of the primal problem as shown on next page.

$$\text{Maximize } Z = \boxed{4}\,X_1 + \boxed{10}\,X_2 + \boxed{25}\,X_3$$

subject to

$$\begin{array}{l} \begin{bmatrix}2\\4\\6\end{bmatrix} \begin{array}{l} X_1 + \\ X_1 + \\ X_1 + \end{array} \begin{bmatrix}4\\9\\8\end{bmatrix} \begin{array}{l} X_2 + \\ X_2 + \\ X_2 + \end{array} \begin{bmatrix}8\\8\\2\end{bmatrix} \begin{array}{l} X_3 \le 25 \longleftarrow Y_1 \\ X_3 \le 30 \longleftarrow Y_2 \\ X_3 \le 40 \longleftarrow Y_3 \end{array} \\ X_1, X_2, X_3 \ge 0 \end{array}$$

The corresponding dual problem may be presented as:

Minimize $Y = 25Y_1 + 30Y_2 + 40Y_3$

subject to

$2Y_1 + 4Y_2 + 6Y_3 \ge 4$

$4Y_1 + 9Y_2 + 8Y_3 \ge 10$

$8Y_1 + 8Y_2 + 2Y_3 \ge 25$

$Y_1 + Y_2 + Y_3 \ge 0$

Form the dual of the following primal problem.

Minimize $Z = 20X_1 + 40X_2$

subject to

$2X_1 + 20X_2 \ge 40$

$20X_1 + 3X_2 \ge 20$

$4X_1 + 15X_2 \ge 30$

X_1 and $X_2 \ge 0$

Solution

Let, Y_i be the dual variable associated with the ith constraint of the given primal problem. The dual of the given above primal problem is:

Maximize $y = 40Y_1 + 20Y_2 + 30Y_3$

subject to

$2Y_1 + 20Y_2 + 4Y_3 \le 20$

$20Y_1 + 3Y_2 + 15Y_3 \le 40$

Y_1, Y_2 and $Y_3 \ge 0$

2.3 Dual simplex methods

Dual simplex method is a specialized form of simplex method in which optimality is maintained in all the iterations. Initially, the solution may not be feasible. Successive iterations will remove the infeasibility.

If the problem is feasible in an iteration, then the procedure will be stopped, because the solution obtained is feasible and optimal at that stage.

This method is an essential subroutine for integer programming method in which it is repeatedly used to remove the infeasibility due to additional constraints known as *Gomory's cuts.* In this section, the dual simplex method is demonstrated through a numerical problem.

Table 1. Initial Table and Different Iterations of Example.

CB,	**C_j**	**20**	**10**	**15**	**0**	**0**	**0**	**$-M$**	**Solution**	**Ratio**
	Basic variable	**X_1**	**X_2**	**X_3**	**S_1**	**S_2**	**S_3**	**R_1**		
0	S_1	8	6	2	1	0	0	0	60	30
$-M$	R_1	5	1	6	0	-1	0	1	40	20/3
0	S_3	2	6	3	0	0	1	0	30	10
	Z_j	$-5M$	$-M$	$-6M$	0	M	0	$-M$	$-40M$	
	C_j-Z_j	$20 + 5M$	$10 + M$	$15 + 6M$	0	$-M$	0	0		
0	S_1	19/3	17/3	0	1	1/3	0	-1/3	140/3	140/19
15	X_3	5/6	1/6	1	0	-1/6	0	1/6	20/3	8
0		-1/2	11/2	0	0	1/2	1	-1/2	10	-

	Z_j	25/2	5/2	15	0	-5/2	0	5/2	100	
	C_j-Z_j	15/2	15/2	0	0	5/2	0	-M-5/2		
20	X_1	1	17/19	0	3/19	1/19	0	-1/19	140/19	140
15	X_3	0	-11/19	1	-5/38	-4/19	0	4/19	10/19	-
0	S_3	0	113/19	0	3/38	10/19	1	-10/19	260/19	26
	Z_j	20	175/19	15	45/38	-40/19	0	40/19	2950/19	
	C_j - Z_j	0	15/19	0	-45/38	40/19	0	-M-40/19		
20	X_1	1	3/10	0	3/20	0	-1/10	0	6	
15	X_3	0	9/5	1	-1/10	0	2/5	0	6	
0	S_2	0	113/10	0	3/20	1	19/10	-1	26	
	Z_j	20	33	15	3/2	0	4	0	210	
	C_j-Z_j	0	-23	0	-3/2	0	-4	-M		

Problems:

Let us solve the following linear programming problem using dual simplex method.

Minimize $Z = 2X_1 + 4X_2$

subject to

$2X_1 + X_2 \geq 4$

$X_1 + 2X_2 \geq 3$

$2X_1 + 2X_2 \leq 12$

X_1 and $X_2 \geq 0$

Solution

Convert the constraints of the given linear programming problem into '≤' type, wherever necessary, as shown below:

Minimize Z = $2X_1 + 4X_2$

Subject to

$-2X_1 - X_2 \leq -4$

$-X_1 - 2X_2 \leq -3$

$2X_1 + 2X_2 \leq 12$

X_1 and $X_2 \geq 0$

The canonical form of the above model is shown below in which S_1 S_2 andS_3 are slack variables.

Minimize Z = $2X_1 + 4X_2$

subject to

$-2X_1 - X_2 + S_1 = -4$

$-X_1 - 2X_2 + S_2 = -3$

$2X_1 + 2X_2 + S_3 = 12$

X_1 X_2, S_1 S_2 and $S_3 \geq 0$

The initial table based on the canonical form of the given problem is shown in Table 2. For minimization problem, if all C_j - Z_j are greater than or equal to 0, then optimality is reached.

Table 2. Iteration 1.

CB_i	**C_j**	**2**	**0**	**0**	**0**	**0**	
CB_i	**Basic**						**Solution**

	variable	X_1	X_2	S_1	S_2	S_3	
0	S_1	-2	-1	1	0	0	-4**
0	S_2	-1	-2	0	1	0	-3
0	S_3	2	2	0	0	1	12
	Z_j	0	0	0	0	0	0
	C_j - Z_j	2*	4	0	0	0	

In Table 2, C_j - Z_j row clearly shows that the problem is optimal. But some of the values under the solution column are negative. These negative values will retain infeasibility in the solution. The guidelines for removing the infeasibility are presented below.

Feasibility condition: The leaving variable is the variable which is having the most negative value (break ties arbitrarily). If all the basic variables are non-negative, then the feasible (optimal) solution is reached. Hence, the procedure ends here.

Optimality condition: The entering variable is selected from among the non-basic variables as follows:

1.Find the ratios of the negative coefficients of the criterion row (C_j - Z_j) equation to the corresponding left-hand side coefficients of the equation associated with the leaving variable. Ignore the ratios associated with positive or zero denominators.

2.The entering variable is the one with the smallest ratio if the problem is minimization, or the smallest absolute value of the ratios if the problem pertains to maximization (break tie arbitrarily). If all the denominators are zero or positive, then the problem has no feasible solution.

In Table 2, the leaving variable is S_1 which has the most negative right-hand side value. The entering variable is determined as shown in Table 3.

Table 3: Determination of Entering Variable

Variable	X_1	X_2	S_1	S_2	S_3

-(C_j - Z_j)	**-2**	**-4**	**0**	**0**	**0**
S_1 equation	**-2**	**-1**	**1**	**0**	**0**
Ratio	**1**	**4**	**-**	**-**	**-**

Since the problem is of the minimization type, entering variable is the one which has the smallest ratio, and the smallest ratio is 1 which is corresponding to the variable X_1. Therefore, the entering variable is X_1.The next iteration is shown in Table 4.

Table 4: Iteration 2

	C_j	**2**	**4**	**0**	**0**	**0**	
CBi	**Basic variable**	**X_1**	**X_2**	**S_1**	**S_2**	**S_3**	**Solution**
2	**X_1**	**1**	**1/2**	**-1/2**	**0**	**0**	**2**
0	**S_2**	**0**	**-3/2**	**-1/2**	**1**	**0**	**-1****
0	**S_3**	**0**	**1**	**1**	**0**	**1**	**8**
	Z_j	**2**	**1**	**-1**	**0**	**0**	**4**
	C_j-Z_j	**0**	**3**	**1***	**0**	**0**	

In Table 4, the solution is optimal but it is not feasible, as the solution value of the row S_2 is negative. So, the leaving variable is S_2 which has negative right-hand side value. The entering variable is determined as in Table 5.

Table 5: Determination of Entering Variable

Variable	**X_1**	**X_2**	**S_1**	**S_2**	**S_3**

-(C_j - Z_j)	**0**	**-3**	**-1**	**0**	**0**
S_2 equation	**0**	**-3/2**	**-1/2**	**1**	**0**
Ratio	**-**	**2**	**2**	**-**	**-**

Since the problem is concerned with minimization, the entering variable is the one which has the smallest ratio. The smallest ratio is 2 corresponding to the variable X_2 and S_1. By breaking the tie randomly, S_1 is selected as the entering variable. The next iteration is shown in Table 6.

In Table 6, the solution values (right-hand side values) are feasible and at the same time, the solution is optimal. The corresponding results are:

$X_1 = 3$, $X_2 = 0$, $S_1 = 2$, $S_3 = 6$ and Z (optimum) = 6

where all other variables are zero.

CB;	***Cj***	***2***	**4**	**0**	**0**	**0**	**Solution**
	Basic variable	**X_1**	**X_2**	**S_1**	**S_2**	**S_3**	
2	X_1	1	2	0	-1	0	3
0	S_1	0	3	1	-2	0	2
0	S_3	0	-2	0	2	1	6
	Z_j	2	4	0	-2	0	6
	C_j-Z_j	0	0	0	2	0	

2.4 Post optimality analysis

Post-optimality analysis of LP model

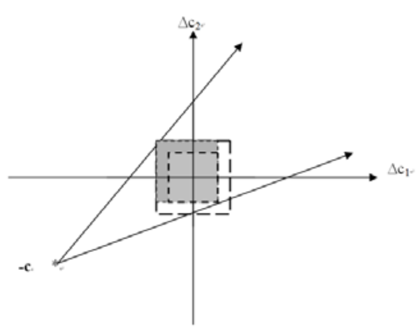
After the optimal solution has been computed for a given model, it is important to know how the solution behaves under different variations in problem parameters. Sensitivity analysis and stability analysis are used to evaluate the effects of variations on the optimal solution or basis of the LP problem. Consider a LP problem in form:

$$Max\ \{ P=\mathbf{c}^{\mathbf{T}}\mathbf{x} : \mathbf{Ax} \leq \mathbf{b},\ \ \mathbf{x} \geq \mathbf{0},\ \ \mathbf{x} \in \mathbf{R}^{\mathbf{n}} \} \quad \_\_\_\_\_\_(1)$$

Sensitivity analysis is usually associated with the determination of the values of the Lagrange multipliers, λ, that describe the change in the optimal solution with respect to the variations in RHS coefficients. The sensitivity relations are important and useful for the decision maker, but the major challenge is to determine when they are valid.

For example, the Lagrange multiplier, λ_i , presents the increase in the optimal value for a maximization problem when the associated RHS coefficient, i.e., b_i , is increased by one unit; however, we do not know by how much the coefficient can be increased under simultaneous variations in vector b before the optimal basis changes and the value of the Lagrange multiplier becomes invalid.

This shows the importance of computing stability limits for each coefficient under simultaneous variations and within which the optimal basis remains unchanged.



Obtained stability limits inside the stability cone.

During the last few decades, many stability approaches have been proposed, for variation in parameters of LP. To date there is no single approach that dominates.

In contrast to other approaches, the tolerance approach leads to easy-to-use results and considers simultaneous and independent variation in the problem parameters.

It basically depends on optimality conditions and uses the concept of basic and non-basic variables to modify matrix A at each iteration.

In this study, the modified tolerance approach proposed by Al-Shammari is used to determine the stability ranges. The proposed method provides a new perspective on the problem and has two steps for computing the stability region or limits.

First, it defines the entire stability region as a cone and studies the relation between the sensitivity information, Lagrange multipliers, and model parameters.

Second, it determines maximum stability limits presented by the maximum rectangular parallelepiped or hyper box that can be built inside the cone.

This hyper box offers flexible and easy-to-use allowable variation limits as shown later on for variation in objective coefficients, i.e. prices or raw materials and products.

To demonstrate the approach, consider problem that has a unique optimal solution. To define the entire stability region or stability cone for variations in the coefficients of the objective function, duality information or Lagrange multipliers are used:

$$\nabla_b P^* = \lambda = c^T A_A^{-1} \quad ..(2)$$

where A_A is the matrix of active constraints. By introducing the perturbations vector, Δc^T:

$$\lambda' = (c^T + \Delta c^T) A_A^{-1} \quad ..(3)$$

by using the non-negativity condition on the optimal solution, there is no change in optimal solution if $\lambda' \geq 0$. By substituting and rearranging:

$$-\Delta c^T A_A^{-1} \leq \lambda \quad ..(4)$$

This inequality relation represents the stability region. This stability region can be defined as a stability cone because it satisfies the definition of a cone. In other words, the optimal solution and basis (not objective value) remain optimal under any scalar positive multiplication in the objective function. The solution remains optimal for any variations that satisfy equation (4).

The stability cone is shown in Figure for a maximization problem with two variables. Next step is defining the largest possible stability ranges starting with computing ordinary (individual) stability limits for Δc_i as follows:

$$\left(\max_h \quad \Delta c_h = \frac{\lambda_g}{A_{Agh}^{-1}} : \Delta c_h < 0\right) < \Delta c_h^{ord} < \left(\min_h \quad \Delta c_h = \frac{\lambda_g}{A_{Agh}^{-1}} : \Delta c_h > 0\right) \quad ...(5)$$

where h and g are indices of objective coefficients and constraints, respectively. In Figure 2, the ordinary stability limits of the coefficient c_i are the intersections between the cone's constraints and Δc_i axis.

The main challenge in LP stability analysis is the presentation of this cone to the decision maker in a simple and useful way, especially for simultaneous variations. The most useful approach is to construct the largest possible hyper box inside the cone.

For variations in the RHS coefficients, a similar stability analysis is employed for the dual problem:

$$min\ \{ \lambda^T b : \lambda^T A \geq c^T \ \&\ \lambda \geq 0 \}$$ ____(6)

to determine the variation limits before the optimal basis changes. In this analysis the optimal solution and slack variables are used in the same manner as the Lagrange multipliers were used in the variations analysis of vector c.

2.5 Transportation and assignment model

1. Transportation Problem (TP)

It involves distribution of any commodity from any group of supply centers, called sources, to any group of receiving centers, called destinations, in such a way as to minimize the total distribution cost (shipping cost).

Total supply must equal total demand. If total supply exceeds total demand, a dummy destination, whose demand equals the difference between the total supply and total demand is created. similarly if total supply is less than total demand, a dummy source is created, whose supply equals the difference.

All units shipping costs of a dummy destination or out of a dummy source are 0.

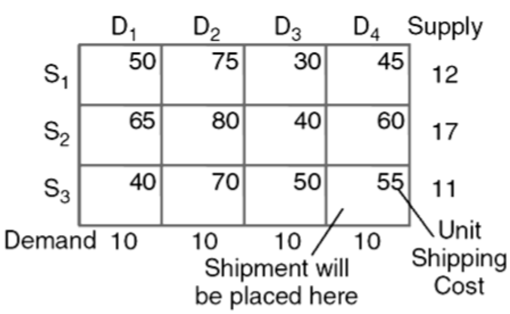
		DESTINATIONS				Supply
		D_1	D_2	D_3	D_4	
Sources	S_1	50	75	30	45	12
	S_2	65	80	40	60	17
	S_3	40	70	50	55	11
Demand		10	10	10	10	

Example 1:

Example 2:

		Destination			Supply
	D1	D2	D3	D4	
S1	50	75	35	75	12
Source S2	65	80	60	65	17
S3	40	70	45	55	11
(D)	0	0	0	0	10
Demand	15	10	15	10	

Transportation Table:



Solving TP – Transportation Simplex Method

1. To find the current $C_{ij}–Z_{ij}$ values for both nonbasic variable and select the one with the most negative $C_{ij}–Z_{ij}$ value as the entering variable; if all $C_{ij}–Z_{ij}$ values are nonnegative, the current solution is optimal.

2. Determine which basic variable reaches 0 first when the entering variable is increased.

3. Determine a new basic solution and repeat the steps.

Step 1: Calculate the $C_{ij}–Z_{ij}$ values for the nonbasic variables.

1. If U_i is the dual variable associated with the i^{th} supply constraint, and V_j is the dual variable associated with the j^{th} demand constraint, then for shipments from node i to node j, one can find the corresponding Z_{ij} value by $Z_{ij} = U_i - V_j$. Thus the $C_{ij}–Z_{ij}$ value for variable X_{ij} is found by

$$C_{ij} - Z_{ij} = C_{ij} - (U_i - V_j) = C_{ij} - U_i + V_j$$

2. Given that there is a redundant equation among the mn constraints (and any of the mn constraints can be considered the redundant one), one can show that the U_i or V_j associated with the redundant equation is 0. Thus one U_i or V_j can arbitrarily be selected and set to 0. Arbitrarily choose $U_1 = 0$.

Since the $C_{ij}–Z_{ij}$ values for basic variables are 0 (i.e., $C_{ij} - U_i + V_j = 0$ for basic variables), we can easily solve for the remaining values of the U_i's and V_j's from the m + n - 1 equations for the basic variables.

3. Once the U_i's and V_j's have been determined, the $C_{ij}–Z_{ij}$ values for the nonbasic variables can be calculated by

$$C_{ij} - Z_{ij} = C_{ij} - U_i + V_j$$

Transportation of simplex method

Find an initial basic feasible solution by some starting procedure. Then,

1. A set U_1= 0. Solve for the other U_i's and $V_{j'}$s by:

$C_{ij} – U_i + V_j = 0$ for normal variables.

Then calculate the $C_{ij}–Z_{ij}$ values for nonbasic variables by:

$$C_{ij} – Z_{ij} = C_{ij} – U_i + V_j$$

Choose the nonbasic variable with the most negative $C_{ij}–Z_{ij}$ value as the entering variable. If all $C_{ij}–Z_{ij}$ values are nonnegative, and STOP; the current solution is optimal.

2. Find the cycle that includes the entering variable and some of the basic variables. Alternating positive and negative changes on this cycle, determine the “change amount” as the smallest allocation on the cycle at which a subtraction will be made.

3. Modify the allocations to the variables of the cycle found in step 2 by the “change amount” and return to the step 1.

Note: There must be m + n - 1 basic variables for the transportation simplex method to work.

• Add dummy source or dummy destination, if necessary

(m=# of sources and n=# of destinations)

2. The Assignment Problem (AP)

A special case of TP with m=n and $s_i=d_j$ for all i and j.

The Hungarian Algorithm

solving the assignment problem of a least cost assignment of m workers to m jobs.

Assumptions:

There is a cost assignment matrix for the m "people" to be assigned to m "tasks."

2. Total costs are nonnegative.

3. The problem is a minimization problem.

Iterative Steps

1. Make as many 0 cost assignments as possible. If all workers are assigned, STOP; this is the minimum cost assignment. Otherwise draw the minimum number of horizontal and vertical lines necessary to cover all 0's in the matrix.

2. Find the smallest value not covered by the lines; this number is the reduction value.

3.Subtract the reduction value from all numbers not covered by any lines. Add the reduction value to any number covered by both a horizontal and vertical line.

GO TO STEP 1.

Example:

	A	B	C	D	E
1	10	4	6	10	12
2	11	7	7	9	14
3	13	8	12	14	15
4	14	16	13	17	17
5	17	11	17	20	19

↓ Row Subtraction

	A	B	C	D	E
1	6	0	2	6	8
2	4	0	0	2	7
3	5	0	4	6	7
4	1	3	0	4	4
5	6	0	6	9	8

↓ Column Subtraction

	A	●B	C	D	E
●1	5	$\boxed{0}$	2	4	4
2	~~3~~	~~0~~	~~$\boxed{0}$~~	~~0~~	~~3~~
●3	4	0	4	4	3
4	~~$\boxed{0}$~~	~~3~~	~~0~~	~~2~~	~~0~~
●5	5	0	6	7	4

Minimum uncovered number

	A	●B	C	D	E
1	~~3~~	~~0~~	~~$\boxed{0}$~~	~~2~~	~~2~~
2	~~3~~	~~2~~	~~0~~	~~$\boxed{0}$~~	~~3~~
●3	2	$\boxed{0}$	2	2	1
4	~~$\boxed{0}$~~	~~5~~	~~0~~	~~2~~	~~0~~
●5	3	0	4	5	2

	A	B	C	D	E
1	3	1	$\boxed{0}$	2	2
2	3	3	0	$\boxed{0}$	3
3	1	0	1	1	$\boxed{0}$
4	$\boxed{0}$	6	0	2	0
5	2	$\boxed{0}$	3	4	1

Optimal Solution

1–C	6
2–D	9
3–E	15
4–A	14
5–B	11
	65

Assignment Problem Example

	Project		
Person	1	2	3
Adams	$11	$14	$6
Brown	$8	$10	$11
Cooper	$9	$12	$7

Find the opportunity cost table by:

a) Subtracting the smallest number in each row of the original cost table or matrix from every number in that row.

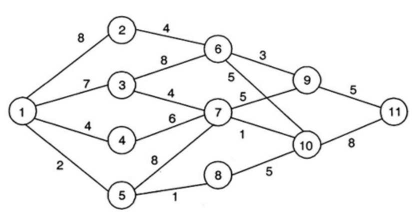
b) Then subtracting the smallest number in each column of the table obtained in part, a) from every number in that column.**2.6 Shortest route problem**

Stage-coach Problem (Shortest-path Problem):

Stage-coach problem is a shortest-path problem in which the objective is to find the shortest distance and the corresponding path from a given source node to a given destination node in a given distance network. Application of dynamic programming technique to this problem is illustrated using Example.

Problems:

A distance network consists of eleven nodes which are distributed as shown in Figure. Let us Find the shortest path from node 1 to node 11 and also the corresponding distances.



Distance network.

Solution

Each pair of adjacent vertical columns of nodes is treated as a stage. As show in Figure 2, there are four stages in this problem. Since the stages are defined from right to left, backward recursive function is to be used.

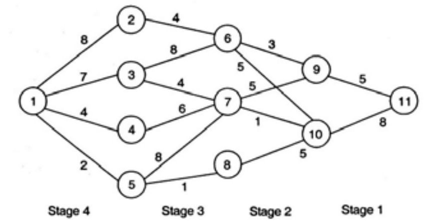


Figure 2: Distance network with stages.

In stage 1, the possible alternative is only one, i.e. node 11. In the same stage (stage 1), the possible state variables are nodes 9 and 10.

The *recursive function* $f_1(x_1)$for a given combination of the state variable,x_1 and alternative, m_l in the first stage is:

$f_1(x_l) = d(x_1\ m_1)$

where $d(x_1\ m_1)$ is the distance between node x_1 and node m_1,

The *recursive function* for a given combination of the state variable, x_i, and alternative, m_i the stage *i* for *i* more than 1 is presented below:

$f_i(x_i) = d(x_i\ m_i) + f_{i\text{-}1}(x_{i\text{-}1} = m_i)$

where $d(x_i, m_i)$ be the distance between the nodes x_i**,** and m_i $f_i(x_i)$ be the shortest distance from node x_i in the current stage *i* to die destination node in stage 1 (node 11 in this example).

Stage 1: The recursive function $f_1(x_1)$ for stage 1 is:

$f_1\text{i}(x_1) = d(x_1, m_1)$

The corresponding distances are shown in Table 1. For each value of the state variable, the best distance and the corresponding alternative are presented in the one and the last columns, respectively of Table 1.

Table 1: Calculations for Stage 1

	Alternative m_1		
State variable			
x_1	11	$f_1(x_1)$*	**m_1***

9	5	5	11
10	8	8	11

Stage 2: The recursive function $f_2(x_2)$ for a given combination of the variable, x_2 and alternative, m_2 in the second stage is

$f_2(x_2) = d(x_2, m_2) + f_1(x_1 = m_2)$

The corresponding distances are summarized in Table 2. For each value of the state variable, the best distance and the corresponding alternative are presented in the last two columns of Table 2, respectively.

Table 2: Calculations for Stage 2.

	Alternative m_2			
State variable	9	10		
x_2			$f_2(x_2)$*	**m_2***
6	3 + 5 = 8	5 + 8 = 13	8	9
7	5 + 5 = 10	1 + 8 = 9	9	10
8	-	5 + 8 = 13	13	10

Stage 3: The recursive function $f_3(x_3)$ for a given combination of the state variable, x_3 and alternative, m_3 is:

$f_3(x_3) = d(x_3, m_3) + f_2(x_2 = m_3)$

The corresponding distances are summarized in Table 3. For each value of the state variable, the best distance and the corresponding alternative are presented in the last but one and the last columns of Table 3, respectively.

Table 3: Calculations for Stage 3.

State variable x_3	Alternative m_3				
	6	7	8	$f_3(x_3)$*	**m_3***
2	4 + **8=** 12	-	-	12	6
3	**8 + 8** = 16	4 + 9=13	-	13	7
4	-	6 + 9 = 15	-	15	7
5	-	**8** + 9 = 17	1 + 13 = 14	14	**8**

Stage 4: The recursive function $f_4(x_4)$ for a given combination of the state variable, x_4 and alternative, m_4 in the fourth stage is:

$f_4(x_4) = d(x_4, m_4) + f_3(x_3 = m_4)$

The corresponding distances are summarized in Table 4. For each value of the state variable, the best distance and the corresponding alternative are presented in the last two columns of Table 4, respectively.

Table 4: Calculations for Stage 4.

	Alternative m_4						
State variable x_4	1	2	3	4	5	$f_4(x_4)$*	**m_4***
1	8 + 12 = 20	7 + 13 = 20	4 + 15 = 19	2 + 4 = 16	2 + 14 = 16	16	5

The final result of the original problem are traced in Table 4 to Table 1 backwards. Therefore, the shortest path is 1-5-8-10-11. Hence, the corresponding shortest distance = 16 units.

3

INTEGER PROGRAMMING

Cutting plan algorithm – Branch and bound methods, Multistage (Dynamic) programming.

3.1 Cutting plan algorithm

An algorithm for solving fractional (pure integer) and mixed integer programming problems has been developed by Ralph E. Gomory.

Fractional (pure integer) algorithm

Step 1: First, relax the integer requirements.

Step 2: Solve the resulting LP problem using simplex method.

Step 3: If all the basic variables (or the required variables) have integer values, optimality of the integer programming problem is reached. So, go to step 7; otherwise go to step 4.

Step 4: Examine the constraints corresponding to the current optimal solution. Also, let m be the number of constraints, n be the number of variables (including slack, surplus and artificial variables), b_i be the right-hand side value of the i^{th} constraint, and a_{ij} be the technological coefficients (matrix of left-hand side constants of the constraints). Then, the constraint equations are summarized as follows:

$$\sum_{j=1}^{n} a_{ij} X_j = b_i, \quad i=1,2,3,....,m$$

For each basic variable with non-integer solution in the current optimal table, find the fractional part, f_i Therefore, $b_i = [b_i] + f_i$ where $[b_i]$ is the integer part of b_i and f_i is the fractional part of b_i.

Step 5: Choose the largest fraction among various f_i's; i.e. Max (f_i). Treat the constraint corresponding to the maximum fraction as the source row. Let the corresponding source row be as follows:

$b_i = X_i + \sum_{j=1}^{n} a_{ij} w_j$ or $X_i = b_i - \sum_{j=1}^{n} a_{ij} w_j$

where variables X_i (i = 1, 2, 3,..., m) represent basic variables and variables W_j (j = 1, 2, 3..., n) are the non-basic variables. This kind of assumption is for convenience only.

Some examples of b_i and a_{ij} into integer and fractional parts are shown as in Table 1.

Table 1: Examples of Integer and Fractional Parts.

b_i **or** a_{ij}	$[b_i]$ **or** $[a_{ij}]$ **(Truncated integer)**	$f_i = b_i - [b_i]$ $f_{ij} = a_{ij} - [a_{ij}]$
$2\frac{2}{3}$	2	$\frac{2}{3}$
$-3\frac{1}{4}$	-4	$\frac{3}{4}$
-5	-5	0
$-\frac{3}{7}$	-1	$\frac{4}{7}$

Based on the source equation, develop an additional constraint (Gomory's constraint or fractional cut) as shown below:

$S_i = \sum_{j=1}^{n} f_{ij} w_j - f_i$ or $-f_i = s_i - \sum_{j=1}^{n} f_{ij} w_j$

where S_i is non-negative slack variable and also an integer.

Step 6: Append the fractional cut as the last row in the latest optimal table and proceed further using dual simplex method, and find the new optimum solution. If this new optimum solution is integer then go to step 7; otherwise go to step 4.

Step 7: Print the integer solution [X_i's and Z values].

Problems:

Let us find the optimum integer solution to the following linear programming problem.

Maximize $Z = 5X_1 + 8X_2$

subject to

$X_1 + 2X_2 \leq 8$

$4X_1 + X_2 \leq 10$

$X_1, X_2 \geq 0$ and integers

Solution

The canonical form of the above problem is as follows:

Maximize $Z = 5X_1 + 8X_2$

subject to

$X_1 + 2X_2 + S_1 = 8$

$4X_1 + X_2 + S_2 = 10$

X_1 X_2, S_1 and $S_2 \geq 0$ and integers

The initial table is shown in Table 2.

Table 2: Initial Table

CB_i	**C_j**	**5**	**8**	**0**	**0**	**Solution**	**Ratio**
	Basic variable	**X_1**	**X_2**	**S_1**	**S_2**		

0	S_1	1	2	1	0	8	8/2 = 4*
0	S_2	4	1	0	1	10	10/1 = 10
	Z_j	0	0	0	0	0	
	C_j - Z_j	5	8*	0	0		

From the above table, the entering variable is X_2, since its C_j - Z_j value is the maximum positive value. The minimum ratio is 4 and the corresponding variable is S_1. Therefore, S_1 leaves the basis. The resulting table is shown as in Table 3.

Table 3: Iteration 1

***CB**$_i$*	***C**$_j$*	**5**	**8**	**0**	**0**	**Solution**	**Ratio**
	Basic variable	***X**$_1$*	***X**$_2$*	***S**$_1$*	**S2**		
8	X_2	1/2	1	1/2	0	4	8
0	S_2	7/2	0	-1/2	1	6	12/7*
	Z_j	4	8	4	0	32	
	C_j - Z_j	1*	0	-4	0		

In Table 3, the maximum positive value for C_j - Z_j is 1. The corresponding variable is X_1. Therefore, X_1 enters the basis. The minimum ratio is for theS_2 row. Therefore, S_2 leaves the basis. The resulting table is shown as in Table 4.

Table 4: Iteration 2 (Optimal Table)

***CB**$_i$*	***C**$_j$*	**5**	**8**	**0**	**0**	**Solution**

	Basic variable	X_1	X_2	S_1	S_2	
8	X_2	0	1	4/7	-1/7	22/7
5	X_1	1	0	-1/7	2/7	12/7
	Z_j	5	8	27/7	2/7	236/7
	C_j - Z_j	0	0	-27/7	-2/7	

In Table 4, all the values in the criterion row (C_j - Z_j row) are 0 or negative. Hence, optimality for linear programming is reached. The results are as follows:

$$X_1 = \frac{12}{7}, \qquad X_2 = \frac{22}{7}, \qquad Z(\text{optimum}) = \frac{236}{7}$$

Since the values of the decision variables X_1 and X_2 are not integers, the solution is not optimal. So, to obtain integer solution for the given problem further steps are carried out.

The integer and fractional parts of the basic variables are summarized in Table 5.

Table 5: Summary of Integer and Fractional Parts

Basic variable in the optimal table	b_i	$[b_i] + f_i$
X_1	12/7	1 + (5/7)
X_2	22/7	3 + (1/7)

The fractional part, f_1 is the maximum. So, select the row X_1 as the source row for developing the first cut.

$$\frac{12}{7} = X_1 - \frac{1}{7}S_1 + \frac{2}{7}S_2$$

or

$$1 + \frac{5}{7} = X_1 + \left(-1 + \frac{6}{7}\right)S_1 + \left(0 + \frac{2}{7}\right)S_2$$

The corresponding fractional cut is

$$-\frac{5}{7} = S_3 - \frac{6}{7}S_1 - \frac{2}{7}S_2 \qquad \text{(Cut 1)}$$

This cut is appended to Table 4 as reproduced in Table 6 and further solved using dual simplex method.

Table 6: Table after Appending Cut 1.

***CB**$_i$*	***C**$_j$*	**5**	**8**	**0**	**0**	**0**	**Solution**
	Basic variable	X_1	X_2	S_1	S_2	S_3	
8	X_2	0	1	4/7	-1/7	0	22/7
5	X_1	1	0	-1/7	2/7	0	12/7
0	S_3	0	0	-6/7	-2/7	1	-5/7*
	Z_j	5	8	27/7	2/7	0	236/7
	C_j - Z_j	0	0	-27/7	-2/7*	0	

Only the third row (containing S_3) has a negative solution value. Therefore, S_3 leaves the basis. The entering variable is determined in Table 7.

Table 7: Determination of Entering Variable

Variable	X_1	X_2	S_1	S_2	S_3
- (C_j - Z_j)	0	0	27/7	2/7	0
Row S_3	0	0	-6/7	-2/7	1

Ratio (absolute value)	-	-	9/2	1	-

The smallest absolute ratio is 1 and the corresponding variable is S_2. So, the variable S_2 enters the basis. The resultant values are shown in Table 8.

Table 8: Table after Pivot Operation

CB_i	C_j	5	8	0	0	0	
	Basic variable	X_1	X_2	S_1	S_2	$S3$	Solution
8	X_2	0 1 0	1 0 0	1 -1 3	0 0 1	-1/2	7/2
5 0	X_1 S_2					1 -7/2	1 5/2
Z_j C_j - Z_j		5	8	3	0	1	33
		0	0	-3	0	-1	

The solution is still non-integer. So, develop a fractional cut. The basic variables, X_2 and S_2 are not integers. The fractional parts of both of them are 1/2. The constraint X_2 is selected randomly as the source row for developing the next cut. Therefore,

$$\frac{7}{2} = X_2 + S_1 - \frac{1}{2}S_3$$

or

$$3 + \frac{1}{2} = (1 + 0)\,X_2 + (1 + 0)\,S_1 + \left(-1 + \frac{1}{2}\right)S_3$$

Therefore, the corresponding fractional cut is

$$-\frac{1}{2} = S_4 - \frac{1}{2}S_3 \qquad \text{(Cut 2)}$$

Append this constraint at the end of Table 8 as shown in Table 9.

CB_i	C_j	5	8	0	0	0	0	Solution
	Basic variable	X_1	X_2	S_1	S_2	S_3	S_4	
8	X_2	0	1	1	0	-1/2	0	7/2
5	X_1	1	0	-1	0	1	0	1
0	S_2	0	0	3	1	-7/2	0	5/2
0	S_4	0	0	0	0	-1/2	1	-1/2*
Z_j		5	8	3	0	1	0	33
C_j - Z_j		0	0	-3	0	-1*	0	

Table 9: Table after Pivot Operation

Only row S_4 has a negative value under the solution column. Therefore, S_4 leaves the basis. The entering variable is determined based on Table 10.

Table 10: Determination of Entering Variable

Variable	X_1	X_2	S_1	S_2	S_3	S_4
- (C_j - Z_j)	0	0	3	0	1	0
Row S_4	0	0	0	0	-1/2	1

Ratio (absolute value)	-	-	-	-	2	-

The smallest absolute ratio (only ratio) is 2 and the corresponding variable is S_3. So, the variable S_3 enters the basis. The resultant values are shown as in Table 11.

Table 11: Table after Pivot Operation

CB_i	C_j	5	8	0	0	0	0	Solution
	Basic variable	X_1	X_2	S_1	S_2	S_3	S_4	
8	X_2	0	1	1	0	0	-1	4
5	X_1	1	0	-1	0	0	2	0
0	S_2	0	0	3	1	0	-7	6
0	S_3	0	0	0	0	1	-2	1
Z_j C_j - Z_j		5	8	3	0	0	2	32
		0	0	-3	0	0	-2	

In Table 11, the values of all the basic variables are integers. So, the optimality is reached and the corresponding results are summarized as follows:

X_1 = 0, X_2 = 4 Z(optimum) = 32.

Directions to solve mixed integer programming problems

In reality, all the decision variables of an integer programming problem need not be integers. In such problem, if the solution values to the linear programming problem are integers, the steps of adding Gomory's cut are not required.

Otherwise, necessary number of iterations is to be carried out by adding different cuts until the values of the required decision variables are made integers.

In each of these iterations, the source row is selected from among the rows corresponding to the decision variables whose values are restricted to integer in the given problem.

Research direction to design superior cut

The effectiveness of the integer programming procedure is determined in terms of number of cuts required to solve a given problem. Instead of simply selecting the source row based on the maximum value of f_i one can use different approaches which will result in a reduced number of cuts required to solve a problem.

This is called the **strength/effectiveness** of the **cut.** Researches have come out with superior cuts which will result in reduced number of cuts to solve a problem.

Let us consider the following integer linear programming problem and solve it.

Maximize $Z = 5X_1 + 10X_2 + 8X_3$

subject to

$2X_1 + 5X_2 + X_3 \leq 10$

$X_1 + 4X_2 + 2X_3 \leq 12$

$X_1 + X_2 + X_3 \geq 0$

Solution:

The canonical form of the given problem is shown below.

Maximize $Z = 5X_1 + 10X_2 + 8X_3 + OS_1 + 0S_2$

subject to

$2X_1 + 5X_2 + X_3 + S_1 = 10$

$X_1 + 4X_2 + 2X_3 + S_2 = 12$

X_1, X_2, X_3, S_1 and $S_2 \geq 0$

The different iterations of the simplex method applied to this problem till the optimality is reached are shown in Table 12. The optimal results of the linear programming problem from the last iteration of the Table 12 are:

$X_1 = 8/3 = 2 + 2/3$, $X_3 = 14/3 = 4 + 2/3$ and Z(optimum) $= 152/3$.

The value of the decision variables, X_1 and X_3 are not integers. Further, the fractional part of X_1 as well as X_3 is 2/3. So, the maximum fractional part is 2/3 and the row X_1 is selected for developing a Gomory's cut.

$8/3 = X_1 + 2X_2 + 2/3S_1 - 2/3S_1 - 1/3S_2$

$2 + 2/3 = (1 + 0)X_1 + (2 + 0)X_2 + (0 + 2/3)S_1 + (-1 + 2/3)S_2$

The corresponding fractional cut is shown below and it is appended to the final iteration of the Table 12 as shown in Table 13.

$-2/3 = S_3 - 2/3S_1 - 2/3S_2 \rightarrow$ Cut 1

In Table 13, the leaving variable is S_3. The entering variable is determined as shown in Table 14 and it is S_1. The corresponding next iteration is presented in Table 15. Since, the results in Table 15 are integers, the optimal integer solution is reached. The optimal solution is:

$X_1 = 2$, $X_3 = 5$, $S_1 = 1$ and Z(optimum) = 50

Table 12: Iterations of Simplex Method Applied to Example

CB_i	C_j	5	10	8	0	0	Solution	Ratio
	Basic variable	X_1	X_2	X_3	S_1	S_2		
0 0	S_1 S_2	2 1	5 4	1 2	1 0	0 1	10 12	2* 3
	Z_j	0	0	0	0	0	0	
	$C_j - Z_j$	5	10*	8	0	0		
10	X_2	2/5	1	1/5	1/5	0	2	10
0	S_2	-3/5	0	6/5	-4/5	1	4	10/3*
	Z_j	4	10	2	2	0	20	

	C_j - Z_j	1	0	6*	-2	0		
10 8	X_2 X_3	1/2 -1/2	1 0	0 1	1/3 -2/3	-1/6 5/6	4/3 10/3	8/3* -
	Z_j	1	10	8	-2	5	40	
	C_j - Z_j	4*	0	0	2	-5		
5 8	X_1 X_3	1 0	2 1	0 1	2/3 -1/3	-1/3 2/3	8/3 = 2 + 2/3 14/3 = 4 + 2/3	
	Z_j	5	18	8	2/3	11/3	152/3	
	C_j - Z_j	0	-8	0	-2/3	-11/3		

Table 13: Table after Appending Cut 1

CB_i	C_j	5	10	8	0	0	0	Solution
	Basic variable	X_1	X_2	X_3	S_1	S_2	S_3	
5 8 0	X_1 X_3 S_3	1 0 0	2 1 0	0 1 0	2/3 -1/3. 1-2/3	-1/3 2/3 -2/3	0 0 1	8/3 14/3 -2/3*
	Z_j	5	18	8	2/3	11/3	0	152/3

	C_j - Z_j	0	-8	0	-2/3	-11/3	0	

Table 14: Determination of Entering Variable

	X_1	X_2	X_3	S_1	S_2	S_3
-(*Cj* - *Zj*)	0	8	0	2/3	11/3	0
Row S_3	0	0	0	-2/3	-2/3	1
Ratio (absolute value)	-	-	-	1*	11/2	-

Table 15: Table after Pivot Operation

CB_i	C_j	5	10	8	0	0	0	Solution
	Basic variable	X_1	X_2	X_3	S_1	S_2	S_3	
5	X_1	1	2	0	0	-1	1	2
8	X_3	0	1	1	0	1	-1/2	5
0	S_1	0	0	0	1	1	-3/2	1
		5	18	8	0	3	1	50
Z_j C_j - Z_j		0	-8	0	0	-3	-1	

3.2 Branch and bound methods

It means that huge majority of the cases are dropped based on consequences obtained from the analysis of the particular numerical problem. The three most important enumerative methods are

(i) implicit enumeration

(ii) dynamic programming

(iii) branch and bound method.

Implicit enumeration and dynamic programming can be applied within the family of optimization problems mainly if all variables have discrete nature. Branch and bound method can easily handle problems having both discrete and continuous variables. Further the techniques of implicit enumeration can be incorporated easily in the branch and bound frame.

Branch and bound method can be applied even in some cases of nonlinear programming.

The Branch and Bound (abbreviated further on as B&B) method is just a frame of a large family of methods. Its sub-steps can be carried out in different ways depending on the particular problem, the available software tools and the skill of the designer of the algorithm.

Boldface letters denote vectors and matrices; calligraphic letters are used for sets. Components of vectors are denoted by the same but non-boldface letter. Capital letters are used for matrices and the same but lower case letters denote their elements. The columns of a matrix are denoted by the same boldface but lower case letters.

Some formulas with their numbers are repeated several times in this chapter. The reason is that always a complete description of optimization problems is provided. Thus the fact that the number of a formula is repeated means that the formula is identical to the previous one.

If the number of decision variables in an integer programming problem is only two, a **branch-and-bound** technique can be used to find its solution graphically. Various terminologies of branch-and-bound technique are explained as under:

Branching: If the solution to the linear programming problem contains non-integer values for some or all decision variables, then the solution space is reduced by introducing constraints with respect to any one of those decision variables.

If the value of a decision variable X_1 is 2.5, then two more problems will be created by using each of the following constraints.

$X_1 \leq 2$ and $X_1 \geq 3$

Lower bound: This is a limit to define a lower value for the objective function at each and every node. The **lower bound** at a node is the value of the objective function corresponding to the truncated values (integer parts) of the decision variables of the problem in that node.

Upper bound: This is a limit to define an upper value for the objective function at each and every node. The **upper bound** at a node is the value of the objective function corresponding to the linear programming solution in that node.

Fathomed subproblem/node: A problem is said to be **fathomed** if any one of the following three conditions is true:

1. The values of the decision variables of the problem are integer.

2. The upper bound of the problem which has non-integer values for its decision variables is not greater than the current best lower bound.

3. The problem has infeasible solution.

This means that further branching from this type of fathomed nodes is not necessary.

Current best lower bound: This is the best lower bound (highest in the case of maximization problem and lowest in the case of minimization problem) among the lower bounds of all the fathomed nodes. Initially, it is assumed as infinity for the root node.

Branch-and-bound algorithm applied to maximization problem

Step 1: Solve the given linear programming problem graphically. Set the current best lower bound, Z_B as ∞.

Step 2: Check, whether the problem has integer solution. If yes, print the current solution as the optimal solution and stop; otherwise go to step 3.

Step 3: Identify the variable X_k which has the maximum fractional part as the branching variable. (In case of tie, select the variable which has the highest objective function coefficient.)

Step 4: Create two more problems by including each of the following constraints to the current problem and solve them.

$X_k \leq$ Integer part of X_k

$X_k \geq$ Next integer of X_k

Step 5: If any one of the new subproblems has infeasible solution or fully integer values for the decision variables, the corresponding node is fathomed. If a new node has integer values for the decision variables, update the current best lower bound as the lower bound of that node.

Step 6: Are all terminal nodes fathomed? If the answer is yes, go to step 7; otherwise, identify the node with the highest lower bound and go to step 3.

Step 7: Select the solution of the problem with respect to the fathomed node whose lower bound is equal to the current best lower bound as the optimal solution.

Problems:

Let us solve the following integer programming problem using branch-and-bound technique.

Maximize $Z = 10X_1 + 20X_2$

subject to

$6X_1 + 8X_2 \leq 48$

$X_1 + 3X_2 \leq 12$

$X_1, X_2 \geq 0$ and integers

Solution

The introduction of the non-negative constraints $X_1 \geq 0$ and $X_2 \geq 0$ will eliminate the second, third and fourth quadrants of the X_1X_2 plane as shown in Figure.

Now, from the first constraint in equation form

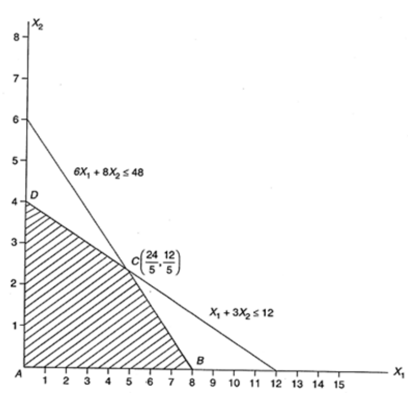
$6X_1 + 8X_2 = 48$

we get $X_2 = 6$, when $X_1 = 0$; and $X_1 = 8$, when $X_2 = 0$. Similarly from the second constraint in equation form

$X_1 + 3X_2 = 12$

we have $X_2 = 4$, when $X_1 = 0$; and $X_1 = 12$, when $X_2 = 0$.

Now, plot the constraints 1 and 2 as shown in Figure.



: Graphical representation for feasible region Example.

The closed polygon ABCD is the feasible region. The objective function value at each of the corner points of the closed polygon is computed as follows by substituting its coordinates in the objective function:

$Z(A) = 10 \times 0 + 20 \times 0 = 0$

$Z(B) = 10 \times 8 + 20 \times 0 = 80$

$$Z(C) = 10 \times \frac{24}{5} + 20 \times \frac{12}{5} = 96$$

$$Z(D) = 10 \times 0 + 20 \times 4 = 80$$

Since, the type of the objective function is maximization, the solution corresponding to the maximum Z value is to be selected as the optimum solution. The Z value is maximum for the comer point C. Hence, the corresponding solution of the continuous linear programming problem is presented below.

$$X_1 = \frac{24}{5},\ X_2 = \frac{12}{5},\ Z(\text{optimum}) = 96$$

These are jointly shown as problem P_1 in Figure. The notations for different types of lower bound are defined as follows:

Z_U = Upper bound = Z(optimum) of LP problem

Z_L = Lower bound w.r.t. the truncated values of the decision variables

Z_B = Current best lower bound

P_1

Problem	Solution
Maximize $Z = 10X_1 + 20X_2$ subject to $6X_1 + 8X_2 \leq 48$ $X_1 + 3X_2 \leq 12$ X_1 and $X_2 \geq 0$ and integers	$X_1 = 24/5$ $X_2 = 12/5$ $Z_U = 96$ $Z_L = 80$ $Z_B = \infty$

Solution of given linear programming problem.

Since both the values of X_1 and X_2 are not integers, the solution is not optimum from the view point of the given problem. So, the problem is to be modified into two problems by including integer constraints one by one.

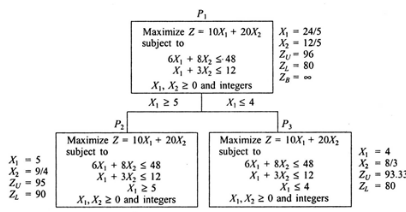
The lower bound of the solution of P_1 is 80. This is nothing but the value of the objective function for the truncated values of the decision variables.

The rule for selecting the variable for branching is explained as follows:

1.Select the variable which has the highest fractional part.

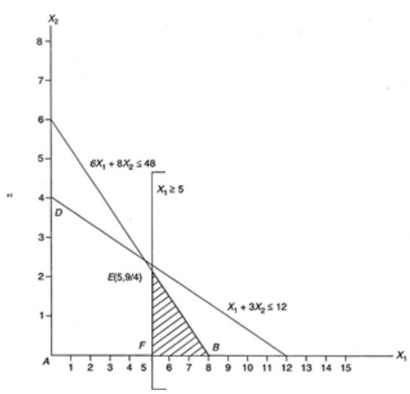
2.If there is a tie, then break the tie by choosing the variable which has the highest objective function coefficient.

In the continuous solution of the given linear programming problem P_1' the variable X_1 has the highest fractional part (4/5). Hence, this variable is selected for further branching as shown in figure.



Branching form P_1.

In Figure , the problems, P_2 and P_3 are generated by adding an additional constraint. The subproblem, P_2 is created by introducing '$X_1 \ge 5$' in problem P_1' and the problem P_3 is created by introducing '$X_1 \le 4$' in problem P_1. The corresponding effects in slicing the non-integer feasible region are shown in Figures respectively.

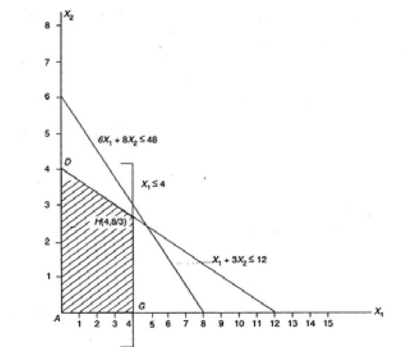


Feasible region of P_2 after introducing $X1 \ge 5$ to P_1.

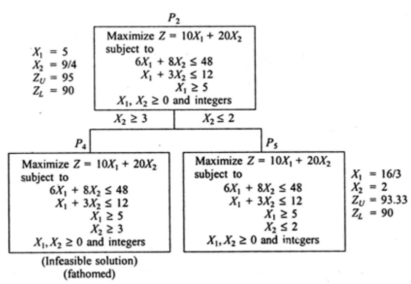
The solution for each of the subproblems, P_2 and P_3 is obtained from Figures , respectively. These are summarized in Figure . The problem P_2 has the highest lower bound of 90 among the unfathomed terminal nodes. So, the further branching is done from this node as shown in Figure.

In Figure, the problems, P_4 and P_5 are generated by adding an additional constraint to P_2. The problem, P_4 is created by including '$X_2 \ge 3$' in problem P_2, and problem P_5 is created by including ‘$X_2 \leq 2$’ in problem P_2. The corresponding effects in slicing the non-integer feasible region are shown in Figures respectively.

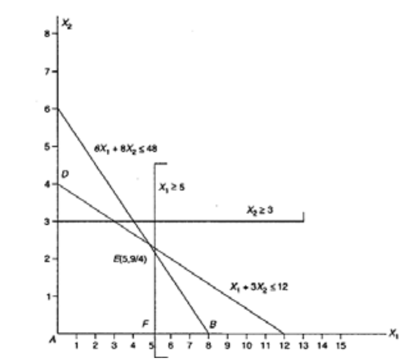
The solution for each of the problems P_4 and P_5 is obtained from Figures respectively. The problem P_4 has infeasible solution. So, this node is fathomed. The lower bound of the node P_5 is 90. But, the solution of the node P_5 is still non-integer.



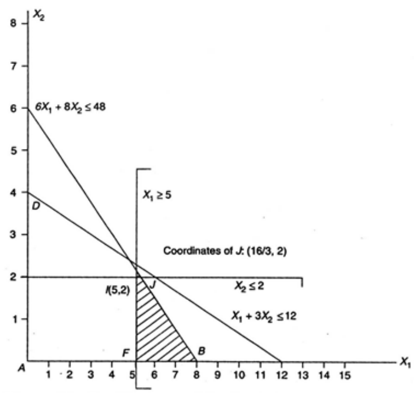
Feasible region of P_3 after introducing $X_1 \leqslant 4$ to P_1.



Branching from P_2.

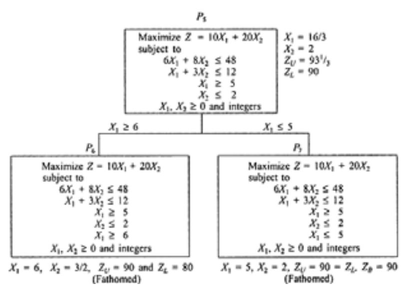


In feasible region of P_4 after introducing $X_2 \geqslant 3$ to P_2.

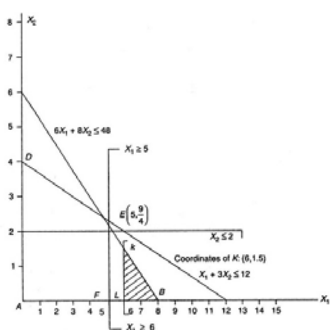


Feasible region of P_5 after introducing $X_2 \le 2$ to P_2

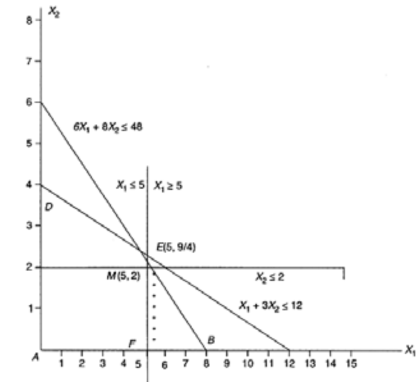
Now, the lower bound of the node P_5 is the maximum when compared to that of all other unfathomed terminal nodes (only P_3) at this stage. So, the further branching should be done from the node, P_5 as shown in Figure.



Branching from P_5.



Feasible region of P_6 after introducing $X1 \ge 6$ to P_5.



Feasible region of P_7 after introducing $X1 \le 5$ to P_5.

[Feasible region is the vertical line from M(5, 2) to F(5, 0) indicated by *s.]

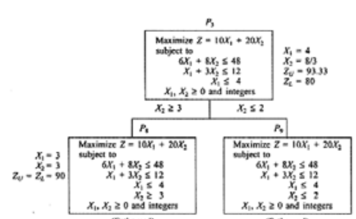
In Figure, the problems, P_6 and P_7 are generated by adding an additional constraint to P_5. The problem P_6 is created by including '$X_1 \ge 6$' in the problem P_5 and problem P_7 is created by including '$X_1 \le 5$' in problem P_5. The corresponding effects in slicing the non-integer feasible region are shown in Figure.

The solution for each of the problems P_6 and P_7 are also obtained from these figures, respectively. The problem P_7 has integer solution. So, it is a fathomed node. Hence, the current best lower bound (Z_B) is updated to its objective function value, 90.

The solution of the node P_6 is non-integer and its lower bound and upper bound are 80 and 90, respectively. Since, the upper bound of the node $P_6$is not greater than the current best lower bound of 90, the node P_6 is also fathomed and it has infeasible solution in terms of not fulfilling integer constraints for the decision variables.

Now, the only unfathomed terminal node is P_3. The further branching from this node is shown in Figure.

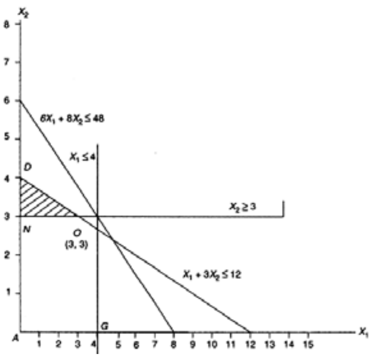
In Figure, the problems P_8 and P_9 are generated by adding an additional constraint to P_3. The problem P_8 is created by including '$X_2 \ge 3$' in problem P_3 and problem P_9 is created by including '$X_2 \le 2$' in problemP_3.



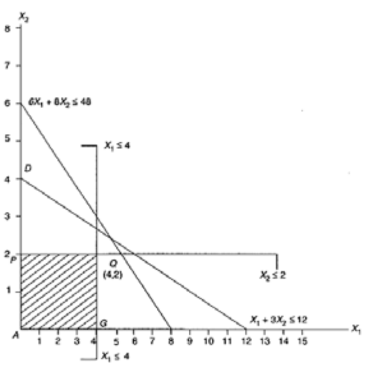
Branching from P_3.

The corresponding effects in slicing the non-integer infeasible region are shown in Figures, respectively. The solution for each of the problems P_8 and P_9 are obtained from the Figures, respectively. The problems P_8 and P_9 have integer solution. So, these two nodes are fathomed.

But the objective function value of these nodes are not greater than the current best lower bound of 90. Hence, the current best lower bound is not updated.



Feasible region of P_8 after introducing $X2 \geq 3$ to P_3.



Feasible region of P_9 after introducing $X2x \leq 2$ to P_3.

Now, all the terminal nodes are fathomed. The feasible fathomed node with the current best lower bound is P_7. Hence, its solution is treated as the optimal solution as listed below. A complete branching tree is shown in Figure.

$X_1 = 5$, $X_2 = 2$, Z(optimum) = 90

Note: This problem has alternate optimum solution at P_8 with $X_1 = 3$, $X_2 = 3$, Z(optimum) = 90.

3.3 Multistage (Dynamic) programming

Dynamic programming is a special kind of optimization technique which subdivides an original problem into as many number of subproblems as the variables, solves each subproblem individually and then obtains the solution of the original problem by integrating the solutions of the subproblems. It is a systematic, complete enumeration technique.

The terminologies used in the dynamic programming problem are presented below:

***Stage* i:** Each subproblem of the original problem is known as a **stage i.**

Alternative, m_i: In a given stage *i,* there may be more than one choice of carrying out a task. Each choice is known as an **alternative**, m_i.

State variable, x_i: A possible value of a resource within its permitted range at a given stage *i* is known as *state variable*, x_i.

Recursive function, $f_i(x_i)$ A function which links the measure of performance of interest of the current stage with the cumulative measure of performance of the previous stages/succeeding stages as a function of the state variable of the current stage is known as the *recursive function* of the current stage. Let

$f_1(x_1) = \max [R(m_1)]$

$f_i(x_i) = \max \{R(m_i) + f_{i-1}[xi - c(m_i)]\}, i = 2, 3, ..., n$

for possible m_i where, n is the total number of stages, $R(m_i)$ is the measure of performance (like, return) due to alternative m_i of the stage *i,* $c(m_i)$ is the cost/resource required for the alternative m_i of the stage *i* and $f_i(x_i)$ is the value of the measure of performance up to the current stage *i* from the stage 1, if the amount of resource allocated up to the current stage is x_i when forward recursion is used.

Best recursive function value: In a given stage *i,* the lowest (minimization problems)/highest (maximization problems) value of the recursive function for a given value of x_i is known as the *best recursive function* value.

Best alternative in a given stage i:

In a given stage **i,** the alternative corresponding to the best recursive function value for a given value of x_i is known as the *best alternative* for that value of x_i.

Backward recursive function:

Here computation begins from the last stage or subproblem, and this stage will be numbered as stage 1, while the first subproblem will be numbered as the last stage. Since the recursion proceeds in a backward direction, this type of recursive function is known as *backward recursive function.*

Forward recursive function:

While defining the stages of the original problem, the first sub-problem will be numbered as stage 1 and last subproblem will be numbered as the last stage. Then, the recursive function will be defined as per this assumption. This type of recursive function is known as *forward recursive function.*

APPLICATION OF DYNAMIC PROGRAMMING

The dynamic programming can be applied to many real-life situations. A sample list of applications of the dynamic programming is given below. The details of these problems are explained while solving them.

1.Capital budgeting problem

2.Reliability improvement problem

3.Stage-coach problem (shortest-path problem)

4.Cargo loading problem

5.Minimizing total tardiness in single machine scheduling problem

6.Optimal subdividing problem

7.Linear programming problem

They are discussed in the following sections:

1. Capital Budgeting Problem

A capital budgeting problem is a problem in which a given amount of capital is allocated to a set of plants by selecting the most promising alternative for each selected plant such that the total revenue of the organization is maximized. This is demonstrated using a numerical problem.

1.Example: An organization is planning to diversify its business with a maximum outlay of Rs. 5 crores. It has identified three different locations to install plants. The organization can invest in one or more of these plants subject to the availability of the fund.

The different possible alternatives and their investment (in crores of rupees) and present worth of returns during the useful life (in crores of rupees) of each of these plants are summarized in Table 1.

The first row of Table1 has zero cost and zero return for all the plants. Hence, it is known as **do-nothing** alternative. Find the optimal allocation of the capital to different plants which will maximize the corresponding sum of the present worth of returns.

Table 1: Example

Alternative	Plant 1		Plant 2		Plant 3	
	Cost	Return	Cost	Return	Cost	Return
1	0	0	0	0	0	0
2	1	15	2	14	1	3
3	2	18	3	18	2	7
4	4	28	4	21	-	-

Solution:

Maximum capital amount, C = Rs. 5 crores. Each plant is treated as a stage. So, the number of stages is equal to 3. The plants 1, 2 and 3 are defined as stage 1, stage 2 and stage 3, respectively. So, the forward recursive function is used for this problem.

Stage 1: The recursive function for a given combination of the state variable, x_1 and alternative, m_1 in the first stage is presented below. The corresponding returns are summarized in Table 2. For each value of the state variable, the best return and the corresponding alternative are presented in the last two columns, respectively.

$f_1(x_1) = R(m_1)$

Table 2: Calculations for Stage 1 (Plant 1)

State variable	**Alternative m_1**				$f_1^*(x_1)$	m_1^*
	1	2	3	4		

	C	*R*	*C*	*R*	*C*	*R*	*C*	*R*		
	0	0	1	15	2	18	4	28		
0	0		-		-		-		0	1
1	0		15		-		-		15	2
2	0		15		18		-		18	3
3	0		15		18		-		18	3
4	0		15		18		28		28	4
5	0		15		18		28		28	4

C = cost, *R* = return.

Stage 2: The recursive function $f_2(x_2)$ for a given combination of the state variable, x_2 and alternative, m_2 in the second stage is given by

$$f_2(x_2) = R(m_2) + f_1\,[x_2 - C(m_2)]$$

The corresponding returns are summarized in Table 3. For each value of the state variable, the best return and the corresponding alternative(s) are presented in the last two columns, respectively.

Table 3: Calculations for Stage 2 (Plant 2)

	Alternative m_2									
State variable	**1**		**2**		**3**		**4**			
	C	*R*	*C*	*R*	*C*	*R*	*C*	*R*		

x_2	0	0	2	14	3	18	4	21	$f_2(x_2)^*$	m_2^*
0	0 0		-		-		-		0	1
1	0 + 15 = 15		-		-		-		15	1
2	0 + 18 = 18		14 + 0 =14		-		-		18	1
3	0 + 18= 18		14 + 15 = 29		18 + 0=18		-		29	2
4	0 + 28 = 28		14 + 18 = 32		18 + 15 = 33		21 + 0 = 21		33	3
5	0 + 28 = 28		14 + 18 = 32		18 + 18 = 36		21 + 15 = 36		36	3 and 4

Stage 3: The recursive function $f_3(x_3)$ for different combinations of the state variable, x_3 and alternative, m_3 in the third stage is:

$f_3(x_3) = R(m_3) + f_2[x_3 - C(m_3)]$

The corresponding returns are summarized in Table 4. For each value of the state variable, the best return and the corresponding alternative(s) are presented in the last two columns, respectively.

Table 4: Calculations for Stage 3 (Plant 3)

	Alternative m_3							
State variable	**1**		**2**		**3**			
	C	*R*	*C*	*R*	*C*	*R*		
x_3	0	0	1	3	2	7	$f_3(x_3)^*$	m_3^*
5	0 + 36 = 36 3 + 33 = 36 7 + 29 = 36						36	1,2 and 3

The final results of the original problem is traced as in Table 5. From this table, one can visualize the fact that the original problem has four alternate optimal solutions.

Table 5: Final Results

Stage 3		**Stage 2**		**Stage 1**		**Optimal alternatives stage**		
C*	m_3*	**C***	m_2*	C*	m_1*	**1**	**2**	**3**
5	1	5 - 0 = 5 $\begin{cases}3\\4\end{cases}$		5 - 3 = 2	3	3 - 3 - 1		
				5 – 4 = 1	2	2 - 4 - 1		
5	2	5 - 1= 4	3	4 - 3 = 1	2	2 - 3 - 2		
5	3	5 - 2= 3	2	3 - 2 = 1	2	2 - 2 - 3		

2. Reliability Improvement Problem

Generally, electronic equipments are made up of several components in series or parallel. Assuming that the components are connected in series, if there is a failure of a component in the series, it will make the equipment inoperative.

The reliability of the equipment can be increased by providing optimal number of standby units to each of the components in the series such that the total reliability of the equipment is maximized subject to a cost constraint. Application of dynamic programming technique to this problem is illustrated in Example below.

2.Example : An electronic item has three components in series. (The reliability of the system is equal to the product of the reliabilities of the three components, i.e. $R = r_1r_2r_3$. It is a known fact that the reliability of the system can be improved by providing standby units at extra cost.) The details of costs and reliabilities for different number of standby units for each of the components of the system are summarized in Table 6.

Table 6: Example

No. of standby units	**Component 1**	**Component 2**	**Component 3**

	Cost (Rs.)	Reliability	Cost (Rs.)	Reliability	Cost (Rs.)	Reliability
1	1	0.75	3	0.84	2	0.80
2	2	0.88	4	0.94	3	0.91
3	4	0.94	6	0.97	5	0.96

The total capital budgeted for this purpose is Rs. 9. Let us Determine the optimal number of standby units for each of the components of the system such that the total reliability of the system is maximized.

Solution

Maximum capital budgeted/unit, K = Rs. 9. Each component is treated as a stage.

So, the number of stages is equal to 3. The components 3, 2 and 1 are defined as stage 1, stage 2 and stage 3, respectively. Hence, the backward recursive function is used for this problem.

Range for state variable x_1 at Stage 1:

The minimum amount of money required to have at least one standby unit for Component 3 is Rs. 2. Therefore, the lower limit of the state variable x_1 = Rs. 2. Similarly, a sum of Rs. 4 (Rs. 3 + Re. 1) is required to have at least one standby unit in Stage 2 and Stage 3 (i.e. the sum of the cost of one standby unit for each of the Component 2 and Component 1).Therefore, the upper limit of the state variable x_1 9 - 4 = Rs. 5.

Based on these guidelines, the effective range of the state variable x_1 is: $2 \leq x_1 \leq 5$

Range for state variable x_2 at Stage 2:

The minimum amount of money required to have at least one standby unit in each of the stages up to the current stage is Rs. 5 (the sum of the cost of one standby unit for each of the Component 3 and Component 2). Therefore, the lower limit of the state variable x_2 = Rs. 5. Similarly, the amount required to have at least one standby unit in stage 3 is Re. 1 (i.e. the cost of one standby unit for Component 1). Therefore, the upper limit of the state variable x_2 = 9 - 1 = Rs. 8.

Based on these guidelines, the effective range of the state variable x_2 is: $5 \leq x_2 \leq 8$

Range for state variable x_3 at Stage 3:

The minimum amount of money required to have at least one standby unit in each of the stages up to the current stage is Rs. 6 (the sum of the cost of one standby unit for each of the Component 3, Component 2 and Component 1). Therefore, the lower limit of the state variable x_3 = Rs. 6. Also, the upper limit of the state variable x_3 = Rs. 9. The effective range of the state variable x_3 is:

$6 \leq x_3 \leq 9$

Stage 1: The recursive function $f_1(x_1)$ for a given combination of the state variable, x_1 and alternative, m_1 in the first stage is:

$f_1(x_1) = r(m_1)$

The corresponding reliabilities are shown in Table 7. For each value of the state variable x_1, the best reliability and the corresponding alternative are presented in the last two columns respectively.

Table 7: Calculations for Stage 1 (Component 3)

State Variable x_1	**Alternative m_1**						**$f_1(x_1)^*m_1^*$**	
	1		2		3			
	C	*R*	*C*	*R*	*C*	*R*		
	2	0.8	3	0.91	5	0.96		
2	0.8		-		-		0.8	1
3	0.8		0.91		-		0.91	2
4	0.8		0.91		-		0.91	2
5	0.8		0.91		0.96		0.96	3

C = cost, *R* = reliability.

Stage 2: The recursive function $f_2(x_2)$ for a given combination of the state variable, x_2 and alternative, m_2 in the second stage is:

$f_2(x_2) = r(m_2) \times f_1[x_2 - C(m_2)]$

The corresponding reliabilities are summarized in Table 8. For each value of the state variable, the best reliability and the corresponding alternative are presented in the last two columns respectively.

Alternative m_2								
State variable x_2	1		2		3		$f_2(x_2)^*$	m_2*
	C	R	C	R	C	R		
	3	0.84	4	0.94	6	0.97		
5	0.84 ×0.8 = 0.672		-		-		0.672	1
6	0.84 × 0.91 = 0.764		0.94 × 0.8 = 0.752		-		0.764	1
7	0.84 × 0.91 = 0.764		0.94 × 0.91 = 0.855		-		0.855	2
8	0.84 × 0.96 = 0.806		0.94 × 0.91 = 0.855		0.97 × 0.8 = 0.776		0.855	2

Table 8: Calculations for Stage 2 (Component 2)

Stage 3: The recursive function for a given combination of the state variable, x_3 and alternative, m_3 in the third stage is presented below:

$f_3(x_3) = r(m3) \times f_2[x_3 - C(m_3)]$

The corresponding reliabilities are summarized in Table 9. For each value of the state variable, the best reliability and the corresponding alternative are presented in the last two columns, respectively.

Table 9: Calculations for Stage 3 (Component 1)

Alternative m3								
State variable$x3$	**1**		**2**		**3**		***f*3(x_3)***	**m_3***
	C	*R*	*C*	*R*	*C*	*R*		
	1	0.75	2	0.88	4	0.94		
6	0.75 × 0.672 = 0.504		-		-		0.504	1
7	0.75 × 0.764 = 0.573		0.88 × 0.672 = 0.591		-		0.591	2
8	0.75 × 0.855 = 0.641		0.88 × 0.764 = 0.672		-		0.672	2
9	0.75 × 0.855 = 0.641		0.88 × 0.855 = 0.752		0.94 × 0.672 = 0.632		0.752	2

The final result of the original problem is traced as shown in Table 10. From the table it is clear that each of the three components requires two standby units to have a maximum reliability of 0.752 with an additional total cost of Rs. 9.

Table 10: Final Results

Stage								
3		2		1		Stage		
*C**	m_3^*	*C**	m_2^*	m_1^*		1	2	3
9	2	9 – 2 = 7	2	7 - 4 = 3	2	$m_1^* = 2$	$m_2^* = 2$	$m_3^* = 2$

3. Cargo Loading Problem

Cargo loading problem is an optimization problem in which a logistic company is left with the option of loading a desirable combination of items in a cargo subject to its weight or volume or both constraints. In this process, the return to the company is to be maximized. The application of dynamic programming to the cargo loading problem which has only the weight constraint is illustrated.

Problems:

Alpha logistic company has to load a cargo out of four items whose details are shown in Table 15. The maximum weight of the cargo is 7 tons. Let us Find the optimal cargo loading using dynamic programming method such that the total return is maximized.

Table 15: Data of Example

Item *i*	1	2	3	4
Weight, w_i/unit (in tons)	2	1	4	3
Return, r_i/unit (in rupees)	1000	400	2100	1400

Solution:

In this problem, each item is treated as a stage starting from stage 1 to stage 4 for the item 1 to item 4 respectively. The maximum weight of the cargo is 7 tons. So, the weights allocated to the alternatives in each of the stages are zero and the multiples of the unit weight (less than or equal to 7) of the item corresponding to that stage. In stage 1, there are four alternatives and the weights allocated to those alternatives are 0, 2, 4 and 6.

In stage 2, there are ten alternatives and the weights allocated to those alternatives are 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.The stage 3 has two alternatives and the weights allocated to them are 0 and 4.The stage 4 has three alternatives and the weights allocated to those alternatives are 0, 3 and 6. The range of values of the state variable in each stage is from 0 to 7 with an increment of 1.

Stage 1: The recursive function of this stage is shown below and the calculations are presented in Table 16.

f_1(xi) $^=$ (Allocated weight/2) × 1000

Table 16: Calculations for Stage 1

	Alternative m_1			

State variable	1	2	3	4		
	Allocated weight					
x_1	0	2	4	6	f_1*	m_1*
0	0	-	-	-	0	1
1	0	-	-	-	0	1
2	0	1000	-	-	1000	2
3	0	1000	-	-	1000	2
4	0	1000	2000	-	2000	3
5	0	1000	2000	-	2000	3
6	0	1000	2000	3000	3000	4
7	0	1000	2000	3000	3000	4

Stage 2: The recursive function of the stage is shown below and the calculations are presented in Table 17.

$f_2(x_2)$ = Allocated weight 400 + $f_1(x_2$ - Allocated weight)

	Alternative m_2									
State variable x_2	1	2	3	4	5	6	7	8		

	Allocated weight									
	0	1	2	3	4	5	6	7	f_2*	m_2
0	0	-	-	-	-	-	-	-	0	1
1	0+0=0	400	-	-	-	-	-	-	400	2
2	0+1000= 1000	400 + 0 =400	-	-	-	-	-	-	1000	1
3	0+1000 = 1000	400+1000 = 1400	800 + 0 =800	-	-	-	-	-	1400	2
4	0+2000 =2000	400+1000 = 1400	800+1000 = 1800	1200+0= 1200	1600	-	-	-	2000	1
5	0+2000 =2000	400+2000 =2400	800+1000 = 1800	1200+1000 =2200	1600+0 = 1600	2000	-	-	2400	2
6	0+3000 =3000	400+2000 =2400	800+2000 =2800	1200+1000 =2200	1600+1000 =2600	2000+0 =2000	2400	-	3000	1
7	0+3000 =3000	400+3000 =3400	800+2000 =2800	1200+2000 =3200	1600+1000 =2600	2000+1000 =3000	2400+0 =2400	2800	3400	2

Table 17: Calculations for Stage 2

Stage 3: The recursive function of this stage is shown below and the calculations are presented in Table 18.

$f_3(x_3)$ = (Allocated weight/4) × 2100 $+f_2(x_3$ - Allocated weight)

Table 18: Calculations for Stage 3

	Alternative m_3		f_3*	m_3*
State variable x_3	1	2		
	Allocated weight			
	0	4		
0	0	-	0	1
1	0+ 400 = 400	-	400	1
2	0+ 1000 = 1000	-	1000	1
3	0+1400 = 1400	-	1400	1
4	0 + 2000 = 2000	2100	2100	2
5	0 + 2400 = 2400	2100+ 400 = 2500	2500	2
6	0 + 3000 = 3000	2100+ 1000 = 3100	3100	2
7	0 + 3400 = 3400	2100+ 1400 = 3500	3500	2

Stage 4: The recursive function of this stage is shown below and the calculations are presented in Table 19.

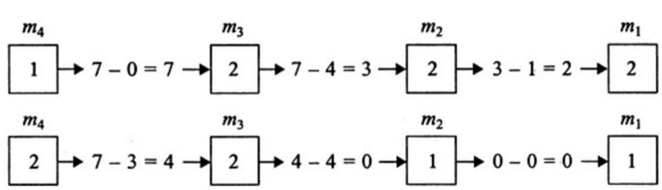
$f_4(x_4)$ = (Allocated weight/3) × 1400 +$f_3(x_4$ - Allocated weight)

Table 19: Calculations for Stage 4

State variablex_4	Alternative m_4		

	1	2	3		
	Allocated weight				
	0	3	6	f_4*	m_4*
0	0	-	-	0	1
1	0 + 400 = 400	-	-	400	1
2	0+ 1000= 1000	-	-	1000	1
3	0+ 1400= 1400	1400	-	1400	1,2
4	0 + 2100 = 2100	1400+ 400= 1800	-	2100	1
5	0 + 2500 = 2500	1400+ 1000 = 2400	-	2500	1
6	0 + 3100 = 3100	1400+ 1400 = 2800	2800	3100	1
7	0 + 3500 = 3500	1400 + 2100 = 3500	2800 + 400 = 3200	3500	1,2

Tracing the solution from the stage 4 gives two solutions as shown below and each of them gives the optimal return of Rs. 3500.



The details of the weights of different items in the cargo are summarized in Table 20.

Table 20: Summary of Weights of Items in the Cargo (Two Solutions)

	Weight (tons)				
Solution number	**Item 1**	**Item 2**	**Item 3**	**Item 4**	**Total return (Rs.)**
1	2	4	4	0	3500
2	0	0	4	3	3500

Solution of Linear Programming Problem through Dynamic Programming

Let us take the generalized linear programming problem for the product-mix problem in which X_j is the volume of production of the product *j.*

Maximize Z = $c_1X_1 + C_2X_2 + ... + C_jX_j + ... + c_nX_n$

subject to

$a_{11}X_1 + a_{12}X_2 + ... + a_{1j}X_j + ... + a_{1n}X_n \leq b_1$

$a2_1X_1 + a_{22}X_2 + ... + a_{2j}X_j + ... + a_{2n}X_n \leq b_2$

. . . . .

. . . . .

. . . . .

$a_{i1}X_1 + a_{i2}X_2 + ... + a_{ij}X_j + ... + a_{in}X_n \leq b_i$

. . . . .

. . . . .

. . . . .

$a_{m1}X_1 + a_{m2}X_2 + ... + a_{mj}X_j + ... + a_{mn}X_n \leq b_m$

$Xj \geq 0$, $j = 1, 2, 3, ..., n$

In this linear programming problem, the product *j* is treated as stage *j*, where *j* varies from 1 to *n.* So, the total number of stages of the problem is equal to *n.* At stage *j*, a value (production volume) of the decision variable X_j is known as an *alternative.*

Since, in the LPP, X_j is a continuous variable, for *j* varying from 1 to *n*, there will be infinite number of alternatives in each of the stages.

The method of backward recursion is used to solve this problem. Product 1 is treated as Stage 1, Product 2 is treated as stage 2 and so on. Since the backward recursion is to be used, the working of the problem will commence from stage *n.*

Let us suppose that b_{ij} be the state of the system with respect to Constraint *i* in Stage *j.* (i.e. the amount of resource *i* allocated to the activities of the current stage and its succeeding stages). So, b_{1j}, b_{2j}, b_{3j},..., b_{ij}, and b_{mj} are the states of the system at stage *j* with respect to the resources 1, 2, 3,..., *i*, and *m* respectively. $f_j(b_{1j}, b_{2j}, b_{3j},..., b_{ij},..., b_{mj})$be the optimum objective function value at stage *j.*

Now, the objective function for stage *n* is,

$$f_n(b_{1n}, b_{2n}, b_{3n},..., b_{in},..., b_{mn}) = \max_{\substack{0 \le a_{in} X_n \le b_{in} \\ \text{for } i = 1,2,...,m}} (c_n X_n)$$

and the objective function for stage *j* is,

$$f_j(b_{1j}, b_{2j}, b_{3j},..., b_{ij},..., b_{mj}) = \max_{\substack{0 \le a_{ij} X_j \le b_{ij} \\ \text{for } i = 1,2,...,m \\ \text{and } j = 1,2,...,n-1}} \begin{bmatrix} c_j X_j + f_{j+1}[(b_{1j} - a_{1j} X_j), (b_{2j} - a_{2j} X_j),..., \\ (b_{ij} - a_{ij} X_j),..., (b_{mj} - a_{mj} X_j)] \end{bmatrix}$$

with

$0 \le b_{ij} \le b_i$, i=l,2,..., *m* and *j* = 1, 2,..., *n*

In the above function, the quantity (b_{ij} - $a_{ij}X_j$) is the sum of the resource *i* allocated to all the succeeding stages (i.e. from stage *j* + 1 through stage *n*) with respect to the current stage.

Let us solve the following LPP using dynamic programming technique:

Maximize Z = 10X_1 + 30X_2

subject to

3X_1 + 6X_2 ≤ 168

12X_2 ≤ 240

X_1 and X_2 ≥ 0

Solution

The number of decision variables in the given problem is equal to 2. So, there will be two stages (i.e. stage 1 is assigned to the decision variable $X_1$and stage 2 is assigned to the decision variable X_2). Since backward recursion is used to solve the problem, stage 2 is to be considered first. The sets of states of different stages are summarized in Table 21.

Table 21: Sets of States of Different Stages

Stage *j*	Decision variable	Set of states
2	X_2	$\{b_{12}, b_{22}\}$
1	X_1	$\{b_{11}\ b_{21}\}$

Recursive function for stage 2 with respect to X_2 is based on the backward recursion. Therefore,

$$f_2(b_{12}, b_{22}) = \max_{\substack{0 \le 6X_2 \le b_{12} \\ 0 \le 12X_2 \le b_{22}}} 30X_2$$

To maintain feasibility, X_2 should be the minimum of $b_{12}/6$ and $b_{22}/12$, the above objective function is modified as follows:

$$f_2(b_{12}, b_{22}) = 30 \min\left(\frac{b_{12}}{6}, \frac{b_{22}}{12}\right) \quad \text{and} \quad X_2^* = \min\left(\frac{b_{12}}{6}, \frac{b_{22}}{12}\right)$$

Recursive function for stage 1 with respect to X_1 is as follows,

$$\begin{aligned} f_1(b_{11}, b_{21}) &= \max_{0 \le 3X_1 \le b_{11}} [10X_1 + f_2(b_{11} - 3X_1, b_{21})] \\ &= \max_{0 \le 3X_1 \le b_{11}} \left[10X_1 + 30 \min\left(\frac{b_{11} - 3X_1}{6}, \frac{b_{21}}{12}\right)\right] \end{aligned}$$

The stage 1 is the last in the series of backward recursion. Therefore, b_{11}= 168, and b_{21} = 240.

To determine the upper limit for X^*_1, we have

$$f_1\left(\frac{X_1}{b_{11}}, b_{21}\right) = \max\left[f_1\left(\frac{X_1}{168}, 240\right)\right]$$

$$= \max\left[10X_1 + 30 \min\left(\frac{168 - 3X_1}{6}, \frac{240}{12}\right)\right]$$

Now, to which the ranges of X_1 is defined, $(168 - 3X_1)/6$ can be as high as 20 or as low as 0. So, equate it to 20 as well as to 0 and solve for X_1 as explained below:

$$\frac{168 - 3X_1}{6} = 20, \quad \text{or} \quad X_1 = 16$$

$$\frac{168 - 3X_1}{6} = 0, \quad \text{or} \quad X_1 = 56$$

The ranges for X_1 are follows:

$0 \le X_1 \le 16$ and $16 \le X_1 \le 56$

Now, $f_1(X_1/b_{11}, b_{21})$ is rewritten as:

$$f_1\left(\frac{X_1}{b_{11}}, b_{21}\right) = \max\begin{bmatrix} 10X_1 + 30\min\left(\frac{168 - 3X_1}{6}, 20\right), & 0 \le X_1 \le 16 \\ 10X_1 + 30\min\left(\frac{168 - 3X_1}{6}, 20\right), & 16 \le X_1 \le 56 \end{bmatrix}$$

$$f_1\left(\frac{X_1}{168}, 240\right) = \max\begin{bmatrix} 10X_1 + 30 \times 20 & 0 \le X_1 \le 16 \\ 10X_1 + 30 \times \frac{168 - 3X_1}{6}, & 16 \le X_1 \le 56 \end{bmatrix}$$

$$= \max\begin{bmatrix} 10X_1 + 600, & 0 \le X_1 \le 16 \\ 10X_1 + 840 - 15X_1, & 16 \le X_1 \le 56 \end{bmatrix}$$

$$= \max\begin{bmatrix} 10X_1 + 600, & 0 \le X_1 \le 16 \\ 10X_1 + 840 - 15X_1, & 16 \le X_1 \le 56 \end{bmatrix}$$

To maximize each of the above cases, substitute 16 for X_1. Now, we get

$$f_1\left(\frac{X_1}{168}, 240\right) = \max(760, 760) = 760$$

Therefore,

$$X_1^* = 16 \quad \text{and} \quad f_1\left(\frac{X_1}{168}, 240\right) = 760$$

For tracing the value of X_2^* we have,

$b_{12} = b_{11} - 3X_1 = 168 - 3 \times 16 = 120$

$b_{22} = b_{21} - 0 = 240 - 0 = 240$

Therefore,

$$X_2^* = \min\left(\frac{b_{12}}{6}, \frac{b_{22}}{12}\right) = \min\left(\frac{120}{6}, \frac{240}{12}\right) = \min(20, 20) = 20$$

The optimal results are:

X_1*=16, X_2*=20,Z(OPTIMUM)=760

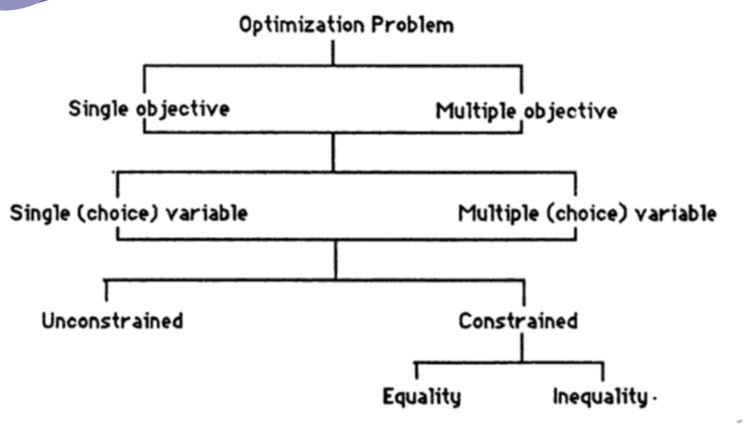
CLASSICAL OPTIMISATION THEORY

Unconstrained external problems, Newton – Ralphson method – Equality constraints – Jacobean methods – Lagrangian method – Kuhn – Tucker conditions – Simple problems.

4.1 Unconstrained external problems

The optimization problem facing a decision-making unit is further classified into unconstrained and constrained problems. In the former, the decision-maker optimizes subject to no constraints, internal or external. In the latter, it has one or more constraints (also called side conditions), imposed either, by itself (internal) or by outside agencies (external) such as government and or market conditions. An example of unconstrained optimization would be one where the firm aims at the maximum possible profit subject to no constraints of any kind. In contrast, under constrained optimization, the firm aims at maximum possible profit subject to one or more constraints such as a fixed production cost budget, a fixed quantity of output to be produced availability of scarce raw-materials in fixed quantity, employment of minimum number of unskilled labour, etc.

The constrained optimization problems are further classified into equality and inequality constrained problems. For example, a profit-maximizing firm might be required to produce a specific quantity of output of all of its multiple products, or if it is a single product firm, it might he faced with a fixed production cost budget or a fixed quantity of a particular scarce raw-material. Under these- conditions, the optimizing firm must strictly adhere to the given number of the constrained variable. An example of optimization subject to inequality constraints would be the one where the firm is seeking maximum possible profits but there is a fixed quantity of skilled labour available in the market; the firm can employ all the skilled labour, any quantity less than it but no more. The various types of optimization problems can, thus, be represented as follows:



• Unconstrained optimization problem:

$\min_x F(x)$ or $\max_x F(x)$

• Constrained optimization problem :

$\min\limits_X$ F (x) or $\max\limits_X$ F(x)

Subjected to g(x) =0

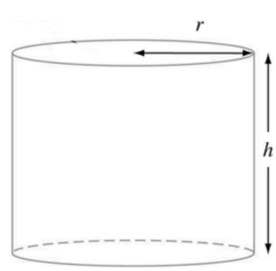
And h(x) <0 or h(x) >0

Example: Minimize the outer area of a cylinder subject to a fixed volume.

Objective function

$F(x) = 2\pi r^2 + 2\pi rh$, $x = \begin{bmatrix} r \\ h \end{bmatrix}$

Constraint: $2\pi r^2 h = V$



Part I: One-dimensional unconstrained optimization consists of

•Analytical method

•Newton's method

•Golden-section search method

Part II: Multidimensional unconstrained optimization consists of

•Analytical method

•Gradient method

•Steepest ascent (descent) method

•Newton's method

1. One-dimensional unconstrained optimization

(i) Analytical approach (1-D)

$\min_x F(x)$ or $\max_x F(x)$

Let $F'(x)=0$ and find $x=x^*$

If $F''(x)>0$, $F(x^*)=\min_x F(x)$, x^* is local minimum of $F(x)$;

If $F''(x^*)<0$, $F(x^*)=\max_x F(x)$, x^* is local maximum of $F(x)$;

If $F''(x^*)=0$, x^* is a critical point of $F(x)$

Example1: $F(x)=x^2$, $F'(x)=2x=0$, $x^*=0$. $F''(x^*)=2>0$.

Therefore,

Example 2: $F(x)=x^3$, $F'(x)=3x^2=0$, $x^*=0$. $F''(x^*)=0$. x^*

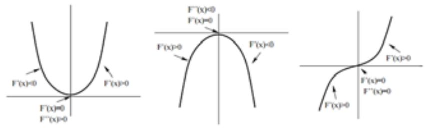
is not a local minimum nor a local maximum.

Example 3: $F(x)=x^4$, $F'(x)=4x^3=0$, $x^*=0$. $F''(x^*)=0$.

In example 2, $F'(x)>0$ when $x<x^*$ and $F'(x)>0$ when $x>x^*$.

In example 3, x^* is a local minimum of $F(x)$.

$F'(x)<0$ when $x<x^*$ and $F'(x)>0$ when $x>x^*$.



Example of constrained Optimization problem

2. Multidimensional Unconstrained Optimization

(i) Analytical Method

Definitions:

If f(x,y)< f(a,b) for all (x,y) near (a,b),f(a,b) is a local maximum.

If f(x,y)> f(a,b) for all (x,y) near by (a,b),f(a,b) is a local minimum.

If f(x,y) has a local maximum or minimum at (a,b) and the first order partial derivative of f(x,y) exist at (a,b), then,

$$\frac{\partial f}{\partial x}|_{(a,b)} = 0, \text{ and } \frac{\partial f}{\partial y}|_{(a,b)} = 0$$

- If
$$\frac{\partial f}{\partial x}|_{(a,b)} = 0 \text{ and } \frac{\partial f}{\partial y}|_{(a,b)} = 0,$$
then (a, b) is a critical point or stationary point of $f(x, y)$.
- If
$$\frac{\partial f}{\partial x}|_{(a,b)} = 0 \text{ and } \frac{\partial f}{\partial y}|_{(a,b)} = 0$$

And the second order partial derivatives of f(x,y) are continuous then

When $|H| > 0$ and $\frac{\partial^2 f}{\partial x^2}|_{(a,b)}$ < 0, f(a,b) is a local maximum of f(x,y)

When $|H| > 0$ and $\frac{\partial^2 f}{\partial x^2}|_{(a,b)}$ > 0, f(a,b) is a local minimum of f(x,y)

When $|H| < 0$, $f(a,b)$ is a saddle point

Hessian of f(x,y):

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

- $|H| = \frac{\partial^2 f}{\partial x^2} \cdot \frac{\partial^2 f}{\partial y^2} - \frac{\partial^2 f}{\partial x \partial y} \cdot \frac{\partial^2 f}{\partial y \partial x}$
- When $\frac{\partial^2 f}{\partial x \partial y}$ is continuous, $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$.
- When $|H| > 0$, $\frac{\partial^2 f}{\partial x^2} \cdot \frac{\partial^2 f}{\partial y^2} > 0$.

Example (saddle point): $f(x, y) = x^2 - y^2$.
$\frac{\partial f}{\partial x} = 2x$, $\frac{\partial f}{\partial y} = -2y$.
Let $\frac{\partial f}{\partial x} = 0$, then $x^* = 0$. Let $\frac{\partial f}{\partial y} = 0$, then $y^* = 0$.

Therefore ,(0,0) is a critical point

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}(2x) = 2, \frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial y}(-2y) = -2$$
$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x}(-2y) = 0, \frac{\partial^2 f}{\partial y \partial x} = \frac{\partial}{\partial y}(2x) = 0$$
$$|H| = \frac{\partial^2 f}{\partial x^2} \cdot \frac{\partial^2 f}{\partial y^2} - \frac{\partial^2 f}{\partial x \partial y} \cdot \frac{\partial^2 f}{\partial y \partial x} = -4 < 0$$

Therefore ,(x*,y*) = (0,0) is a saddle maximum

4.2 Newton – Raphson method

The Newton-Raphson method, or Newton Method, is a powerful technique for solving equations numerically. Like so much of the dierential calculus, it is based on the simple idea of linear approximation

The Newton-Raphson technique requires only one inital value x_0, which we will refer to as the initial guess for the root. To see how the Newton-Raphson method works, we can be rewrite the function f(x) using a Taylor series expansion in $(x-x_0)$:

$f(x) = f(x_0) + f'(x_0)(x-x_0) + 1/2\, f''(x_0)(x-x_0)^2 + ... = 0$ (1)

Here f'(x) denotes the first derivative of f(x) with respect to x, f''(x) is the second derivative, and so 4^{th} . Suppose the initial guess is pretty close to the real root. Then $(x-x_0)$ is small, & only the first few terms in the series are important to get an accurate estimate of true root, given x_0. By truncating the series at the 2^{nd} term , we obtain the Newton Raphson iteration formula for getting a better estimate of the true root:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Thus the Newton-Raphson method finds the tangent to the function f(x) at $x=x_0$ & extrapolates it to intersect the x axis to get x_1. This point of intersection is taken as the new approximation to the root and the procedure is repeated until convergence is obtained whenever possible. Mathematically, given the value of $x = x_i$ at the end of the ith iteration, we obtain x_{i+1} as

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

Assume that the derivative does not vanish for any of the x \$_{k,\ k=0,1,......,\ i+1}$. The result obtained from this method with x_0 = 0.1 for the equation of Example 1, x*sin(pi x)-exp(-x)=0, is graphically shown in Fig 1.

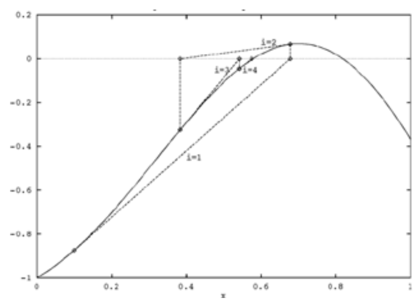


FIG 1. Newton Raphson method

Here also, when multiple roots are present, the root evenutally identified by the algorithm depends on the starting conditions supplied by the user. Example: if we had started with x_0 = 0.0, the Newton- Raphson method will converge to the larger root, as shown in Fig 2.

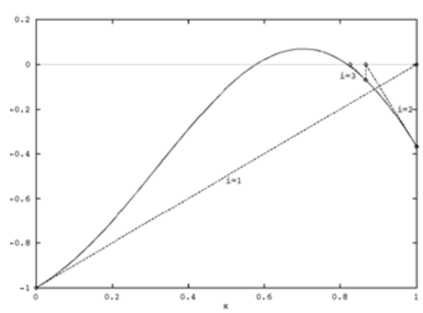


FIG 2. Newton Raphson method

So, before using any of the methods. An approximate idea of the roots may be developed from the physics the equation represents, preliminary mathematical analysis & using plotting routines.

Newton's Method:

Extend the Newton's method for 1-D case to multidimensional case.

Given $f(\vec{X})$, approximate $f(\vec{X})$ by a second order Taylor series at $\vec{X} = \vec{X}_i$:

$$f(\vec{X}) \approx f(\vec{X}_i) + \nabla f'(\vec{X}_i)(\vec{X} - \vec{X}_i) + \frac{1}{2}(\vec{X} - \vec{X}_i)' H_i (\vec{X} - \vec{X}_i)$$

where H_i is the Hessian matrix .

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ & \cdots & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

At the maximum (or minimum) point, $\frac{\partial f(\vec{X})}{\partial x_j} = 0$ for all j = 1, 2, . . . , n, or $\nabla f = \vec{0}$.

Then if H_i is non-singular,

$$\nabla f(\vec{X}_i) + H_i(\vec{X} - \vec{X}_i) = 0$$

$$\vec{X} = \vec{X}_i - H_i^{-1} \nabla f(\vec{X}_i)$$

Iteration:

$$\vec{X}_{i+1} = \vec{X}_i - H_i^{-1} \nabla f(\vec{X}_i)$$

Example:

$$f(\vec{X}) = 0.5x_1^2 + 2.5x_2^2$$

$$\nabla f(\vec{X}) = \begin{bmatrix} x_1 \\ 5x_2 \end{bmatrix}$$

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}$$

$$\vec{X}_0 = \begin{bmatrix} 5 \\ 1 \end{bmatrix}, \vec{X}_1 = \vec{X}_0 - H^{-1} \nabla f(\vec{X}_0) = \begin{bmatrix} 5 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{5} \end{bmatrix} \begin{bmatrix} 5 \\ 5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Comments: Newton’s method

• Converges quadratically near the optimum .

- Sensitive to initial point .
- Requires matrix inversion.
- Requires first and second order derivatives .

4.3 Equality constraints

Two variables and One Equality Constraint

Theorem 1 Suppose $x^* = (x^*_1, x^*_2)$ is a solution of the problem:

maximize $f(x_1, x_2)$ subject to $h(x_1, x_2) = c$.

Suppose further that (x^*_1, x^*_2) is not a critical point of h:

$$\nabla h(x^*_1, x^*_2) = \quad \nabla h(x_1^*, x_2^*) = \left(\frac{\partial h}{\partial x_1}(x_1^*, x_2^*), \frac{\partial h}{\partial x_2}(x_1^*, x_2^*)\right) \neq (0, 0),$$

(this condition is called constraint qualification at the point (x^*_1, x^*_2)). Then, there is a real number μ^* such that (x^*_1, x^*_2, μ^*) is a critical point of the Lagrangian function

$L(x_1, x_2, \mu) = f(x_1, x_2) - \mu[h((x_1, x_2) - c]$.

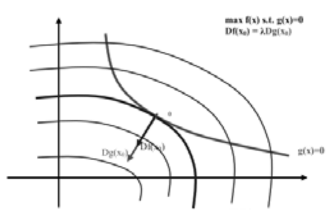
In other words, at (x^*_1, x^*_2, μ^*) we have

$$\frac{\partial L}{\partial x_1} = 0, \quad \frac{\partial L}{\partial x_2} = 0, \quad \frac{\partial L}{\partial \mu} = 0.$$

Proof. The solution (x^*, y^*) lies highest-valued level curve of f which meets the constraint set

$C = \{(x_1, x_2), h(x_1, x_2) = c\}$.

In other words at the point (x^*_1, x^*_2) the level set of f and C have common tangent, that is their tangents have equal slopes, or equivalently, parallel gradient vectors:



These slopes are

$$\frac{-\frac{\partial f}{\partial x_1}(x_1^*, x_2^*)}{\frac{\partial f}{\partial x_2}(x_1^*, x_2^*)} \quad and \quad \frac{-\frac{\partial h}{\partial x_1}(x_1^*, x_2^*)}{\frac{\partial h}{\partial x_2}(x_1^*, x_2^*)}.$$

The fact that these two slopes equal at (x^*_1, x^*_2) means

$$\frac{-\frac{\partial f}{\partial x_1}(x_1^*, x_2^*)}{\frac{\partial f}{\partial x_2}(x_1^*, x_2^*)} = \frac{-\frac{\partial h}{\partial x_1}(x_1^*, x_2^*)}{\frac{\partial h}{\partial x_2}(x_1^*, x_2^*)}.$$

Let us rewrite this equality as

$$\frac{\frac{\partial f}{\partial x_1}(x_1^*, x_2^*)}{\frac{\partial h}{\partial x_1}(x_1^*, x_2^*)} = \frac{\frac{\partial f}{\partial x_2}(x_1^*, x_2^*)}{\frac{\partial h}{\partial x_2}(x_1^*, x_2^*)}.$$

Let us denote by μ^* the common value of these two quotients:

$$\frac{\frac{\partial f}{\partial x_1}(x_1^*, x_2^*)}{\frac{\partial h}{\partial x_1}(x_1^*, x_2^*)} = \mu^* = \frac{\frac{\partial f}{\partial x_2}(x_1^*, x_2^*)}{\frac{\partial h}{\partial x_2}(x_1^*, x_2^*)}.$$

Rewrite this as the two equations

$$\frac{\partial f}{\partial x_1}(x_1^*, x_2^*) - \mu^* \frac{\partial h}{\partial x_1}(x_1^*, x_2^*) = 0,$$
$$\frac{\partial f}{\partial x_2}(x_1^*, x_2^*) - \mu^* \frac{\partial h}{\partial x_2}(x_1^*, x_2^*) = 0.$$

This two equations together with the third one

$h(x_1, x_2) - c = 0$

are exactly the conditions

$$\frac{\partial L}{\partial x_1}(x_1^*, x_2^*, \mu^*) = 0, \quad \frac{\partial L}{\partial x_2}(x_1^*, x_2^*, \mu^*) = 0, \quad \frac{\partial L}{\partial \mu}(x_1^*, x_2^*, \mu^*) = 0.$$

This proof was based on the fact that the level curves of f and of h at (x^*_1, x^*_2) and have equal slopes. We now present another version of the proof using the gradient vectors $\nabla f(x^*_1, x^*_2)$ and $\nabla h(x^*_1, x^*_2)$. Since gradient is orthogonal to level curve, then the level curves of f and h at (x^*_1, x^*_2) are tangent if and only if the gradients $\nabla f(x^*_1, x^*_2)$ and $\nabla h(x^*_1, x^*_2)$ line up at (x^*_1, x^*_2), that is the gradients are scalar multiples of each other

$$\nabla f(x^*_1, x^*_2) = \mu^* \nabla h(x^*_1, x^*_2)$$

(note that $\nabla f(x^*_1, x^*_2)$ and $\nabla h(x^*_1, x^*_2)$ can point in the same direction, in this case $\mu^* > 0$, or point in opposite directions, in this case $\mu^* < 0$). This equality immediately implies the above condition

$$\frac{\partial f}{\partial x_1}(x_1^*, x_2^*) - \mu^* \frac{\partial h}{\partial x_1}(x_1^*, x_2^*) = 0,$$
$$\frac{\partial f}{\partial x_2}(x_1^*, x_2^*) - \mu^* \frac{\partial h}{\partial x_2}(x_1^*, x_2^*) = 0.$$

Equality Constraints (Lagrangians)

$5-(x_1-2)^2-2(x_2-1)^2$

Maximize

subject to

$x_1 + 4x_2 = 3$

If we ignore the constraint, we get the solution $x_1 = 2$, $x_2 = 1$, which is too large for the constraint. Let us penalize ourselves λ for making the constraint too big. We end up with a function,

$L(x_1,x_2,\lambda)=5 - (x_1 -2)^2 -2 (x_2-1)^2+\lambda(3-x_1-4x_2)$

This function is called the Lagrangian of the problem. The main idea is to adjust λ so that we use exactly the right amount of the resource.

$\lambda=0$ leads to (2,1).

$\lambda=1$ leads to (3/2,0) which uses to minimum of the resource.

$\lambda = \frac{2}{3}$ gives (5/3, 1/3) and the constraint is satisfied exactly.

We now explore this idea more formally. Given a nonlinear program (P) with equality constraints:

Minimize (or maximize) f(x)

subject to

$g_1(x)=b_1$

$g_2(x)= b_2$

$g_m(x)= b_m$

A solution can be found using the Lagrangian as follows:

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i (b_i - g_i(x)))$$

(Note: this can also be written as $f(x) - \sum_{i=1}^{m} \lambda_i (g_i(x) - bi)$.

Each λ_i gives the price associated with constraint i.

The reason L is of interest is due to the following:

Assume $x^* = (x_1{}^*, x_2{}^*, \ldots\ldots, x_n{}^*)$ maximizes or minimizes f(x) subject to the constraint gi(x)=bi, for i=1,2,...,m.Then either

(i) The vectors $\nabla g_1(x^*), \nabla g_2(x^*), \ldots\ldots, \nabla g_m(x^*)$ are linearly dependent,

(ii) There exists a vector $\lambda^* = (\lambda_1{}^*, \lambda_2{}^*, \ldots\ldots, \lambda m^*)$ such that $\nabla L(x^*, \lambda^*) = 0.$

$$\frac{\partial L}{\partial x_1}(x^*,\lambda^*) = \frac{\partial L}{\partial x_2}(x^*,\lambda^*) = \ldots\ldots = \frac{\partial L}{\partial x_n}(x^*,\lambda^*) = 0$$

and $\frac{\partial L}{\partial \lambda_1}(x^*,\lambda^*) = \ldots = \frac{\partial L}{\partial \lambda_m}(x^*,\lambda^*) = 0$

Case (i) Above cannot occur when there is only one constraint. The following example shows how it might occur.

Problems:

1.Minimize $x_1+x_2+x_3{}^2$

Subject to

$X_1=1$

$X_1^2+x_2^2=1.$

It is easy to check directly that the minimum is achieved at $(x_1,x_2,x_3) = (1,0,0)$. The associated Lagrangian is given by,

$L(x_1,x_2,x_3,\lambda_1,\lambda_2)=x_1+x_2+x_3^2+\lambda_1(1-x_1)+\lambda_2(1-x_1^2-x_2^2)$

Observe that

$$\frac{\partial L}{\partial x_2}(1,0,0,\lambda_1,\lambda_2) = 1 \quad \text{for all} \quad \lambda_1, \lambda_2,$$

and consequently $\dfrac{\partial L}{\partial x_2}$ does not vanish at the optimal solution.

The reason for this is the following. Let $g_1(x_1,x_2,x_3) = x_1$ and $g_1(x_1,x_2,x_3) = x_1^2 + x_2^2$ denote the left hand sides of the constraints. Then $\nabla g_1(1,0,0) = (1,0,0)$ and $\nabla g_2(1,0,0) = (2,0,0)$ are linearly dependent vectors. So Case (i) occurs here.

When solving optimization problems with equality constraints, we will only look for solutions x* that satisfy Case (ii).

Note that the equation,

$$\frac{\partial L}{\partial \lambda_i}(x^*,\lambda^*) = 0$$

is nothing more than

$$b_i - g_i(x^*) = 0 \quad \text{or} \quad g_i(x^*) = b_i$$

In other words, taking the partials with respect to λ does nothing more than returning the original constraints.

Once we have found candidate solutions x*, it is not always easy to figure out whether they correspond to a minimum, a maximum or neither. If f(x) is concave and all of the $g_i(x)$ are linear, then any feasible x*with a corresponding λ* making $\nabla L(x^*, \lambda^*) = 0$ maximizes f(x) subject to the constraints. Similarly, if f(x) is convex and each $g_i(x)$ is linear, then any x*with a λ*making $\nabla L(x^*, \lambda^*) = 0$ minimizes f(x) subject to the constraints.

Minimize $2x_1^2 + x_2^2$

Subject to $X_1 + X_2=1$

$L(x_1,x_2,\lambda)=(2x_1^2+x_2^2+\lambda_1(1-x_1-x_2)$

$$\frac{\partial L}{\partial x_1}(x_1^*, x_2^*, \lambda^*) = 4x_1^* - \lambda_1^* = 0$$

$$\frac{\partial L}{\partial x_2}(x_1^*, x_2^*, \lambda^*) = 2x_2^* - \lambda_1^* = 0$$

$$\frac{\partial L}{\partial \lambda}(x_1^*, x_2^*, \lambda^*) = 1 - x_1^* - x_2^* = 0$$

Now, the first two equations imply $2x_1^*=x_2^*$.Substituting into the final equation gives the solution $x_1^*=1/3$,$x_2^*=2/3$ and $\lambda^* =4/3$with function value 2/3.

Since $f(x_1,x_2)$is convex (its Hessian matrix $H(x1, x2) = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$ is positive definite) and $g(x_1,x_2)=x_1+x_2$ is a linear function, the above solution minimizes $f(x_1,x_2)$subject to the constraint.

4.4 Jacobean methods

Two assumptions made on Jacobi Method:

1. The system given by

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots a_{nn}x_n &= b_n \end{aligned}$$

Has a unique solution.

2. The co-efficient matrix has non zeros on its main diagonal, namely, a_{11},a_{22}......a_{nn} are non zeros.

Main idea of Jacobi

To , solve the first equation for ,x_1 the second equation for x_2 & so on to obtain the rewritten equations,

$$x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 - \cdots a_{1n}x_n)$$
$$x_2 = \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3 - \cdots a_{2n}x_n)$$
$$\vdots$$
$$x_n = \frac{1}{a_{nn}}(b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots a_{n,n-1}x_{n-1})$$

Then make an initial guess of the solution $\boldsymbol{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, \dots x_n^{(0)})$. Substitute these values into the right hand side the of the rewritten equations to obtain the first approximation, $\left(x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \dots x_n^{(1)}\right)$.

This accomplishes one iteration.

The second approximation $\left(x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, \dots x_n^{(2)}\right)$ is computed by substituting the first approximation's vales into the right hand side of rewritten equations.

$$\boldsymbol{x}^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \dots x_n^{(k)}\right)^t, \quad k = 1,2,3, \dots$$

By repeated iterations, form a sequence of approximations

The Jacobi Method.

For each k ≥1 generate the components $x_i^{(k)}$ of $\boldsymbol{x}^{(k)}$ from $\boldsymbol{x}^{(k-1)}$ by

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[\sum_{\substack{j=1,\\ j\neq i}}^{n}(-a_{ij}x_j^{(k-1)}) + b_i\right], \qquad \text{for } i = 1,2,, \dots n$$

Optimization problem

One of the well known method to solve this system of equations is a Newton – Raphson method, which is one of so called Householder's methods in numerical analysis.

For the function of one variable it is based on the fact that for a differentiable function f(x) we have the following approximation:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Similarly, for the system of n functions of n variables:

$$X_{n+1} = X_n - [F'(x_n)]^{-1} F(X_n)$$

$F'(x_n)$,often called Jacobean matrix, is a matrix of first order partial derivatives of all the functions.

The possible ways to implement this algorithm:

(i) Define a function that calculates values at a given location.

(ii) Define a function that evaluates a Jacobean matrix.

(iii) Select a "best guess" starting value.

(iv) Evaluate the function and Jacobean at the current location.

(v) Find inverse Jacobean matrix.

(vi) Calculate the next position.

(vii) Iterate through steps 4 – 6 until the root is found with desired precision.

Optimization c :

void get F () {…}

void find Jacobean() {…}

void matrix Inverse () {…}

void matrix Vector Mult () {…}

int is Converge() {…}

main () {…}

4.5 Lagrangian method

Let P(b) denote the optimization problem, minimize f(x) subject to h(x) = b, x ∈ X. Let X(b) = {x ∈ X : h(x) = b}. We say that x is feasible if x ∈ X(b).

Define the Lagrangian as $L(x, \lambda) = f(x) - \lambda^T(h(x) - b)$.

Typically, $X \subseteq \mathbb{R}^n$, $h: \mathbb{R}^n \to \mathbb{R}^m$, with $b, \lambda \in \mathbb{R}^m$. Here λ is called a **Lagrangian multiplier**.

Theorem: (Lagrangian Sufficiency Theorem)

If $\bar{x}$ is feasible for P(b) and there exists $\bar{\lambda}$ such that $\inf_{x \in X} L(x, \bar{\lambda}) = L(\bar{x}, \bar{\lambda})$ then $\bar{x}$ is optimal for P(b).

Proof of the LST: For all x ∈ X(b) and λ we have ,

$f(x) = f(x) - \lambda^T(h(x) - b) = L(x, \lambda)$.

Now $\bar{x} \in X(b) \subseteq X$ and so by assumption,

$f(\bar{x}) = L(\bar{x}, \bar{\lambda}) \leqslant L(x$

Thus $\bar{x}$ is optimal for the optimization problem P(b).

Example: Minimize $x_1^2 + x_2^2$ subject to $a_1x_1 + a_2x_2 = b$, $x_1, x_2 \geq 0$. Here $L = x_1^2 + x_2^2 - \lambda(a_1x_1 + a_2x_2 - b)$. We consider the problem.

$\underset{x_1, x_2 \geq 0}{\min imize}[x_1^2 + x_2^2 - \lambda(a_1x_1 + a_2x_2 - b)]$. This has a stationary point where $(x_1, x_2) = (\lambda a_1/2, \lambda a_2/2)$. Now we choose λ such that $a_1x_1 + a_2x_2 = b$.

This happens for $\lambda = 2b/(a_1^2 + a_2^2)$. We have a minimum since $\partial^2L/\partial x_i^2 > 0$, $\partial^2L/\partial x_1\partial x_2 = 0$. Thus with this value of λ, the conditions of the LST are satisfied and the optimal value is $b^2/(a_1^2 + a_2^2)$ at $(x_1, x_2) = (a_1b, a_2b)/(a_1^2 + a_2^2)$.

4.6 Kuhn – Tucker conditions

The Kuhn-Tucker conditions discussed with examples for a point to be a local optimum in case of a function subject to inequality constraints.

The Kuhn Tucker conditions are each necessary & sufficient if the objective function is concave and each constraint is linear or each constraint function is concave, i.e.., the problems belong to a class called as the convex programming problems.

Consider the following problem,

Minimize f(X) subject to $g_j(X) \le 0$ for j = 1,2,...., p ; here X = (x_1 x_2 .... X_n)

Then the Kuhn-Tucker conditions for X* = (x_1 * x_2 * . .. . x_n *) to be a local minimum are ,

$$\begin{aligned} \frac{\partial f}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \frac{\partial g}{\partial x_i} &= 0 \qquad i = 1,2,\ldots,n \\ \lambda_j g_j &= 0 \qquad j = 1,2,\ldots,m \\ g_j &\le 0 \qquad j = 1,2,\ldots,m \\ \lambda_j &\ge 0 \qquad j = 1,2,\ldots,m \end{aligned}$$

Optimization Methods:

Optimization using Calculus – Kuhn-Tucker Conditions

If the constraints are of the form gj(X) ≥ 0 then λ_j have to be non positive in 1. On the other hand, if the problem is one of maximization with the constraints in the form $g_j(X) \ge 0$, then λ_j have to be non negative. It will be noted that sign convention has to be strictly followed for the Kuhn-Tucker conditions to be applicable.

Karush – Kuhn – Tucker (KKT) conditions:

Consider the problem ,

Maximize z = f(X) = f(x_1, x_2,..., x_n) subject to g(X) ≤ 0 [$g_1(X) \le 0$

$g_2(X) \le 0$

$g_m(X) \le 0$]

(the non-negativity restrictions, if any, are included in the above).

We define the Lagrangian function as,

L(X, S, $\vec{\lambda}$) = f(X) – λ[g(X) + s2]

(where s_1^2, s_2^2,..,s_m^2 are the non negative slack variables added to $g_1(X) \le 0$,.... $g_m(X) \le 0$ to make them into equalities) = f(X) – [λ_1\{g_1(X) + s_1^2\} + λ_2\{g_2(X) +s_2^2\}+..+λ_m \{g_m(x) + s_m^2\}]

KKT necessary conditions for optimality are given by $\vec{\lambda} \geq 0$,

$$\frac{\partial L}{\partial X} = \nabla f(X) - \lambda \nabla g(X) = 0$$
$$\frac{\partial L}{\partial S_i} = -2\lambda_i S_i = 0, i = 1,2,\ldots,m$$
$$\frac{\partial L}{\partial \lambda} = -(g_i(X) + S^2) = 0$$

These are equivalent to the following conditions: $\vec{\lambda} \geq 0$

$$\nabla f(X) - \lambda \nabla g(X) = 0$$
$$\lambda_i g_i(X) = 0 \qquad i=1,2\ldots m$$
$$g_i(X) \leq 0$$

In scalar notation this is given by,

$\lambda_i \geq 0$ i=1,2,….m

$$\frac{\partial f}{\partial x_j} - \lambda_1 \frac{\partial g_1}{\partial x_j} - \lambda_2 \frac{\partial g_2}{\partial x_j} - \ldots - \lambda_m \frac{\partial g_m}{\partial x_j} = 0$$
$$j = 1,2,\ldots,n$$
$$\lambda_i g_i(X) = 0 \quad i=1,2,\ldots m$$
$$g_i(X) \leq 0 \quad i=1,2,\ldots m$$

Economical Interpretation of Lagrange Multipliers

As with LPs, there is actually a whole area of duality theory that corresponds to NLPs.

we can view Lagrangians as shadow prices for the constraints in NLP (corresponding to the y vector in LP).

KKT Conditions

KKT conditions may not lead directly to a very efficient algorithm for solving NLPs. However, they do have a number of benefits:

1. They give insight into what optimal solutions to NLPs look like.

2. They provide a way to set up and solve small problems.

3. They provide a method to check solutions to large problems.

4. The Lagrangian values can be seen as shadow prices of the constraints.

4.7 Simple problems

Let us use the KKT conditions to derive an optimal solution for the following problem:

Maximize $f(x_1,x_2) = x_1+2x_2-x_2^3$

Subject to $x_1+x_2 \leq 1$

$X_1 \geq 0$

$X_2 \geq 0$

Solution:

Here there are three constraints namely,

$g_1(x_1,x_2)=x_1+x_2-1 \leq 0$

$g_2(x_1,x_2)= -x_1 \leq 0$

$g_3(x_1,x_2)= -x_2 \leq 0$

Hence the KKT conditions become,

$$\frac{\partial f}{\partial x_1} - \lambda_1 \frac{\partial g_1}{\partial x_1} - \lambda_2 \frac{\partial g_2}{\partial x_1} - \lambda_3 \frac{\partial g_3}{\partial x_1} = 0$$

$$\frac{\partial f}{\partial x_2} - \lambda_1 \frac{\partial g_1}{\partial x_2} - \lambda_2 \frac{\partial g_2}{\partial x_2} - \lambda_3 \frac{\partial g_3}{\partial x_2} = 0$$

$\lambda_1 g_1(x_1,x_2)=0$

$\lambda_1 g_2(x_1,x_2)=0$

$\lambda_1 g_3(x_1,x_2)=0$

$g_1(x_1,x_2)\leq 0$

$g_2(x_1,x_2)\leq 0$

$g_3(x_1,x_2)\leq 0$

(note: f is concave ,gi are convex,maximization problem

$\Rightarrow$these KKT conditions are sufficient at the optimum point and $\lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_3 \geq 0$

i.e.

$1 - \lambda_1 + \lambda_2 = 0$ (1)

$2 - 3x_2^2 - \lambda_1 + \lambda_3 = 0$ (2)

$\lambda_1(x_1 + x_2 - 1) = 0$ (3)

$\lambda_2 x_1 = 0$ (4)

$\lambda_3 x_2 = 0$ (5)

$X_1 + x_2 - 1 \leq 0$ (6)

$X_1 \geq 0$ (7)

$X_2 \geq 0$ (8)

$\lambda_1 \geq 0$ (9)

$\lambda_2 \geq 0$ (10)

$\lambda_3 > 0$ (11)

(1)gives $\lambda_1 = 1 + \lambda_2 \geq 1 > 0$ (using 10)

Hence (3) gives $x_1 + x_2 = 1$ (12)

Thus both x_1, x_2 cannot be zero.

So let $x_1 > 0$ (4) gives $\lambda_2 = 0$. therefore $\lambda_1 = 1$

if now $x_2 = 0$, then (2) gives $2 - 0 - 1 + \lambda_3 = 0$ or $\lambda_3 < 0$ not possible

Therefore $x_2 > 0$

hence (5) gives $\lambda_3 = 0$ and then (2) gives $x_2^2 = 1/3$ so $x_2 = 1/\sqrt{3}$ And so $x_1 = 1 - 1/\sqrt{3}$

Max $f = 1 - 1/\sqrt{3} + 2/\sqrt{3} - 1/3\sqrt{3} = 1 + 2/3$

Maximize $f(x) = 20x_1 + 10x_2$

Subject to $x_1^2 + x_2^2 \leq 1$

$x_1 + 2x_2 \leq 2$

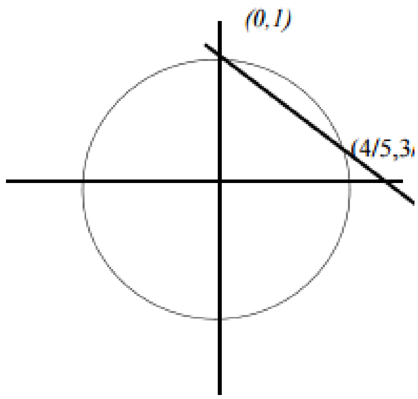
$x_1 \geq 0$, $x_2 \geq 0$

KKT conditions become

$20 - 2\lambda_1 x_1 - \lambda_2 + \lambda_3 = 0$

$10 - 2\lambda_1 x_2 - 2\lambda_2 + \lambda_4 = 0$

$\lambda_1(x_1^2 + x_2^2 - 1) = 0$



$\lambda_1(x_1 + 2x_2 - 2) = 0$

$\lambda_3 x_1 = 0$

$\lambda_4 x_2 = 0$

$x_1^2 + x_2^2 \leq 1$

$X_1 + 2X_2 \leq 2$

$X_1 \geq 0$

$X_2 \geq 0$

$\lambda_1 \geq 0$

$\lambda_2 \geq 0$

$\lambda_3 \geq 0$

$\lambda_4 \geq 0$

From the figure it is clear that max f occurs at (x_1, x_2) where $x_1, x_2 > 0$.

$\lambda_3 = 0, \lambda_4 = 0$

suppose $x_1 + 2x_2 — 2 \neq 0$

$\lambda_2 = 0$, therefore we get $20 - 2\lambda_1 x_1 = 0$

$10- 2\lambda 1, x2 = 0$.

$\lambda_1 x_1 = 10$ and $\lambda_1 x_2 = 5$, squaring and adding we get

$\lambda_l^2 = 125$ $\lambda_1 = 5\sqrt{5}$

therefore $x_1 = 2/\sqrt{5}$, $x_2 = 1/\sqrt{5}$, $f= 50/\sqrt{5} >22$

$\lambda_2 \neq 0 \Rightarrow x_1+X_2-2=0$

Therefore $x_1=0, x_2=1, f=10$

Or $x_1=4/5, x_2=3/5, f=22$

Therefore max f occurs at $x_1 =2 / \sqrt{5}$,$x_2=1/ \sqrt{5}$

Let us use the KKT conditions to derive an optimal solution for the following problem:

minimize $f(x_1,x_2) = x_1^2 +x_2$

Subject to $x_1^2 +x_2^2 \leq 9$

$x_1+x_2 \leq 1$

Solution:

Here there are two constraints namely,

$g_1 (x_1,x_2)=x_1 2+x_2 2-9\leq 0$

$g_2(x_1,x_2)=x_1+x_2-1\leq 0$

1: $\lambda_1\leq 0,\lambda_2\leq 0$ as it is a minimization problem

2: $2x_1-2\lambda_1 x_1-\lambda_2=0$

$1-2\lambda_1 x_2-\lambda_2=0$

Thus the KKT condition are:

3:$\lambda_1(x_1^2+x_2^2-9)=0$

$\lambda_2(x_1+x_2-1)=0$

4:$x_1^2+x_2^2 \leq 9$

$x_1+x_2 \leq 1$

Now $\lambda_1 = 0$ (from 2) gives $\lambda_2 = 1$ Not possible.

Hence $\lambda_1 \neq 0$ and so $x_l^2 + x_2^2 = 9$ (5)

Assume $\lambda_2 = 0$ So (1st equation of) (2) gives

$2x_1(1-\lambda_1) = 0$ Since $\lambda_1 \leq 0$ we get $x_1 = 0$

From (5), we get $x_2 = \pm 3$

2nd equation of (2) says (with $\lambda_l < 0$, $\lambda_2 = 0$) $x_2 = -3$

Thus the optimal solution is:

$x_1 = 0$, $x_2 = -3$, $\lambda_1 = -1/6$, $\lambda_2 = 0$

The optimal value is :

Z = - 3

OBJECT SCHEDULING

Network diagram representation – Critical path method – Time charts and resource leveling – PERT.

5.1 Network diagram representation

Your browser does not support the video tag.

GUIDELINES FOR NETWORK CONSTRUCTION

The terminologies which are used in network construction are explained as follows:

Node:

Generally, a node represents the starting or ending of an activity.

Branch/arc:

A *branch* represents the actual activity which consumes some kind of resource.

Precedence relations of activities:

For any activity, the *precedence relations* provide the information about the activities that precede it, the activities that follow it and the activities that may be concurrent with it.

The network construction requires a detailed list of individual activities of a project, estimates of activity duration and specifications of precedence relationships among different activities of the project.

Rules for network construction:

The following are the primary rules for constructing AOA diagram.

1. Dummy activity is an imaginary activity indicating precedence relationship only. Duration of a dummy activity is zero.

2. The network should have a unique starting node (tail event).

3. The network should have a unique completion node (head event).

4.No activity should be represented by higher than one arc in the network.

5. No two activities should have the same starting node to ending node.

5.2 Critical path method

CRITICAL PATH METHOD (CPM)

In 1957, DuPont developed a project management method designed to address the challenge of shutting down chemical plants for maintenance and then restarting the plants once the maintenance had been completed. Given a complexity of the process, they developed the Critical Path Method (CPM) for managing such projects.

CPM provides the following benefits:

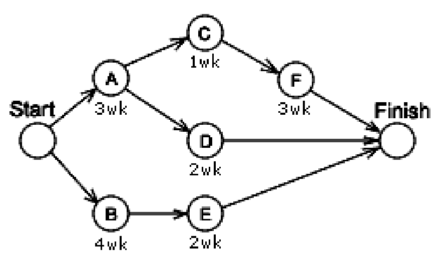
• It provides a graphical view of the project.

• Shows which activities are critical to maintaining the schedule and which are not.

• Predicts the time required to complete the project.

CPM models the activities and events of a project as a network. Activities are depicted as nodes on the network and events that signify the beginning or ending of activities are depicted as arcs or lines between the nodes.

The following is an example of a CPM network diagram:



CPM network diagram

Problems:

Consider Table summarizing the details of a project involving 14 activities.

Table : Project Details

Activity	Immediate predecessor(s)	Duration (months)
A	-	2
B	-	6
C	-	4
D	B	3
E	A	6
F	A	8
G	B	3
H	C, D	7
I	C, D	2
J	E	5
K	F, G, H	4
L	F, G, H	3
M	I	13
N	J, K	7

Let us

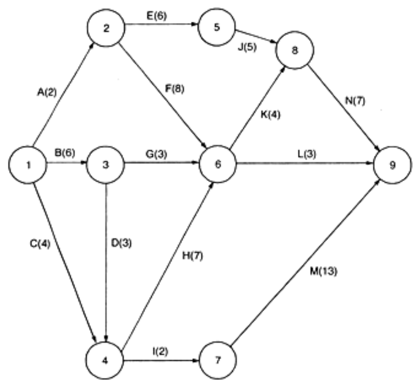
(a) Construct the CPM network.

(b) Determine the critical path and project completion time.

(c) Compute total floats and free floats for non-critical activities.

Solution

(a) The CPM network is shown in figure(a) below.



(a): CPM network for Example.

(b) The *critical path* of a project network is the longest path in the network. This can be identified by simply listing out all the possible paths from the starting node of the project (node 1) to the end node of the project (node 9) and select the path with the maximum sum of activity times on that path.

Hence, a different approach is to be used to identify the critical path. This consists of two phases: Phase 1 determines *earliest start times* (ES) of all the nodes. This is called *forward pass*; Phase 2 determines *latest completion times* (LC) of various nodes. This is called *backward pass.*

Let D_{ij} be the duration of the activity (*i,j*). ES_j be the earliest start times of all the activities which are emanating from node *j* (this is shown in the lower square which is by side of node *j*). LC_j be the latest completion times of all the activities which are ending at node *j* (this is shown in the upper square which is by the side of node *j*).

Determination of earliest start times **(ES_j).** During forwards pass use the following formula to compute earliest start times for all nodes.

$$ES_j = \max_i (ES_i + D_{ij})$$

The calculations of ES_j are summarized below:

Node 1: For node 1, $ES_1 = 0$

Node 2: $ES_2 = ES_1 + D_{1,2} = 0 + 2 = 2$

Node 3: $ES_3 = ES_1 + D_{1.3} = 0 + 6 = 6$

Node 4: $ES4 = \max_{i=1,3} (ES_i+D_{i4})$

$= \max (ES_1 + D_{1.4}, ES_3 + D_{3,4})$

$= \max (0 + 4, 6 + 3) = 9$

Node 5: $ES_5 = ES_2 + D_{2,5} = 2 + 6 = 8$

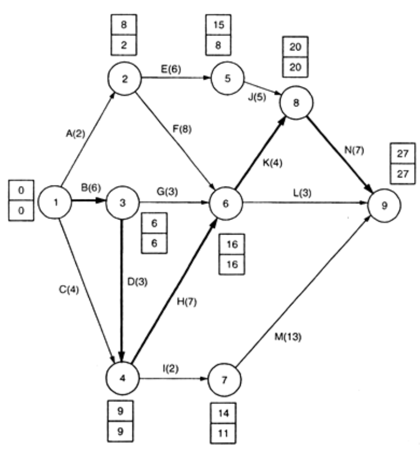
Node 6: $ES4 = \max_{i=2,3,4} (ES_i+D_{i4})$

$= \max (ES_2 + D_{2\,6}, ES_3 + D_{3.6}, ES_4 + D_{4,6})$

$= \max (2 + 8, 6 + 3, 9 + 7)$

$= \max (10, 9, 16)$

$= 16$



Similarly, the ES_j values for all other nodes are computed and summarized in below figure.

(b) Network with critical path calculations

***Determination of latest completion times* (LC_i).** During backward pass, the following formula is used to compute latest completion times (LC_i):

$LC_i = \max_{j} (LC_j - D_{ij})$

Node 9: For the last node 9, $LC_9 = ES_9 = 27$

Node 8: $LC_8 = LC_9 - D_{8,9} = 27 - 7 = 20$

Node 7: $LC_7 = LC_9 - D_{7,9} = 27 - 13 = 14$

Node 6: $LC_6 = \max_{j=8,9} (LS_i + D_{6,j})$

$= \min (LC_8 - D_{6,8}, LC_9 - D_{6,9})$

$= \min (20 - 4, 27 - 3)$

$= 16.$

Node 5: $LC_5 = LC_8 - D_{5\,8} = 20 - 5 = 15$

Node 4: $LC_4 = \max_{j=6,7} (LS_j + D_{4,j})$

$= \min (LC_6 - D_{4,6}, LC_7 - D_{4,7})$

$= \min (16-7, 14-2)$

$= 9.$

Similarly, the LC_i values for all other nodes are summarized in figure (b).

An activity (*i,j*) is said to be *critical* if all the following conditions are satisfied.

$ES = LC_i$, $ES_j = LC_j$, $ES_j - ES_i = LC_j - LC_i = D_{ij}$

By applying the above conditions to the activities in figure(b), the critical activities are identified and are shown in the same figure with thick lines on them. The corresponding critical path is 1-3-4-6-8-9 (B-D-H-K-N). The project completion time is 27 months.

(c) Total floats:

It is the amount of time that the completion time of an activity can be delayed without affecting the project completion time. Therefore,

$TF_{ij} = LC_j - ES_i - D_{ij} = LC_j - (ES_i + D_{ij}) = LC_j - EC_{ij}$

where, EC_{ij}, the earliest completion of the activity (*i,j*). Also,

$TF_{ij} = LS_{ij} - ES_i$

Where LS_{ij}, the latest start of the activity (*ij*) is given by,

$LS_{ij} = LC_j - D_{ij}$

(d) Free floats:

It is the amount of time that the activity completion time can be delayed without affecting the earliest start time of immediate successor activities in the network.

$FF_{ij} = ES_j - ES_i - D_{ij} = ES_j - (ES_i + D_{ij}) = ES_j - EC_{ij}$

The calculation of total floats and free floats of the activities are summarized in Table.

Table : Summary of Total Floats and Free Floats

Activity (*i, j*)	**Duration (D_{ij})**	**Total float (TF_{ij})**	**Free float (FF_{ij})**
1-2	2	6	0
1-3	6	0	0
1-4	4	5	5
2-5	6	7	0
2-6	8	6	6
3-4	3	0	0
3-6	3	7	7
4-6	7	0	0
4-7	2	3	0

5-8	5	7	7
6-8	4	0	0
6-9	3	8	8
7-9	13	3	3
8-9	7	0	0

Any critical activity will have zero total float and zero free float. Based on this property one can determine the critical activities.

From Table one can check that the total floats and free floats for the activities (1,3), (3, 4), (4, 6), (6, 8) and (8, 9) are zero. Hence they are critical activities. The corresponding critical path is 1-3-4-6-8-9 (B-D-H-K-N).

Let us consider the details of a project are as shown in Table. And try to determine the critical path and the corresponding project completion time.

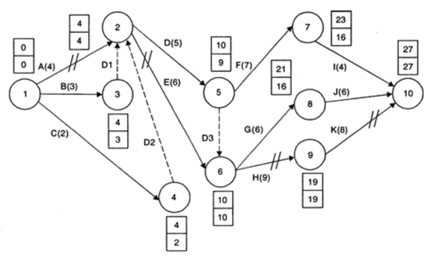
Table : Data of Example

Activity	**Immediate predecessor(s)**	**Duration (weeks)**
A	-	4
B	-	3
C	-	2
D	A, B, C	5
E	A, B, C	6
F	D	7

G	D, E	6
H	D, E	9
I	F	4
J	G	6
K	H	8

Solution

The network for this example is shown in figure(a). The critical path calculations are also summarized in the same figure. From Figure(a), the critical path is A-E-H-K and the corresponding project completion time is 27 weeks.



Note: D1, D2 and D3 are dummy activities.

(a) Project network and critical path calculations

A project consists of activities from A to J as shown in Table . The immediate predecessor(s) and the duration in weeks of each of the activities are given in the same table. Let us Draw the project network and find the critical path and the corresponding project completion time. Also find the total float as well as free float for each of the non-critical activities.

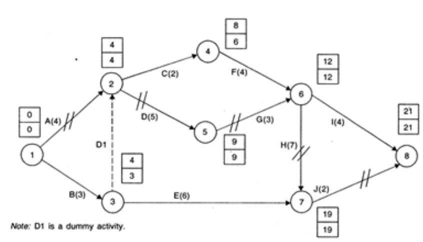
Table : Data of Example

Activity	Immediate predecessor(s)	Duration (weeks)
A	-	4

B	-	3
C	A, B	2
D	A, B	5
E	B	6
F	C	4
G	D	3
H	F, G	7
I	F, G	4
J	E, H	2

Solution

The project network for the given problem is constructed as in figure. The critical path calculations are shown in the same figure. The critical path is A-D-G-H-J and the corresponding project completion time is 21 weeks. The total float and the free float of each of the non-critical activities are shown in Table.



Project network and CPM calculations

Table : Details of Floats of Non-Critical Activities

Non-critical activity	Total float (weeks)	Free float (weeks)
B	1	0
C	2	0
E	10	10
F	2	2
I	5	5

5.3 Time charts and resource leveling

Your browser does not support the video tag.

GANTT CHART (TIME CHART)

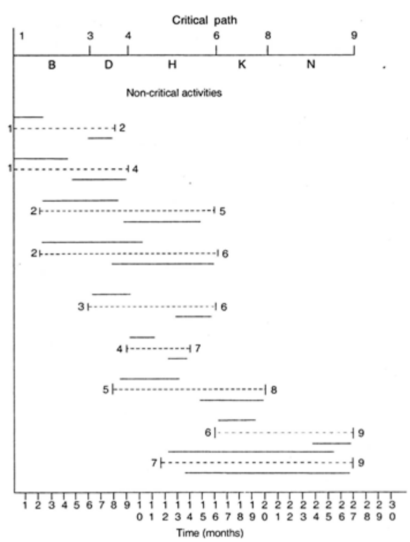
The network calculations is to draw Gantt chart (time chart). The start time and completion time of both and every activity will be represented in this chart. This chart gives clear calendar schedule for the whole project.

This chart is also used for resource levelling purpose. When there are limitations on the available resource (manpower, money, equipment, etc.), using this chart, one can adjust the schedule of non-critical activities depending upon their total floats to minimize the peak requirement of resource.

This helps to level the resource requirements smoothly throughout the project execution. Even if there is no restriction on any resource, it is usual practice to smooth out the resource requirements by resource levelling technique.

The time chart for Example is shown in Figure.

In Figure, the top-most horizontal line represents the schedule for various critical activities: (1, 3), (3, 4), (4, 6), (6, 8) and (8,9). The horizontal dotted lines represent the total time span over which a non-critical activity can be performed by,



Gantt chart (time chart)

A non-critical activity will have float in excess of its time duration. So, it is possible to adjust the start and completion time of any non-critical activity over its entire range shown by the dotted line without delaying the project completion time. Infinite number of schedules are possible for the non-critical activities.

But to give a finite number of schedules, we consider only two types of schedules: earliest start schedule and latest start schedule. These are shown by continuous horizontal lines over and below the dotted lines respectively. The time range for execution of a given non-critical activity is represented by the dotted line.

Problems:

A project consists of activities from A to H as shown in Table . The immediate predecessor(s) and the duration in months of each of the activities are given in the same table.

Let us

(a) Draw the project network and find the critical path and the corresponding project completion time.

(b) Also draw a Gantt-chart/Time chart for this project.

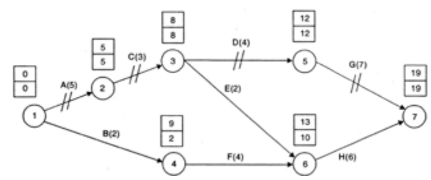
Table : Data of Example

Activity	Immediate predecessor(s)	Duration (months)
A	-	5
B	-	2
C	A	3
D	C	4
E	C	2
F	B	4
G	D	7

H	E, F	6

Solution

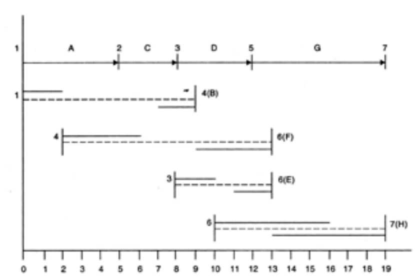
(a) The project network for the given problem is as shown in Figure (a) . The critical path calculations are shown in the same figure. The critical path is A-C-D-G and the corresponding project completion time is 19 months.



(a): Project network with CPM calculations of Example

(b) The Gantt-chart for the project shown in Figure(a) is given in Figure(b). The critical activities (A, C, D and G) are represented at the top. One can notice the fact that there is no time gap between the completion of one critical activity and the start of another critical activity. The total span during which each non-critical activity can be executed is shown by a dotted line.

A continuous line on the left hand side just above each dotted line shows the earliest start schedule of that non-critical activity. Another continuous line on the right hand side just below each dotted line shows the latest start schedule of that non-critical activity.

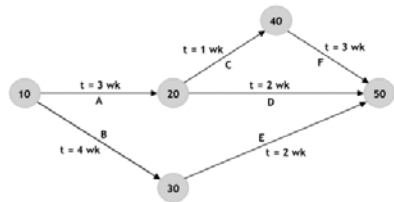


(b): Gantt-chart

5.4 PERT

PROJECT EVALUATION AND REVIEW TECHNIQUE (PERT)

CPM uses a fixed time estimate for each project activity. While easy to understand and use, it does not take account of the time variations that can impact on the completion time of a complex project. The Program Evaluation and Review Technique (PERT) is a network model that allows for variations in activity completion times. In a PERT network model, each activity is represented by a line (or arc), and each milestone (i.e. The completion of an activity) is represented by a node. A simple example is shown below.



A Simple PERT network

Milestones are numbered so that the end node of an activity has a higher number than the start node. Incrementing the numbers by 10 allows for additional nodes to be inserted without modifying the numbering of the entire network. The activities are labelled alphabetically, and the expected time required for each activity is also indicated. The critical path is the pathway through the project network that takes the longest to complete, and will determine the overall time required to complete the project. Bear in mind that for a complex project with many activities and task dependencies, there can be more than one critical path through the network, and that the critical path can change.

PERT planning involves the following steps:

▪ **Step:1 Identify activities and milestones** - the tasks required to complete the project, and the events that mark the beginning and end of each activity, are listed in a table.

▪ **Step 2: Determine the proper sequence of the activities** - this step may be combined with step 1, if the order in which activities must be performed is relatively easy to determine.

▪ **Step 3: Construct a network diagram** - using the results of steps 1 and 2, a network diagram is drawn which shows activities as arrowed lines, and milestones as circles. Software packages are available that can automatically produce a network diagram from tabular information.

▪ **Step 4: Estimate the time required for each activity** - any consistent unit of time can be used, although days and weeks are a common.

▪ **Step 5: Determine the critical path** - the critical path is determined by adding the activity times for each sequence and determining the longest path in the project. If the activity time for activities in other paths is significantly extended, the critical path may change. The amount of time that a non-critical path activity can be extended without delaying the project is referred to as its slack time.

▪ **Update the PERT chart as the project progresses** - as the project progresses, estimated times can be replaced with actual times.

Because the critical path determines the completion date of the project, the project can be completed earlier by allocating additional resources to the activities on the critical path. PERT also identifies activities that have slack time, and which can therefore lend resources to critical path activities. One drawback of the model is that if there is little experience in performing an activity, the activity time estimate may simply be a guess. Another more serious problem is that, because another path may become the critical path if one or more of its associated activities are delayed, PERT often tends to underestimate the to time required to complete the project.

PERT incorporates uncertainty by making it possible to schedule a project while not knowing precise details and durations of all activities. The time shown for each project activity when creating the network diagram is the time that the task is expected to take based on a range of possibilities that can be defined as:

▪ **The optimistic time** - the minimum time required to complete a task

▪ **The most likely time** - an estimate of how long the task will actually take

▪ **The pessimistic time** - the maximum time required to complete a task

The expected time (the time that will appear on the network diagram) is defined as the average time the task would require if it were repeated a number of times over a period of time, and can be calculated using the following formula:

The information included on the network diagram for each activity may include:

▪ The activity name

▪ The earliest start - ES

▪ The latest start - LS

▪ The expected duration

▪ The latest finish - LF

▪ The slack

▪ The earliest finish - EF

In order to determine these parameters, the project activities must have been identified and the expected duration of each calculated. The earliest start (ES) for any activity will depend on the maximum earliest finish(EF) of all predecessor activities (unless the activity is the first activity, in which case the ES is zero). The earliest finish for the activity is the earliest start plus the expected duration. The latest start (LS) for an activity will be equal to the maximum earliest finish of all predecessor activities. The latest finish (LF) is the latest start plus the expected duration. The slack in any activity is defined as the difference between the earliest finish and the latest finish, and represents the amount of time that a task could be delayed without causing a delay in subsequent tasks or the project completion date. Activities on the critical path by definition have zero slack.

A PERT chart provides a realistic estimate of the time required to complete a project, identifies the activities on the critical path, and makes dependencies (precedence relationships) visible. It can also identify the earliest and latest start and finish dates for a task, and any slack available. Resources can thus be diverted from non-critical activities to those that lie on the critical path should the need arise, in order to prevent project slippage. Variance in the project completion time can be calculated by summing the variances in the completion times of the activities in the critical path, allowing the probability of the project being completed by a certain date to be determined (this will depend on the number of activities in the critical path being great enough to allow a meaningful normal distribution to be derived).PERT charts can become unwieldy, however, if the number of tasks is too great. The accuracy of the task duration estimates will also depend on the experience and judgment of the individual or group that make them.

Time durations of various activities in a project Is assumed to be deterministic estimates. But, in reality, activity durations may be probabilistic.Therefore probabilistic considerations are incorporated while obtaining time durations of the activities in a project.

The following three estimates are used: *a* optimistic time, *b* pessimistic time and *m* most likely time.

The optimistic time is a time estimate if the execution goes extremely good. The pessimistic time is a time estimate if the execution goes very badly. The most likely a time estimate if execution is normal.

The probabilistic data for project activities commonly follow *beta distribution.*

The formula for mean (μ) and variance (σ^2) of the beta distribution are given below:

$$\mu = \frac{a + 4m + b}{6} \quad \text{and} \quad \sigma^2 = \left(\frac{b - a}{6}\right)^2$$

The range for the time estimates is from *a* to *b.* The most likely time will be everywhere in the range from *a* to *b.* The *expected project completion time* is $\sum \mu_i$, where μ_i is the expected duration of the i^{th} critical activity. The *variance of the project completion time* is $\sum_i \sigma_i^2$, where σ_i^2 is the variance of the i^{th} critical activity in the critical path.

As a part of statistical analysis, it may be interested in knowing the probability of completing the project on or before a given due date (*C*) or we may be interested in knowing the expected project completion time if the probability of completing the project is given.

For this, the beta distribution is approximated to standard normal distribution whose statistic is given below:

$$z = \frac{x - \mu}{\sigma}$$

Where, X is the actual project completion time, μ is the expected project completion time (sum of the expected durations of the critical activities) and σ is the standard deviation of the expected project completion time .

Therefore $P\ (x \leq C)$ represents the probability that the project will be completed on or before the C time units. This can be converted into the standard normal statistic z as:

$$P\left(\frac{x-\mu}{\sigma} \leq \frac{C-\mu}{\sigma}\right) = P\left(z \leq \frac{C-\mu}{\sigma}\right).$$

Problems:

Consider the Table summarizing the details of a project involving 11 activities.

Table : Details of Project with 11 Activities

Activity	**Predecessor(s)**	**Duration (weeks)**		
		a	*m*	*b*
A	-	6	7	8
B	-	1	2	9
C	-	1	4	7
D	A	1	2	3

E	A, B	1	2	9
F	C	1	5	9
G	C	2	2	8
H	E, F	4	4	4
I	E, F	4	4	10
J	D, H	2	5	14
K	I, G	2	2	8

Let us

(a) Construct the project network.

(b) Find the expected duration and variance of each activity.

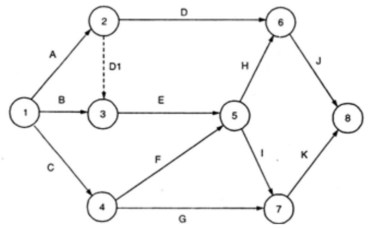
(c) Find the critical path and the expected project completion time.

(d) What is the probability of completing the project on or before 25 weeks?

(e) If the probability of completing the project is 0.84, find the expected project completion time.

Solution

(a) The project network is shown in Figure .



Network for Example .

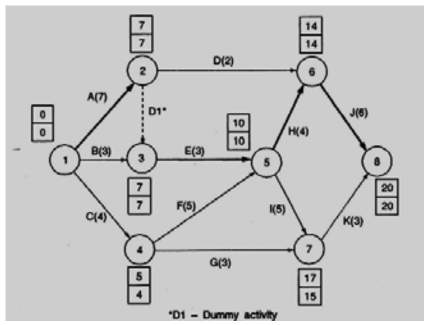
(b) The expected duration and variance of each activity are shown in Table .

Table : Computations of Expected Duration and Variance

Activity	**Duration (weeks)**			**Mean duration**	**Variance**
	a	*m*	*b*		
A	6	7	8	7	0.11
B	1	2	9	3	1.78
C	1	4	7	4	1.00
D	1	2	3	2	0.11
E	1	2	9	3	1.78
F	1	5	9	5	1.78
G	2	2	8	3	1.00
H	4	4	4	4	0.00
I	4	4	10	5	1.00

J	2	5	14	6	4.00
K	2	2	8	3	1.00

(c) The calculations of critical path based on expected durations are summarized in Figure. The critical path is A-D1-E-H-J and the corresponding project completion time is 20 weeks.



Network for Example.

(d) The sum of the variances of all the activities on the critical path is:

0.11 + 1.78 + 0.00 + 4.00 = 5.89 weeks.

Therefore $\sigma = \sqrt{5.89}$ = 2.43 weeks.

Also

$$P(x \leq 25) = P\left(\frac{x-\mu}{\sigma} \leq \frac{25-20}{2.43}\right)$$

= $P(z \leq 2.06)$ = 0.9803

This value is obtained from standard normal table. Therefore, the probability of completing the project on or before 25 weeks is 0.9803.

(e) We also have $P(x \leq C)$ = 0.84. Therefore,

$$P\left(\frac{x-\mu}{\sigma} \leq \frac{C-\mu}{\sigma}\right) = 0.84$$

$$P\left(z \leq \frac{C-20}{2.43}\right) = 0.84$$

From the standard normal table, the value of z is 0.99, when the cumulative probability is 0.84. Therefore,

$$\frac{C-20}{2.43} = 0.99 \quad \text{or} \quad C = 22.4 \text{ weeks}$$

The project will be completed in 22.41 weeks (approximately 23 weeks) if the probability of completing the project is 0.84.

Consider the data of a project summarized in Table .

Table : Data of Example

Activity	**Immediate Predecessor(s)**	**Duration (weeks)**		
		a	*m*	*b*
A	_	4	4	10
B	-	1	2	9
C	-	2	5	14
D	A	1	4	7
E	A	1	2	3
F	A	1	5	9
G	B, C	1	2	9

H	C	4	4	4
I	D	2	2	8
J	E, G	6	7	8

Let us

(a) Construct the project network.

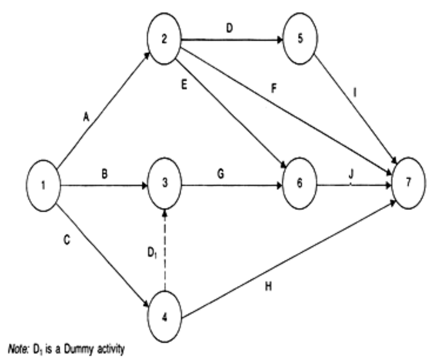
(b) Find the expected duration and the variance of each activity.

(c) Find the critical path and the expected project completion time.

(d) Find probability of completing the project on or before 35 weeks

Solution

(a) The project network for the given problem is shown in Figure.



Project network of Example .

(b) The mean and variance of each activity in the project are shown in Table .

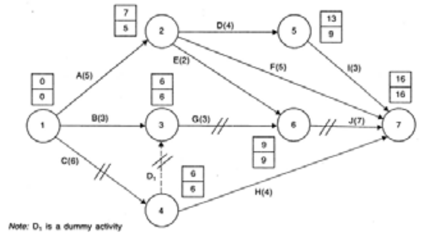
Table : Data of Example

Activity	Immediate predecessor(s)	Duration (weeks) *a m b*	Mean (Weeks)	Variance (Weeks)

		a	**m**			
A	—	4	4	10	5	1.00
B	-	1	2	9	3	1.78
C	-	2	5	14	6	4.00
D	A	1	4	7	4	1.00
E	A	1	2	3	2	0.11
F	A	1	5	9	5	1.78
G	B, C	1	2	9	3	1.78
H	C	4	4	4	4	0.00
I	D	2	2	8	3	1.00
J	E, G	6	7	8	7	0.11

(c) The critical path calculations are shown in figure .

From figure, the critical path is C-D_1-G-J and the corresponding expected project completion time is 16 weeks.



Project network with critical path calculations.

(d) The probability of completing the project on or before 35 weeks is computed as shown below:

The sum of the variance of the critical activities = 4 + 1.78 + 0.11 = 5.89 weeks. Hence the standard deviation, $\sigma = (5.89)^{0.5} = 2.427$ weeks.

$P(X \leq 35) = P[(X - \mu)/\sigma \leq (35 - 16)/2.427]$

$= P(Z \leq 7.82) = 0.9999.$