COURSE CODE: *MTS 363*

COURSE TITLE: *Introduction to Operations Research*

NUMBER OF UNITS: *2 Units*

COURSE DURATION: *Two hours per week*

COURSE DETAILS:

Course Coordinator: Dr. I.A.Osinuga

Email: osinugaia@unaab.edu.ng

Office Location: Departmental Office

Other Lecturers: Nil

COURSE CONTENT:

Phases of operations research, study, Modeling, linear, dynamic and integer programming. Probabilistic models. Decision theory and games. Traffic flow, network flow, project controls, Inventory models.

COURSE REQUIREMENTS:

This is a compulsory course for Mathematics, Statistics and Computer Science students in the University. In view of this, students are expected to participate in all the course activities and have minimum of 75% attendance to be able to write the final examination.

READING LIST:

1. Hamdy, Taha A. Operations Research (an introduction), New York: Macmillan Publishing Company, 1976.
2. Anderson, D.R., Sweeney,D.J. and Williams, T.A. An introduction to Management Science (7/ed.), St.Paul: West Publishing Company, 1994.
3. Lucey, T., Quantitative Techniques (5/ed.), London: Letts Educational Aldine Place, 1996.

4. Budnick, F.S.Apply Mathematics for Business, Economics and the Social Sciences (4/ed.), Singapore: McGraw-Hill Book Company, 1993.

5. Sharma, J.K ., Operations Research (Problems and Solutions, 2/ed.), Delhi: Macmillan India Ltd., 2004.

LECTURE NOTES

THE NATURE OF OPERATION RESEARCH

Definition

The phrase operation research is used to describe the application of scientific method (observe, model, predict) to the operation and management of organizations involving people and other resources (for example, money and machinery).

The nature and origin of OR

The activity of OR really got under way in England and then in the United States, during the second world war, when the motivation was the optimization of the allocation of scarce military resources. The application of science to warfare dates back to ancient times and it involves such luminous names as Archimedes and Leonardo da Vinci. These applications of science were devoted to developing ingenious new tools of warfare, new armaments. The modern development of operations research was organized on the principle of improving the utilization of existing stocks of resources (armament) by careful application of the scientific methods. Lanchester's law represents an early effort in the direction of present day operations research being the first modern attempt to model the interaction of opposing armies at war.

The scope of OR has expanded in the last 30 years and it has been found in civil as well as military settings. Some of the civil applications are to hospital management, criminal justice system operation, and a variety of commercial enterprises. A hallmark of OR is that it is directed toward achieving optimal solutions to problems. E.g., toward finding the mix of products for a manufacturer that will maximize profits, towards helping the manufacturer choose the right distribution of products among various outlets locations so as to minimize transportation costs and so on.

Phases of OR study

The major phases through which an OR team will proceed in order to effect an OR study include:

1. Definition of the problem
2. Construction of the model

3. Solution of the model
4. Validation of the model
5. Implementation of the final results

Models and Modelling in OR

Definition: A model is a general term denoting any abstract or idealized representation of a real life system or situation or simply idealized representation of a real life system or situation. Models provide a concise framework for analyzing a decision problem in a systematic manner and in this respect two basic components are essential for constructing a model: the ***objective of the system and the constraints imposed on the system.

Types of OR Models

The most important of OR models is the symbolic or mathematical model. It assumes that all the relevant variables, parameters and constraints as well as the objective are quantifiable. Thus if x_j, j = 1, 2, …, n are the n decision variable of the problem under study, and if the system is subject to m constraints, the general mathematical model can be written in the form.

$$\text{Optimize } z = f(x_1, \dots x_n) \qquad \text{(objective)}$$

s.t.

$$g_1(x_1, \dots x_n) \le b_1 \qquad i = 1, 2, \dots m$$

$$x_1, x_2, \dots, x_n \ge 0. \qquad \text{constraints}$$

In addition to mathematical model, simulation and heuristic models are used. Simulation models ‘initiate’ the behavior of the system over a period of time. This is achieved by specifying a number of events that are points in time whose occurrence signifies that important information pertaining to the behaviour of the system can be gathered.

LINEAR PROGRAMMING MODELS

Introduction

A linear programming is a mathematical technique applicable when the relationship among variables can be expressed as directly proportional (Linear) functions.

Every linear programming problem consists of a mathematical statement called an objective function. This function is to be either maximized or minimize, depending on the nature of the problem. For example, profit would be maximized but cost would be minimized. Also involved is a set of constrains equations which can either be in form of inequalities or equalities of the difference resources available and the proportion of each resource necessary to make a unit of the item of interest such as manufactured parts, personal policies, inventories etc.

FORMULATION OF LINEAR PROGRAMMING MODELS

The usefulness of linear programming as tool for optimal decision making and resource allocation is based on its applicability to many diversified decision problems as determining the most profitable product mix, scheduling inventory, planning manpower management etc. it has been used for pollution control, personal allocation, capital budgeting and financial personnel selection.

The effective use and application require, as a first step the formulation of the model when the problem is presented. The three basic steps in formulating a linear programming are as follows:

Step 1

Identify the decision variables to be determined and express them in terms of algebraic equations.

Step 2

Identify all the limitations or constraints in the given problem and then express them as linear inequalities or equalities, in terms of above identified decision variables

Step 3

Identify the objective (criterion) which is to be optimized (maximized or minimized) and express it as a linear function of the above defined decision variables.

We present below some illustrations on the formulation of linear programming models in various situations drawn from different area of management.

Example 1: Production Planning Problem

A tailor has the following materials available. 16 square meters of cotton, 11 square meters of silk, and 15 square meters of wool. He can make out two products from these three materials, namely dress and suite. A dress requires the following: 2 square meters of cotton, 1 square meter of silk and 1 square meter of wool. A suite requires 1 square meter of cotton, 2 square meter of silk and 3 square meter of wool. If the gross profit realized from a dress and as suite is respective N30 and N50. Formulate the above as a linear programming model.

Solution:

The information needed to formulate the above problem is summarized in the table below

Table 1

Product	Dress	Suite	Material
Material			
Cotton	2	1	16
Silk	1	2	11
Wool	1	3	15
Profit	N30	N50	

Let x_1 be number of dresses to be made

x_2 be the number of suites to be made

Then the total profit $z = 30x_1 + 50x_2$ which is the objective function of the problem.

The following are constraints or limitations of the problem:

(a) Only 16 sq. meter available for cotton to be used hence we have $2x_1 + x_2 \leq 16$

(b) Limitation on silk would imply that $x_1 + 2x_2 \leq 11$

(c) That on wool also means $x_1 + 3x_2 \leq 15$

(d) And finally, at worst the tailor would make no garment implies that $x_1, x_2 \geq 0$

Rewriting all together we have their linear programming models to be as follows:

Maximize $z = 30x_1 + 50x_2$ (objective function) s.t.

$2x_1 + x_2 \leq 16$ (cotton constraint)

$x_1 + 2x_2 \leq 111$ (silk constraint)

$x_1 + 3x_2 \leq 15$ (wool constraints)

$x_1, x_2 \geq 0$ (non negative constraints)

Example 2: Cost Minimization Problem:

A manufacturer is to market a new fertilizer which is to be mixture of two ingredients A and B. The properties of the two ingredients are as follows: ingredient A contains 20% bone meal, 30% nitrogen, 40% lime and 10% phosphate and it cost N2.40 per kilogram. Ingredient B contains 40% bone meal, 10% nitrogen, 45% lime and 5% phosphate, and it costs N1.60 per kilogram. Furthermore it is decided that:

(i) The fertilizer will be sold in bags containing a minimum of 50 kilograms

(ii) It must contain at least 12% nitrogen

(iii) It must contain at least 6% phosphate

(iv) It must contain at least 20% bone meal

Formulate the above problem as a LPM.

Solution:

Let x_1 = number of kilogram of ingredient A

x_2 = number of kilogram of ingredient B

The objective function is

Minimize $z = 2.4x_1 + 1.6x_2$

The following are constraints or limitations

(a) Total weight constraints: $x_1 + x_2 \geq 50$

(b) Bone meal constraints: $0.2x_1 + 0.4x_2 \geq 0.20$

(c) Nitrogen constraints: $0.3x_1 + 0.1x_2 \geq 0.12$

(d) Phosphate constraints: $0.1x_1 + 0.05x_2 \geq 0.06$

(e) Non negative constraints: $x_1, x_2 \geq 0$.

GRAPHICAL SOLUTION METHOD

One way of solving a linear programming problem after its formulation is the use of graph. This is possible if the decision variables are not more than three. It is very handy when the decision variables are two since a graph in 2 dimensions is easier to draw than that in 3 dimensions; anything above three dimensions may be difficult.

Example 3:

Suppose a manufacturer produce 2 liquids, X and Y because of past sales experience the market research estimates that at least as much Y as X is needed. The manufacturing capacity of the plant allows for a total of 9 units to be manufactured. If each unit of liquid X results in a profit of N2 and the profit for each unit of Y is N1, how much of each should be produced to maximize profit?

Solution:

Let x be the amount of x to be produced

y be the amount of Y to be produced

We may state the given problem mathematically as follows.

$$\text{Max } z = 2x + y$$

s.t

$$2x - y \leq 0$$

$$x + y \leq 9$$

$$x, y \geq 0$$

in the next section we develop a technique (the simplex algorithm) for solving such problems in general, but we can solve this particular one geometrically. First we sketch the set of point in IR^2 that satisfy the set of constraints (shaded part of fig. 1). This region is called the feasible region.

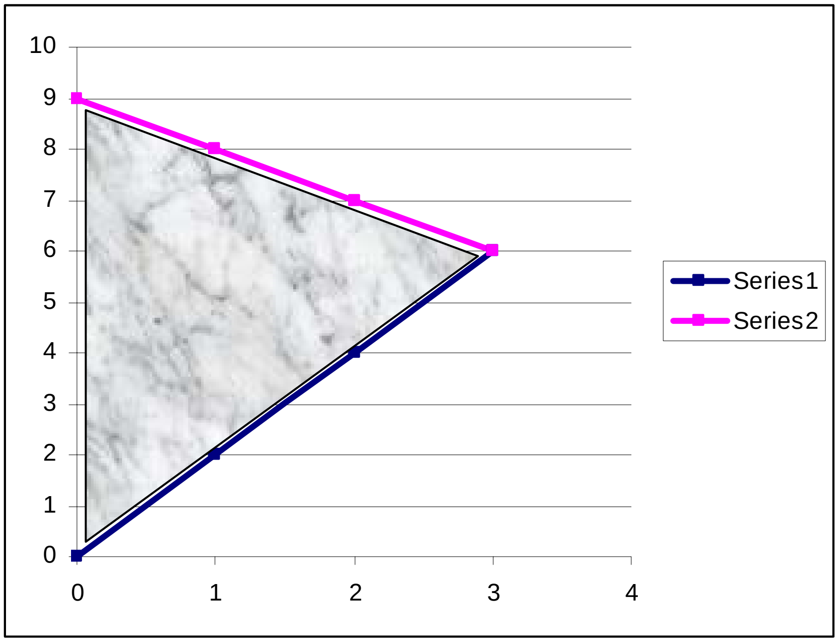


Figure 1

If a point is in the region, it satisfies all of the constraints and is called feasible, if it is not in the region; it violates at least one of them and is called infeasible.

Different points in the infeasible region give different values of the objective function. For example, the region (0, 0) is feasible and gives a profit 2 (0) – 0 = 0. Similarly, (1, 5) is feasible and gives profit of 2 (1) + 5 = 7, which is better than 0, we seek the point or points of the feasible region that yield the maximal profit.

Corresponding to a fixed value of z, the set of solutions to the equation z = 2x + y is the line (called the objective line) with the slope -2 and y intercept z. In other words, all points along the line y = -2x + z correspond to the same z value. Remember that we are interested only in those points that lie in the feasible region, and we want z to be as large as possible (fig. 2)

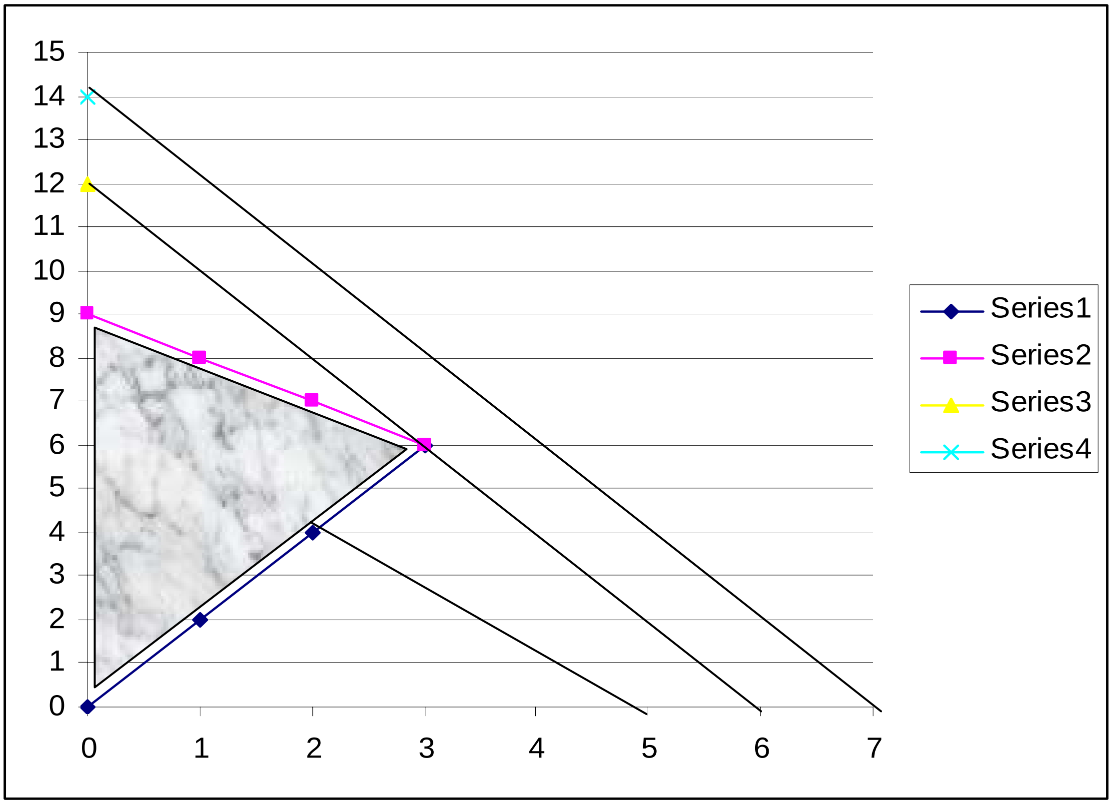


Figure 2

2.0 THE SIMPLEX METHOD

Simplex method is a general purpose approach employed in solving linear programming problems that are too large to be solved graphically. At this time, the simplex method is by far the most widely used algebraic procedure for solving LPP. Computer programs based on this method can routinely solve LPP with thousands of variables and constraints.

2.1 An Algebraic Overview of the Simplex Method

We introduce the problem we will use to demonstrate the simplex method.

Lowtech industries import electronic components that are used to assemble two different models of personal computers. One model is called the LT Deskpro computer and the other model is called LT portable computer. Lowtech management is currently interest in developing a weekly production schedule for both products.

The Deskpro generates a profit contribution of N50 per unit, and the portable generates a profit contribution of N40 per unit. For next week's production, a maximum of 150 hours of assembly time can be made available. Each unit of the Deskpro requires 3 hours of assembly time, and each unit of the portable requires 5 hours of assembly time. In addition, Lowtech currently has only 20 portable display components in inventory; thus no more than 20 units of the portable may be assembled. Finally, only 300 square feet of warehouse space can be made available for new production. Assembly of each Deskpro requires 8 square feet of warehouse space; similarly, each portable requires 5 square feet.

Let x_1 = no of units of the Deskpro assembled

x_2 = no of units of the Portable assembled

The complete mathematical model for this problem is presented below:

Max $50x_1 + 40x_2$

s.t

$3x_1 + 5x_2 \leq 150$	Assembly Time
$1x_2 \leq 20$	Portable display
$8x_1 + 5x_2 \leq 300$	Warehouse capacity

Adding a slack variable to each of the constraints permits us to write the problem in standard form.

Max $50x_1 + 40x_2 + 0s_1 + 0s_2 + 0s_3$

s.t.

$3x_1 + 5x_2 + s_1$	=	150
$1x_2$	=	20
$8x_1 + 5x_2 + s_3$	=	300

$x_1, x_2, s_1, s_2, s_3 \geq 0$

2.2 Basic Solution

To determine a basic solution, set n-m of the variables equal to zero, and solve the m linear constraints equations for the remaining m variables. We shall refer to the n-m variables set equal to zero as the non basic variables and the remaining m variables allowed (to be non zero) as the basic variables.

A basic solution can either be feasible or infeasible. A basic infeasible solution is a basic solution that also satisfies the non negativity condition otherwise it is infeasible.

For example, if we set $x_2 = 0$ and $s_1 = 0$ we obtain the following solution to the three-equation, five variable set of linear equation determined by the Lowtech constraints.

$x_1 = 50$ $s_1 = 0$ $s_3 = -100$ $x_2 = 0$ $s_2 = 20$

The above solution is referred to as a basic solution for the Lowtech LPP but the basic solution is not a feasible solution because $s_3 = -100$. However, suppose we had chosen to make x_1 and x_2 non basic variables by setting $x_1 = 0$ and $x_2 = 0$. Then the complete solution corresponding to $x_1 = 0$ and $x_2 = 0$ is:

$x_1 = 0, x_2 = 0, s_1 = 150, s_2 = 20, s_3 = 300.$

This solution is a basic solution since it was obtained by setting two of the variables equal to zero and solving for the other three variables. Moreover, it is a basic feasible solution since all of the variables are greater than or equal to zero.

2.3 Setting the Initial Simplex Tableau

To provide a convenient means for performing the calculation required by the simplex solution procedure, we will first develop what is referred to as the initial simplex tableau.

We adopt the general notation below in the tableau for representation of a linear program.

c_j = objective function coefficient for variable j

b_i = right hand side value for constraint i

a_{ij} = coefficient associated with variable j in constraint i

We can show this portion of the initial simplex tableau as follows:

Table 2

(a)

c_1	c_2	...		c_n	
a_{11}	a_{12}	…	a_{in}	b_1	
a_{21}	a_{22}	…	a_{2n}	b_2	
.	.	…	.	.	
.	.	…	.	.	
.	.	…	.	.	
a_{m1}	a_{m2}	…	a_{mn}	b_m	

(b)

c row	
A	b
Matrix	Column

Thus for the Lowtech problem, we obtain the following partial initial simplex tableau

Table 3

x_1	x_2	s_1	s_2	s_3	
50	40	0	0	0	
3	5	1	0	0	150
0	1	0	1	0	20
8	5	0	0	1	300

2.4 Improving the Solution

To improve the initial basic feasible solution, the simplex method must generate a new basic feasible solution (extreme point) that yields a better value for the objective functions. To do so require changing the set of basic variables and the simplex methods provides an easy way to carry out this change of variables. For computational convenience, we will add two new columns to the present form of the simplex tableau and two rows to the bottom of the tableau.

Table 4

		x_1	x_2	s_1	s_2	s_3		
Basis	C_B	50	40	0	0	0		
s_1	0	3	5	1	0	0	150	
s_2	0	0	1	0	1	0	20	
s_3	0	8	5	0	0	1	300	
	z_j	0	0	0	0	0	0	← Profit
	$c_j - z_j$	50	40	0	0	0		

Criterion for entering a new variable into the basis

Look at the evaluation row (cj – zj) and select the variable to enter the basis that will cause the larger per-unit improvement on the value of the objective function. In the case of a tie, we follow the convention of selecting the variable to enter the basis that corresponds to the leftmost f the columns.

Criterion for removing a variable from the current basis (minimum ratio test)

Suppose the incoming basic variable corresponds to column j in the A portion of the simplex tableau, for each row I, compute the ratio b_i/a_{ij} for each aij greater than zero. The basic variable that will be removed from the basis corresponds to the minimum of these ratios. In the case of a tie, we follow the convention of selecting the variable that corresponds to the uppermost of the tied rows.

Table 5

Entering Variable ↘ (x_1)

Leaving Variable ↘ (s_3)

		x_1	x_2	s_1	s_2	s_3		b_i/a_{ij}
Basis	C_B	50	40	0	0	0		
s_1	0	3	5	1	0	0	150	150/3 = 50
s_2	0	0	1	0	1	0	20	20/0 = ∞ (ignore)
s_3	0	8	5	0	0	1	300	300/8 = 37.5
z_j		0	0	0	0	0	0	
$c_j - z_j$		50	40	0	0	0		

Adopting the usual programming terminology refer to this squared element as the pivot element. The column and the row containing pivot element are called the pivot column and pivot row respectively

Table 6

		x_1	x_2	s_1	s_2	s_3	
Basis	C_B	50	40	0	0	0	
s_1	0	0	25/8	1	0	-3/8	75/2
s_2	0	0	1	0	1	0	20
x_1	50	1	5/8	0	0	1/8	75/2
	z_j						1875 ← Profit
	$c_j - z_j$						

Complete Second Tableau

Table 7

Basis	C_B	x_1 50	x_2 40	s_1 0	s_2 0	s_3 0	
s_1	0	0	25/8	1	0	-3/8	75/2
s_2	0	0	1	0	1	0	20
x_1	50	1	5/8	0	0	1/8	75/2
z_j		50	250/8	0	0	50/8	1875
$c_j - z_j$			70/8	0	0	-50/8	

Moving Towards a better Solution

Table 8

Basis	C_B	x_1 50	x_2 40	s_1 0	s_2 0	s_3 0		b_i/a_{ij}
s_1	0	0	25/8	1	0	0	-3/8	75/2 / 25/8 = 12
s_2	0	0	1	0	1	0	20	20/1 = 20
x_1	50	1	5/8	0	0	1/8	75/2	75/2 / 5/8 = 60
z_j		50	250/8	0	0	50/8	1875	
$c_j - z_j$		0	70/8	0	0	-50/8		

The new tableau resulting from there row operations is as follows

Table 9

Basis	C_B	x_1 50	x_2 40	s_1 0	s_2 0	s_3 0	
s_1	40	0	1	8/25	0	-3/25	12
s_2	0	0	0	8/25	1	3/25	8
x_1	50	1	0	-5/25	0	5/25	30
z_j		50	40	14/5	0	26/5	1980
$c_j - z_j$		0	0	-14/5	0	-26/5	

Note that the values of the basic variables are x_2 = 12, s_2 = 8, and x_1 = 30, and the corresponding profit is 40 (2) + 0 (8) + 50 (30) = 1980.

Optimally Criterion

The optimal solution to a LPP has been reached when all of the entries in the act evaluation row (cj – zj) are zero or negative. In such cases, the optimal solution is the current basic feasible solution.

Interpreting the optimal solution

The optimal solution of the Lowtech problem, consisting of the basic variables x_1, x_2 and s_2 and non basic variables s_1 and s_3 is written as follows.

$x_1 = 30, x_2 = 12, s_1 = 0, s_2 = 8, s_3 = 0.$

Indicating that there are no idle unit of the assembly time constraint and the warehouse capacity constraint, in other words, these constraints are both binding. Moreover, if management wants to maximize total profit contribution, Lowtech should produce 30 units of the Deskpro and 12 units of the portable. Since s_2 = 8, management should note that there will be eight unused portable display units.

SUMMARY OF THE SIMPLEX METHOD

The steps followed to solve a linear program using the simplex method can now be summarized as follows:

We assume that the problem has all less than or equal to constraints and involves maximization.

Step:

1. Formulate a LP model of the problem
2. Add slack variables to each constraints to obtain standard form
3. Set up the initial simplex tableau
4. Choose the non basic variable with the largest entry in the net evaluation row to bring into the basis. This identifies the pivot column; the column associated with the incoming variable.
5. Choose as the pivot row that row with the smallest ratio of b_i/ a_{ij} for $a_{ij} > 0$ where j is the pivot column

6. Perform the necessary elementary row operations to convert the column for the incoming variable to a unit column with a 1 in the pivot row.
7. Test for optimally. If $c_j - z_j \le 0$ for all columns, we have the optimal solution. If not, return to Step 4.

Problems

1. Suppose a manufacturer produces to liquids, X and Y. Because of the past sales experience the market researcher estimates that at least twice as much Y as X is needed. The manufacturing capacity of the plant allows for a total of 9 units to be manufactured. If each unit of liquid X results in a profit of N2 and the profit for each unit of Y is N1, how much of each should be produced to maximize the profit?
2. Solve the following LP using the Simplex method.

$$\text{Max } 4x_1 + 6x_2 + 3x_3 + x_1$$

s.t

$$3/2x_1 + 2x_2 + 4x_3 + 3x_4 \le 550$$
$$4x_1 + x_2 + 2x_3 + x_4 \le 700$$
$$2x_1 + 3x_2 + x_3 + 2x_4 \le 200$$
$$x_1, x_2, x_3, x_4 \ge 0.$$

3. The following partial initial simplex tableau is given:

		x_1	x_2	x_3	s_1	s_2	s_3	
Basis	C_B	5	20	25	0	0	0	
		2	1	0	1	0	0	40
		0	2	1	0	1	0	30
		3	0	-1/2	0	0	1	15
	z_j							
	$c_j - z_j$							

(a) Complete the initial tableau

(b) Write the problem in tableau form

(c) What is the initial basis, does this correspond to the origin? Explain.

(d) What is the value of the objective function at this initial solution?

(e) For the next iteration, what variable should enter the basis, and what variable should leave the basis?

2.5 Tableau Form: The General Case

(a) Grater – Than or – Equal – to Constraints

Suppose that in the Lowtech industry problem, management wanted to ensure that the combined total production for both models would be at least 25 units. That enables us to add another constraint to the constraint equation i.e. total production constraint. $1x_1 + 1x_2 \geq 25$

Max $50x_1 + 40x_2$

s.t

$3x_1 + 5x_2 \leq 150$

$1x_2 \leq 20$

$8x_1 + 5x_2 \leq 300$

$1x_1 + 1x_2 \geq 25$

$x_1, x_2 \geq 0.$

Standard Form:

Max $50x_1 + 40x_2 + 0s_1 + 0s_2 + 0s_3 + 0s_4$

s.t

$3x_1 + 5x_2 + s_1 = 150$

$1x_2 + s_2 = 20$

$8x_1 + 5x_2 + s_3 = 300$

$x_1, x_2, s_1, s_2, s_3, s_4 \geq 0$

Tableau form:

$3x_1 + 5x_2 + s_1 = 150$

$1x_2 + s_2 = 20$

$8x_1 + 5x_2 + s_3 = 300$

$1x_1 + 1x_2 – 1s_4 + 1a_4 = 25$

We can write the objective function for the tableau form of the problem as follows:

Max $50x_1 + 40x_2 + 0s_1 + 0s_2 + 0s_3 + 0s_4 – Ma_4$

Exercise: ***Prepare the initial simplex tableau for the problem. Hence obtain the optimal solution***

(b) Equality Constraints

When an equality constraints occurs in a LPP, we need to add an artificial variable to obtain tableau form and an initial basic feasible solution for example, if the equality constraint is $6x_1 + 4x_2 – 5x_3 = 30$

With the artificial variable, the above equation becomes

$$6x_1 + 4x_2 – 5x_3 + 1a_1 = 30$$

(c) Negative Right-Hand Sides

Case 1: Equality constraints: $6x_1 + 3x_2 – 4x_3 = -20$

$$6x_1 – 3x_2 + 4x_3 = 20$$

Case 2: ≥ Constraints: $6x_1 + 3x_2 – 4x_3 \geq -20$

$$6x_1 – 3x_2 + 4x_3 \leq 20$$

Case 3: ≤ Constraints: $6x_1 + 3x_2 – 4x_3 \leq -20$

$$6x_1 – 3x_2 + 4x_3 \geq 20$$

Example: Convert the following example problem into tableau form and then set up the initial simplex tableau:

Max $6x_1 + 3x_2 + 4x_3 + 1x_4$

s.t

$-2x_1 – 1/2x_2 + 1x_3 – 6x_4 = -60$

$1x_1 + 1x_3 + 2/3x_4 \leq 20$

$-1x_2 – 5x_3 \leq - 50$

$x_1, x_2, x_3, x_4 \geq 0$

We later obtain the tableau form:

Max $6x_1 + 3x_2 + 4x_3 + 1x_4 + 0s_2 + 0s_3 – Ma_1 – Ma_3$

s.t.

$2x_1 + 1/2x_2 – 1x_3 + 6x_4 \qquad 1a_1 = 60$

$1x_1 + 1x_3 + 2/3x_4 + 1s_2 \leq 20$

$1x_2 + 5x_3 - 1s_3 + 1a_3 \geq 50$

$x_1, x_2, x_3, x_4, s_2, s_3, a_1, a_3 \geq 0$

The initial simplex tableau corresponding to this tableau form is:

Table 10

Basis	C_B	x_1 6	x_2 3	x_3 4	x_4 1	s_2 0	s_3 0	a_1 -M	a_3 -M	
a_1	-M	2	½	-1	6	0	0	1	0	60
s_2	0	1	0	1	2/3	1	0	0	0	20
a_3	-M	0	1	5	0	0	-1	0	1	50
z_j		-2M	-3/2M	-4M	-6M	0	M	-M	-M	-110M
$c_j - z_j$		6+2M	3 + 3/2M	4 + 4M	1 + 6M	0	-M	0	0	

Note that the squared element is the pivot element indicating that x_4 will enter and a_1 will leave the basis at the first iteration.

2.6 Solving a Minimization Problem

There are two different ways of solving a minimization problem

(1) Reverse the maximization rule i.e. select the variable with the most negative cj – zj as the one to introduce into solution. In this case, optimal solution is reached when every value in the cj – zj row is non-negative

(2) Minimize z subject to a set of constraints should be changed to maximize (-z) subject to a set of the same constraints.
i.e. min z = max (-z).

Example: Solve the LPP:

Min $2x_1 + 3x_2$

s.t.

$1x_1 \geq 125$

$1x_1 + 1x_2 \geq 350$

$2x_1 + 1x_2 \leq 600$

$x_1, x_2 \geq 0.$

Solution:

Convert to an equivalent maximization problem:

Max $-2x_1 - 3x_2$

s.t

$1x_1 \geq 125$

$1x_1 + 1x_2 \geq 250$

$2x_1 + 1x_2 \leq 600$

$x_1, x_2 \geq 0.$

The tableau form for this problem is as follows:

Max $-2x_1 - 3x_2 + 0s_1 + 0s_2 + 0s_3 - Ma_1 - Ma_2$

s.t

$1x_1 - 1s_1 + 1a_1 = 125$

$1x_1 + 1x_2 - 1s_2 + 1a_2 = 350$

$2x_1 + 1x_2 + 1s_3 = 600$

$x_1, x_2, s_1, s_2, s_3, a_1, a_2 \geq 0$

The initial simplex tableau is shown below:

Table 11

Basis	C_B	x_1 -2	x_2 -3	s_1 0	s_2 0	s_3 0	a_1 -M	a_2 -M	
a_1	-M	1	0	-1	0	0	1	0	125
s_2	-M	1	1	0	-1	0	0	1	350
a_3	0	2	1	0	0	1	0	0	600
z_j		-2M	-M	M	M	0	-M	-M	-475M
$c_j - z_j$		-2+2M	-3+M	-M	-M	0	0	0	

At the first iteration, x_1 is brought into the basis and a_1 is eliminated. After dropping the a_1 column from the tableau, the result of the first iteration is as follows:

Table 12

Basis	C_B	x_1 -2	x_2 -3	s_1 0	s_2 0	s_3 0	a_2 -M	
X_1	-2	1	0	-1	0	0	0	125
a_2	-M	0	1	1	-1	0	1	225
s_3	0	0	1	2	0	1	0	350
z_j		-2	-M	2 – M	M	0	-M	
$c_j - z_j$		0	-3 + M	-2+M	-M	0	0	-250 – 225M

Continuing with two more iterations of the simplex method provides the final simplex tableau shown below:

Table 13

Basis	C_B	x_1 -2	x_2 -3	s_1 0	s_2 0	s_3 0	
x_1	-2	1	0	0	1	1	250
x_2	-3	0	1	0	-2	-1	100
s_1	0	0	0	1	1	1	125
z_j		-2	-M	2 – M	M	0	
$c_j - z_j$		0	-3 + M	-2+M	-M	0	-800

The simplex method provides the optimal solution with x_1 = 250, x_2 = 100, s_1 = 125, s_2 = 0, and s_3 = 0. Note however that the value of the objective function is -800 in the final simplex tableau. We must now multiply this value by -1 to obtain the value of the objective function to the original minimization problem (total cost = 800).

2.7 Special Class

1. **Infeasibility:** This occurs when an optimal solution is reach but one or more artificial variables remained in the solution with a positive value. Degeneracy occurs whenever there is no solution to the linear program that satisfies all the constraints, including the non negativity constraints.

2. **Unboundedness:** Occurs if the value of the solution may be made infinitely large without violating any constraints. Unboundness indicates error in the problem formulation. In this case, the simplex rule for determining the variable to be removed from the solution will not work.

3. **Alternate Optimal Solutions:** Exercise of two or more optimal solutions. This occurs whenever the objective function line is parallel to one of the constraint lines. If the linear programming problem has alternate optimal solution, cj – zj will equal zero for one or more of the variables nor in solution.

4. **Degeneracy:** Occurs when one or more of the variables in the basic solution has a value of zero. This does not create problem when it occurs at optimal solution. But if the situation occurs at the early stage of iterations, an optimal solution may never be obtained since each successive iteration will alternative between the same set of non optimal points.

DUALITY THEORY AND SENSITIVITY ANALYSIS

Duality is an important concept in LP problem. Every LP problem has and associated LP problem called the **Dual**. If the original problem known as the primal problem, is a maximizing one then the dual formation is a minimizing one and vice versa. A fundamental property of the primal-dual relationship is that the optimal solution to either the primal or dual problem also provides that optimal solution to the other. In case where the primal and dual problems differ in terms of computational difficulty, we can choose the easier problem to solve.

Formulation of the Dual Problem

Consider the high tech industries problem described above:

Max $50x_1 + 40x_2$

s.t.

$3x_1 + 5x_2 \leq 150$ assembly time

$1x_2 \leq 20$ portable display

$8x_1 + 5x_2 \leq 300$ warehouse space

$x_1, x_2 \geq 0$

The High Tech dual problem is as follows:

Min $150u_1 + 20u_2 + 300u_3$

s.t.

$3u_1 + 8u_3 \geq 50$

$5u_1 + 1u_2 + 5u_3 \geq 40$

$u_1, u_2 \geq 0$

The following general statements can be borne in mind about the dual problem.

1. The dual is a minimization problem
2. When the primal has n decision variables, the dual will have n constraints. The first constraint of the dual is associated with variable x_1 in the primal and so on.
3. When the primal has m constraints, the dual will have m decision variables. Dual variable u is associated with the first primal constraint and soon.

4. The r.h.s of the primal constraints becomes the objective function co-efficient in the dual.
5. The objective function co-efficient of the primal become the r.h.s. of the dual constraints.
6. The constraint co-efficient of the i^{th} primal variable becomes the co-efficient in the i^{th} constraint of the dual.

The Optimal Dual Solution

Obtain the standard and tableau form as follows:

Max $(-150u_1 - 20u_2 - 300u_3 + 0s_1 + 0s_2 - Ma_1 - Ma_2)$

s.t.

$3u_1 + 8u_2 - s_1 + a_1 \leq 50$

$5u_1 + u_2 + 5u_3 - s_2 + a_2 \leq 40$

$u_1, u_2, u_3, s_1, s_2, a_1, a_2 \geq 0$

The Initial Simplex Tableau is:

Table 14

Basis	C_B	U_1 -150	U_2 -20	U_3 -300	S_1 0	S_2 0	a_1 -M	a_2 -M	
a_1	-M	3	0	8	-1	0	1	0	50
a_2	-M	5	1	5	-1	-1	0	1	40
Z_j		-8M	-M	-13M	M	M	-M	-M	
$c_j - z_j$		-150 + M	-20 + M	-300 + 13M	-M	-M	0	0	-90M

At the first iteration, u_3 is brought into the basis of a_1 is removed.

The second tableau, with the a_1 column dropped, is shown below.

Table 15

Basis	C_B	U_1 -150	U_2 -20	U_3 -300	S_1 0	S_2 0	a_2 -M	
u_3	300	3/8	0	8	-1/8	0	0	50/8
a_2	-M	25/8	1	0	5/8	-1	1	70/8
		$\frac{(-900 - 25m)}{8}$	-M	-300	$\frac{300 - 5M}{8}$	M	-M	$\frac{-15,000 - 70M}{8}$
Z_j $c_j - z_j$		$\frac{(-300+25M)}{8}$	-20+M	0	$\frac{-300+5M}{8}$	-M	0	

At the second iteration u_1 is brought into the basis, and a_2 is removed. The third tableau, with the a_2 column removed is:

Table 16

		U_1	U_2	U_3	S_1	S_2	
Basis	C_B	-150	-20	-300	0	0	
u_3	-300	0	-3/25	1	5/25	3/25	26/5 = 5.20
u_1	-150	1	8/25	0	5/25	-8/25	14/5 = 2.80
Z_j		-150	-12	-300	30	12	-1980
$c_j - z_j$		0	-8	0	30	-12	

The optimal solution has been reached with $u_1 = 14/5$, $u_2 = 0$, $u_3 = 26/5$, $s_1 = 0$, $s_2 = 0$. The value of the objective function for the optimal dual solution be – (-1980) = 1980.

Observe that the original High Tech Industries problem yield also the same value of the objective function. Thus, the optimal value of the objective function is the same for both. This relationship is true for all optimal and dual LP problems.

Interpretation of Dual Prices

The dual price of a binding constraint provides valuable guidance because it indicates to management, the extra contribution they would gain from increasing by one unit the amount of scarce resource. As an example, the dual price of Assembly time is N2.8 i.e. each unit of first resources contributes N2.80 in the objective function, while each unit of 3rd resource contributes N5.20 in the objective function. This conclusion is of course true so long as the current primal (dual) solution is optimal and feasible.

Using the dual to identify the primal solution

Recall the fundamental property of the primal-dual property, stated in the preamble, also observe that when the primal problem is solve by the simplex method, the optimal values of the primal variables appear in the right-most column of the final tableau and the dual prices (values of the dual variables) are found in the zj row. Since the final simplex tableau of the dial problem provides the optimal values of the dual variables, the values of the primal variable should be found in the zj row of the optimal dual tableau. This is infact the case and is formally stated as follows: **Given the simplex tableau corresponding to the optimal dual solution, the optimal values of the primal decision variable are given by the Zj entries for the surplus variables; furthermore, the**

optimal values of the primal slack variables are given by the negative of the Cj – Zj entries for the Uj variables.

This property enables us to use the final simplex tableau for the dual of the High Tech problem to determine the optimal solution of x_1 = 30 units of the Deskpro and x_2 = 12 units of the portable. These optimal values of x_1 and x_2 as well as the values for all primal slack variables are given in the Zj and Cj – Zj rows of the final simplex tableau of the dual problem, which are shown again below.

Table 17

Basis	C_B	U_1	U_2	U_3	S_1	S_2	
		-150	-20	-300	0	0	
u_3	-300	0	-3/25	1	5/25	3/25	26/5
u_1	-150	1	8/25	0	5/25	-8/25	14/5
Z_j		-150	-12	-300	30	12	
$c_j - z_j$		0	-8	0	30	-12	-1980

Post-Optimal Analysis (Sensitivity Analysis)

After the optimal solution of a LPP has been reached, it is desired to study the effect of discrete changes in the different co-efficient of the problem on the current optimal solution. This study is referred to as sensitivity analysis. The usual sensitivity analysis for LP programs involving computing ranges for the objective function co-efficient and the right-hand-side value, as well as the dual prices.

In general, these changes may result in one of these cases:

1. The optimal solution remains unchanged i.e. the basic variable and their value remains essentially unchanged.
2. The basic variables remain the same but their values are changed
3. The basic solution changes completely.

Next, we illustrate sensitivity analysis with the objective function co-efficient.

Changes in the co-efficient of the Objective Function

Changes in the coefficients of the objective function can only affect the net evaluation row cj – zj equation and hence the optimality of the problem. In order to carry out the sensitivity analysis, a range is placed on the coefficients value called the range of optimality.

Recall the High Tech industries problem, the LPP is restarted below

Max $50x_1 + 40x_2$

s.t.

$3x_1 + 5x_2 \leq 150$ assembly time

$1x_2 \leq 20$ portable display

$8x_1 + 5x_2 \leq 300$ warehouse space

$x_1, x_2 \geq 0$

The final simplex tableau of the above LPP is reproduced below.

Table 18

		X_1	X_2	S_1	S_2	S_3	
Basis	C_B	50	40	0	0	0	
X_2	40	0	1	8/25	0	-3/25	12
S_2	0	0	0	-8/25	1	3/25	8
X_1	50	1	0	-5/25	0	5/25	30
Zj		50	40	14/5	0	26/5	1980
$C_j - z_j$		0	0	-14/5	0	-26/5	

The range of optimality for an objective function coefficient, then, is determined by those coefficient values that maintain

$$Cj - zj \leq 0 \text{ for every } j \qquad \ldots (*)$$

Thus, to determine the range of optimality for an objective function coefficient, say, c_k, we complete the value of the left-hand-side of inequality (*), using c_k as the objective function coefficient for x_k.

Let us compute the range of quantity for c_1, the profit contribution per unit of the Deskpro using c_1 (instead of 50) as the objective function coefficient of x_1, the revisd final simplex tableau is as follows.

Table 19

		X_1	X_2	S_1	S_2	S_3	
Basis	C_B	50	40	0	0	0	
X_2	40	0	1	8/25	0	-3/25	12
S_2	0	0	0	-8/25	1	3/25	8
X_1	50	1	0	-5/25	0	5/25	30
Zj		C_1	40	$\frac{64 - c_1}{5}$	0	$\frac{C_1 - 24}{5}$	$480 + 30c_1$
$C_j - z_j$		0	0	$\frac{C_1 - 64}{5}$	0	$\frac{24 - c_1}{5}$	

The current solution will remain optimal as long as the value of c_1 results in $Cj - zj \leq 0$ for the two non-basic variables s_1 and s_3.

Hence we must have

$c_1 - 64 / 5 \leq 0$ and $24 - c_1 / 5 \leq 0$. Using these inequalities,

We obtain $c_1 \leq 64$ or $24 \leq c_1$ … (**)

Since c_1 must satisfy (**), the range of optimality for c_1 is given by $24 \leq c_1 \leq 64$.

To verify this range, we recomputed the final simplex tableau after reducing the valueo f c_1 to 30.

Table 20

		X_1	X_2	S_1	S_2	S_3	
Basis	C_B	30	40	0	0	0	
X_2	40	0	1	8/25	0	-3/25	12
S_2	0	0	0	-8/25	1	3/25	8
X_1	50	1	0	-5/25	0	5/25	30
Zj		30	40	34/5	0	6/5	1380
$C_j - z_j$		0	0	-34/5	0	-6/5	

Since $cj - zj \leq 0$ for all variable, the original solution is still optimal i.e. the optimal solution with $c_1 = 30$ is the same as the optimal solution with $c_1 = 50$. Note, however, the decrease in total profit from N1980 to N1380.

Exercise:

Try and investigate the range of optimality for c_1 if the profit contribution per unit were reduced even further say to N20

LINEAR PROGRAMMING: TRANSPORTATION PROBLEM

This section presents the transportation model and its variants. In the obvious sense, the model deals with the determination of a minimum-cost plan for transporting a single commodity from a number of sources (e.g. factories) to a number of destinations (e.g. warehouses). The model can be applied to a wide variety of problems including inventory control, employment scheduling, personnel assignment, scheduling dam reservoir and many others.

It is basically a special type of linear programming problem that can as well be solved by the regular simplex method. However, its special structure allows the development of a solution procedure called the transportation technique i.e. computationally more efficient. The model also can be modified to account for multiple commodities.

The Transportation Model (definition and application)

The transportation model is formulated for a class problem with the following unique characteristics:

1. A product is transported from a number of sources to a number of destinations at the minimum possible cost, and
2. Each source is able to supply a fixed number of units of the product to each destination, which has a fixed demand for the product.

Since there is only one commodity a destination can receive its demand from one or more sources. The objective of the model is to determine the amount to be shipped from each source to each destination such that the total transportation cost is minimized.

Let x_{ij} represents the amount transported from source i to destination j: then the LP model representing the transportation problem is given generally as

$$\text{Minimize } z = \sum\sum c_{ij}x_{ij}$$

s.t.

$\sum x_{ij} \le ai$ $\quad$ i = 1, 2, …, m $\quad$ Supply

$\sum x_{ij} \le b_j$ $\quad$ j = 1, 2, …, n $\quad$ Demand

$Xij \ge 0$ for all i and j.

We use the following notations in the above LPM of the transportation problem:

i = index for origins: i = 1, 2, …, m.

j = index for destinations: j = 1, 2, …, n.

cij = cost per unit of shipping from origin i to destination j.

aij = supply or capacity in units at origin i.

bj = demand in units at destination j.

The first set of constraints stipulates that the sum of the shipments from a source cannot exceed its supply; similarly, the second set requires that the sum of the shipments to a destination must satisfy its demand.

The model described above implies that the total supply ∑ai must at least equal total demand ∑bj. When the total supply equals total demand (∑ai = ∑bj) the resulting formulation is called a balanced transportation model. The model will be used to demonstrate the solution of a transportation problem.

Solution of the Transportation Problem

In this section, we introduce the details for solving the transportation model. The method uses the steps of the simplex method directly and differs only in the details of implementing the optimal and feasibility conditions.

The Transportation Technique

The basic steps of the TT are:

Step 1: Determine a starting feasible solution

Step 2: Determine an entering variable from among the non basic variables. If all such variables satisfy the optimality condition (of the simplex method), stop; otherwise go to Step 3.

Step 3: Determine the leaving variable of the current basic solution; then find the new basic solution. Return to Step 2.

Determination of the Starting Solution

Transportation model are solved within the context of a tableau. When the transportation tableau is used an initial solution can be found by several alternative methods, including the North West corner method, the minimum cell cost method (least cost) and Vogel's approximation method.

The steps of the North West corner method can be summarized as:

1. Allocate as much as possible to the cell in the upper left – hand corner subject to the supply and demand constraints.
2. Allocate as much as possible to the next adjacent feasible cell.
3. Repeat step 2 until all rim requirement are met.

Also the steps of the least cost method can be summarized as

1. Allocate as much as possible to the feasible cell with the minimum transportation cost.
2. Repeat step 1 until all rim requirements are met.

For the Vogel's approximation method, the steps are:

1. Determine the penalty cost for each row and column by subtracting the lowest cell cost in the row or column from the next lowest cell cost in the same row or column.
2. Select the row or column with the highest penalty cost.
3. Allocate as much as possible to the feasible cell with the lowest transportation cost in the row or column having the highest penalty cost.
4. Repeat steps 1, 2, and 3 until all rim requirements have been met.

Once the initial basic feasible solution has been determined by any of the previous three methods, the next step is to solve the model for the optimal (i.e. minimum cost) solution. There are two basic solution methods: the stepping stone method and the modified distribution method (MODI) or method of multipliers. We will illustrate with the following example.

Example 9

Wheat is harvested in the Midwest and stored in grain elevators in the three cities – Kansas City, Omaha and Desmonies. These grains elevators supply three mills that produce flour, Chicago, St. Louis and Cincinnati. Grain is shipped to the mills in railroad can such capable of holding one ton of wheat. Each grain elevator is able to supply the following number of tons (i.e. railroad cars) of wheat to the mill on the monthly basis.

Table 21

(a)

Grain Warehouse		Supply
1.	Kansas City	100
2.	Omaha	175
3.	Desmoines	275
		600 tons

Each mill demands the following tons of wheat per month:

(b)

Mill		Demand
A.	Kansas City	200
B.	Omaha	100
C.	Desmoines	300
		600 tons

The cost of transporting one ton of wheat from each grain elevator (source) to each mill (destination) differs according to the distance and rail system. These costs are:

Table 22

	Mill		
Grain Elevator	Chicago A	St. Louis B	Cincinnati C
1. Kansas City	6	8	10
2. Omaha	7	11	11
3. Desmoins	4	5	12

Determine how many tons of wheat to transport from each grain elevator to each mill on a monthly basis in order to minimize the total cost of transportation.

Solution:

The linear programming model for the problem is formulated as follows

$$\text{Minimize} \quad Z = 6x_{1A} + 8x_{1B} + 10x_{1C} + 7x_{2A} + 11x_{2B} + 11x_{2C} + 4x_{3A} + 5x_{3B} + 12x_{3C}$$

s.t.

$$\left.\begin{aligned} x_{1A} + x_{1B} + x_{1C} &= 150 \\ x_{2A} + x_{2B} + x_{2C} &= 175 \\ x_{3A} + x_{3B} + x_{3C} &= 275 \end{aligned}\right\} \quad \text{supply constraints}$$

$$\left.\begin{aligned} x_{1A} + x_{1B} + x_{1C} &= 200 \\ x_{2A} + x_{2B} + x_{2C} &= 100 \\ x_{3A} + x_{3B} + x_{3C} &= 200 \end{aligned}\right\} \quad \text{demand constraints}$$

$$x_{ij} \geq 0$$

In this model, the decision variables, xij represent the number of tons of wheat transported from each grain elevator, i (where i = 1, 2, 3) to each mill j (where j = A, B, C).

Table 23

	A	B	C	Supply
1	6 X_{1A}	8 X_{1B}	10 X_{1C}	150
2	7 X_{2A}	11 X_{2B}	11 X_{2C}	175
3	4 X_{3A}	5 X_{3B}	12 X_{3C}	275
Demand	200	100	300	600

Determine a feasible solution: Using the northwest corner method

Table 24

The Initial NW corner allocation

	A	B	C	Supply
1	6 150	8	10	150
2	7	11	11	175
3	4	5	12	275
Demand	200	100	300	600

Table 25

The Second NW corner allocation

	A	B	C	Supply
1	(6) 150	(8)	(10)	150
2	(7) 50	(11)	(11)	175
3	(4)	(5)	(12)	275
Demand	200	100	300	600

Table 26

The Third NW corner allocation

	A	B	C	Supply
1	(6) 15	(8)	(10)	150
2	(7) 50	(11) 100	(11)	175
3	(4)	(5)	(12)	275
Demand	200	100	300	600

Table 27

The Initial Solution from NWC is:

	A	B	C	Supply
1	(6) 150	(8) [redacted]	(10) [redacted]	150
2	(7) 50	(11) 100	(11) 25	175
3	(4) [redacted]	(5) [redacted]	(12) 275	275
Demand	200	100	300	600

The transportation cost of the solution is computed by substituting the cell allocations (i.e. the amounts transported)

$x_{1A} = 150, x_{2C} = 25, x_{2A} = 50, x_{3C} = 275, x_{2B} = 100$

$Z = 6x_{1A} + 8x_{1B} + 10x_{1C} + 7x_{2A} + 11x_{2B} + 11x_{2C} + 4x_{3A} + 5x_{3B} + 12x_{3C}$

$Z = 6\ (150) + 8\ (0) + 10\ (0) + 7\ (50) + 11\ (100) + 11\ (25) + 4\ (0) + 5\ (0) + 12\ (275).$

= N5,925

Using the minimum cell cost method (least cost rule) and following the steps highlighted above strictly.

Table 28

The Initial Solution minimum cell cost allocation

	A	B	C	Supply
1	(6) [redacted]	(8)	(10)	150
2	(7) [redacted]	(11) 100	(11) 25	175
3	(4) 200	(5)	(12)	275
Demand	200	100	300	600

Table 29

The second minimum cell cost allocation

	A	B	C	Supply
1	6 [redacted]	8	10	150
2	7 [redacted]	11	11	175
3	4 200	5 75	12 [redacted]	275
Demand	200	100	300	600

Table 30

The initial Solution from MCC

	A	B	C	Supply
1	6 [redacted]	8 25	10	150
2	7 [redacted]	11 [redacted]	11 175	175
3	4 200	5 75	12 [redacted]	275
Demand	200	100	300	600

$x_{3A} = 200$, $x_{3C} = 75$, $x_{2C} = 175$, $x_{1B} = 25$, $x_{1C} = 125$.

The total cost of this initial solution is N4,550 as compared to the N5,925 total cost of the NW corner initial solution.

The third method for determining an initial solution, Vogel's approximation method (also called VAM), is based on the concept of **penalty cost** or regret.

Table 31

The VAM penalty cost

	A	B	C	Supply
1	6	8	10	150
2	7	11	11	175
3	4	5	12	275
Demand	200	100	300	600
	2	3	1	

Table 32

The Initial VAM allocations

	A	B	C	Supply
1	6	8	10	150
2	7 175	11	11	175
3	4	5	12	275
Demand	200	100	300	600
	2	3	2	

Table 33

The Second VAM allocation

	A	B	C	Supply
1	[6]	[8] [redacted]	[10]	150
2	[7] 175	[11] [redacted]	[11] [redacted]	175
3	[4]	[5]	[12]	275
Demand	200	100	300	600
	2		2	

Table 34:

The third VAM allocation

	A	B	C	Supply
1	[6] [redacted]	[8] [redacted]	[10]	150
2	[7] 175	[11] [redacted]	[11] [redacted]	175
3	[4] 25	[5] 100	[12]	275
Demand	200	100	300	600
			2	

Table 35

The Initial VAM solution

	A	B	C	Supply
1	6 [redacted]	8 [redacted]	10 150	150
2	7 175	11 [redacted]	11 [redacted]	175
3	4 25	5 100	12 150	275
Demand	200	100	300	600

$x_{1C} = 150$, $x_{2A} = 175$, $x_{3A} = 25$, $x_{3B} = 100$, $x_{3C} = 150$

The total cost of this initial solution is N5,125 which is not high as the N5,925 obtained by the NWC initial solution or as low as the MCC solution of N4,550. Like the MCC method, VAM typically results in a lower cost for the initial solution than the NWC method.

Step 2: Determination of Optimal Solution (Modified Distribution Method [MODI] or Method of Multipliers)

In order to demonstrate MODI, we will again use the initial solution obtained by the minimum cell cost method.

Table 36

The minimum cell cost initial solution

$V_A = V_B = V_C =$

From	A	B	C	Supply
To U_1 1	6 [redacted]	8	10	150
U_2 2	7 [redacted]	11 [redacted]	11 175	175
U_3 3	4 200	5 75	12 [redacted]	275
Demand	200	100	300	600

The extra left – hand column with the u_1 symbols and the extra row with the v_j symbols represents column and row values that must be computed in MODI. These values are computed for all cells with allocation by using the following formula: $u_i + v_j = c_{ij}$

The value of cij is the transportation cost for cell ij. For example, the formula for cell 1B is $u_1 + v_B = C_{1B}$ and since $C_{IB} = 8$

$u_1 + v_B = 8$

The formulas for the remaining presently allocated cells are:

x_{1C}: $u_1 + v_c = 10$

x_{2C}: $u_2 + v_c = 11$

x_{3A}: $u_3 + v_A = 4$

x_{3B}: $u_3 + v_B = 5$

Thus, if we let $u_1 = 0$, then we can solve for all remaining u_i and v_j values

	$v_B = 8$
x_{1C}: $u_1 + v_c = 10$	$v_c = 10$
x_{2C}: $u_2 + v_c = 11$	$u_2 = 1$
x_{3A}: $u_3 + v_A = 4$	$v_A = 7$
x_{3B}: $u_3 + v_B = 5$	$u_3 = -3$

Now, all the ui and vj values can be substituted into our table as shown in Table 2

Table 37

The Initial Solution with all ui and vj values

$V_A = 7$ $V_B = 8$ $V_C = 10$

From		A	B	C	Supply
To $U_1 = 0$	1	6	- 8 25	10 125	150
$U_2 = 1$	2	7	11	11 175	175
$U_3 = -3$	3	- 4 200	+ 5 75	12	275
Demand		200	100	300	600

Next we use the following formula to evaluate all empty cells (non – basic variables) $c_{ij} - u_i - v_j = k_{ij}$

Where k_{ij} equals the cost increase or decrease that would occur by allocating to a cell.

For the empty cells in (Table 2)

$x_{1A} : k_{1A} = c_{1A}$ $\quad u_1 - v_A = 6 - 0 - 7 = -1$

$x_{2A} : k_{2A} = c_{2A}$ $\quad u_2 - v_A = 7 - 1 - 7 = -1$

$x_{2B} : k_{2B} = c_{2B}$ $\quad u_2 - v_B = 11 - 1 - 8 = +2$

$x_{3C} : k_{3C} = c_{3C}$ $\quad u_3 - v_C = 12 - (-3 - 10) = +5$

This indicates that either cell 1A or 2A will decrease cost by N1 per allocated ton. We can select either cell 1A or 2A to allocate since they are tied at -1. If cell 1A is selected as the entering non basic variable, then the stepping stone path (loop) for the cell must be determine so that we know how much to reallocate.

Table 38

The second iteration of the MOD solution method

$V_A = V_B = V_C =$

From	A	B	C	Supply
To U_1 = 1	+ 6 25	[redacted] 8 [redacted]	10 125	150
U_2 = 2	[redacted] 7 [redacted]	[redacted] 11 [redacted]	11 175	175
U_3 = 3	- 4 175	+ 5 100	[redacted] 12 [redacted]	275
Demand	200	100	300	600

The ui and vj values for (table 3) must now be recomputed using our formula for the allocated cells

Let $u_1 = 0$

$x_{1A} : u_1 + v_A = 6$ $\quad v_A = 6$

$x_{1C}: u_1 + v_C = 10$ $\quad v_C = 10$

x_{2C} : $u_2 + v_C = 11$ $u_2 = 1$

x_{3A} : $u_3 + v_A = 4$ $u_3 = -2$

x_{3b} : $u_3 + v_B = 4$ $v_B = 7$

Table 39

The new ui and vj values for second iteration

$V_A = 6$ $V_B = 7$ $V_C = 10$

From	A	B	C	Supply
To $U_1 = 0$ 1	+ 6 25	8	10 125	150
$U_2 = 1$ 2	7	11	11 175	175
$U_3 = 2$ 3	4 175	+ 5 100	12	275
Demand	200	100	300	600

The cost changes for the empty cells are not computed using the formula

$c_{ij} - u_i - v_j = k_{ij}$

x_{1B} : $k_{1B} = c_{1B}$ $u_1 - v_B = 8 - 0 - 7 = -1$

x_{2A} : $k_{2A} = c_{2A}$ $u_2 - v_A = 7 - 1 - 6 = 0$

x_{2B} : $k_{2B} = c_{2B}$ $u_2 - v_B = 11 - 1 - 7 = +3$

x_{3C} : $k_{3C} = c_{3C}$ $u_3 - v_C = 12 - (-2) - 10 = +4$

Since none of these values is negative, the solution shown above in (table 39) is optimal.

Thus the optimal solution is calculated as: 6 (25) + 10 (125) + 11 (175) + 4 (175) + 5 (100) = N4,525

Steps of MODI (Methods of Multipliers)

(1) Develop an initial solution using one of the three methods available

(2) Compute u_1 and v_j values for each row and column by applying the formula $u_i - v_j = c_{ij}$ to each cell that has an allocation

(3) Compute the cost change, k_{ij}, for each allocated cell using the formula $c_{ij} - u_i - v_j = k_{ij}$.

(4) Allocate as much as possible to the empty cell that will result in the greatest net decrease in cost (k_{ij}). Allocate according to the stepping stone path or the selected cell.

(5) Repeat steps 2 through 4 until all k_{ij} values are positive or zero.

DEGENERACY

In the entire tableau showing solution to our wheat transportation problem, the following condition was met. m row + n columns – 1 = no of cells with allocation.

For example, in any of the balanced tableau for wheat transportation, the no of rows are 2 (i.e. m = 3) and the no of columns are 3 (i.e. n = 3), thus:

3 + 3 – 1 = 5 cells with allocations.

Five cells with allocation always existed for these tableaus thus our condition for normal solution was met. When this condition is not met and less then m + n – 1 cells have allocations, the tableau is said to be degenerate.

THE ASSIGNMENT MODEL

Consider the situation of assigning m jobs (or workers) to n machines. A job i (= 1, 2, …, m) when assigned to machine j (= 1, 2, …, n) incurs a cost c_{ij}. The objective is to assign the jobs to the machines (one job per machine) at the least total cost. The situation is known as the assignment problem.

Table 40

	machine						
	1	2	.	.	.	n	
1	c_{11}	c_{12}	.	.	.	c_{1n}	1
2	c_{21}	c_{22}	.	.	.	c_{2n}	1
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
m	c_{m1}	c_{m2}	.	.	.	c_{mn}	1

The formulation of this problem may be regarded as a special case of the transportation model. Here jobs represents “sources” and machines represent “destinations”. The supply available at each sources is 1; i.e. $a_i = 1$ for all i, similarly, the demand required at each destination is 1; i.e. bi = 1 for all j. the cost of (assigning) job i to machine j is c_{ij}.

The mode is thus given by:

Minimize z: $\sum\sum c_{ij}x_{ij}$

Subject to

$\sum x_{ij} = 1, \qquad i = 1, 2, \ldots, m$

$\sum x_{ij} = 1, \qquad j = 1, 2, \ldots, n$

$x_{ij} = 0$ or 1

$$x_{ij} = \begin{cases} 0 & \text{if the } j^{th} \text{ job is not assigned to the } i^{th} \text{ machine} \\ 1 & \text{otherwise} \end{cases}$$

The Steps of the assignments solution method can be summarized as:

(1) Perform row reduction by subtracting the minimum value in each row from all other row values.

(2) Person column reductions by subtracting the minimum value in each column from all other column values

(3) In the completed opportunity cost table, cross out all zeros using the minimum number of horizontal and/or vertical lines

(4) If less than m lines are required (where m = the no of rows or columns), subtract the minimum uncrossed value from all other uncrossed values, and add this same minimum value to all cells where two lines intersect. All other values are unchanged.

(5) If m lines are required, the optimal solution exists and m unique assignments are made. If less m lines are required, repeat step 4.

Example 1: Court Scheduling

A court administrator is in the process of scheduling four court dockets. Four judges are available to be assigned, one judge to each docket. The court administrator has information regarding the types of cases on the dockets as well as data indicating the relative efficiency of each of the judges in processing different types of court cases. Based on this information, the court administrator has compiled the data in the Table below:

Table 41

	Docket			
Judge	1	2	3	4
1	14	13	17	14
2	16	15	16	15
3	18	14	20	17
4	20	13	15	18

The table shows the estimates of the no of court days each judge would require in order to completely process each court docket. The court administrator would like to assign the four judges so as to minimize the total no of court – days needed to process all four dockets

Table 42

	Docket			
Judge	1	2	3	4
1	1	0	4	1
2	1	0	1	0
3	4	0	6	3
4	7	0	2	5

Table 43

	Docket			
Judge	1	2	3	4
1	0	0	3	1
2	0	0	0	0
3	3	0	5	3
4	6	0	1	5

Table 44

	Docket			
Judge	1	2	3	4
1	0	0	3	1
2	0	0	0	0
3	2	0	4	2
4	5	0	0	4

Table 45

	Docket			
Judge	1	2	3	4
1	[0]	0	3	1
2	0	0	0	[0]
3	2	[0]	4	2
4	5	0	[0]	4

Final Assignment

Judge 1 to Docket 1	14
Judge 2 to Docket 4	15
Judge 3 to Docket 2	14
Judge 4 to Docket 3	15

Total days 58

Thus the optimal assignment of the four judges results in 58 judges – days to clear the four dockets.

The Assignment Technique for Maximizing

A maximizing assignment problem typically involves making assignments so as to maximize contribution. To maximize only step 1 from the above differs, the column are reduced by the largest number of each column. From then on the same rules apply that are used for minimizing.

Example: Suppose the figures below relate to contribution and that it is required to maximize contribution.

Table 46

	W	X	Y	Z
A	25	18	23	14
B	38	15	53	23
C	15	17	41	30
D	26	28	36	29

Table 47

Column

	W	X	Y	Z
A	13	10	30	16
B	0	13	0	7
C	23	11	12	0
D	12	0	17	1

Table 46

~ Row

	W	X	Y	Z
A	3	0	20	6
B	0	13	0	7
C	23	11	12	0
D	12	0	17	1

Table 49

~

	W	X	Y	Z
A	3	0	20	6
B	0	13	0	7
C	23	11	12	0
D	12	0	17	1

Table 50

~

	W	X	Y	Z
A	0	0	17	6
B	0	16	0	10
C	20	11	9	0
D	9	0	14	1

Table 51

~

	W	X	Y	Z
A	0	0	17	6
B	0	16	0	10
C	20	11	9	0
D	9	0	14	1

Table 52

~

	W	X	Y	Z
A	0	0	17	6
B	0	16	0	10
C	20	11	9	0
D	9	0	14	1

The assignments

C to Z	20
D to X	28
A to W	25
B to Y	53
	136

Unequal sources and destinations

To solve assignment problems in the manner described the matrix must be square i.e. the supply must equal requirement. Where the supply and requirements are not equal, an artificial source or destination must be created to square the matrix. The costs/mileage / contribution etc. for the fictitious row or column will be zero throughout.

Solution method, having mad the sources equal to the destination, the solution method will be as normal, treating the fictitious elements as though they are real. The solution method will automatically assign a source or destination to the fictitious row or column and the resulting assignment will incur zero or gain zero contribution.

Example:

A foreman has four fitters and has been asked to deal with five jobs. The times or each job are estimated as follows:

Table 53

	Fitters			
Job	Alf	Bill	Charlie	Dave
1	6	12	20	12
2	22	18	15	20
3	12	16	18	15
4	16	8	12	20
5	18	14	10	17

Dummy fitter inserted to square the matrix

Table 54

	A	B	C	D	Dummy
1	6	12	20	12	0
2	22	18	15	20	0
3	12	16	18	15	0
4	16	8	12	20	0
5	18	14	10	17	0

Table 55

	A	B	C	D	Dummy
1	0	15	10	0	0
2	16	10	5	8	0
3	16	8	8	3	0
4	10	0	2	8	0
5	12	6	0	5	0

Table 56

	A	B	C	D	Dummy
1	0	4	10	0	3
2	13	7	2	5	0
3	3	5	5	0	0
4	10	0	2	8	3
5	12	6	0	5	3

Table 57

	A	B	C	D	Dummy
1	0	4	10	0	3
2	13	7	2	5	0
3	13	5	5	0	0
4	10	0	2	8	3
5	12	6	0	5	3

5 lines so optimum assignments:
B to 4
C to 5
A to 2
Dummy to 2
D to 3
Hence Job 2 is not done

NETWORK ANALYSIS/PROJECT MANAGEMENT

1. INTRODUCTION TO NETWORK ANALYSIS

Definition

Network analysis is a generic term for a family or group of related techniques developed to aid management to plan and control project.

Why Network Analysis?

It is most valuable when project are complex, large and restricted i.e. completed within stipulated or cost limits. It illustrates the way in which parts of the project are organized and determine the time duration of these projects. It also aids planning and scheduling of projects.

Background

In the 1950's, a basic form of Network analysis was being used in the UK and USA in order to reduce project times i.e. the amount of time spent to complete a project.

In 1950, the US Naval Special Projects Office set up a team to devise a means of controlling the planning of complex projects. The team came up with a network technique known as PERT i.e. Programme Evaluation and Review Technique. This technique was used in planning and controlling the development of the Polaris missile and credited with saving two years in the development of missiles.

Since 1958, the technique has been improved upon and nowadays many variants exist which handle, in addition to basic time, costs, resources, probabilities and combination of these factors. Variety of names exists and among the commonly used are: CPP (Critical Path Planning), CPA (Critical Path Analysis), CPM (Critical Path Method), PERT etc.

Network Techniques (CPM/PERT)

The network techniques that are used for project analysis are CPM (Critical Path Method) and PERT (Programme Evaluation Review Techniques). They were developed at approximately the same time, although independently during the late 1950's. The fact that they have been frequently and widely used in such a short of period of time attests to their value as management science techniques.

The basic difference between them is that CPM is a deterministic technique while PERT is a probabilistic technique.

2. BASIC NETWORK TERMINOLOGY

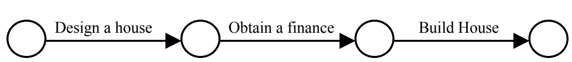
i. Activity: This is a task or job which takes time and resources e.g. Building of house, construction of bridges etc. It represented in a network by an arrow (→) in which the head indicates the end of an activity and the tail indicates the beginning of an activities.

ii. Event: It is a point in time and it indicates the start and finish of an activity. It is represented in a network by a circle or nodes (O). Note that the establishment of activities automatically determines the event.

iii. Dummy Activities: This is an activity which does not take time or resources. It is used merely to show logical dependences or sequences between activities so as not violate the rules of drawing network. It is represented in a network by a dotted arrow (--►)

iv. Predessor Activity/Successor Activity: This is an activity that precedes/succeeds an activity in a network or an activity that must be completed before the next activity can start (predecessor activity).

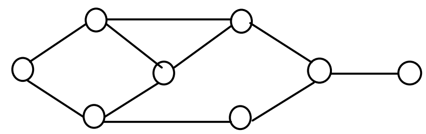
An activity that can start after the one before it has been completed (successor activity).

E.g 

From example, "design a house" is a predecessor activity for "obtain finance" while "building house" is a successor activity for "obtain finance".

v. Network: This is the combination of activities, events, dummy activities in a logical sequence according to the rules of drawing network.

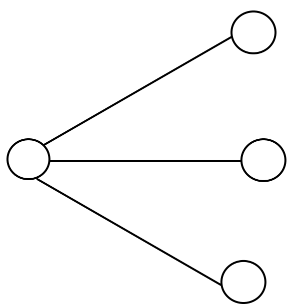
Example of a network;



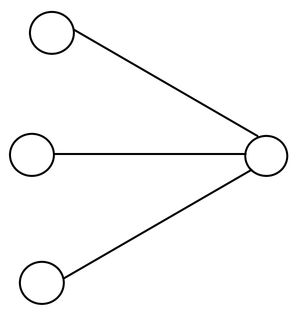
3. **RULES OF NETWORK CONSTRUCTION**

1. A complete network must have only one point of entry (start event) and only one point of exit (finish event).
2. Every activities must have one preceeding event (tail event) and one succeeding event (head event).

Note: Many activities may use the same tail event or the same head event. E.g.



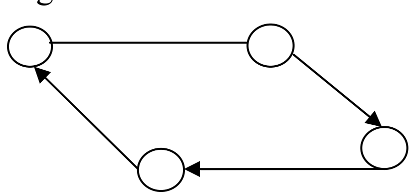
Tail event



Head Event

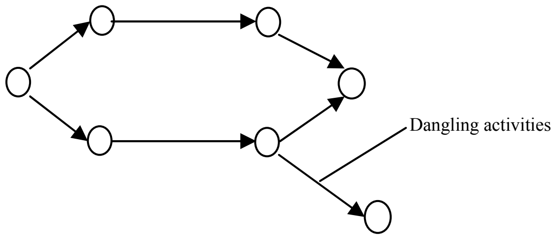
i.e. they can't use the same tail and head event.

3. No activity can start until its tail event is reached
4. An Event is not complete until all activities leading into it are complete
5. Loops are not allowed i.e. series of activities which leads back to the same event e.g



6. All activities must be tied into a network i.e. danglers is not allowed

Note: Danglers are the activities that do not link into the overall network. e.g



4. CONVENTION FOR DRAWING NETWORKS

1. Networks proceeds from left to right.
2. Networks are not drawn to scale but must be drawn neatly.
3. Arrows need to be drawn in the horizontal plane unless totally unavoidable, they must proceed from left to right.
4. Events should be progressively numbered from left to right.

Activity Identification

Activities can be identified in several ways, typical of the method to be found includes:

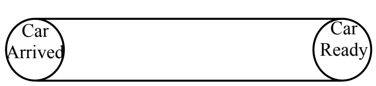
1. Shortened description of jobs e.g order materials
2. Alphabetic or numeric code e.g A,B,C or 100,130 etc.

Dummy Activities

Dummy activities are the one that does not consume time or resources but shoe logical relationship in a network. It is shown on a network by a dotted arrow.

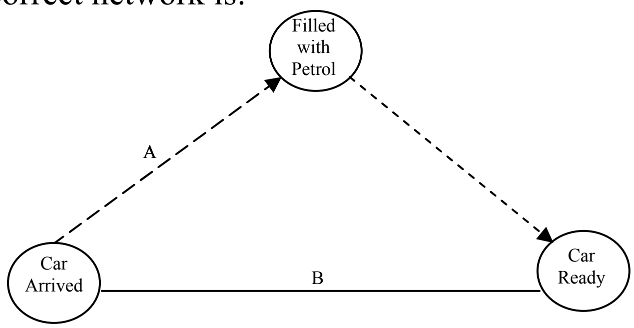
Example: Assume that a car arrive at a service station during which two independent activities takes place, filling with petrol (A) and topping up with oil (B).

This could be



Which is incorrect because it contradicts the rule that says two or more activities can't have the same head and tail event?

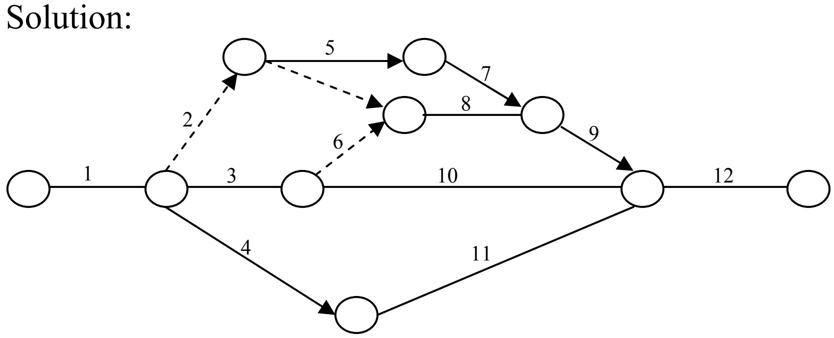
The correct network is:



Example 1: Draw the network of the following problem:

Activity	**Preceding Activities**
1	-
2, 3, 4	1
5	2
6	3
7	5
8	6 and 2
9	7 and 8
10	3
11	4
12	9, 10, 11

Solution:



Summary

1. Network analysis is used for the planning and controlling of large complex projects.

2. Network comprises activities which consumes time and resources and also event which is point in time. Activities is represented by an arrow (→) while event by a circle (O) in a network.

3. Network has one start and one head event. An event is not complete if the activity leading to it is not complete.

4. Two or more activities can't have the same head and tail event.

5. The length of the arrows representing the activities is not important because networks are not drawn to scale.

6. Dummy activities (represented by- - - -►) are necessary to show logical relationship. They do not consume time or resources. They become necessary as the network is drawn.

EXERCISE:

1. Draw the network of the following:

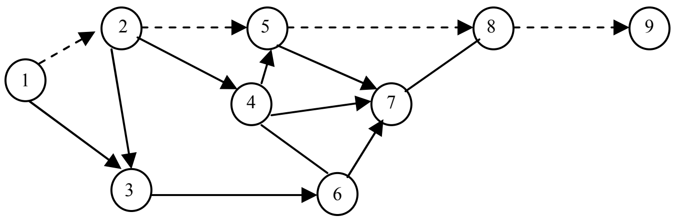
1→2	2→3	4→6	7→8
1→3	3→6	5→7	8→0
2→5	4→5	5→8	
2→4	4→7	6→7	

2.

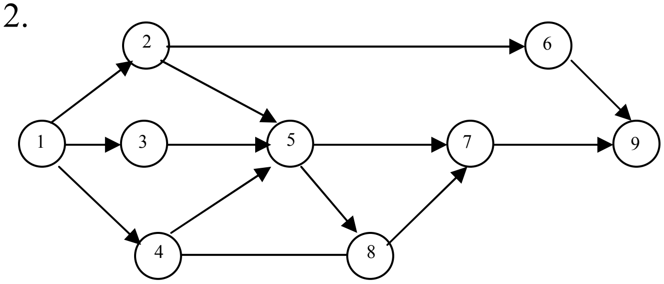
1→2	4→5	7→9
1→3	4→8	
1→4	5→7	
2→5	5→8	
2→6	8→7	
3→5	6→9	

Solution to Exercise

1.



2.



5. TIME ANALYSIS IN PROJECT SCHEDULING

Critical Path Calculations

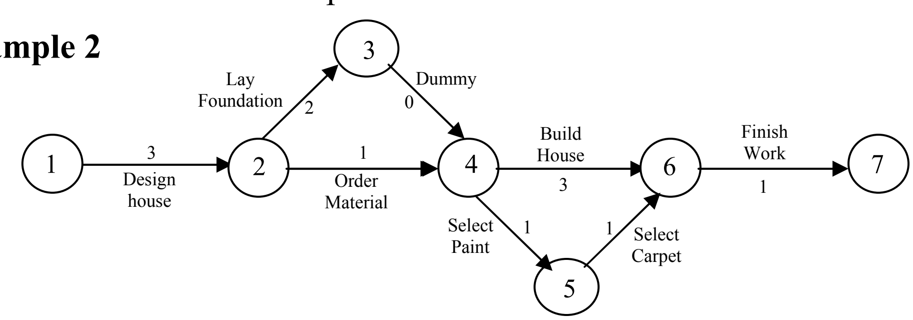
The critical path of network gives the shortest time in which the whole project can be completed. It is the chain of activities with the longest duration time.

It can be calculated using Inspection/Tree diagram method or Forward/Backward pass method.

Inspection/Tree Diagram Method

This is done by writing out all the paths in a network then the path with the longest duration time is the critical path.

Example 2



The paths are:

1. $1\xrightarrow{3}2\xrightarrow{2}3\xrightarrow{0}4\xrightarrow{3}6\xrightarrow{1}7$

Duration times 3 + 2 + 0 + 3 + 1 = 9

2. $1\xrightarrow{3}2\xrightarrow{2}3\xrightarrow{0}4\xrightarrow{1}5\xrightarrow{1}6\xrightarrow{1}7$

Duration times: 3 + 2 + 0 + 1 + 1 + 1 = 8

3. $1\xrightarrow{3}2\xrightarrow{1}4\xrightarrow{1}5\xrightarrow{1}6\xrightarrow{1}7$

Duration times: 3 + 1 + 1 + 1 + 1 = 7

4. $1\xrightarrow{3}2\xrightarrow{1}4\xrightarrow{3}6\xrightarrow{1}7$

Duration times: 3 + 1 + 3 + 1 = 8

The longest duration is 9 months

Therefore, the critical path is 1→2→3→4→6→7

5.2 Forward/Backward Pass Method

1. **Earliest Start Time,** which is also the forward pass. This is the earliest possible time at which a succeeding activity can start. The Formula is

$ET_j = \max(ET_i + t_{ij})$ where i is the starting node number of all activities ending at node j.

t_{ij} is the time for activity i → j

2. **Latest Start Time**, which is also the backward pass. This is the latest possible time at which a preceding activities can be completed without delaying beyond project duration time. The formula is:

$LT_i = \max(LT_i - t_{ij})$ where j is the ending node number of activities starting at i t_{ij} is the time for activity i→ j.

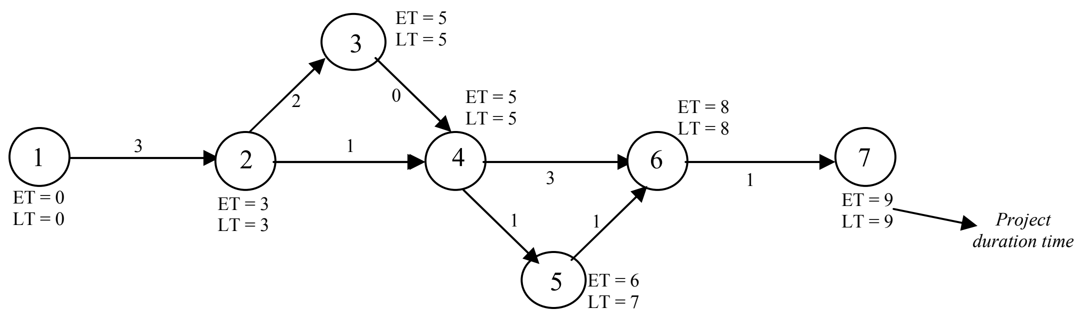
CRITICAL ACTIVITIES

This is the activities that have the same earliest start time and latest start time.

Critical Path (Estimation of project completion time)

The path that has the same earliest start time and latest start time is called the critical path which also corresponds to the chain of activities with the longest duration time. It is indicated in a network by double line or different colour in the network.

Example 3: Use backward/forward method to calculate critical path of example 2



For the earliest start time: $ET_j = \max (ET_i + t_{ij})$

$ET_1 = 0$, $ET2 = ET_1 + t_{12} = 0 + 3 = 3$

$ET_3 = (ET_2 + t_{23}) = 3 + 2 = 5$

$ET_4 = \max (ET_2 + t_{24}, ET_3 + t_{34}) = \max (3 + 1, 5 + 0) = 5$

$ET_5 = ET_4 + t_{45} = 5 + 1 = 6$

$ET_6 = \max (ET_4 + t_{46}, ET_5 + t_{56}) = \max (5 + 3, 6 + 1) = 8$

$ET_7 = ET_6 + t_{67} = 8 + 1 = 0$

For latest start time: $LTi = \min (LT_j – t_{ij})$

$LT_7 = 9, LT_6 = LT_7 + t_{67} = 9 – 1 = 8$

$LT_5 = LT_6 – t_{56} = 8 – 1 = 7$

$LT_4 = \min (LT_5 – t_{45}, LT_6 – t_{46}) = \min (7 – 1, 8 – 3) = 5$

$LT_3 = LT_4 – t_{34} = 5 – 0 = 5$

$LT_2 = \min (LT_3 – t_{23}, LT_4 – t_{24}) = \min (5 – 2, 5 – 1) = 3$

$LT_1 = LT_2 – t_{12} = 3 – 3 = 0$

Therefore, the critical path is 1→2→3→4→6→7 and the project duration time = 9 months.

Slack of an Activity

This is the time an activity can be delayed without affecting the overall project duration i.e. extra time available for completing an activities

Slack for activity $i - j = LT_j - ET_j - t_{ij}$

Where LT_j = Latest start time for activity j

ET_i = Earliest start time for activity i

t_{ij} = Activity time between i and j

Slack of an event

This is difference between the earliest start time and the latest start time of each event.

Note: The activities or event on the critical path does not have activity slack or event slack.

From the example above, the slack activity for 2→4 = 5 – 3 – 1 = 1 months i.e. there is extra one month. The slack event for (5); = $LT_5 - ET_5 = 7 - 6 = 1$ month

Note: Events is the only event not on the critical path.

6. PROBABILITY CONSIDERATIONS IN PROJECT SCHEDULING

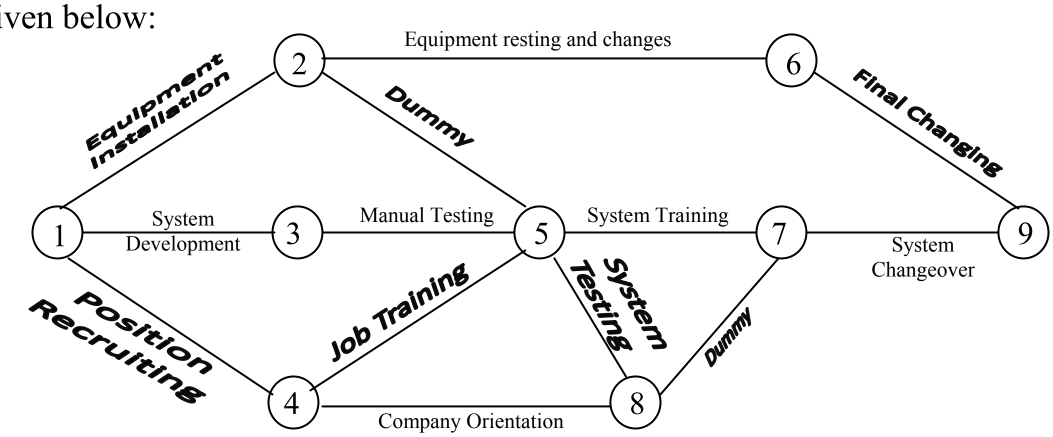
PERT Analysis

In CPM, all the activities time estimate has a single value i.e. we are assuming that the activity time estimate is known with certainty, but in reality, it is rare that activity time estimate can be known with certainty. This is true since projects that are networked tends to be unique as such there is little historical evidence to be used as basis to predict future occurrence. As an alternative to CPM, PERT (Project Evaluation and Review Technique) uses probabilistic activity times.

Example 4

Using the example below to demonstrate PERT

A Southern Textile Company has decided to install a new computerized order-processing system. In the past, order for the cloth the company produce were processed manually, which contributes to delay in delivering orders and as a result, lost sales. The company wants to know how long it will take to install the new system. The network is given below:



All this state in CPM network, we should assign a single-time estimate to each network activity. However, in PERT networks we will determine three time estimates for each activity which will enable us to estimate the mean and variance for a beta distribution of the activity times.

We assume that the activity time estimate can be described by beta distribution for several reasons:

1. The beta distribution mean and variance can be approximated by three estimates
2. The beta distribution is continuous and has no predetermined shape it will take on the shape that is indicated by the time estimate given.
3. It has become traditional to use beta distribution for PERT analysis

The three estimates are (1) most likely time (m) (2) optimistic time (a) (3) pessimistic time (b)

1. **Most Likely Time (m):** Is the time would most frequently occur if the activities were repeated many times
2. **Optimistic Time (a):** Is the shortest possible time that an activity could be completed assuming everything went right
3. **Pessimistic Time (b):** Is the longest possible time required for an activity to complete if everything went wrong.

These 3 estimates will be determined the mean and variance of beta distribution:

Mean (Expected time) = $\frac{a+4m+b}{6}$

Variance = $\left(\frac{b-a}{6}\right)^2$

The three estimates mean and variance of the southern textile network is given below:

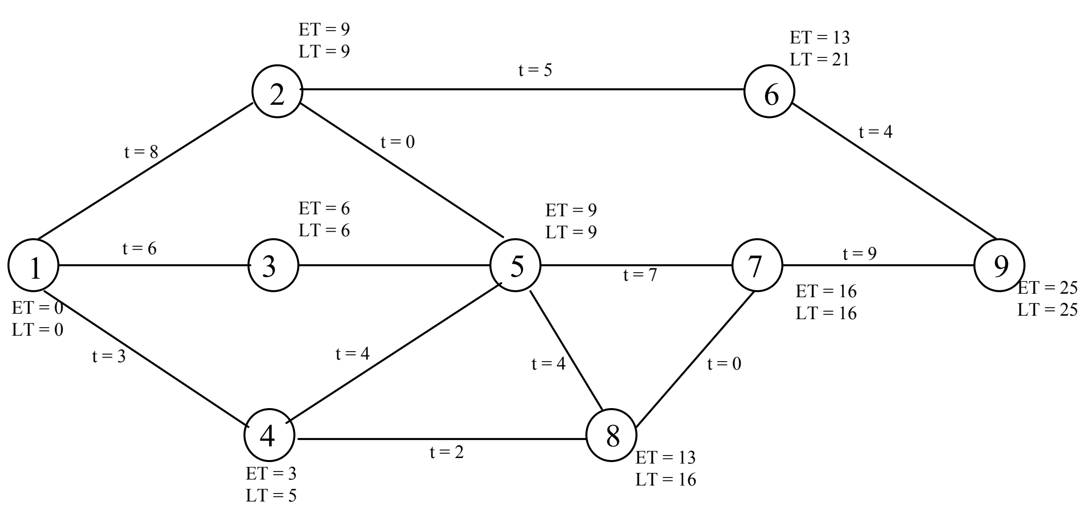
	Time estimate (week)			mean time	
Activities	**a**	**m**	**b**	**t**	**Variance**
1 – 2	6	8	10	8	4/9
1 – 3	3	6	9	6	1
1 – 4	1	3	5	3	4/9
2 – 5	0	0	0	0	0
2 – 6	2	4	12	5	25/9
3 – 5	2	3	4	3	1/9
4 – 5	3	4	5	4	1/9
4 – 8	2	2	2	2	0
5 – 7	3	7	11	7	16/9
5 – 8	2	4	6	4	4/9
8 – 7	0	0	0	0	0
6 – 9	1	4	7	4	1
7 – 9	1	10	13	9	1

To calculate the mean for activity 1 → 2, a = 6, m = 8, b = 10

$t = = \frac{6+4(8)+10}{6} = \frac{48}{6} = 8$

Variance = $\left(\frac{b-a}{6}\right)^2 = \left(\frac{10-6}{6}\right)^2 = \frac{4}{9}$

Calculate the critical path using EST and LST



$ET_j = \max(ET_i + t_{ij})$

$LT_i = \min(LT_j - t_{ij})$

The critical path is 1→3→5→7→9 and the project duration = 25 weeks

The project duration or expected time and the variance can be computed by summing the, expected time and variance for each activities in the critical path because PERT method assumes that activities are statistically independent

Variance activities	**Variance**		
1 – 3	1		
3 – 5	1/9	1 + 1/9 + 16/9 + 4	= 62/9 weeks
5 – 7	16/9		= 6.9 weeks
7 – 9	4		

PERT also assumes that mean and variance of the network is normally distributed

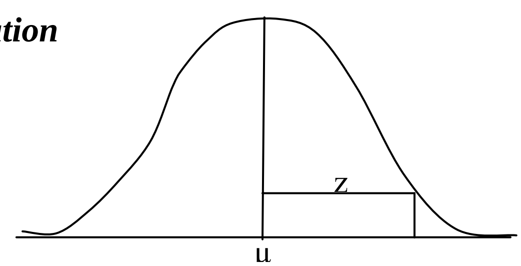
7. ESTIMATION ON PROJECT COMPLETION TIME USING STANDARD NORMAL VARIANCE VALUE

We can interpret the expected time and variance as mean and variance as mean and variable of normal distribution.

From above, mean μ = 25 weeks

Variance σ^2 = 6.9 weeks

Normal Distribution



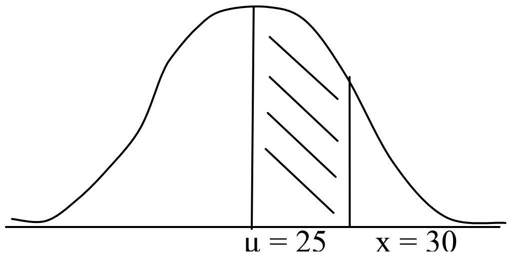
$z = \frac{x-\mu}{\sigma}$ *where σ is standard deviation*

x – proposed project completion time

Example 5: Suppose the textile company manager told customers the new order-processing system would be completely installed in 30 weeks. What is the probability that it will in fact be ready by that time?

Solution:

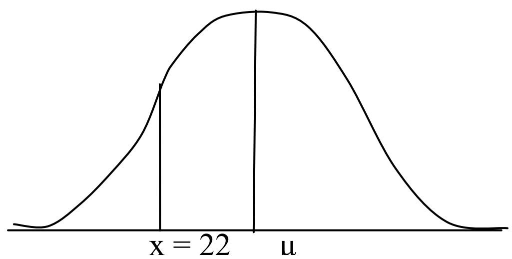
x = 30 weeks, μ = 25 weeks, σ^2 = 6.9 weeks $\Rightarrow \sigma$ = 2.63



$z = \frac{30-25}{2.63} = 1.90$

The z value 1.90 corresponds to 0.4713 on the normal distribution table, this means there is a possibility 0.5 + 0.4713 = 0.9713 of completing the project in 30 weeks. Alternatively, suppose one customer is so frustrated with delay order has told the textile company that if she does not have the new ordering system working in 22 weeks, she will trade elsewhere.

Solution:



$z = \frac{22-25}{2.63}$

The value on the normal table corresponds to 0.3729. This mean the probability of completing the work in 22 weeks is 0.5 – 0.3729. 0.1271.

Summary

1. Basic time analysis is the calculation of critical path which is the shortest time which the project is completed.
2. To determine the critical path, calculate the EST and LST for each event and compare them. The chain activities in which the LST and EST are the same is the critical path.
3. The event or activities on the critical path has zero slack
4. Given the 3 time estimates: Optimistic, most likely time, pessimistic the project time and variance can be computed.
5. The probability of the project time is calculated using = $\frac{x-\mu}{\sigma}$

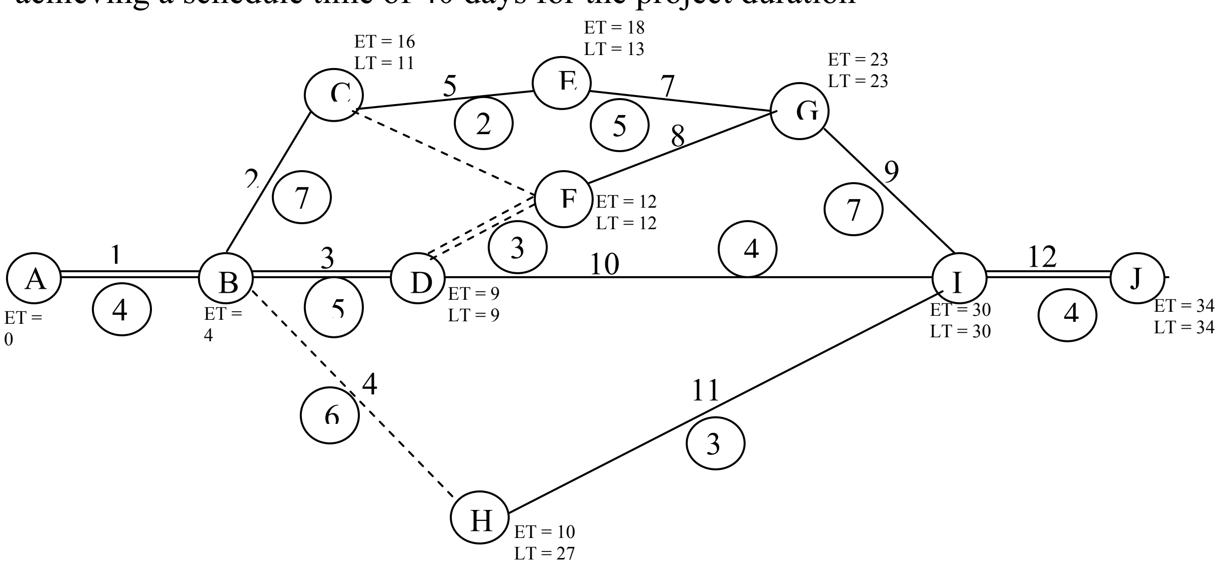
Exercises

a. Find the critical path of the following network using the EST/LST.

Activity	Preceeding Activity	Duration (Days)
1	-	4
2	7	7
3	1	5
4	1	6
5	2	2
6	3	3
7	5	5
8	2,6	11
9	7.8	7
10	3	4
11	4	3
12	9, 10, 11	4

b. Find the activity slack and event slack in the network

2. Assuming the variance of the activities on the critical path in question 1 are 1, 4, 2.25, 9, 6.25, 9 respectively. Based on this value calculated the responsibility of achieving a schedule time of 40 days for the project duration



The double line indicates he critical path

b. Activity slack: $LT_j - ET_i - t_{ij}$

$B \rightarrow C = LT_C - ET_B - t_{BC} = 16 - 4 - 7 = 5$ days

$C \rightarrow E = LT_E - ET_C - t_{CE} = 18 - 11 - 2 = 5$ days

$E \rightarrow G = 23 - 13 - 5 = 5$ days

$D \rightarrow I = 30 - 9 - 4 = 17$ days

$B \rightarrow H = 27 - 4 - 6 = 17$ days

$H \rightarrow I = 30 - 10 - 3 = 17$ days

Slack Event:

Event C = 16 – 11 = 5 days

Event E = 18 – 13 = 5 days

Event H = 22 – 10 = 17 days

Variance = 1 + 4 + 2.25 + 9 + 6.25 + 9

$\sigma^2 = 31.5$

INTEGER PROGRAMMING

In this section, we focus on a class of problems that are modeled as linear programs with the additional requirement that some requirement that some or all of the decision variable must be integer. Such problems are called **Integer Linear Programming Problems.** The use of integer variables provides additional modeling flexibility. As a result, the number of practical applications that can be addressed with LP methodology is enlarged and includes capital budgeting, distribution system Design, Location problems etc.

Types of integer Linear Programming Models

Formally, the general integer programming problem is

Maximize $f(x)$

s.t

$g_j(x) = 0, \quad j = 1, 2, \ldots, m$

$h_i(x) \le 0, \quad i = 1, 2, \ldots, k$

$X = (x_1, x_2, \ldots, x_q, x_{q+1}, \ldots, x_n)$

Where $x_1, x_2, \ldots, x_q$ are integers for a given q. As problem remains essentially unsolved in the general case we confine our attentive to a useful simplification. We assume f and the hi's are **linear,** these are no gj's and all the variables in X must be non-negative. Then the formulation can be expressed in matrix notation as

Maximize CX … (1)

s.t

$AX \le b$ … (2)

$X \ge 0$ … (3)

$x_1, x_2, \ldots, x_q$, integers … (4)

Where $X = (x_1, x_2, \ldots, x_{q+1}, \ldots, x_n)^T$, C is a 1 x n real vector, b is an m x 1 real vector, A is an m x n real matrix.

If q = n, the problem is termed an **all- integer linear programming problem**

If 1 < q < n, the problem is termed a **mixed- integer linear programming problem.** If $(x_{1,}\ x_2, \ldots, x_q)$ is replaced by xj = 0 or 1, i =1, 2…..n problem is termed a **zero-one programmed problem.**

Rounding (Graphical Solution)

One obvious approach to (1) – (4) is to neglect (4) and solve the resulting problem graphically. If the solution produced satisfies (4) then it must be optimal. If it does not then there are a number of options available. One straight forward strategy is to round the values of non integer values either up or down to achieve n integer solution.

Let us explore this idea on the following integer – programming problem:

(1) Max $z = 2x_1 + 3x_2$

s.t

$195x_1 + 273x_2 \leq 1365$

$4x_1 + 40x_2 \leq 140$

$x_1 \leq 4$

$x_1\ x_2 \geq 0$ and integer

The Linear programming version of this problem has been solved graphically in (fig 1.0) it can be been that the optimal solution is

$(x_1, x_2) = (2.44, 3.26)$ Z = 14.66

Rounding the decision variable to the nearest integer value yields a solution of x_1 =2 and x_2 = 3 for an objective function value of 13 or #13,000 annual cash flow. In fig 1, we shows the feasible solution points that provide integer values for x_1 and x_2. Is the rounded solution $(x_1, x_2) = (2, 3)$ the optimal solution? The answer is no! as can be seen that the optimal integer solution is x_1 = 4 and x_2 = 2, with object function value of 14.00 or #14,000 annual cash flow.

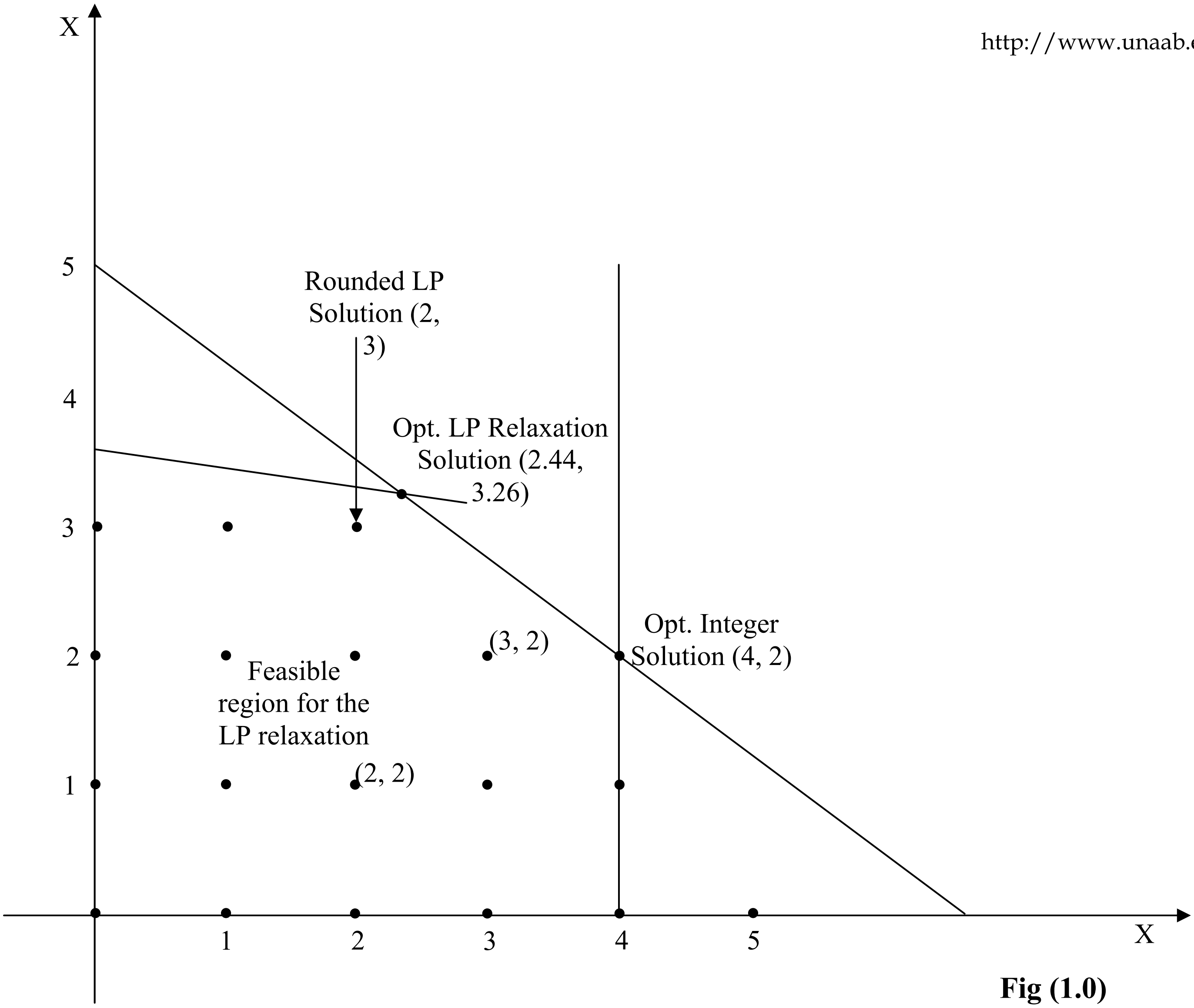


Fig (1.0)

Graphical Solution to the LP Relaxation

Methods of Integer Programming

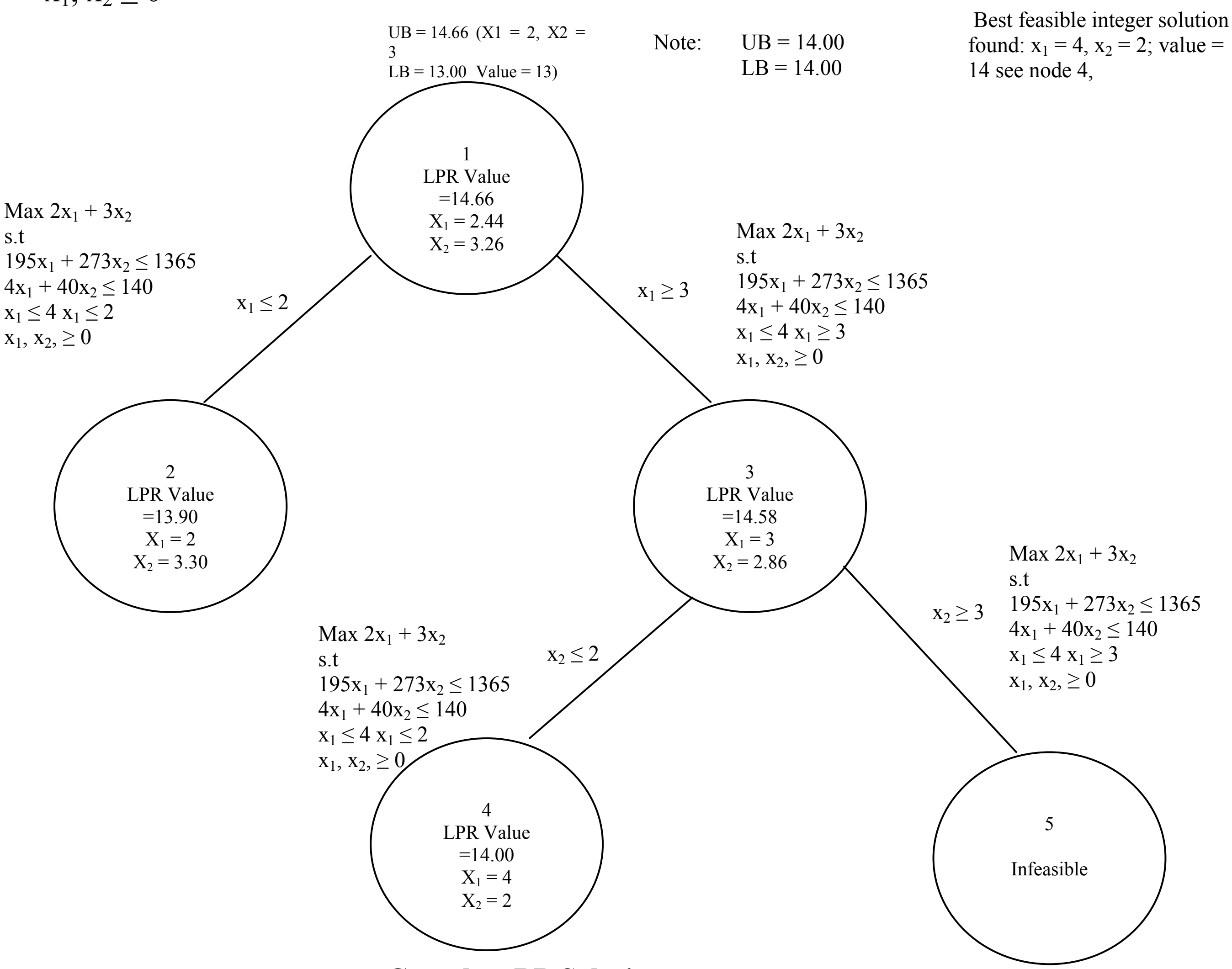
The section unveils the methods that guarantees to find an optimal solution (if one exists) to any integer – programming problem. The two broad approaches **Branch and Bound Technique** and **Cutting plane method**. The earlier technique starts with the continuous optimum; but systematically "partitions" the solution space into sub problems by deleting parts that contain no feasible integer points. The cutting methods systematically adding special "secondary" constraints, which essentially represent necessary conditions for integrality, the continuous solution space is gradually modified until its continuous optimum extreme points satisfies the integer conditions.

The branch and bound algorithm was originally developed by A.H. Larid and A. G. Doig. However, R.J. Dakin's modification offers greater computational advantage and his version will be presented here.

Branch and Bond Solution

BB is currently the most efficient general-purpose solution procedure or integer linear programs. The BB procedure begins by solving the LP Relaxation of the integer linear program. The LP Relaxation of the above problem is stated below,

Max $2x_1 + 3x_2$	x_1 = blocks of town houses purchased
s.t	x_2 = apartment building
$195x_1 + 273x_2 \leq 1365$	funds available
$4x_1 + 40x_2 \leq 140$	Manager's time
$x_1 \leq 4$	Town house availability
$x_1, x_2 \geq 0$	



Complete BB Solution

The Branch and bound solution procedure could be summarized as follows

1. Solve the LP Relaxation of the IP at node 1 set UB value equal to the value of the LP Solution
2. Find n feasible integer solution. Set LB equal to the valueo f the feasible integer solution
3. Is UB = LB? If yes, the optimal solution is the feasible solution with value = LB
4. Otherwise branch from the node with the greatest LP value. Find the variable (call it x_j) that is furthest from being integral. Create two branches and two descendant nodes; one with $x_j \leq (x_j)$ and one with $x_j \geq (k_j) + 1$.
5. Solve the LP Relaxation at each of the descendant nodes, and record its LP value
6. Re compute the upper bound by finding the maximum over all nod from which there are no branches
7. Re compute the lower bound as the max value of all feasible integer solution found to date. Test for 93) and treat the decision arrive at accordingly.

Extension to Mixed-Integer Integer Programs

One of the advantage of BB Solution procedure for integer programming is that it is applicable to both all-integer and mixed-integer linear programs. To see how the BB solution approach can be applied to a mixed-integer linear program, let us return to problem (1) and suppose that x_2 was not required to be integer i.e.

Max $2x_1 + 3x_2$

s.t

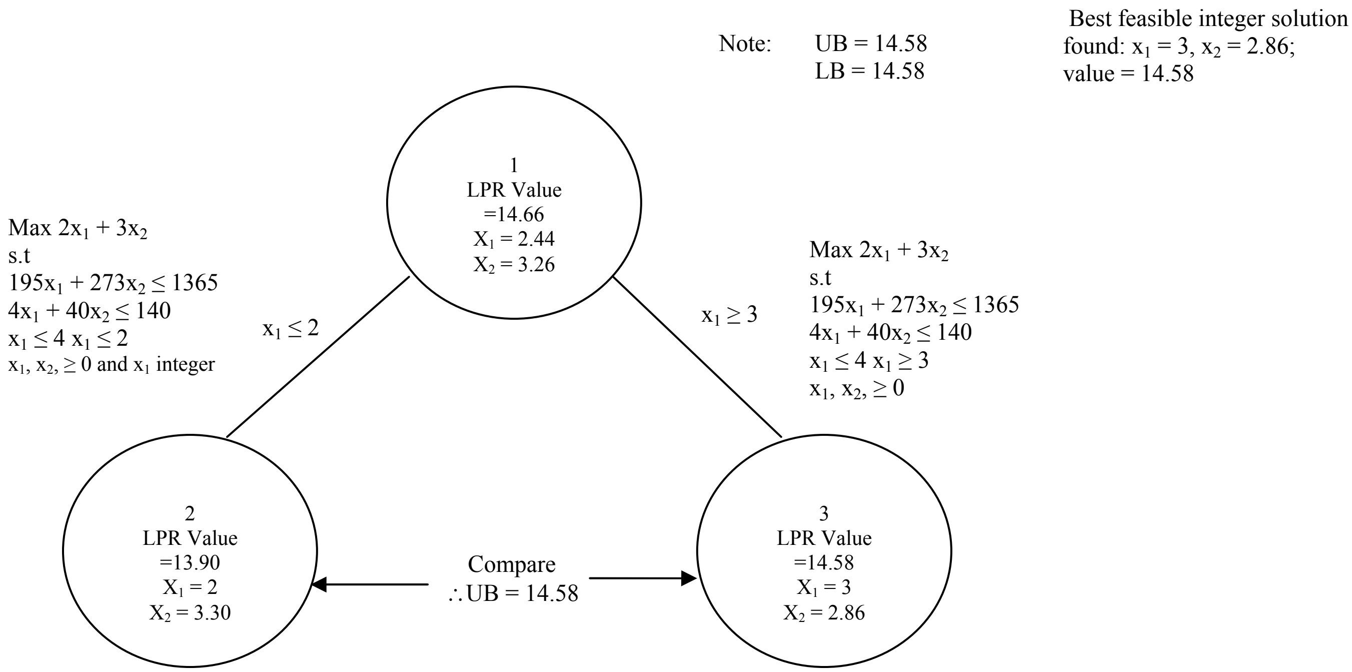
$195x_1 + 273x_2 \leq 1365$

$4x_1 + 40x_2 \leq 140$

$x_1 \leq 4$

$x_1, x_2, \geq 0$ and x_1 integer

The BB solution procedure is illustrated using the decision tree below:



From the above, since the upper and lower bound are equal, the optimal solution to the problem (original) with only x_1 required to be integer has been found. It is given by $x_1 = 3$ and $x_2 = 2.86$, with an objective function value of 14.58

INVENTORY PROBLEMS

INTRODUCTION

An inventory can be defined as any idle resource of an enterprise. An inventory problem exists when it is necessary to stock physical goods or commodities for the purpose of satisfying demand over a specified time horizon (finite or infinite). Almost every business must stock goods to ensure smooth and efficient running of its operation. Decisions regarding how much and when to order are typical of every inventory problem. The required demand maybe satisfied by stocking once for the entire time horizon or by stocking separately for every time unit of the horizon. The two extreme situations (overstocking and under-stocking) are costly. Decisions may thus be based on the minimization of an appropriate cost function that balances the total costs resulting from over-stocking and under-stocking.

A GENERALIZED INVENTORY MODEL

The ultimate objective of an inventory model is to answer two questions.

1. How much to order?
2. When to order

The answer to the first question is expressed in terms of what we call the order quantity and the when-to-order decision is the inventory level at which a new order should be placed usually expressed in terms of re-order point.

The order quantity and re-order pint are normally determined by minimizing the total inventory cost that can be expressed as a function of these two variables. We can summarize the total cost of a general inventory model as a function of its principal components in the following manner:

(Total inventory cost) = (purchasing cost) + (setup cost) + (holding cost)
+ (shortage cost)

TYPES OF INVENTORY MODELS

In general, inventory models are classified into two categories:

1. Deterministic model and
2. Stochastic model

Definitions:

1. Delivery lags or lead times is the time between the placement of an order and its receipt and may be deterministic or stochastic.
2. Time Horizon defined the period over which the inventory level will be controlled. This horizon may be finite or infinite depending on the time period which demand can be forecast reliably.
3. Stock replacement: Although an inventory system may operate with delivery lags. The actual replenishment of stock may occur instantaneously or uniformly. Instantaneous replenishment can occur when the stock is purchased from outside sources. Uniform replenishment may occur when the product is manufactured locally within the organization.

DETERMINISTIC MODEL (Single item static model)

The simplest type of inventory model occurs when demand is constant over time with instantaneous replenishment and no shortages. Typical situations to which this model may apply are:

1. The use of light bulbs in a building
2. The use of clerical supplies, such as paper, pads and pencil in a large company
3. The consumption of staple food items, such as bread and milk

Fig (1) illustrates the variation of the inventory level. It is assumed that demand occurs at the rate β (per unit time). The highest level of inventory occurs when the inventory level reaches zero level y/β time units after the order quantity y is received.

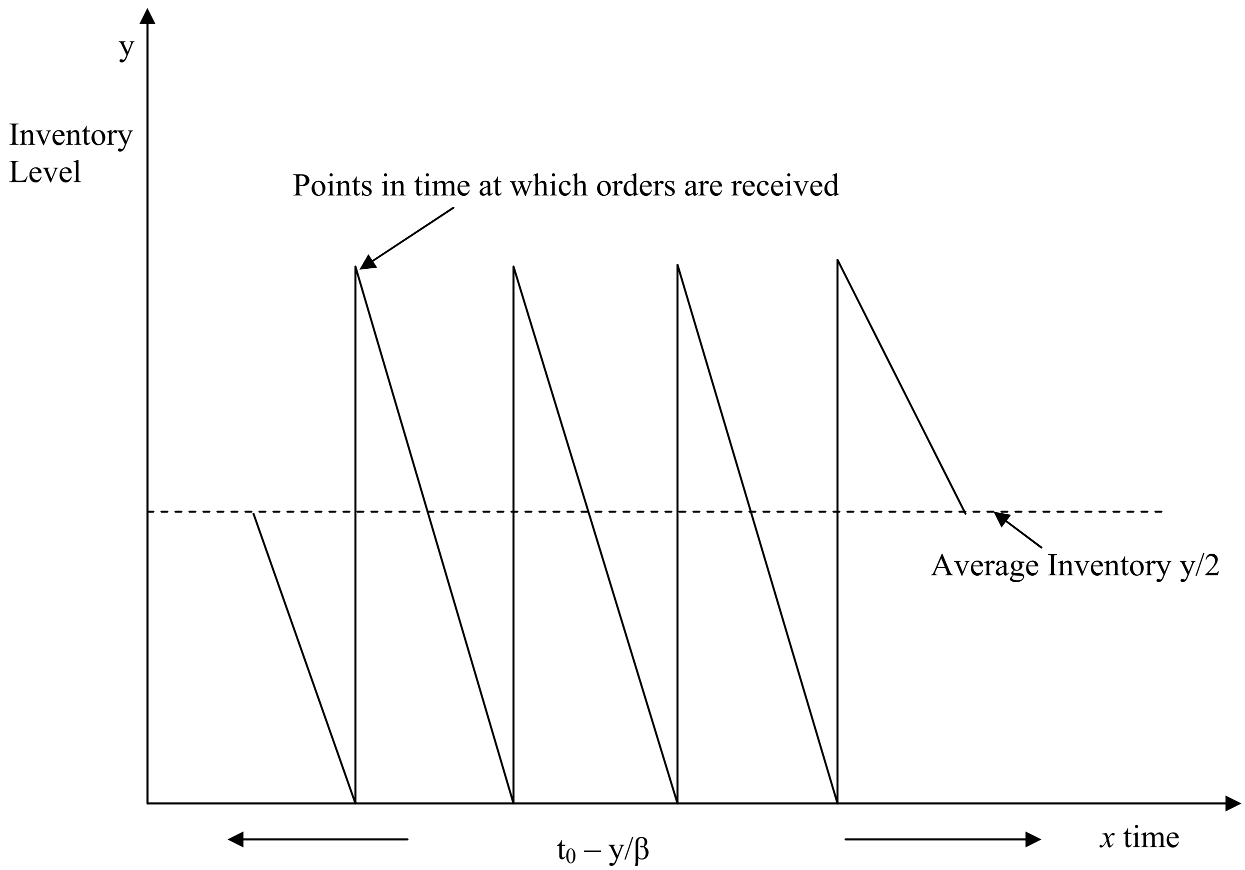


Fig. 1

The smaller the order quantity y, the more frequent will be the placement of new orders. However, the average level of inventory hold in stock will be reduced. On the other hand, larger order quantities indicate larger inventory level but less frequent placement of order (see fig. 2).

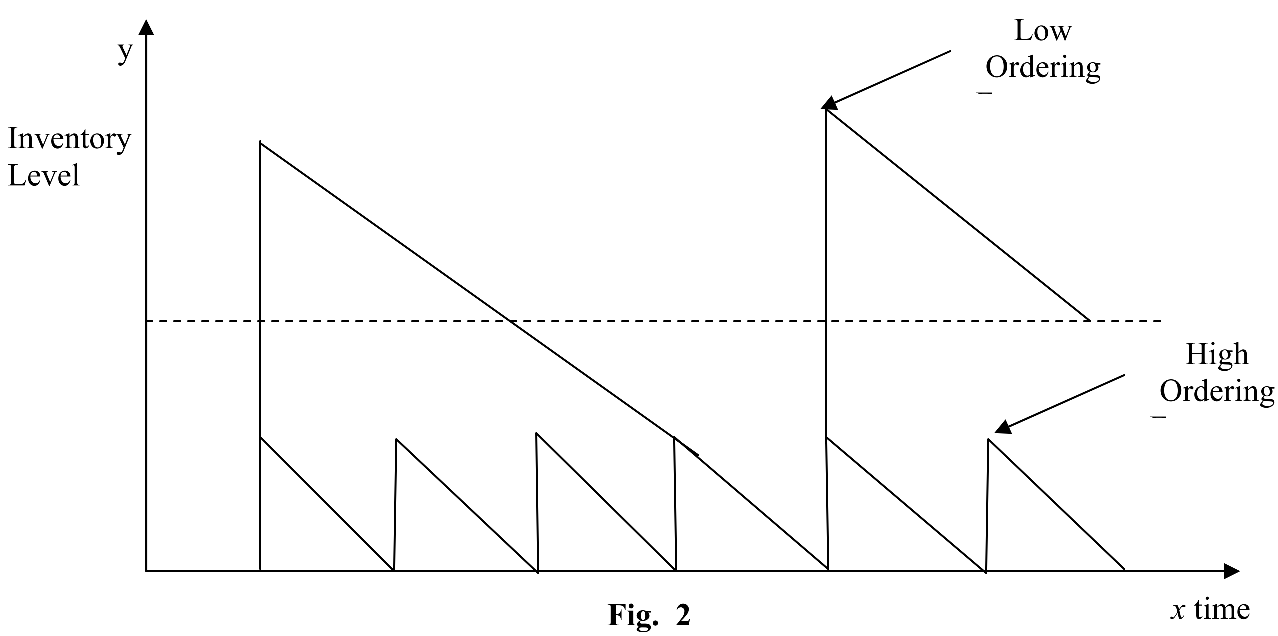


Fig. 2

Because there are costs associated with placing orders and holding inventory in stock, the quantity y is selected to allow a compromise between the two types of costs. This is the basis for formulating the inventory model.

Let K be the set up cost incurred every time an order is placed and assume that the holding cost per unit inventory per unit time is k. Hence, the total cost per unit time TCU as a function of y may be written as

TCU (y) = Setup cost/unit time + holding cost/unit time

$$TCU\ (y) = \frac{k}{y/\beta} + h\ (y/2)$$

As seen from fig. (1), the length of each inventory cycle is to = y / β and the average inventory in stock is y/2

The optimum value of y is obtained by minimizing TCU (y) with respect to y. Thus, assuming that y is a continuous variable, we have

$$\frac{dTCU\ (y)}{dy} = k\beta/y + h/2 = 0$$

Which yields the optimum order quantity as

$$y = \sqrt{\frac{2k\beta}{h}}$$

(It can be proved that y minimizes TCU(y) by showing that the 2^{nd} derivates at y is strictly positive). The order quantity above is usually referred to as Wilson's economic lot size.

The optimum policy of the model calls for ordering units every to time units. The optimum cost TCU (y) obtained by direct substitution is √2k/β.

EXAMPLE 1:

The daily demand for a commodity is approximately 100 units. Every time an order dared, fixed cost is N100 is incurred. The daily holding cost per unit inventory if N0.02. If the lead time is 12 days. Determine the economic lot size and the re-order point

SOLUTION:

From the earlier formula the economic lot size is

$$y = \sqrt{\frac{2k\beta}{h}}$$

= √2* 100*/0.02 = 1000 units

The associated optimum cycle length is this given as

$b_0* = \frac{y*}{\beta}$ = 1000/100 = 10 days

Since the lead time is 12 days add the cycle length is 10 days re-ordering occurs when the level of inventory is sufficient to satisfy the demand for two (- 12, 10) days. Thus the quantity y* = 1000 is ordered when the level of inventory reaches 2 * 100 = 200 units.

Notice that the "effective" lead time is taken equal to 2 days rather than 12 days. This result occurs because the lead time is longer than t_0*.

EXAMPLE 12:

A manufacturer has to supply his customer' with 600 units of his product per year. Shortages are not allowed and the shortage cost amounts to N0.60 per and per year. The setup cost per run is N80.00. Find the optimum run size and minimum average yearly cost.

Solution:

Since β = 600 units/year

K = N80.00

h = N0.06

$y = \sqrt{\frac{2k\beta}{h}}$

√2 * 80 * 600 / 0.06 = 400 units opt. run time

And the minimum average yearly cost = √2kβh

= √2 * 80 * 60 * 0.60

= N240.00

EXERCISES

1. XYZ Company purchases a component used in the manufacturing automobile generators directly from the suppliers. XYZ's generator production which is operated at a constant rate will required 1000 components per month throughout the year (12,000 units annually). If ordering cost is N25 per order, unit cost is

N200 per component and annual inventory holding costs are charged at 20%. Answer the following inventory policy question for XYZ

a. What is the economic order quantity (EOQ) for this component?

b. What is the length of cycle time in months?

c. What are the total annual inventory building and ordering cost associated with your recommended EOQ?

2. The demand for a particular item is 18,000 units per year. The holding cost per unit is N1.20 per year, and the cost of the replenishment rate is instantaneous. Determine

a. Optimum order quantity

b. Number of orders per year

c. Time between orders and

d. Total cost per year when the cost of 1 unit is N1.00

GAME THEORY

A game is defined to be a contest between opponents in which each has a no (finite of infinite) of courses of action, strategies and the outcome of any combination of strategies is known beforehand.

Definitions:

1. A two – person zero – sum game is one played by 2 persons or groups where the gain of one person will be exactly equal to the loss of the other so that the sum total of gains and losses will be equal to zero. A 2 – person zero-sum game can be formulated in the form of a matrix payoff matrix shown below:

Player C Strategies

		C_1	C_2	C_3	. . .	C_1
	R_1	a_{11}	a_{12}	a_{13}	. . .	a_{1n}
Player R	R_2	a_{21}	a_{22}	a_{23}	. . .	a_{2n}
Strategy	R_3	a_{31}	a_{32}	a_{33}	. . .	A_{3n}
	.	.	.	.	. . .	.
	.	.	.	.		.
	.	.	.	.		.
	R_m	a_{m1}	a_{m2}	a_{m3}	. . .	a_{mn}

The player R controls the rows R_1, R_2, …, R_m which represents his strategies while player C controls the columns C_1, C_2, … C_n which represents his strategies. If player R chooses be ith strategy and player C the jth strategy then the element aij is assumed to represent the payoff from player C to player R i.e. if aij is a +ve no, it represents payment of C to R and if –ve it denotes the payment of R to C.

2. The element *aij* of the payoff matrix (of order mx n) of a game is called a saddled point if it is minimum along the *ith* row elements and the maximum along the *jth* column elements. Thus a game is said to be strictly determined, if an only if it has a saddled value. The value of the game is equal to the saddle point. The optimal strategies for the 2 players are given by the row that contains the saddled point for the player R, and the column that contains the saddle point for the Player C.

Determination of Saddle Point – Minimax (Maximum) Principle

In any game problem, each player is interest in determining his own optimal strategy. However, because of the lack of information regarding the specific strategies selected by the other players, optimality has to be based on a conservative principle so, each player selects his strategy which guarantees a payoff that can never be worse by the selection of his opponents, this idea is called the **Minimax (or Maximum)** principle and is illustrated below.

Example:

Let the payoff matrix of a 2-person zero-sum game be as in the matrix below. Find the optimal strategies of the players.

Player C Strategies

		C_1	C_2	C_3	C_4	Row Minimum
Player R strategy	R_1	7	1	8	4	1
	R_2	5	4	6	7	[4]
	R_3	6	2	-3	6	-3
	Column Maximum	7	[4]	8	7	

If player *R* selects his strategy R_1, he may gain 7, 1, 8, 4 depending on the strategy selected by player C. However, player R is guaranteed a gain of at least 1 = (7, 1, 8, 4) irrespective of the strategy of C. Similarly R is guaranteed a gain of at least 42min (5, 4, 6 7) if he choose strategy R_2 and at least -3 = min (6, 2, -3, 6) if strategy R_3 is selected. Thus if player R wants to maximize his gain irrespective of strategy selected by C, he has to maximize the minimum gain i.e. max (1, 4, -3) = 4. Thus the strategy R_2 is to be chosen by R based on the maximum principle with 4 as the maximum value of the game.

On the othe hand, if player C chooses strategy C, he loses 7, 5, 6 depending on the strategy selected by player R. however he can lose no more than 7 = max (7, 5, 6) regardless of R's strategies. In similar manner, player C loses no more than 4 = max (1, 4, 2), 8 = max (8, 6, -3) and 7 = max (4, 7, 6) by choosing strategies C_2, C_3 and C_4 respectively regardless of the strategies selected by R. Thus C selects the particular strategy which minimizes his maximum losses irrespective of the strategies of R, i.e. min (7, 4, 8, 7). The minimum of maximum loss is given by strategy C_2 based on minimax principle.

A game as in the e.g. above, where the minimax value = maximin value the corresponding pure strategies are called optimal strategies and the game is said to have a saddle point.

As illustrated below, it is possible to have a situation where there is no saddle point and hence one cannot find a pure strategy solution consider a game for which the pay off matrix is as below:

Player C Strategies

		C_1	C_2	C_3	Row Minimum
Player R strategy	R_1	6	2	7	2
	R_2	3	4	8	3
	R_3	5	4	1	1
	Column Maximum	7	4	8	

So, in general maximin ≤ value of game ≤ minimax value. (3. 27)

As in the last example above, mixed strategies in which both players will search for a correct strategy mixture to find equilibrium, has to be used. The value of the game at equilibrium is uniquely determined by the right strategy mixture and satisfies the inequality (3.27).

The correct strategy mixture for each player is determined by assigning to each strategy its probability of being chosen. Let r_1, r_2, ……….. r_n represents the probabilities with which player C selects the pure strategies R_1, R_2 …………. R_m respectively and let C_1, C_2 ……… C_m be the probabilities with which player C selects the pure strategies respectively. The sum of the probabilities for the strategies of each player may be equal to one i.e.

m

$$\sum_{i=1} r_i = 1, \qquad r_i \geq 0 \; i = 1, 2, \ldots, m$$

and (3.28)

$$\sum_{j=1}^{m} c_i = 1, \qquad c_i \geq 0 \; j = 1, 2, \ldots, n$$

The solution of mixed strategy problems is also based on the minimax principle in the sense that R selects the values of r_i so as to maximize the smallest expected value of pay off in a column, where as C chooses the values of c_j so as to minimize the largest expected value of pay off in a row.

Thus, the player R finds r_i which will

$$\text{maximize}\left\{\text{minimize of}\left(\sum_{i=1}^{m} a_{i1}\, r_i, \sum_{i=1}^{m} a_{i1}\, r_i, \ldots \sum_{i=1}^{m} a_{in}\, r_i,\right)\right.$$

w.r.t r_i *with* $r, \geq 0$

$$\text{and} \sum_{i=1}^{m} r_i = 1 \qquad (3.29)$$

and the player C selects cj which will

$$\text{maximize}\left\{\text{minimize of} \sum_{i=1}^{n} a_{ij}\, c_j, \sum_{i=1}^{n} a_{2j}\, c_j, \ldots \sum_{i=1}^{n} a_{mj}\, c_j,\right.$$

w.r.t c_i *with* $r, \geq 0$

$$\text{and} \sum_{j=1}^{m} r_i = 1 \qquad (3.30)$$

Let r_i (i = 1, 2, …, m) and c_j (j = 1, 2, …, n) denote the optimal solution, since each pay off element a_{ij} is associated with a probability combination (r_i, c_j) the optimal expected value of the game is given optimum expected value of the game =

$$\text{and} \sum_{i=1}^{m} \sum_{j=1}^{n} = a_{ij} r_{\,i} c \qquad (3.31)$$

Solution of a Game Problem

Equation (3.29) can be transformed to a LPP as follows

$$\left(\;{}^{m} \qquad\qquad {}^{m}\;\right)$$

Let $v = \min \sum_{i=1} a_{i1}\, r_i, \ldots \sum_{i=1} a_{in}\, r_i,$

Then (3.29) becomes

Maximize $f(r_1, r_2, \ldots r_m) = v$ subject to

$$\sum_{i=1}^{m} a_{i1}\, r_i, \geq v$$

$$\sum_{i=1}^{m} a_{i2}\, r_i, \geq v \qquad (3.32)$$

.

.

.

$$\sum_{i=1}^{m} a_{in}\, r_i, \geq v$$

And $r_i \geq 0$, $i = 1, 2, \ldots m$

Clearly, (3.32) is a LPP, whose solution gives the value of the game. The simplex method is used to solve the game problem below

Example

There are 2 competing departmental stores R and C in a city. Both the stores customers are equally divided between the two. Both the store, plan to run annual discount sales in the last week of December for this they want to attract more number of customers by using advertisement through newspaper, radio and television. By seeing the market trend, the store R constructed the following pay off matrix below, where the number in the matrix includes a gain or loss of customer.

Strategies of Store C

	C_1	C_2	C_3	Row Minimum

Strategies Store C	R_1	40	50	-70	-70
	R_2	10	25	-10	-10
	R_3	100	30	60	30
	Column Maximum	100	50	60	

Assuming that a gain of customers to store R means a loss to C, find the optimal strategies for both the store along with the value of the gains.

Solution

The minimax and maximin values for this pay off matrix are 50 and 30 respectively and hence the game does not have saddle point. Now, dividing the constraint in (3.32) by v we have

Maximize $f(r_1, r_2, \dots r_m) = v$

$$\sum_{i=1}^{m} a_{i1} \frac{r_i}{v} \geq 1$$

$$\sum_{i=1}^{m} a_{i2} \frac{r_i}{v} \geq 1 \qquad (3.33)$$

.

.

.

$$\sum_{i=1}^{m} a_{in} \frac{r_i}{v} \geq v$$

And $r_i \geq 0$, $i = 1, 2, \dots m$

Defining new variables x_i as

$$x_i = \frac{r_i}{v} \quad i = 1, 2, \dots m$$

Note that

$$\textit{Maximum of } v = \textit{minimum of } \frac{1}{v} = \textit{minimum of } \sum_{i=1}^{m} x_i$$

The LPP (3.32) can be restated as

Maximize $f(x_1, x_2, \dots x_m) = v$

$$\sum_{i=1}^{m} x_i \text{ subject to}$$

$$\sum_{i=1}^{m} a_{i1} x_i, \geq 1$$

$$\sum_{i=1}^{m} a_{i2} x_i, \geq 1 \qquad (3.33)$$

v

.

.

.

$$\sum_{i=1}^{m} a_{in} x_i, \geq v$$

And $x_i \geq 0$, i = 1, 2, … m

Respectively from (3.30) we have the LPP

Maximize $g(y_1, y_2, \dots y_m) =$

$$\sum_{j=1}^{n} y_j \text{ subject to}$$

$$\sum_{i=1}^{m} a_{1j} y_j, \leq 1$$

$$\sum_{i=1}^{m} a_{2j} y_j, \leq 1 \qquad (3.34)$$

v

.

.

.

$$\sum_{i=1}^{m} a_{1j} y_j, \leq 1$$

And $y_i \geq 0$, j = 1, 2, … n

Where $g = \frac{1}{v}$, $y_j = \frac{cj}{v}$, $j = 1, 2, \dots n$

For this example, the problem of store C can state as LPP as follows:

Maximize $g = y_1 + y_2 + y_3$ subject to

$a_{11}y_1 + a_{12}y_2 + a_{12}y_3 \leq 1$

$a_{21}y_1 + a_{22}y_2 + a_{23}y_3 \leq 1$

$a_{31}y_1 + a_{32}y_2 + a_{33}y_3 \leq 1$

And $\quad y1 \geq 0, \ i = 1, 2, 3$

This is restated as

Maximize $\quad f = y_1 + y_2 + y_3$ subject to

$40y_1 + 50y_2 - 70y_3 + y_4 = 1$

$10y_1 + 25y_2 - 10y_3 + y_5 = 1$

$100y_1 + 30y_2 + 60y_3 + y_6 = 1$

And $\quad y_1 \geq 0, \ i = 1, 2, 3$

Basis	C_B	y_1 +1	y_2 +1	y_3 +1	y_4 0	y_5 0	y_6 0		Ratios
y_4	0	40	50	-70	1	0	0	1	$\frac{1}{40}$
y_5	0	10	25	-10	0	1	0	1	$\frac{1}{10}$
y_6	0	[100]	30	60	0	0	1	1	[$\frac{1}{100}$] ← smallest value
z_j		0	0	0	0	0	0	0	
$c_j - z_j$		+1	+1	+1	0	0	0		
		↑							

Basis	C_B	y_1 +1	y_2 +1	y_3 +1	y_4 0	y_5 0	y_6 0		Ratios
y_4	0	0	[38]	-94	1	0	$\frac{4}{10}$	$\frac{6}{10}$	[$\frac{6}{380}$] ← smallest value
y_5	0	0	22	-16	0	1	$\frac{1}{10}$	$\frac{9}{10}$	$\frac{9}{220}$

y_1	1	1	$\frac{3}{10}$	$\frac{3}{5}$	0	0	$\frac{1}{100}$	$\frac{1}{100}$	$\frac{1}{30}$
z_j		0	$\frac{3}{10}$	$\frac{3}{5}$	0	0	$\frac{1}{100}$	$\frac{1}{100}$	
$c_j - z_j$		0	$\frac{7}{10}$ ↑	$\frac{2}{5}$	0	0	$\frac{-1}{100}$		

Basis	C_B	y_1 +1	y_2 +1	y_3 +1	y_4 0	y_5 0	y_6 0		Ratios
y_2	1	0	1	$\frac{-47}{19}$	$\frac{1}{38}$	0	$\frac{-1}{95}$	$\frac{3}{190}$	
y_5	0	0	0	$\frac{730}{19}$	$\frac{-11}{19}$	1	$\frac{5}{38}$	$\frac{21}{38}$	$\frac{21}{1460}$
y_1	1	1	0	$\boxed{\frac{51}{38}}$	$\frac{-3}{360}$	0	$\frac{1}{76}$	$\frac{1}{190}$	$\boxed{\frac{1}{255}}$ ← smallest value
z_j		1	1	$\frac{43}{38}$	$\frac{7}{380}$	0	$\frac{1}{380}$	$\frac{2}{95}$	
$c_j - z_j$		0	0	$\frac{81}{38}$ ↑	$\frac{-7}{380}$	0	$\frac{-1}{380}$		

Basis	C_B	y_1 +1	y_2 +1	y_3 +1	y_4 0	y_5 0	y_6 0		Ratios
y_4	0	$\frac{94}{51}$	1	0	$\frac{1}{85}$	0	$\frac{7}{510}$	$\frac{13}{510}$	
y_5	0	$\frac{-1460}{51}$	0	0	$\frac{-114}{323}$	1	$\frac{-25}{102}$	$\frac{41}{102}$	
y_6	1	$\frac{38}{51}$	0	1	$\frac{-1}{170}$	0	$\frac{1}{102}$	$\frac{1}{255}$	
z_j		$\frac{132}{51}$	1	1	$\frac{1}{170}$	0	$\frac{2}{85}$	$\frac{1}{34}$	
$c_j - z_j$		$\frac{-81}{51}$	0	0	$\frac{-1}{170}$	0	$\frac{-12}{510}$		

$$v^* = \frac{1}{g} = 34$$

Thus, the optimum solution for C

$$v^* = -\frac{1}{g} = 34,$$

$c_2^* = y_2^* v^* = \frac{13}{510} * 34 = \frac{13}{15}$

$c_3^* = y_3^* v^* = \frac{1}{510} * 34 = \frac{2}{15}$

$c_1^* = 0$

The optimal strategy for R is given by

v* = 34,

$r_1^* = x_1 * v^* = \frac{1}{170} * 34 = \frac{1}{5},$

$r_2^* = x2 * v^* = 0,$

$r_3^* = \frac{2}{85} * 34 = \frac{4}{5}$

Thus, store R will gain 34 customers from C when both apply the optimal advertising strategies during the annual reduction sale period.

Exercises

1. Find the optimal strategies and the value of the 2 games whose payoff matrices are given below:

P_1/P_2	T_1	T_2	T_3	T_4
S_1	7	8	12	14
S_2	5	6	-10	-12
S_3	4	-4	-3	4
S_4	7	9	13	12

	C_1	C_2	C_3	C_4
R_1	1	9	6	0
R_2	2	3	8	4
R_3	-5	-2	10	-3
R_4	7	4	-2	-5

2. An m *x* n matrix is called a Latin square if each row and column contains each of the integers from 1 to m. Show that a game which has this as its payoff matrix has the value ½ (1 + m)

3. Two players fight a duel, they face other 2n paces apart and each has a single bullet in his gun. At a signal each may-fire, if either is hit or if both fire the game ends. Otherwise both advance one pace and may again fire. The game of course

ends anyway by the time n paces has been taken. The probability of either hitting his target if he fires after the ith pace forward is 1/n. The payoff is +1 to a player who survives after his opponent is hit, and 0 if neither or both are hit; the guns are silent so that neither knows whether or not his opponent has fired.

Show that if n = 4, the strategy shoot after taking two steps is optimal for both but that if n = 4, a mixed strategy is optimal.

DECISION THEORY

The ultimate purpose of any operations research analysis is to enable operations to be run efficiently and effectively, and this in turn involves selecting the best of the alternative ways and means of conducting operations. Fundamental, then, to any operations research exercise is the final step of making a decision between alternatives, and the principles underlying such decision making are referred to as **Decision Theory**

Structuring the Decision Problem

To illustrate the decision analysis approach, let us consider the case of Political Systems, Inc (PSI), a newly formed computer service from specializing in information services such as surveys and data analysis for individuals running for political office. The firm is in the final stages of selecting a computer system for its Midwest branch, located in Lagos. While PSI has decided on a computer manufacturer, it is currently attempting to determine the size of the computer system that would be most economical. We will use decision theory to help PSI make its computer decision.

The first step is to identify the alternatives considered by the decision maker. For PSI, the final decision will be to select one of the three computer systems, which differ in size and capacity. The three decision alternatives denoted by D_1, D_2, and D_3 are as follows:

D_1 – large computer system

D_2 – Medium computer system

D_3 – Small computer system

The second step is to identify the future events that might occur. There events, which are not under the control of the decision maker, are referred to as the **States of nature**. Thus, the PSI states of nature denoted S_1 and S_2 are as follows:

S_1 – high customer acceptance of PSI services

S_2 – low customer acceptance of PSI services

Given the three decision alternatives and the two states of nature, **which computer system should PSI select?** To answer this question, we will need information on the profit associated with each combination of a decision alternative and a state of nature.

Payoff Tables

We denote the decision alternatives by D_1, D_2, …, D_m, the states of nature by S_1, S_2, … S_n; and the return associated with decision Di and state Sj by Vij (I = 1, 2, …, m i j = 1, 2, …, n). A process requiring the implementation of just one decision is defined completely by Table 1. A table of this form is referred as a payoff table. In general, entries in (a) can be stated in terms of profits, costs etc. Using the best information available, management has estimated the payoffs or profits for the PSI problem. There estimates are presented in Table 2

Table 1: States of Nature

	S_1	S_2	…	S_n
D_1	V_{11}	V_{12}	…	V_{1n}
D_2	V_{21}	V_{22}	…	V_{2n}
…	….	…	…	…
D_m	V_{m1}	V_{m2}	…	V_{mn}

Table 2:

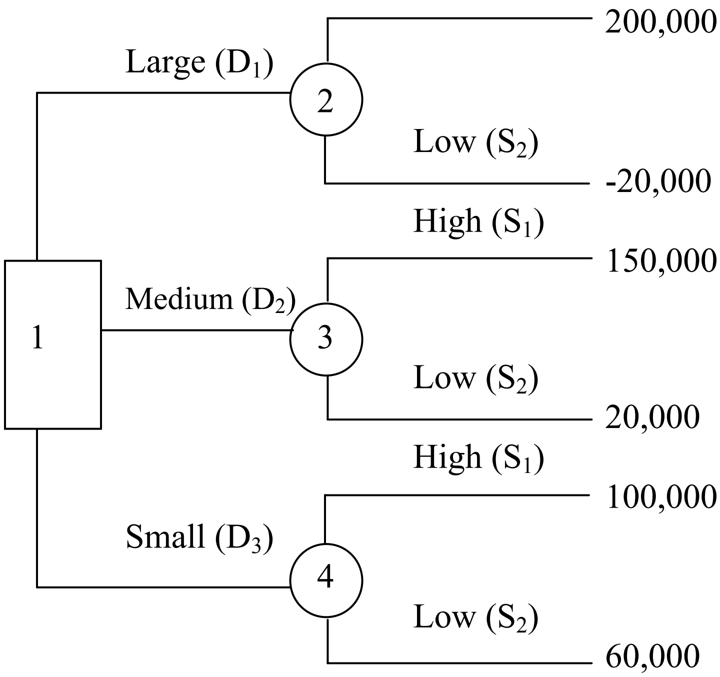
Decision alternatives		High Acceptance S_1	Low Acceptance S_2
Large system	D_1	200,000	20,000
Medium system	D_2	150,000	20,000
Small system	D_3	100,000	60,000

Decision Trees

A decision tree provides a graphical presentation of the decision-making process. Figure 1 shows a decision tree for the PSI problem. Note that the three shows the natural or logical progress that will occur overtime.

Figure 1

High (S_2)



Using the general terminology associated with decision trees, we will refer to the intersection or junction points of the tree as **nodes** and the arcs or connectors between the nodes as **branches**. Fig 1 shows the PSI decision tree with the nodes numbered 1 to 4. When the branches leaving a given node are decision branches, we refer to the nodes as decision node. Decision nodes are denoted by squares. Similarly, when the branches leaving a given node are state-of-nature branches, we refer to the node as a state-of-nature node. State-of-nature nodes are denoted by circles. Using the node-labelling procedure, node 1 is a decision node, where as nodes 2, 3 and 4 are states of nature nodes.

Decision Making without Probabilities

This section consider approaches to decision making that do not require knowledge of the probabilities of the states of nature

Optimistic Approach

The (~) evaluates each decision alternative in terms of the best payoff that can occur. The decision alternative that is recommended is the one that provides the best possible payoff. For a problem in which maximum profit is desired, as it is in the PSI problem, the optimistic approach would lead the decision maker to choose the alternative corresponding to the largest profit. For problems involving minimization, this approach leads to choosing the alternative with the smallest payoff. To illustrate the use of the ~, we will show how it can be used to develop a recommendation for the PSI problem.

Table 3

Decision alternatives		Maximum Payoff	
Large system	D_1	200,000	←Maximum of the maximum payoff values
Medium system	D_2	150,000	
Small system	D_3	100,000	

Conservative Approach

The conservative approach evaluates each decision alternatives in terms of the most payoff that can occur. The decision alternative recommended is the one that provides the best of the worst possible payoffs. For a problem in which the output measure is profit, as it is in PSI problems. The (~) would lead the decision maker to chose the alternative that maximizes the minimum possible profit that could be obtained. For problems involving minimization, this approach identifies the alternative that will minimize the maximum payoff.

Table 4

Decision alternatives		Maximum Payoff	
Large system	D_1	-20,000	
Medium system	D_2	20,000	
Small system	D_3	60,000 ←	Maximum of the maximum payoff values

Minimax Regret Approach

(~) is another approach to decision making with certainty. This approach is neither purely optimistic nor purely conservative. We illustrate the (~) for the PSI problem. In maximization problem, the general expression for opportunity loss or regret is given by the formula:

Opportunity loss or Regret

$Rij = Vj - Vij$ --- (*)

Where

Rij = reject associated with decision alternative Di and state of nature Sj

Vj = best payoff value under state of nature Sj

Vij = payoff associated with decision alternative D1 and state of nature Sj

Using eq (*) and the payoff in Table 2, we can compute the regret associated with all combinations of decision alternatives Di and States of nature Sj

Table 5

Regret or opportunity loss for the PSI problem		States of Nature	
Decision Alternatives		High Acceptance S_1	Low Acceptance S_2
Large System	D_1	0	80,000
Medium System	D_2	50,000	40,000
Small System	D_3	100,000	0

Table 6

Decision Alternatives		Maximum Regret or Opportunity Loss
Large system	D_1	80,000
Medium system	D_2	50,000 ← Minimum of the maximum
Small system	D_3	100,000 regret

For the PSI problem, the decision to select a medium-computer system, with a corresponding regret of N50,000, is the recommended minimax regret decision.

Rank: In cost minimization problems, Vj will be the smallest entry in column j, and equation (*) must be changed to

$$Rij = Vij - Vj$$

Decision Making with Probabilities

In many decision-making situations, it is possible to obtain probability estimates for each of the states of nature. When such probabilities are available, the **expected value**

approach can be used to identify the best decision alternative. The expect value approach evaluates each decision alternative in terms of its expected value. The recommended decision alternative is the one that provides the best expected value.

Let

N = the number of states of nature

P (Sj) = the probability of state of nature Sj

P (Sj) $\geq$ 0 for all states of nature

P (Sj) = P (S_1) + P (S_2) + … + P (S_N) = 1

Expected Value of Decision Alternative Di

$$EV\ (Di) = \sum_{j=1}^{N} P\ (Sj)\ Vij \qquad **$$

Using the payoff values Vij shown in Tale 1 and supposes that PSI management believes that S_1, the high acceptance state of nature, has a 0.3 probability of occurrence and that S2, the low-acceptance state of nature, has a 0.7 probability. Thus, P (S_1) = 0.3 and P (S_2) = 0.7 and equation (**), expected values for the three decision alternatives can be calculated:

EV (D_1) = 0.3 (200,000) + 0.7 (-20,000) = N46,000

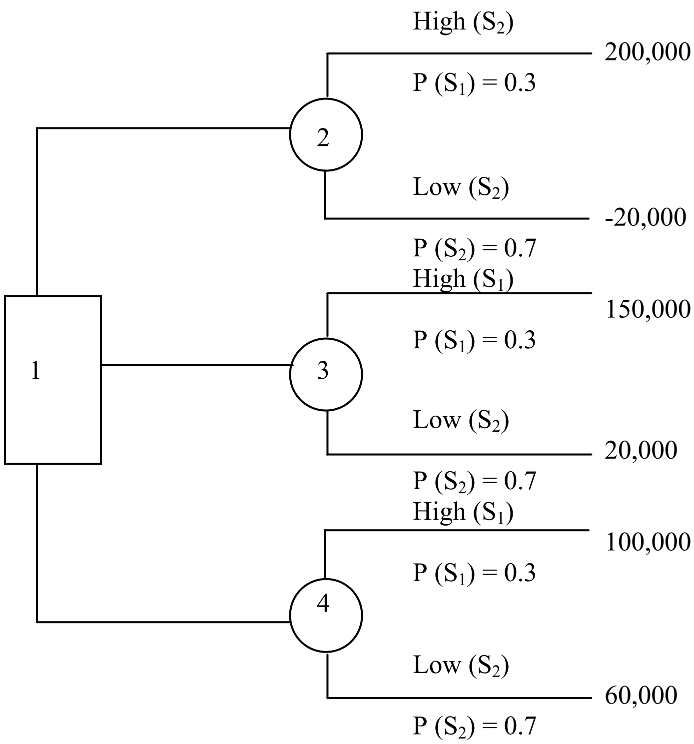
EV (D_2) = 0.3 (150,000) + 0.7 (20,000) = N59,000

EV (D_3) = 0.3 (100,000) + 0.7 (60,000) = N72,000

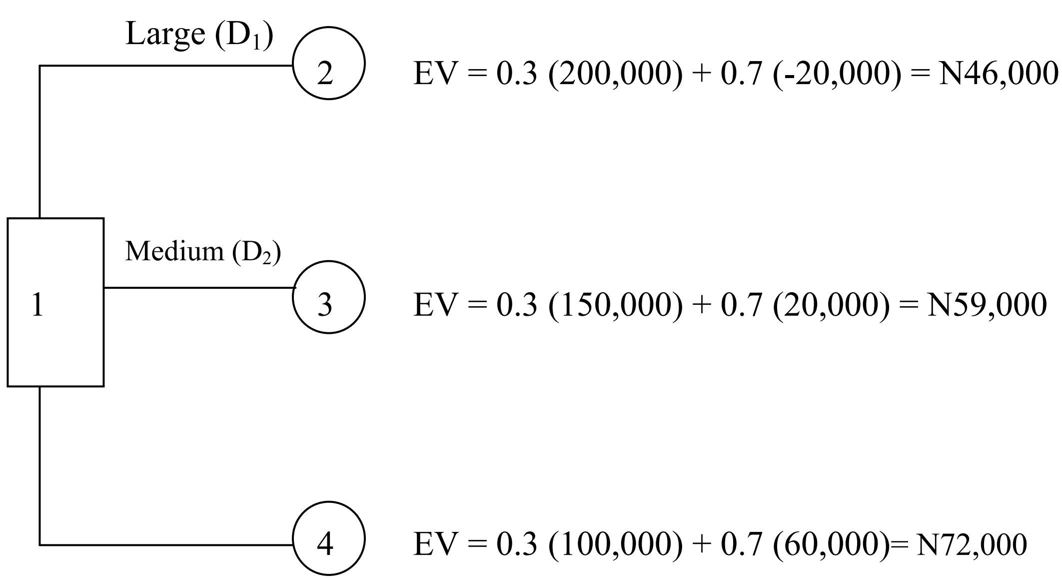
Thus, according to the expected value approach, D3 is the recommended decision since D_3 has the highest expected value (N72,000)

Using Decision Tree

Figure 2



PSI Decision Tree with State-of-Nature Branch Probabilities



Applying the Expected Value Approach Using Decision Trees

Sensitivity Analysis

In this section, we consider how changes in the probability estimates for the states of nature affect or utter the recommended decision. The study of the effect of such changes is referred to as ~. One approach to ~ is to consider different probabilities for the states of nature and then recompute the expected value for each decision alternative. Repeating this several times, we can begin to learn how changes in the probabilities for the states of nature affect the recommended decision. For example, suppose that we consider a change in the probabilities for the states of nature such that P (S_1) = 0.6 and P (S_2) = 0.4. Using these probabilities and repeating the expected value computations, we find the following:

EV (D_1) = 0.6 (200,000) + 0.4 (-20,000) = N112,000

EV (D_2) = 0.6 (150,000) + 0.4 (20,000) = N98,000

EV (D_3) = 0.6 (100,000) + 0.4 (60,000) = N84,000

Thus, with these probabilities, the recommended decision alternative is D_1, with an expected value of N112,000.

The only drawback to this approach is the numerous calculations required to evaluate the effect of several possible changes in the state-of-nature probabilities.

DYNAMIC PROGRAMMING AND MULTISTAGE OPTIMIZATION

Optimization problems consist in selecting from among the feasible alternatives one which is economically optimal. A problem of this nature is solved by formulating a mathematical model of the problem, typically a maximization model in which a preference function is to be maximized subject to a number of side conditions, and applying a method of solution tailored to the particular kind of problem. The variables of the model, interdependent through the side relations, are determined simultaneously in the solution.

Consider, for example, the linear programming problem

$$f = \quad 8x_1 + 10x_2 = \max$$

$$4x_1 + 2x_2 \leq 12 \qquad (1)$$

$$x_1, x_2 \geq 0$$

to be interpreted as a problem of optimal capacity utilization. x_1 and x_2 are quantities produced per period of two commodities which require 4 and 2 machine hours per unit, and the right-hand side of the side condition is maximum available machine time per period, the coefficients in the preference function are unit profits. Solving by the simples method (or, what is simpler in such a trivial case, geometrically or by numerical inspection) we get.

$x_1 = 0$, $x_2 =$, $f = 60$

The optimal value of the two decision variables is found simultatiously in the solution procedure.

An alternative approach is to determine the variable one at a time (sequentially), decomposing the problem into a series of stages each corresponding to a sub problem in only one variable, and solving the two single-variable problem (1). This is the basic idea underlying dynamic programming (DP).

The decomposition of the problem (1) can be illustrated as shown in fig. 8. Let us assume for convenience that commodity no. 1 is produced "first" (stage 1). We might as well have started with the second commodity; the order in which they are arranged is purely formal in a case like this where the decomposition into stages does not reflect a sequence in time.

Now, for the production of the first commodity, 12 units of the capacity factor (machine hours) are available as shown in the flow diagram. If x_1 units are produced, $12 - 4x_1$ machine hours are available as input for the second stage. After producing x_2 units of commodity

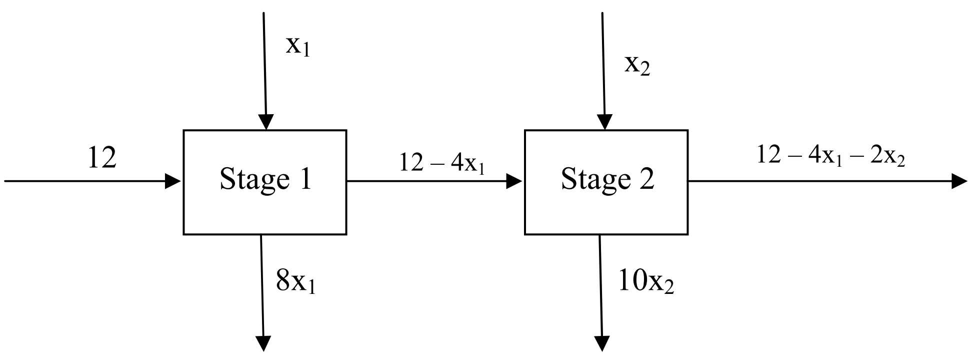


Figure 8

No 2, we are left with $12 - 4x_1 - 2x_2$ machine hours, corresponding to the slack variable in (1) which represents unutilized capacity. The two stages contribute $8x_1$ and $10x_2$ respectively to total profit.

We can now solve the problem *backwards*, treating x_1 as a *parameter* and optimizing *stage 2* with respect to the variable x_2. For parametric $x_1 = x_2$ the maximal capacity left to stage 2 is $12 - 4x_1$ machine hours so that the (parametric) subproblem of stage 2 is

$$f_1 = 10x_2 = \max$$
$$2x_2 \leq 12 - 4x_1$$
$$x_2 \geq 0 \qquad (3)$$

which is a linear programming problem like the total problem (1), only it is single-variable problem. The solution is obviously.

$$x_2 = 6 - 2x_1; f_1^{max} = 60 - 20x_1 \qquad (4)$$

where x_1 is a parameter.

Next we optimize *stage 1* with respect to its decision variable, x_1. The capacity available is 12 machine hours. Production of x_1 units contributes $8x_1$ to total profit, but against this we have to consider that machine hours left over and used by stage 2 also affect total profit, contributing $f_1^{max} = 60 - 20x_1$ which also depends on x1. The optimization problem of stage 1 becomes

$$f_2 = 8x_1 + f_1^{max} = 60 - 12x_1 = max$$
$$4x_1 \leq 12 \qquad (5)$$
$$x_1 \geq 0$$

which is also a linear problem. Because f_2 – which expresses profit contributed by the first stage plus the (parametric) maximum profit earned by the second – is a decreasing function of x1, the solution obviously is

$$x_1 = 0; f_2^{max} = 60 \qquad (6)$$

Having thus found the optimal value of x_1, which is also a parameter in the solution for x_2, we substitute it into (4) to get

$$x_2 = 6 - 2x_1 = 60 \qquad (7)$$

The solutions which we have found for the two variables are seen to agree with (2), and total profit $f = 60$ is seen to be equal to f_2^{max} – as it should be seen f_2^{max} was calculated as the total of stage contributions to profit. What we did in solving (5) was to maximize the profit of stage 1 – a function of x_1 – plus the maximum profit earned by stage 2 for any given value of x_1.

In this way we have solved an optimization problem in two variables by transforming it into a series, or sequence, or two single-variable problems. This is an example of dynamic programming. The subproblems corresponding to the individual stage are of the same type as the total problem (1) – in the present case, a linear programming problem – and they are solved by the same method as that applied in the simultaneous solving of the total problem (e.g. the simplex method). In other words, "dynamic programming" does not refer to a particular class of optimization problems (e.g. linear programming problems) or to a specific method of solution (like the simplex method); rather, it indicates a general procedure for decomposing a problem into a series of subproblems involving fewer variables, and combining their solutions to get the solution of the original problem.

3. When an optimization problem is formulated as a multistage problem to be solved by dynamic programming, it is convenient to introduce *state variables* y_n associated with the individual stages (numbered n = 1, 2, .., n). The production process of Figure 8 may be thought of as starting in a (given) *initial state* where $y_0 = 12$ machine hours are available; this is the input state of stage 1. Producing x_1 units of the first commodity, each of which requires 4 machine hours, changes the state of the system: available capacity is reduced by $4x_1$ machine hours so that the output state of stage 1 – which is also the input state to stage 2 – becomes $y_1 = y_0 - 4x_1 = 12 - 4x_1$. Producing x_2 units of the second product, available capacity is further reduced to $y_2 = y_1 - 2x_2 = y_0 - 4x_1 - 2x_2$, which in this case represents the final state. ($y_2 \geq 0$ by definition but otherwise unknown).

Thus, the input state of stage no. n, y_{n-1}, is transformed into an output state y_n, the change being brought about by the decision variable

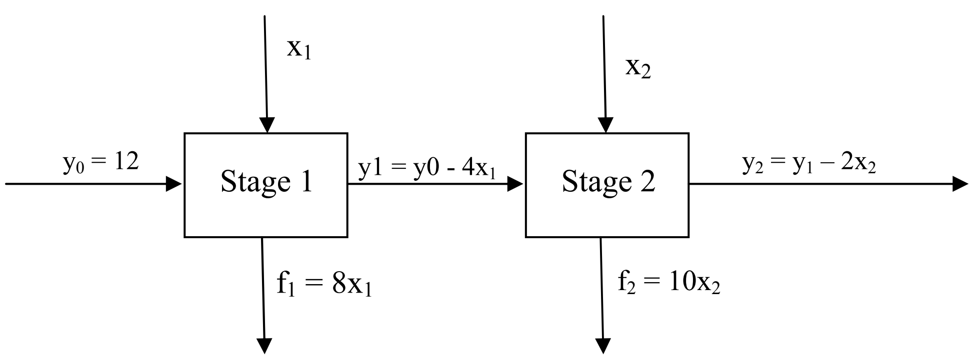


Figure 9

of the stage, xn. The successive changes of the state of the system can formally be described by transformation equations of the form

$$y_n = t_n (y_{n-1}, x_n) \ (n = 1, 2, \ldots, n).$$

In the example they have the form

$$y_1 = y_0 - 4x_1$$
$$y_2 = y_1 - 2x_2 \qquad (8)$$

together with the nonnegativity requirement $x_1, x_2, y_1, y_2 \geq 0$ they are equivalent to the restrictions of the original problem (1), y_0 being = 12 and y2 representing the slack variable.

The stage returns, i.e. the contributions of the individual stages to the preference function *f,* will in the general case depend on the input state and the decision variable:

$$r_n = r_n\,(y_{n-1}, x_n)\ (n = 1, 2, \ldots, n);$$

in the present example these *return functions* are of the simple form

$$r_1 = 8x_1$$
$$r_2 = 10x_2 \qquad (9)$$

Introducing these symbols into Fig. 8, the flow diagram of the two-stage problem has the form of Fig. 9. The backward solution now proceeds as follows.

At the *first* stage of the calculations – corresponding to the last stage in the production system, n = N = 2 – the input style y_1 is considered as a parameter, "inherited" from the previous stage of the system. The stage is optimized by maximizing its decision function, f_1 – here equal to the stage return, $r_2\,(x_2)$ – subject to the parametric capacity restriction $2x_2 \leq y1^1$:

$$f_1 = r_2\,(x_2) = 10x_2 = \max$$
$$2x_2 \leq y_1$$
$$x_2 \geq 0 \qquad (10)$$

The solution to this parametric single-variable LP problem is

$$x_2\,(y_1) = 0.5y_1$$
$$f_1\,(y_1) = 5y_1 \qquad (11)$$

where f_1 denotes the maximum value of the stage decision function, $F_1 = f_1^{max}$

At the *second stage* of the computations (production stage 1) we maximize the decision function $f_2 = r_1\,(x_1) + f_1\,(y_1)$ subject to the capacity limitation $4x_1 \leq y_0$ (i.e., y_1 =

$y_0 - 4x_1 \geq 0$); substituting the stage transformation $y_1 = y_0 - 4x_1$, f_2 becomes a function of x_1 and y_0 so that we have the LP problem.

$$f_2 = r_1(r_1) + F_1(y_1) = 8x_1 + 5y_1$$

$$= 8x_1 + 5(y_0 - 4x_1) = 5y_0 - 12x_1 = \max \qquad (12)$$

$$4x_1 \leq y_0$$

$$x_1 \geq 0.$$

The solution is

$$x_1(y_0) = 0$$

$$f_2(y_0) = 5y_0 \qquad (13)$$

when $F_2 = f_2^{\max}$.

The solution to the complete problem (1) – called *the optimal policy* – can now be determined by solving the *recursive equation system* formed by the *parametric optimum solutions* (11) and (13) and the *transformation equations* (8), starting from the initial state $y_0 = 12$:

Transformation Equations $y_n = t_n(y_{n-1}, x_n)$	Parametric Optimal solutions $x_n = x_n(y_{n-1})$	Maximum of decision function
$y_0 = 12$		
$y_1 = y_0 - 4x_1 = 12$	$x_1(y_0) = 0$	$f_2(y_0) = 5y_0 = 60$
$y_2 = y_1 - 2x_2 = 0$	$x_2(y_1) = 0.5y_1 = 6$	$f_1(y_1) = 5y_1 = 60$

where the direction of the computations is the opposite of that followed above in the optimization of stages. The optimal policy emerges as $x_1 = 0$, $x_2 = 6$ and total profit is $f = F_2 = 60$. If y_0 had been a parameter, the solution – now parametric – would have been $x_1 = 0$, $x_2 = 0.5y_0$, $f = 5y_0$[1].

The decomposition by which we solved problem (1) can be described as follows. Replacing the side conditions by the equivalent formulation

$$y_1 = y_0 - 4x_1,\ y_1 \geq 0$$

$$y_2 = y_1 - 2x_1,\ y_2 \geq 0,$$

this together with the nonnegativity requirements implies

$$0 \le x_1 \le \tfrac{1}{4} y_0 \quad (14)$$

$$0 \le x_2 \le \tfrac{1}{2} y_1 \quad (15)$$

Then we can write (1) in the form

$$F^{max} = \max_{x_1, x_2} (8x_1 + 10x_2) = \max_{x_1} (8x_1 + \max_{x_2} 10x_2)$$

subject to (14) – (15); clearly this maximization problem can be decomposed into two single-value problems, corresponding to (10) and (12):

$$F_1 (y_1) = \max 10x_2 \ (0 \le x_2 \le \tfrac{1}{2} y_1)$$

$$F_2 (y_0) = \max \{8x_1 + F_1 (y_1)\}$$

$$= \max \{8x_1 + F_1 (y_0 - 4x_1)\} \ (0 \le x_1 \le \tfrac{1}{4} y_0),$$

Where $f^{max} = F_2 (y_0)$. In a more general formulation the decomposition of a two-stage problem

$$F^{max} = \max [r_1 (y_0, x_1) + r_2 (y_1, x_2)]$$

can be expressed in the *recursion equations*

$$F_1 (y_1) = \max [r_2 (y_1, x_2)] \quad (16)$$

$$F_2 (y_0) = \max [r_1 (y_0, x_1) + F_1 \{t_1 (y_0, x_1)\}]. \quad (17)$$

This procedure of solving a dynamic programming problem by *backward recursion* can be generalized to any number of variables. A flow diagram for an N-stage system is shown in Fig. 10.

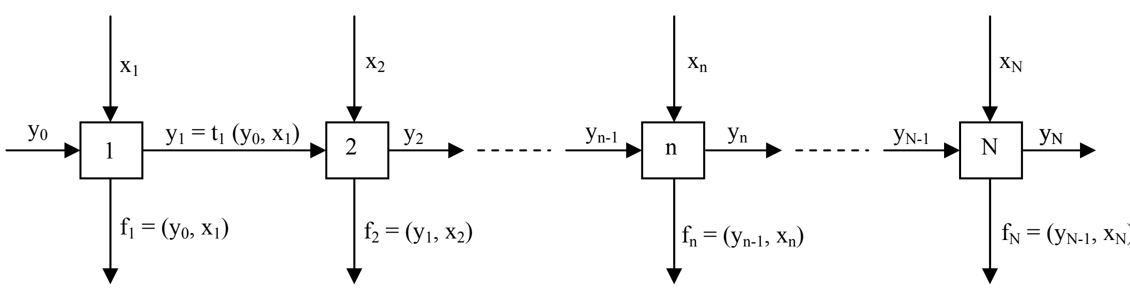


Figure 10

The decision functions of stages N, N – 1, …, 2, 1 are respectively

$f_1 = r_N (y_{N-1}, x_N)$

$f_2 = r_{N-1} (y_{N-2}, x_{N-1}) + F_1 (y_{N-1})$ where $y_{N-1} = t_{N-1} (y_{N-2}, x_{N-1})$ (18)

…

$f_N = r_1 (y_0, x_1) + F_{N-1} (y_1)$ where $y_1 = t_1 (y_0, x_1)$,

F_j being the maximum of f_j ($j = 1, 2, …, N$). Maximizing the decision function of each stage with respect to its decision variable, treating the input state as a parameter, we get the parametric stage solutions

$$x_n = x_n (y_{n-1}) \ (n = N, N-1, …, 1) \qquad (19)$$

which can be "sewn together" by means of the transformation equations so that we get the parameters determined. y_0 now determines x_1 and together they determine y_1; this gives x_2 which with y_1 determines y_2; and so forth as illustrated by Fig. 11

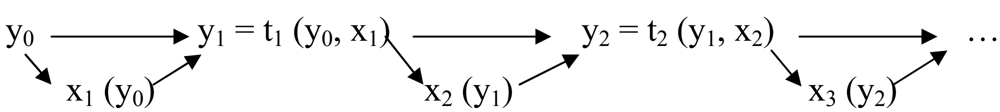


Figure 11

It follows from (18) that the maximum of f_N represents the accumulated value of the stage returns, e.g. $F_N = f^{max}$.

If the variables of a DP problem are allowed to take *discrete* values only, or if the stage returns and/or the transformation functions are given in *tabular form* for discrete values of the variables, the problem will have to be solved by *tabular computations.*

For example, the decision variables may be required to have *integral* values (x_n = 0, 1, 2, …) because the interpretation of the problem is such that fractional values would be meaningless. This is so in problem (1): strictly speaking it is impossible to produce a fractional number of units of a commodity, e.g. $x_1 = 2.6$. Thus, although in this case the analytical shape of the *return functions* is known, r_1 and r_2 are defined only for integral values of x_1 (≤ 3) and x_2 (≤ 6):

x_1	0	1	2	3
r_1 (= $8x_1$)	0	8	16	24

x_2	0	1	2	3	4	5	6

r_2 (= $10x_2$)	0	10	20	30	40	50	60

The *transformation functions* in tabular form are as follows:

	y_1 (= $y_0 - 4x_1$)			
y_0\ x_1	0	1	2	3
12	12	8	4	0

	y_2 (= $y_1 - 2x_2$)						
y_1 \ x_2	0	1	2	3	4	5	6
0	0						
4	4	2	0				
8	8	6	4	2	0		
12	12	10	8	6	4	2	0

where y_1 is confined to the values 0, 4, 8, and 12 resulting from the first table. The black cells in the last table correspond to combinations of values of y_1 and x_2 which are not feasible because they would imply a negative value of y_2 (cf. the sign restriction $y_2 = y_1 - 2x_2 \geq 0$).

The solution procedure, using backward recursion, now proceeds as follows. For the *last stage* (n = N = 2) we have:

Stage 2	$f_1 = r_2(x_2)$ (= $10x_2$)									
y_1 \ x_2	0	1	2	3	4	5	6	$F_1(y_1)$	$x_2(y_1)$	$y_2(y_1)$
0	0							0	0	0
4	0	10	20					20	2	0
8	0	10	20	30	40			40	4	0
12	0	10	20	30	40	50	60	60	6	4

where the maximal value of the decision function f_1 for each of the possible input states (values of y_1) is shown in bold-faced type. This parametric optima and the optimal values of the decision variable are listed in the F_1 and x_2 columns to the right. The last column gives the resulting values of the output state y_2, computed from the transformation function (or table).

For *stage 1*, we have the decision function $f_2 = r_1\ (x_1) + F_1\ (y_1)$ where the transformation gives y_1 for each value of x_1; for example, $x_1 = 2$ implies $y_1 = y_0 - 4x_1 = 4$, and the preceding table then gives $F_1\ (y_1) = 20$. The stage computations are done in the following table:

Stage 1	$f_2 = r_1\ (x_1) + F_1\ (y_1)$						
	$\{ = 8x_1 + f_1\ (y_1)\}$						
$y_0 \setminus x_1$	0	1	2	3	$F_2\ (y_0)$	$x_1\ (y_0)$	$y_1\ (y_0)$
12	0+60	8+40	16+20	24+0	60	0	12

These tables correspond to (10) – (11) and (12) – (13) respectively, and the optimal solution can be found similarly, starting with the last table. $y_0 = 12$ gives $x_1 = 0$ (the optimal stage solution) which leads to $y_1 = 12$ (by transformation equation). Proceeding to the first table, $y_1 = 12$ (as just determined) gives the stage optimum $x_2 = 6$ and the output state $y_2 = 0$. These values are indicated in italics. The maximal total return is $f_2\ (y_0) = 60$.

It will now be clear why it was expedient to solve the decomposed version of problem (1) *backwards,* starting with the optimization of the last stage. The procedure led to a recursive system which has the *initial state* y_0 as it starting point – as shown in Fig. 11 – and it was y_0 that was given ($y_0 = 12$). This suggests that problems in which the *final state* y_N is given may be solved in the opposite direction, proceeding *forwards* from the first stage.

To show how this is done, let us redefine the state variables y_n in the example so that the final state y_2 now represents total accumulated "use" of capacity, including idle capacity; the latter is put first as the input state y_0 of the first stage of the production system. Then y_1 represents the accumulated "use" of capacity, including capacity not utilized, after the first commodity has been produced. This leads to the transformations.

$$y_1 = y_0 - 4x_1$$

$$y_2 = y_1 - 2x_2 \qquad (20)$$

where $y_0 \geq 0$ and $y_2 = 12$. The stage return functions are the same as above, (9).

This dynamic programming problem is another decomposed version of problem (1). To solve it by *forward recursion* we reverse the direction

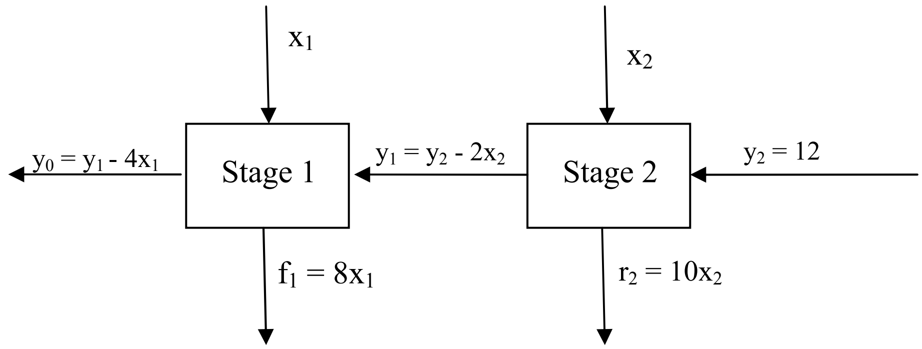


Figure 12

of the system as shown in Fig. 12, where y_n is now to be formally treated as the input state of stage n (n = 1, 2) whereas the y_{n-1} become output states; we therefore write the transformation functions (20) in the inverse form

$$y_0 = y_1 - 4x_1$$
$$y_1 = y_2 - 2x_2 \qquad (21)$$

(more generally,

$$y_{n-1} = t_n^* (y_n, x_n)$$

where t_n^* is the inverse transformation equation of stage n).

the procedure starts with the optimization of stage 1. The decision function $f_1 = r_1 (x_1)$ is to be maximized subject to $y_0 = y_1 - 4x_1 \geq 0$, $x_1 \geq 0$, where y_1 – now the input state of the stage – is a parameter:

$$f_1 = r_1 (x_1) = 8x_1 = \max$$
$$4x_1 \leq y_1 \qquad (22)$$
$$x_1 \geq 0;$$

the solution depends on the parameter,

$$x_1 (y_1) = 0.25y_1 \qquad (23)$$
$$F_1 (y_1) = 2y_1$$

where $F_1 = f_1^{max}$. At the *second stage* we have the decision function $f_2 = r_2 (x_2) + F_1 (y_1)$ where $y_1 = y_2 - 2x_2 \geq 0$, $x_2 \geq 0$ so that the stage optimization problem becomes

$$f_2 = 10x_2 + 2y_1 = 6x_2 + 2y_2 = \max \tag{24}$$

$$2x_2 \leq y_2$$

$$x_2 \geq 0.$$

The solution is

$$x_2 (y_2) = 0.5y_2$$

$$F_2 (y_2) = 5y_2 \tag{25}$$

Working backwards from y_2 (= 12) through the recursive system (21), (23), (25) we obtain the optimal solution:

$$y_2 = 12$$

$$y_1 = y_2 - 2x_2 = 0 \qquad x_2 (y_2) = 0.5y_2 = 6$$

$$y_0 = y_1 - 4x_1 = 0, \qquad x_1 (y_1) = 0.25y_1 = 0,$$

and $f^{max} = F_2 (y_2) = 5y_2 = 60$.

In the general case of N stages the decision functions are

$f_1 = r_1 (y_0, x_1)$ where $y_0 = t_1^* (y_1, x_1)$

$f_2 = r_2 (y_1, x_2) + F_1 (y_1)$ where $y_1 = t_2^* (y_2, x_2)$

…

$f_N = r_N (y_{N-1}, x_N) + F_{N-1} (y_{N-1})$ where $y_{N-1} = t_N^* (y_N, x_N)$.

The parametric stage solutions

$$x_n = x_n (y_n) \qquad (n = 1, 2, \ldots, N)$$

and the inverse transformation equations determine the optimal policy for given y_N as shown in Fig. 13

It is often possible to solve by forward recursion when the initial state y_0 is given, or to apply backward recursion to DP problems with given final state y_N, using a slightly modified procedure. However, when the order of the stages is arbitrary and the transformations can be inverted so that we are free to choose the direction, *backward recursion* is generally a more efficient procedure for *given y_0*, and *forward recursion* for *given y_N*, (which amounts to the same thing, the only difference being the numbering of stages and variables).

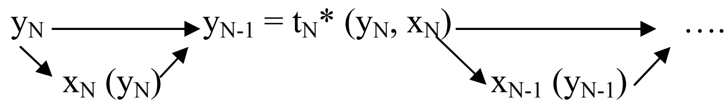


Figure 13

In order for an optimization problem to be solvable by dynamic programming, the "*technical*" structure of the problem (as represented by the *restrictions*) must be such that it can be described by a series of successive changes of the state of the system, from the initial state y_0 to the final state y_N, each change being effected by a particular decision variable. The two-stage system of Fig. 9 above is an example of this; the transformation equations (8) together with x_1, y_1, x_2, $y_2 \geq 0$ and $y_0 = 12$ are an equivalent reformulation of the restriction in problem (1), i.e., $4x_1 + 2x_2 \leq 12$ and x_1, $x_2 \geq 0$, so that the set of feasible solutions is preserved. In some cases the multistage structure represents a sequence in time – hence the name "dynamic" programming – where the stages correspond to actual processes and the direction indicates the order in which the transformations take place. This would be so in our example (1) if the two commodities were produced in separate processes and commodity no. 1 had to be made first. In many applications, however, the sequence of stages and the order in which the system passes through them are an artificial device, introduced in order make the problem solvable by DP methods. In either case we can choose between forward and backward recursion if the direction is mathematically arbitrary.

Decomposition of a problem also requires that the *objective function* satisfies certain conditions. In general, the function to be maximized in some function of the stage returns r_n,

$$f = \varphi \{r_1 (y_0, x_1), \ldots, r_N (y_{N-1}, x_N)\}. \qquad (26)$$

It can be shown that two conditions on the function – separability and monotonicity – together are sufficient for decomposition, i.e., for solution by means of a system of recursive equations. These conditions are automatically satisfied by a class of functions including the case of *additive returns*,

$$f = r_1 (y_0, x_1) + \ldots + r_N (y_{N-1}, x_N).$$

In this case, as we have seen above, the objective function is obviously decomposable and the recursion equations have the form (16) – (17) for a two-stage problem, readily generalized to any number of stages [cf. (18)].

The recursion equations can be thought of as a mathematical expression of an intuitive principle known as the *"principle of optimality"*. Consider the two-stage problem shown in Fig. 9, and let $(x_1, x_2) = (x_1, x_2)$ be the optimal policy. The first decision, $x_1 = x_1$, changes the state of the system from the initial state $y_0 = 12$ to $y_1 = y_0 - 4x_1 = y_1$. The principle of optimally now says that the remaining decision, $x_2 = x_2$, must represent an optimal policy with respect to the state y_1, i.e. it must be an optimal solution to the remaining one-step DP problem with the initial state y_1. The proof is simple: if this were not so, $(x_1, x_2) = (x_1, x_2)$ could not be an optimal policy.

The backward recursive procedure, as expressed in (16) – (17), by which we solved the problem, is based directly on this principle. Starting by optimizing the last stage (n = 2), we do not know its input stage y_1, but we do know that whatever it is – i.e., whatever the first decision is – r_2 must be optimal with regard to y_1. Hence we optimize $f_2 = r_2 (y_1, x_2)$ for parametric y_1 as expressed in (16). Proceeding backwards to stage one, we optimize $f_2 = r_1 (y_0, x_1) + F_1 (y_1)$, i.e. the return of stage 1 plus the parametric optimal return of stage 2, where $y_1 = t_1 (y_0, x_1)$.

The principle of optimality in its general form states that any part of an optimal policy must be optimal; specifically, the decisions remaining after stage no. n (i.e., $x_{n+1}, \ldots, x_N$) constitute an optimal policy for the series of stages n + 1, …, N with regard to the state y_n resulting from the first n decisions.

The multistage structures dealt with above (cf. Fig. 10) are *serial systems,* i.e., ordered sequences of stages where the output state of stage no. n is the input state to stage no. n + 1. Moreover, they are special in that there is *only one decision variable* x_n and *only one (output) state variable* y_n for each stage.

As an obvious generalization, the x_n and the y_n may be vectors so that there are *several decision and state variables per stage*. Cases of this kind are treated in Chapter VII below.

Nonserial multistage systems – important in the chemical industries – represent another generalization, characterized by branches or loops in the flow diagram.

A third extension of the N-stage serial multistage structure is an *infinite-stage system* where N tends to infinity. This case, relevant to some applications, is treated below in Chapter VIII.

The advantage of dynamic programming as a procedure for solving optimization problems is the simplification obtained by decomposition. It is often simpler and easier to solve a series of single-variable stage optimization problems, and in some cases this is the only possible procedure because "simultaneous" solution is mathematically or computationally difficult or downright impossible. Certain classes of optimization problems, however, such as linear programming problems are more efficiently solved by special algorithms without decomposition, e.g. the simplex method.

Dynamic programming has been particularly successful in its discrete version. Tabular computations are well suited for computer solution and can be used to handle problems involving irregular functions, and maybe the only practicable way of solving a problem in which the variables are required to be integers. As a rough illustration of the computational advantages of DP, consider an optimization problem in N decision variables, each of which can assume m alternative discrete values (e.g. 0, 1, 2, …, m – 1). If, for lack of other methods of solution, we had to solve the problem by total enumeration, we would have to examine each of m^N alternative solutions for feasibility and optimality, whereas in a dynamic programming procedure – assuming that decomposition is possible – the number of alternatives to be enumerated would be reduced to mN, namely m for each of the N stages. Thus, roughly speaking, the computational labour increases exponentially with the number of decision variables in the case of total exponentially with the number of decision variables in the case of total enumeration, but only proportionately if the problem is decomposed. For large values of m and N the computational advantages become enormous; for example, in a problem with 20 variables each of which can assume integral values from 0 to 9 we have $m^N = 10^{20}$ –

an astronomic number – as against $mN = 200$ possible solutions if the problem is reformulated as a 20-stage DP problem.

APPLICATIONS OF DYNAMIC PROGRAMMING

The Shortest Path through a Network

Perhaps the simplest and most straightforward application of dynamic programming is the determination of *the shortest path or whole through a network.*

Consider the (stylized) road map shown in Fig. 144. A driver wants to find the shortest route from point P to point Q. There are six intermediate junctions A, B, …, F. The lengths of all existing road sections connecting two points in the area are indicated on the map. Any unbroken chain of road sections starting at P and ending at Q represents a possible route through this network of roads.

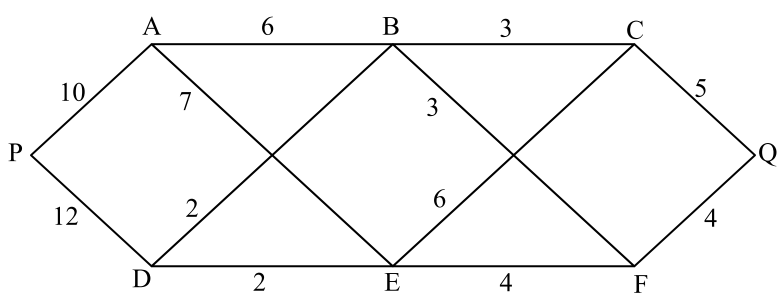


Figure 14

Assuming that the direction of travel is always from left to right having arrived at, say, B the driver never travels back to A or D but proceeds to either C or F – the number of possible routes is finite. The problem can therefore be solved by enumerating the alternative routes and comparing their total lengths.

Any route from P to Q is the result of *three successive decisions.* Starting at point P, the driver must decide whether he will go to A or to D. If he chooses to drive to A, say, he can proceed from A to either B or E, each of which in turn leaves him with two alternatives, C and F. Having arrived at either C or F, he has no choice but no proceed to the destination, Q. Since each decision is a choice between two alternatives and there are three consecutive decisions to be made, there are $2^3 = 8$ possible combinations, i.e. 8 alternative routes. The alternative decisions and the resulting routes can be illustrated graphically by a *decision tree* as shown in Fig. 15, where the root represents the starting point P and the branches are composed of road sections (with lengths indicated). Comparing the total lengths from root to top, it will be seen that PDBFQ is the shortest route, the total length being 12 + 2 + 3+ 4 = 21.

Figure 15

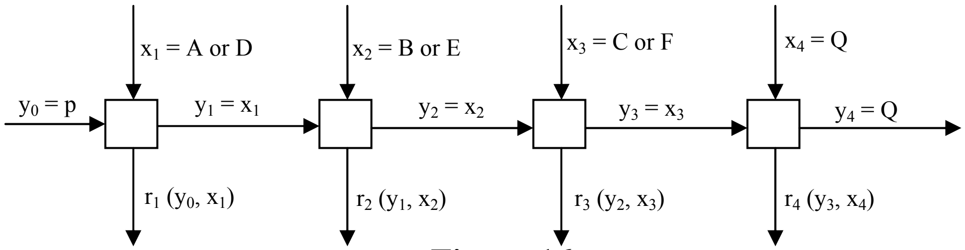


Figure 16

This decision structure clearly represents a multistage decision system. The driver's geographical position – the points on the map – represents the state of the system, which is to be changed from the given initial state P to the given final state Q through a sequence of stage decisions in such a way as to minimize the total "return", i.e., the total distance covered. Starting from point P, i.e., the initial state y_0 = P, the two alternatives open to the driver can be represented by two values of a decision variable x_1 : x_1 = A (i.e., drive to point A) and x_1 = D (drive to D). If he chooses x_1 = A, he will get to this point so that the output state of the first stage will be y_1 = A and the corresponding "return" r_1 (P, A) will be the distance PA = 10. Similarly, x_1 D leads to y_1 = D and the return will be r_1 (P, D) = 12. Proceeding in this fashion, the problem can be represented by a four-stage decision structure as shown in Fig. 16. There is no choice at stage 4 since the destination – i.e., the final state y_4 = Q = is given.

The transformation functions are $y_n = x_n$ (n = 1, 2, 3, 4). The return functions $r_n = r_n (y_{n-1}, x_n)$ can be written in tabular form as follows

$r_1 (y_0, x_1)$		
$y_0 \setminus x_1$	A	D
P	10	12

$r_2 (y_1, x_2)$		
$y_1 \setminus x_2$	B	E
A	6	7
D	2	2

$r_3 (y_2, x_3)$		
$y_2 \setminus x_3$	C	F
B	3	3
E	6	4

$r_4 (y_3, x_4)$	
$y_3 \setminus x_4$	Q
C	5
F	4

Using backward recursion, the problem is solved by tabular computations as follows:

Stage 4 $y_3 \backslash x_4$	$f_1 = r_4\ (y_3, r_4)$ Q	$F_1\ (y_3)$	$x_4\ (y_3)$	$y_4\ (y_3)$
C	5 (= CQ)	5	Q	Q
F	4 (= FQ)	4	Q	Q

Stage 3 $y_2 \backslash x_3$	$f_2 = r_3\ (y_2, x_3) + F_1\ (y_3)$ C	F	$F_2\ (y_2)$	$x_3\ (y_2)$	$y_3\ (y_2)$
B	3 + 5	3 + 4	7	F	F
E	6 + 5	4 + 4	8	F	F

Stage 2 $y_1 \backslash x_2$	$f_3 = r_2\ (y_1, x_2) + F_2\ (y_2)$ B	E	$F_3\ (y_1)$	$x_2\ (y_1)$	$y_2\ (y_1)$
A	6 + 7	7 + 8	13	B	B
D	2 + 7	2 + 8	9	B	B

Stage 1 $y_0 \backslash x_1$	$f_4 = r_1\ (y_0, x_1) + F_3\ (y_1)$ A	D	$F_4\ (y_0)$	$x_1\ (y_0)$	$y_1\ (y_0)$
P	10 + 13	12 + 9	21	D	D

Starting with y_0 = P, the solution is determined by the recursive equation system formed by the parametric stage solutions $x_n = x_n\ (y_{n-1})$ and the transformations $y_n = x_n$. The last table gives $x_1\ (y_0)$ = D and $y_1 = x_1$ = D; for y_1 = D, the table for stage 2 gives x_2 = B, y_2 = B; and so on. The optimal values – italicized in the tables – are

n	y_n	x_n
0	P	
1	D	D
2	B	B
3	F	F
4	Q	Q

The optimal sequence of states y_0, y_1, …, y_4 indicates that the shortest path through the network is PDBFQ; the total length of this route is F_4 (y_0) = 21.

This solution procedure can be translated into a graphical method, making use of a decision tree like that of Fig. 15. After making the three first decisions the driver finds himself at point C or F. No matter how he got there he will have to proceed to Q, so we draw the eight top branches of the tree; none of them can be eliminated at this stage since the preceding decisions have not yet been determined.

Going one stage back, the first two decisions have taken the driver to either B or E. If he has arrived at B, he can get to his destination Q either through C or F; the best course is to go to F since BFQ = 3 + 4 = 7 whereas BCQ = 3 + 5 = 8. Therefore, no matter which way he may have got to B, he will never proceed to C, so the branches from B to C can be eliminated and we need only draw the branches going from B to F. This is an application of Bellman's principle of optimality: if the optimal route from P to Q passes through B, the remaining part of the route (from B to Q) must also be optimal; the optimal route from P to Q cannot contain BCQ because there is a shorter route from B to the destination. Similarly, if the driver is at point E after the first two decisions, he will proceed to F because EFQ = 4 + 4 + 8 < ECQ = 6 + 5 = 11, so ECQ can be eliminated.

Applying a similar reasoning to the state attained after the first decision, it is seen that the branches AE and DE need not be drawn; the shortest path from A to Q is ABFQ = 6 + 7 = 13 and the shortest path from D to Q is DBFQ = 2 + 7 = 9. Finally, at the starting point the choice between PA + ABFQ = 10 + 13 = 23 and PD + DBFQ = 12 + 9 = 21; the latter alternative represents the shortest total route and there is no need to draw the branch PA.