LECTURE NOTES

ON

OPTIMIZATION TECHNIQUES

V Semester

Mr. R M Noorullah

Associate Professor, CSE

Dr. K Suvarchala

Professor, CSE

Mr. J Thirupathi

Assistant Professor, CSE

Ms. B Geethavani

Assistant Professor, CSE

Ms. A Soujanya

Assistant Professor, CSE

INFORMATION TECHNOLOGY

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

UNIT - 1

INTRODUCTION TO OR

TERMINOLOGY

The British/Europeans refer to "operational research", the Americans to "operations research" - but both are often shortened to just "OR" (which is the term we will use). Another term which is used for this field is "management science" ("MS"). The Americans sometimes combine the terms OR and MS together and say "**OR/MS**" or "ORMS".

Yet other terms sometimes used are "industrial engineering" ("IE"), "decision science" ("DS"), and “problem solving”.

In recent years there has been a move towards a standardization upon a single term for the field, namely the term "OR".

“Operations Research (Management Science) is a scientific approach to decision making that seeks to best design and operate a system, usually under conditions requiring the allocation of scarce resources.”

A system is an organization of interdependent components that work together to accomplish the goal of the system.

THE METHODOLOGY OF OR

When OR is used to solve a problem of an organization, the following seven step procedure should be followed:

Step 1. Formulate the Problem

OR analyst first defines the organization's problem. Defining the problem includes specifying the organization's objectives and the parts of the organization (or system) that must be studied before the problem can be solved.

Step 2. Observe the System

Next, the analyst collects data to estimate the values of parameters that affect the organization's problem. These estimates are used to develop (in Step 3) and evaluate (in Step 4) a mathematical model of the organization's problem.

Step 3. Formulate a Mathematical Model of the Problem

The analyst, then, develops a mathematical model (in other words an idealized representation) of the problem. In this class, we describe many mathematical techniques that can be used to model systems.

Step 4. Verify the Model and Use the Model for Prediction

The analyst now tries to determine if the mathematical model developed in Step 3 is an accurate representation of reality. To determine how well the model fits reality, one determines how valid the model is for the current situation.

Step 5. Select a Suitable Alternative

Given a model and a set of alternatives, the analyst chooses the alternative (if there is one) that best meets the organization's objectives.

Sometimes the set of alternatives is subject to certain restrictions and constraints. In many situations, the best alternative may be impossible or too costly to determine.

Step 6. Present the Results and Conclusions of the Study

In this step, the analyst presents the model and the recommendations from Step 5 to the decision making individual or group. In some situations, one might present several alternatives and let the organization choose the decision maker(s) choose the one that best meets her/his/their needs.

After presenting the results of the OR study to the decision maker(s), the analyst may find that s/he does not (or they do not) approve of the recommendations. This may result from incorrect definition of the problem on hand or from failure to involve decision maker(s) from the start of the project. In this case, the analyst should return to Step 1, 2, or 3.

Step 7. Implement and Evaluate Recommendation

If the decision maker(s) has accepted the study, the analyst aids in implementing the recommendations. The system must be constantly monitored (and updated dynamically as the environment changes) to ensure that the recommendations are enabling decision maker(s) to meet her/his/their objectives.

HISTORY OF OR

In physics or engineering (for example) at university it would not have been possible to study OR, indeed the term OR did not exist then. It was only really in the late 1930's that operational research began in a systematic fashion, and it started in the UK.

Early in 1936 the British Air Ministry established Bawdsey Research Station, on the east coast, near Felixstowe, Suffolk, as the centre where all pre-war radar Z experiments for both the Air Force and the Army would be carried out. Experimental radar equipment was brought up to a high state of reliability and ranges of over 100 miles on aircraft were obtained.

It was also in 1936 that Royal Air Force (RAF) Fighter Command, charged specifically with the air defense of Britain, was first created. It lacked however any effective fighter aircraft - no Hurricanes or Spitfires had come into service - and no radar data was yet fed into its very elementary warning and control system.

It had become clear that radar would create a whole new series of problems in fighter direction and control so in late 1936 some experiments started at Biggin Hill in Kent into the effective use of such data. This early work, attempting to integrate radar data with ground based observer data for fighter interception, was the start of OR.

The first of three major pre-war air-defense exercises was carried out in the summer of 1937. The experimental radar station at Bawdsey Research Station was brought into operation and the information derived from it was fed into the general air-defense warning and control system. From the early warning point of view this exercise was encouraging, but the tracking information obtained from radar, after filtering and transmission through the control and display network, was not very satisfactory.

In July 1938 a second major air-defense exercise was carried out. Four additional radar stations had been installed along the coast and it was hoped that Britain now had an aircraft location and control system greatly improved both in coverage and effectiveness. Not so! The exercise revealed, rather, that a new and serious problem had arisen. This was the need to coordinate and correlate the additional, and often conflicting, information received from the additional radar stations. With the out-break of war apparently imminent, it was obvious that something new - drastic if necessary - had to be attempted. Some new approach was needed.

Accordingly, on the termination of the exercise, the Superintendent of Bawdsey Research Station, A.P. Rowe, announced that although the exercise had again demonstrated the technical feasibility of the radar system for detecting aircraft, its operational achievements still fell far short of requirements. He therefore proposed that a crash program of research into the operational - as opposed to the technical -aspects of the system should begin immediately. The term "operational research" [RESEARCH into (military) OPERATIONS] was coined as a suitable description of this new branch of applied science. The first team was selected from amongst the scientists of the radar research group the same day.

In the summer of 1939 Britain held what was to be its last pre-war air defense exercise. It involved some 33,000 men, 1,300 aircraft, 110 antiaircraft guns, 700 searchlights, and 100 barrage balloons. This exercise showed a great improvement in the operation of the air defense warning and control system. The contribution made by the OR teams was so apparent that the Air Officer Commander-in-Chief RAF Fighter Command (Air Chief Marshal Sir Hugh Dowding) requested that, on the outbreak of war, they should be attached to his headquarters at Stanmore.

On May 15th 1940, with German forces advancing rapidly in France, Stanmore Research Section was asked to analyze a French request for ten additional fighter squadrons (12 aircraft a squadron) when losses were running at some three squadrons every two days. They prepared graphs for Winston Churchill (the British Prime Minister of the time), based upon a study of current daily losses and replacement rates, indicating how rapidly such a move would deplete fighter strength. No aircraft were sent and most of those currently in France were recalled.

This is held by some to be the most strategic contribution to the course of the war made by OR (as the aircraft and pilots saved were consequently available for the successful air defense of Britain, the Battle of Britain).

In 1941 an Operational Research Section (ORS) was established in Coastal Command which was to carry out some of the most well-known OR work in World War II.

Although scientists had (plainly) been involved in the hardware side of warfare (designing better planes, bombs, tanks, etc) scientific analysis of the operational use of military resources had never taken place in a systematic fashion before the Second World War. Military personnel, often by no means stupid, were simply not trained to undertake such analysis.

These early OR workers came from many different disciplines, one group consisted of a physicist, two physiologists, two mathematical physicists and a surveyor. What such people brought to their work were "scientifically trained" minds, used to querying assumptions, logic, exploring hypotheses, devising experiments, collecting data, analyzing numbers, etc. Many too were of high intellectual caliber (at least four wartime OR personnel were later to win Nobel prizes when they returned to their peacetime disciplines).By the end of the war OR was well established in the armed services both in the UK and in the USA.OR started just before World War II in Britain with the establishment of teams of scientists to study the strategic and tactical problems involved in military operations. The objective was to find the most effective utilization of limited military resources by the use of quantitative techniques.

Following the end of the war OR spread, although it spread in different ways in the UK and USA.

You should be clear that the growth of OR since it began (and especially in the last 30 years) is, to a large extent, the result of the increasing power and widespread availability of computers. Most (though not all) OR involves carrying out a large number of numeric calculations. Without computers this would simply not be possible.

BASIC OR CONCEPTS

"OR is the representation of real-world systems by mathematical models together with the use of quantitative methods (algorithms) for solving such models, with a view to optimizing."

We can also define a mathematical model as consisting of:

- Decision variables, which are the unknowns to be determined by the solution to the model.
- Constraints to represent the physical limitations of the system
- An objective function
- An optimal solution to the model is the identification of a set of variable values which are feasible (satisfy all the constraints) and which lead to the optimal value of the objective function.

An optimization model seeks to find values of the decision variables that optimize (maximize or minimize) an objective function among the set of all values for the decision variables that satisfy the given constraints.

Two Mines Example

The Two Mines Company own two different mines that produce an ore which, after being crushed, is graded into three classes: high, medium and low-grade. The company has contracted to provide a smelting plant with 12 tons of high-grade, 8 tons of medium-grade and 24 tons of low-grade ore per week. The two mines have different operating characteristics as detailed below.

Mine	Cost per day (£'000)	Production (tons/day)		
		High	Medium	Low
X	180	6	3	4
Y	160	1	1	6

Consider that mines cannot be operated in the weekend. How many days per week should each mine be operated to fulfill the smelting plant contract?

Guessing

To explore the Two Mines problem further we might simply guess (i.e. use our judgment) how many days per week to work and see how they turn out.

- work one day a week on X, one day a week on Y

This does not seem like a good guess as it results in only 7 tons a day of high-grade, insufficient to meet the contract requirement for 12 tones of high-grade a day. We say that such a solution is infeasible.

- work 4 days a week on X, 3 days a week on Y

This seems like a better guess as it results in sufficient ore to meet the contract. We say that such a solution is feasible. However it is quite expensive (costly).

We would like a solution which supplies what is necessary under the contract at minimum cost. Logically such a minimum cost solution to this decision problem must exist. However even if we keep guessing we can never be sure whether we have found this minimum cost solution or not. Fortunately our structured approach will enable us to find the minimum cost solution.

Solution:

What we have is a verbal description of the Two Mines problem. What we need to do is to translate that verbal description into an equivalent mathematical description. In dealing with problems of this kind we often do best to consider them in the order:

Variables Constraints
Objective

This process is often called formulating the problem (or more strictly formulating a mathematical representation of the problem).

Variables

These represent the "decisions that have to be made" or the "unknowns".

We have two decision variables in this problem:

- x = number of days per week mine X is operated y = number of days per week mine Y is operated

Note here that $x \geq 0$ and $y \geq 0$.

Constraints

It is best to first put each constraint into words and then express it in mathematical form.Ore production constraints - balance the amount produced with the quantity required under the smelting plant contract

Ore	
High	$6x + 1y \geq 12$
Medium	$3x + 1y \geq 8$
Low	$4x + 6y \geq 24$

Days per week constraint - we cannot work more than a certain maximum number of days a week e.g. for a 5 day week we have

$x \leq 5$

$y \leq 5$

Inequality constraints

Note we have an inequality here rather than equality. This implies that we may produce more of some grade of ore than we need. In fact we have the general rule: given a choice between an equality and an inequality choose the inequality

For example - if we choose an equality for the ore production constraints we have the three equations 6x+y=12, 3x+y=8 and 4x+6y=24 and there are no values of x and y which satisfy all three equations (the problem is therefore said to be "over-constrained"). For example the values of x and y which satisfy 6x+y=12 and 3x+y=8 are x=4/3 and y=4, but these values do not satisfy 4x+6y=24.

The reason for this general rule is that choosing an inequality rather than an equality gives us more flexibility in optimizing (maximizing or minimizing) the objective (deciding values for the decision variables that optimize the objective).

Implicit constraints

Constraints such as days per week constraint are often called implicit constraints because they are implicit in the definition of the variables

Objective

Again in words our objective is (presumably) to minimize cost which is given by 180x + 160y

Hence we have the **complete mathematical representation** of the problem:

Minimize

$$180x + 160y$$

Subject to

$$6x + y \geq 12$$

$$3x + y \geq 8$$

$$4x + 6y \geq 24$$

$$x \leq 5$$

$$y \leq 5$$

$$x, y \geq 0$$

Some notes

The mathematical problem given above has the form all variables continuous (i.e. can take fractional values) a single objective (maximize or minimize) the objective and constraints are linear i.e. any term is either a constant or a constant multiplied by an unknown (e.g. 24, 4x, 6y are linear terms but xy or x^2 is a non-linear term)

Any formulation which satisfies these three conditions is called a linear program (LP). We have (implicitly) assumed that it is permissible to work in fractions of days - problems where this is not permissible and variables must take integer values will be dealt with under Integer Programming (IP).

Discussion

This problem was a decision problem.

We have taken a real-world situation and constructed an equivalent mathematical representation - such a representation is often called a mathematical model of the real-world situation (and the process by which the model is obtained is called formulating the model).

Just to confuse things the mathematical model of the problem is sometimes called the formulation of the problem.

Having obtained our mathematical model we (hopefully) have some quantitative method which will enable us to numerically solve the model (i.e. obtain a numerical solution) - such a quantitative method is often called an algorithm for solving the model.

Essentially an algorithm (for a particular model) is a set of instructions which, when followed in a step-by-step fashion, will produce a numerical solution to that model.

Our model has an objective that is something which we are trying to optimize. Having obtained the numerical solution of our model we have to translate that solution back into the real-world situation.

"OR is the representation of real-world systems by mathematical models together with the use of quantitative methods (algorithms) for solving such models, with a view to optimizing."

LINEAR PROGRAMMING

It can be recalled from the Two Mines example that the conditions for a mathematical model to be a linear program (LP) were:

- all variables continuous (i.e. can take fractional values) single objective (minimize or maximize)
- The objective and constraints are linear i.e. any term is either a constant or a constant multiplied by an unknown.

Many practical problems can be formulated as LP's there exists an algorithm (called the simplex algorithm) which enables us to solve LP's numerically relatively easily

We will return later to the simplex algorithm for solving LP's but for the moment we will concentrate upon formulating LP's.

Some of the major application areas to which LP can be applied are:

Work scheduling

- Production planning & Production process
- Capital budgeting

Financial planning

- Blending (e.g. Oil refinery management) Farm planning
- Distribution

Multi-period decision problems

- Inventory model
- Financial models
- Work scheduling

Note that the key to formulating LP's is practice. However a useful hint is that common objectives for LP's are maximized profit/minimize cost.

There are four basic assumptions in LP:

- **Proportionality**

  - The contribution to the objective function from each decision variable is proportional to the value of the decision variable (The contribution to the objective function from making four soldiers (4 $3=$12) is exactly four times the contribution to the objective function from making one soldier ($3)
  - The contribution of each decision variable to the LHS of each constraint is proportional to the value of the decision variable (It takes exactly three times as many finishing hours (2hrs 3=6hrs) to manufacture three soldiers as it takes to manufacture one soldier (2 hrs)

- **Additivity**

    - The contribution to the objective function for any decision variable is independent of the values of the other decision variables (No matter what the value of train (x2), the manufacture of soldier (x1) will always contribute 3x1 dollars to the objective function)

    - The contribution of a decision variable to LHS of each constraint is independent of the values of other decision variables (No matter what the value of x1, the manufacture of x2 uses x2 finishing hours and x2 carpentry hours)

        - 1st implication: The value of objective function is the sum of the contributions from each decision variables.

        - 2nd implication: LHS of each constraint is the sum of the contributions from each decision variables.

- **Divisibility**

    - Each decision variable is allowed to assume fractional values. If we actually can not produce a fractional number of decision variables, we use IP (It is acceptable to produce 1.69 trains)

- **Certainty**

    - Each parameter is known with certainty

FORMULATING LP

Giapetto Example

Giapetto's wooden soldiers and trains. Each soldier sells for $27, uses $10 of raw materials and takes $14 of labor & overhead costs. Each train sells for $21, uses $9 of raw materials, and takes $10 of overhead costs. Each soldier needs 2 hours finishing and 1 hour carpentry; each train needs 1 hour finishing and 1 hour carpentry. Raw materials are unlimited, but only 100 hours of finishing and 80 hours of carpentry are available each week. Demand for trains is unlimited; but at most 40 soldiers can be sold each week. How many of each toy should be made each week to maximize profits?

Solution:

Decision variables completely describe the decisions to be made (in this case, by Giapetto). Giapetto must decide how many soldiers and trains should be manufactured each week. With this in mind, we define:

x1 = the number of soldiers produced per week x2 = the number of trains produced per week

Objective function is the function of the decision variables that the decision maker wants to maximize (revenue or profit) or minimize (costs). Giapetto can concentrate on maximizing the total weekly profit (z).

Here profit equals to (weekly revenues) – (raw material purchase cost) – (other variable costs). Hence Giapetto's objective function is:

Maximize z = 3x1 + 2x2

Constraints show the restrictions on the values of the decision variables. Without constraints Giapetto could make a large profit by choosing decision variables to be very large. Here there are three constraints:

Finishing time per week

Carpentry time per week

Weekly demand for soldiers

Sign restrictions are added if the decision variables can only assume nonnegative values (Giapetto can not manufacture negative number of soldiers or trains!)

All these characteristics explored above give the following **Linear Programming** (LP) model

max z = 3x1 + 2x2		(The Objective function)
s.t.2x1 + x2 100		(Finishing constraint)
x1 +	x2 80	(Carpentry constraint)
x1	40	(Constraint on demand for soldiers)
x1, x2 > 0		(Sign restrictions)

A value of (x1, x2) is in the **feasible region** if it satisfies all the constraints and sign restrictions.

Graphically and computationally we see the solution is (x1, x2) = (20, 60) at which z =180 (**Optimal solution**)

<u>Report</u>

The maximum profit is $180 by making 20 soldiers and 60 trains each week. Profit is limited by the carpentry and finishing labor available. Profit could be increased by buying more labor.

Advertisement Example

Dorian makes luxury cars and jeeps for high-income men and women. It wishes to advertise with 1 minute spots in comedy shows and football games. Each comedy spot costs $50K and is seen by 7M high-income women and 2M high-income men. Each football spot costs $100K and is seen by 2M high-income women and 12M high-income men. How can Dorian reach 28M high-income women and 24M high-income men at the least cost?

Solution:

The decision variables are

x1 = the number of comedy spots

x2 = the number of football spots

The model of the problem:

Min z = 50x1 + 100x2

St 7x1 +2x2 >=28

2x1 +12x2 >+24

x1, x2 >= 0

The graphical solution is z = 320 when (x1, x2) = (3.6, 1.4). From the graph, in this problem rounding up to (x1, x2) = (4, 2) gives the best integer solution.

Report

The minimum cost of reaching the target audience is $400K, with 4 comedy spots and 2 football slots. The model is dubious as it does not allow for saturation after repeated viewings.

Diet Example

Ms. Fidan's diet requires that all the food she eats come from one of the four "basic food groups". At present, the following four foods are available for consumption: brownies, chocolate ice cream, cola, and pineapple cheesecake. Each brownie costs 0.5$, each scoop of chocolate ice cream costs 0.2$, each bottle of cola costs 0.3$, and each pineapple cheesecake costs 0.8$. Each day, she must invest at least 500 calories, 6 oz of chocolate, 10 oz of sugar, and 8 oz of fat. The nutritional content per unit of each food is shown in Table. Formulate an LP model that can be used to satisfy her daily nutritional requirements at minimum cost.

		Calories	Chocolate	Sugar	Fat
			(ounces)	(ounces)	(ounces)
Brownie		400	3	2	2

Choc. ice cream (1 scoop)	200	2	2	4
Cola (1 bottle)	150	0	4	1
Pineapple cheesecake (1 piece)	500	0	4	5

Solution:

The decision variables:

$x1$: number of brownies eaten daily

$x2$: number of scoops of chocolate ice cream eaten daily

$x3$: bottles of cola drunk daily

$x4$: pieces of pineapple cheesecake eaten daily

The objective function (the total cost of the diet in cents):

$$\min w = 50x1 + 20x2 + 30x3 + 80x4$$

Constraints:

$400x1 + 200x2 + 150x3 + 500x4 > 500$ (daily calorie intake)

$3x1 + 2x2 > 6$ (daily chocolate intake)

$2x1 + 2x2 + 4x3 + 4x4 > 10$ (daily sugar intake)

$2x1 + 4x2 + x3 + 5x4 > 8$ (daily fat intake)

$xi \geq 0, \; i = 1, 2, 3, 4$ (Sign restrictions!)

Report

The minimum cost diet incurs a daily cost of 90 cents by eating 3 scoops of chocolate and drinking 1 bottle of cola (w = 90, x2 = 3, x3 = 1

Post Office Example

A PO requires different numbers of employees on different days of the week. Union rules state each employee must work 5 consecutive days and then receive two days off. Find the minimum number of employees needed.

	Mon	Tue	Wed	Thur	Fri	Sat	Sun
Staff Needed	17	13	15	19	14	16	11

Solution:

The decision variables are xi (# of employees starting on day i)

Mathematically we must

$$
\begin{aligned}
\min z = \; & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \\
\text{s.t.} \; & x_1 + x_4 + x_5 + x_6 + x_7 \ge 17 \\
& x_1 + x_2 + x_5 + x_6 + x_7 \ge 13 \\
& x_1 + x_2 + x_3 + x_6 + x_7 \ge 15 \\
& x_1 + x_2 + x_3 + x_4 + x_7 \ge 19 \\
& x_1 + x_2 + x_3 + x_4 + x_5 \ge 14 \\
& x_2 + x_3 + x_4 + x_5 + x_6 \ge 16 \\
& x_3 + x_4 + x_5 + x_6 + x_7 \ge 11 \\
& x_t \ge 0, \; \forall t
\end{aligned}
$$

The solution is (xi) = (4/3, 10/3, 2, 22/3, 0, 10/3, 5) giving z = 67/3.

We could round this up to (xi) = (2, 4, 2, 8, 0, 4, 5) giving z = 25 (may be wrong!). However restricting the decision ver. to be integers and using Linda again gives (xi) = (4, 4, 2, 6, 0, 4, 3) giving z = 23.

Sailco Example

Sailco must determine how many sailboats to produce in the next 4 quarters. The demand is known to be 40, 60, 75, and 25 boats. Sailco must meet its demands. At the beginning of the 1st quarter Sailco starts with 10 boats in inventory. Sailco can produce up to 40 boats with regular time labor at $400 per boat, or additional boats at $450 with overtime labor. Boats made in a quarter can be used to meet that quarter's demand or held in inventory for the next quarter at an extra cost of $20.00 per boat.

Solution:

The decision variables are for t = 1,2,3,4

xt = # of boats in quarter t built in regular time yt = # of boats in quarter t built in overtime

For convenience, introduce variables:

it = # of boats in inventory at the end quarter t dt = demand in quarter t

We are given that d1 = 40, d2 = 60, d3 = 75, d4 = 25, i0 =10

xt ≤ 40, t

By logic it = it-1+ xt + yt - dt, t.

Demand is met if it ≥ 0, t

(Sign restrictions xt, yt ≥ 0, t)

We need to minimize total cost z subject to these three sets of conditions where z = 400 (x1 + x2 + x3 + x4) + 450 (y1 + y2 + y3 + y4) + 20 (i1 + i2 + i3 + i4)

Report:

Linda reveals the solution to be (x1, x2, x3, x4) = (40, 40, 40, 25) and (y1, y2, y3, y4) = (0, 10, 35, 0) and the minimum cost of $78450.00 is achieved by the schedule

		Q1	Q2	Q3	Q4
Regular time (xt)		40	40	40	25
Overtime (yt)		0	10	35	0
Inventory (it)	10	10	0	0	0
Demand (dt)		40	60	75	25

Customer Service Level Example

CSL services computers. Its demand (hours) for the time of skilled technicians in the next 5 months is

t	Jan	Feb	Mar	Apr	May
dt	6000	7000	8000	9500	11000

It starts with 50 skilled technicians at the beginning of January. Each technician can work 160 hrs/ month. To train a new technician they must be supervised for 50 hrs by an experienced technician for a period of one month time. Each experienced technician is paid \$2K/month and a trainee is paid \$1K/month. Each month 5% of the skilled technicians leave. CSL needs to meet demand and minimize costs.

Solution:

The decision variable is

x_t = # to be trained in month t

We must minimize the total cost. For convenience let y_t = # experienced tech. at start of t^{th} month d_t = demand during month t

Then we must

$$\text{Min } z = 2000 (y_1+...+ y_5) + 1000 (x_1 +...+ x_5)$$

Subject to

$160y_t - 50x_t \geq d_t$ for t = 1,..., 5

$y_1 = 50$, $d_1 = 6000$, $d_2 = 7000$, $d_3 = 8000$, $d_4 = 9500$, $d_5 = 11000$

$y_t = .95y_{t-1} + x_{t-1}$ for t = 2,3,4,5

$x_t, y_t \geq 0$

SOLVING LP

LP Solutions: Four Cases

When an LP is solved, one of the following four cases will occur:

1. The LP has a **unique optimal solution**.

2. The LP has **alternative (multiple) optimal solutions**. It has more than one (actually an infinite number of) optimal solutions

3. The LP is **infeasible**. It has no feasible solutions (The feasible region contains no points).

4. The LP is **unbounded**. In the feasible region there are points with arbitrarily large (in a max problem) objective function values.

The Graphical Solution

Any LP with only two variables can be solved graphically

Example 1. Giapetto

Since the Giapetto LP has two variables, it may be solved graphically.

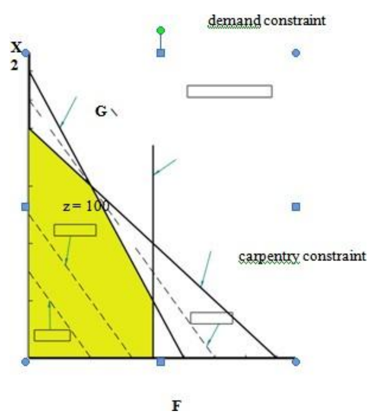
Solution:

The feasible region is the set of all points satisfying the constraints.

max z = 3x1 + 2x2				
s.t.	2x1	+ x2	≤ 100	(Finishing constraint)
	x1	+ x2	≤ 80	(Carpentry constraint)
	x1		≤ 40	(Demand constraint)
	x1, x2		≥ 0	(Sign restrictions)

The set of points satisfying the LP is bounded by the five sided polygon DGFEH. Any point **on** or **in** the interior of this polygon (the shade area) is in the **feasible region**. Having identified the feasible region for the LP, a search can begin for the **optimal solution** which will be the point in the feasible region with the largest z-value (maximization problem).

To find the optimal solution, a line on which the points have the same z-value is graphed. In a max problem, such a line is called an **isoprofit** line while in a min problem; this is called the **isocost** line. (The figure shows the isoprofit lines for z = 60, z = 100, and z = 180).



		F				
				z = 180		
z = 60						
		E		A	C	
10	20	40	50	60	80	X1

In the unique optimal solution case, isoprofit line last hits a point (vertex - corner) before leaving the feasible region.

The optimal solution of this LP is point G where (x1, x2) = (20, 60) giving z = 180.

A constraint is **binding** (active, tight) if the left-hand and right-hand side of the constraint are equal when the optimal values of the decision variables are substituted into the constraint.

A constraint is **nonbinding** (inactive) if the left-hand side and the right-hand side of the constraint are unequal when the optimal values of the decision variables are substituted into the constraint.

In Giapetto LP, the finishing and carpentry constraints are binding. On the other hand the demand constraint for wooden soldiers is nonbinding since at the optimal solution x1 < 40 (x1 = 20).

Example 2. Advertisement

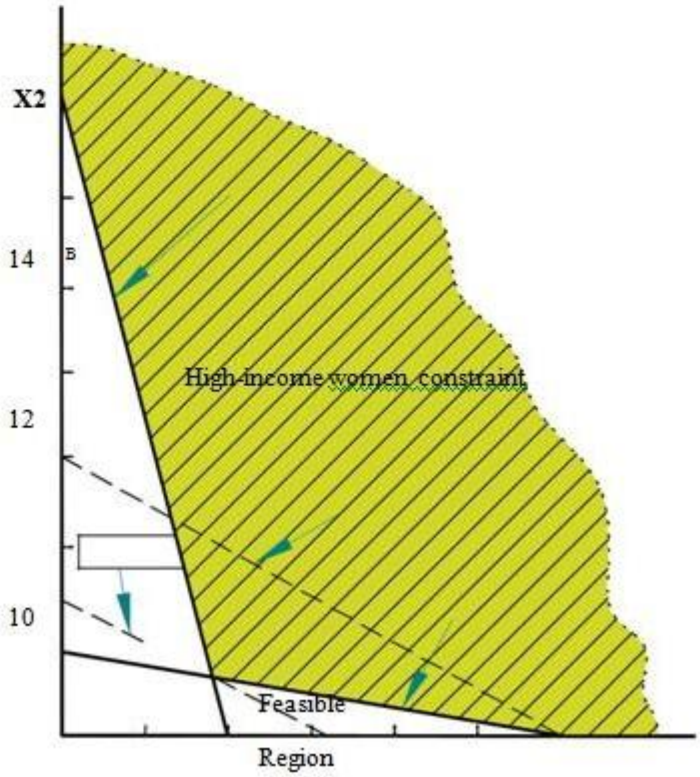
Since the Advertisement LP has two variables, it may be solved graphically.

Solution:

The feasible region is the set of all points satisfying the constraints.

min z = 50x1 + 100x2

s.t.	$7x1 + 2x2 \geq 28$	(high income women)
	$2x1 + 12x2 \geq 24$	(high income men)
	$x1, x2 \geq 0$	



6	
	z = 600
4	z = 320

	A				C		
2	4	6	8	10	12	14	**X1**

Since Dorian wants to minimize total advertising costs, the optimal solution to the problem is the point in the feasible region with the smallest z value.

An isocost line with the smallest z value passes through point E and is the optimal solution at x1 = 3.6 and x2 = giving z = 320.

Both the high-income women and high-income men constraints are satisfied, both constraints are binding

Example 3. Two Mines

min 180x + 160y

st $6x + y \geq 12$

$3x + y \geq 8$

$4x + 6y \geq 24$

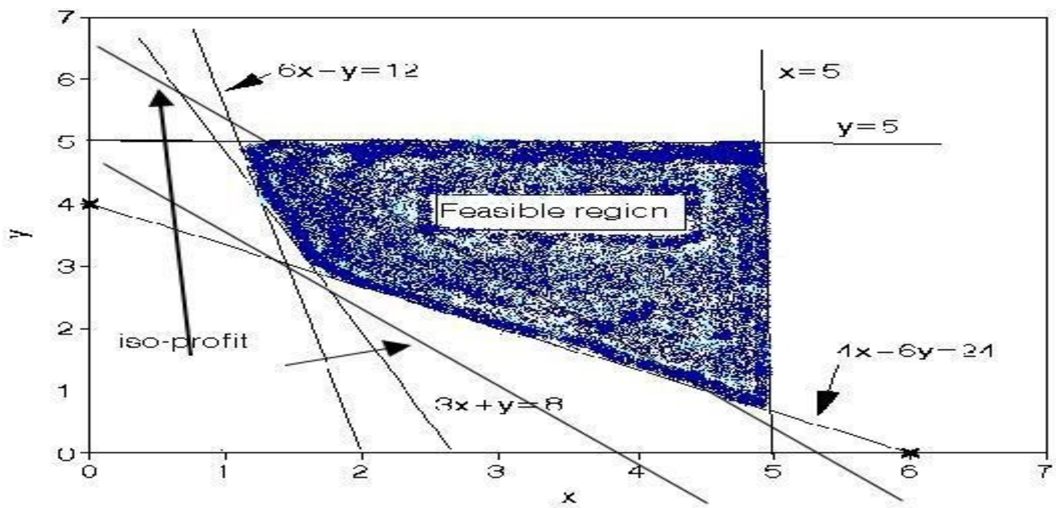
$x \leq 5$

$y \leq 5$

$x, y \geq 0$

Solution:

.



Optimal sol'n is 765.71. 1.71 days mine X and 2.86 days mine Y are operated.

Example 4. Modified Giapetto

max $z = 4x1 + 2x2$

s.t.	$2x1 + x2$	≤ 100	(Finishing constraint)
	$x1 + x2$	≤ 80	(Carpentry constraint)
	$x1$	≤ 40	(Demand constraint)
	$x1, x2 \geq 0$		(Sign restrictions)

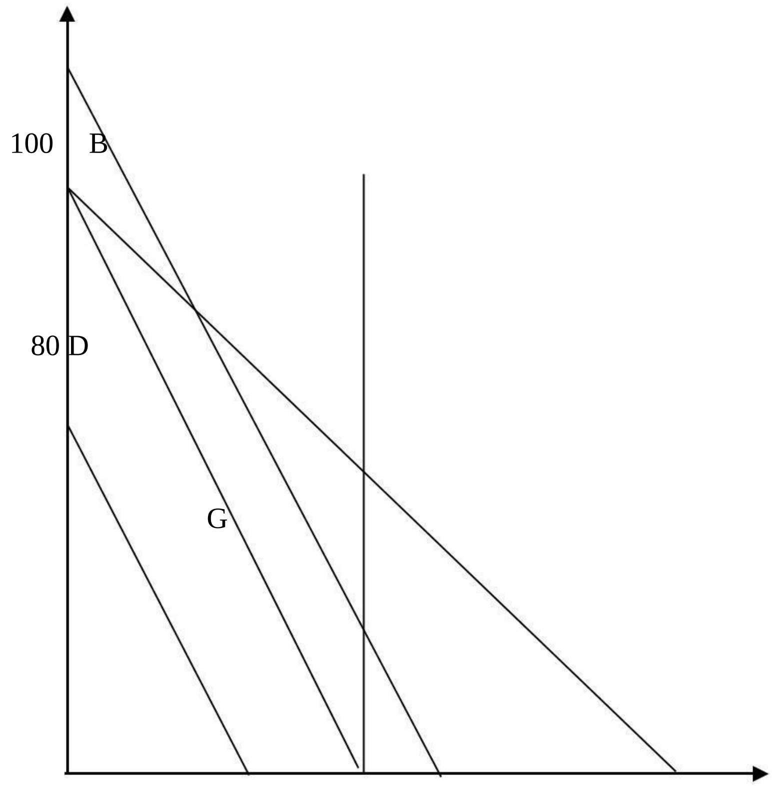
Solution:

Points on the line between points G (20, 60) and F (40, 20) are the **alternative optimal solutions** (see figure below).

Thus, for $0 \leq c \leq 1$,

[20 60] + (1 - c) [40 20] = **[40 - 20c, 20 + 40c]** will be optimal

For all optimal solutions, the optimal objective function value is 200



	F		
	E	A	C
H	40	50	x1
			80

Example 5. Modified Giapetto

Add constraint x2 ≥ 90 (Constraint on demand for trains).

Solution:

No feasible region: Infeasible LP

Example 6. Modified Giapetto

Only use constraint x2 ≥ 90

Solution:

Isoprofit line never lose contact with the feasible region: **Unbounded LP**

The Simplex Algorithm

Note that in the examples considered at the graphical solution, the unique optimal solution to the LP occurred at a vertex (corner) of the feasible region. In fact it is true that for any LP the optimal solution occurs at a vertex of the feasible region. This fact is the key to the simplex algorithm for solving LP's.

Essentially the simplex algorithm starts at one vertex of the feasible region and moves (at each iteration) to another (adjacent) vertex, improving (or leaving unchanged) the objective function as it does so, until it reaches the vertex corresponding to the optimal LP solution.

The simplex algorithm for solving linear programs (LP's) was developed by Dantzig in the late 1940's and since then a number of different versions of the algorithm have been developed. One of these later versions, called the revised simplex algorithm (sometimes known as the "product form of the inverse" simplex algorithm) forms the basis of most modern computer packages for solving LP's.

Steps

- Convert the LP to standard form

1. Obtain a basic feasible solution (bfs) from the standard form
2. Determine whether the current bfs is optimal. If it is optimal, stop.
3. If the current bfs is not optimal, determine which non basic variable should become a basic variable and which basic variable should become a non basic variable to find a new bfs with a better objective function value
4. Go back to Step 3.

Related concepts:

- Standard form: all constraints are equations and all variables are nonnegative
- bfs: any basic solution where all variables are nonnegative
- Non basic variable: a chosen set of variables where variables equal to 0
- Basic variable: the remaining variables that satisfy the system of equations at the standard form

Example 1. Dakota Furniture

Dakota Furniture makes desks, tables, and chairs. Each product needs the limited resources of lumber, carpentry and finishing; as described in the table. At most 5 tables can be sold per week. Maximize weekly revenue.

Resource	Desk	Table	Chair	Max Avail.
Lumber (board ft.)	8	6	1	48
Finishing hours	4	2	1.5	20
Carpentry hours	2	1.5	.5	8
Max Demand	unlimited	5	unlimited	
Price ($)	60	30	20	

LP Model:

Let x1, x2, x3 be the number of desks, tables and chairs produced.

Let the weekly profit be $z. Then, we must

max z = 60x1		+ 30x2 +		20x3	
s.t.	8x1	+	6x2 +	x3	≤ 48
	4x1	+	2x2 + 1.5 x3		≤ 20
	2x1	+ 1.5x2 +		.5 x3	≤ 8
			x2		≤ 5
	x1, x2, x3 ≥ 0				

Solution with Simplex Algorithm

First introduce slack variables and convert the LP to the standard form and write a canonical for

R0	z	-60x1	-30x2	-20x3				= 0
R1		8x1	+ 6x2	+ x3	+ s1			= 48
R2		4x1	+ 2x2	+1.5x3	+ s2			= 20
R3		2x1	+ 1.5x2 + .5x3			+ s3		= 8
R4			x2				+ s4	= 5

x1, x2, x3, s1, s2, s3, s4 ≥

Obtain a starting bfs.

As (x1, x2, x3) = 0 is feasible for the original problem, the below given point where three of the variables equal 0 (the **non-basic variables**) and the four other variables (the **basic variables**) are determined by the four equalities is an obvious bfs:

x1 = x2 = x3 = 0, s1 = 48, s2 = 20, s3 = 8, s4 = 5.

Determine whether the current bfs is optimal.

Determine whether there is any way that z can be increased by increasing some nonbasic variable.

If each nonbasic variable has a nonnegative coefficient in the objective function row (**row 0**), current bfs is optimal.

However, here all nonbasic variables have negative coefficients: It is not optimal.

Find a new bfs

Increases most rapidly when x1 is made non-zero; i.e. x1 is the **entering variable**.

Examining R1, x1 can be increased only to 6. More than 6 makes s1 < 0. Similarly R2, R3, and R4, give limits of 5, 4, and no limit for x1 (**ratio test**). The smallest ratio is the largest value of the entering variable that will keep all the current basic variables nonnegative. Thus by R3, x1 can only increase to x1 = 4 when s3 becomes 0. We say s3 is the **leaving variable** and R3 is the **pivot equation**.

Now we must rewrite the system so the values of the basic variables can be

	read off.							
The new pivot equation (R3/2) is								
	R3': x1+.75x2+.25x3+			.5s3	= 4			
Then use R3' to eliminate x1 in all the other rows.								
	R0'=R0+60R3',		R1'=R1-8R3', R2'=R2-4R3',				R4'=R4	
R0'	z	+ 15x2	- 5x3			+ 30s3	= 240	z = 240
R1'			- x3		+ s1	- 4s3	= 16	s1 = 16
R2'		- x2	+ .5x3		+ s2	- 2s3	= 4	s2 = 4

R3'	x1	+ .75x2	+ .25x3		+ .5s3	= 4	x1 = 4
R4'		x2			+ s4 = 5		s4 = 5

The new bfs is x2 = x3 = s3 = 0, x1 = 4, s1 = 16, s2 = 4, s4 = 5 making z = 240.

<u>Check optimality of current bfs. Repeat steps until an optimal solution is reached</u>

We increase z fastest by making x3 non-zero (i.e. x3 enters).

x3 can be increased to at most x3 = 8, when s2 = 0 (i.e. s2 leaves.)

Rearranging the pivot equation gives

R2'' - 2x2 + x3 + 2s2 - 4s3 = 8(R2' × 2).

Row operations with R2'' eliminate x3 to give the new system

R0''= R0' + 5R2'',R1'' = R1' + R2'',R3'' = R3' - .5R2'', R4'' = R4'

The bfs is now x2 = s2 = s3 = 0, x1 = 2, x3 = 8, s1 = 24, s4 = 5 making z = 280.

Each non basic variable has a nonnegative coefficient in row 0 (5x2, 10s2, 10s3).

THE CURRENT SOLUTION IS OPTIMAL

Report: Dakota furniture's optimum weekly profit would be 280$ if they produce 2 desks and 8 chairs.

This was once written as a tableau.

<u>(Use tableau format for each operation in all HW and exams!!!)</u>

	max z = 60x1 +		30x2 + 20x3							
	s.t.	8x1 +	6x2 +	x3	≤ 48					
		4x1 +	2x2 + 1.5x3 ≤ 20							
		2x1 + 1.5x2 +		.5x3 ≤ 8						
			x2	≤ 5						
		x1, x2, x3 > 0								
Initial tableau:										
z	x1	x2	x3	s1	s2	s3	s4	RHS	BV	Ratio
1	-60	-30	-20	0	0	0	0	0	z = 0	
0	8	6	1	1	0	0	0	48	s_1 = 48	6

0	4	2	1.5	0	1	0	0	20	s_2 = 20	5
0	2	1.5	0.5	0	0	1	0	8	s_3 = 8	4
0	0	1	0	0	0	0	1	5	s_4 = 5	-

<u>First tableau:</u>

z	x_1	x_2	x_3	s_1	s_2	s_3	s_4	RHS	BV	Ratio
1	0	15	-5	0	0	30	0	240	z = 240	
0	0	0	-1	1	0	-4	0	16	s_1 = 16	-
0	0	-1	0.5	0	1	-2	0	4	s_2 = 4	8
0	1	0.75	0.25	0	0	0.5	0	4	x_1 = 4	16
0	0	1	0	0	0	0	1	5	s_4 = 5	-

<u>Second and optimal tableau:</u>

z	x_1	x_2	x_3	s_1	s_2	s_3	s_4	RHS	BV	Ratio
1	0	5	0	0	10	10	0	280	z = 280	
0	0	-2	0	1	2	-8	0	24	s_1 = 24	
0	0	-2	1	0	2	-4	0	8	x_3 = 8	
0	1	1.25	0	0	-0.5	1.5	0	2	x_1 = 2	

0	0	1	0	0	0	0	1	5	s	= 5	

Example 2. Modified Dakota Furniture

Dakota example is modified: $35/table

new z = 60 x1 + 35 x2 + 20 x3

Second and optimal tableau for the modified problem:

	z	x1	x2	x3	s1	s2	s3	s4	RHS	BV	Ratio
	1	0	**0**	0	0	10	10	0	280	z=280	
	0	0	-2	0	1	2	-8	0	24	s1=24	-
	0	0	-2	1	0	2	-4	0	8	x3=8	-
	0	1	1.25	0	0	-0.5	1.5	0	2	x1=2	2/1.25
	0	0	1	0	0	0	0	1	5	s4=5	5/1

Another optimal tableau for the modified problem:

	z	x1	x2	x3	s1	s2	s3	s4	RHS	BV	
	1	0	0	0	0	10	10	0	280	z=280	
	0	1.6	0	0	1	1.2	-5.6	0	27.2 s1=27.2		
	0	1.6	0	1	0	1.2	-1.6	0	11.2 x3=11.2		
	0	0.8	1	0	0	-0.4	1.2	0	1.6	x2=1.6	
	0	-0.8	0	0	0	0.4	-1.2	1	3.4	s4=3.4	

Therefore the optimal solution is as follows:

z = 280 and for $0 \le c \le 1$

x1		2		0		2c
x2	= c	0	+ (1 – c)	1.6	=	1.6 – 1.6c

x3		8		11.2		11.2 – 3.2c

Example 3. Unbounded LPs

	z	x1	x2	x3	s1	s2	z	RHS	BV	Ratio
	1	0	2	**-9**	0	12	4	100	z=100	
	0	0	1	-6	1	6	-1	20	x4=20	None
	0	1	1	-1	0	1	0	5	x1=5	None

Since ratio test fails, the LP under consideration is an unbounded LP.

The Big M Method

If an LP has any ≥ or = constraints, a starting bfs may not be readily apparent.

When a bfs is not readily apparent, the Big M method or the two-phase simplex method may be used to solve the problem.

The Big M method is a version of the Simplex Algorithm that first finds a bfs by adding "artificial" variables to the problem. The objective function of the original LP must, of course, be modified to ensure that the artificial variables are all equal to 0 at the conclusion of the simplex algorithm.

Steps

1. Modify the constraints so that the RHS of each constraint is nonnegative (This requires that each constraint with a negative RHS be multiplied by -1. Remember that if you multiply an inequality by any negative number, the direction of the inequality is reversed!). After modification, identify each constraint as a ≤, ≥ or = constraint.

2. Convert each inequality constraint to standard form (If constraint i is a ≤ constraint, we add a slack variable si; and if constraint i is a ≥ constraint, we subtract an excess variable ei).

3. Add an artificial variable ai to the constraints identified as ≥ or = constraints at the end of Step 1. Also add the sign restriction ai ≥ 0.

4. Let M denote a very large positive number. If the LP is a min problem, add (for each artificial variable) Mai to the objective function. If the LP is a max problem, add (for each artificial variable) -Mai to the objective function.

5. Since each artificial variable will be in the starting basis, all artificial variables must be eliminated from row 0 before beginning the simplex. Now solve the transformed problem by the simplex (In choosing the entering variable, remember that M is a very large positive number!).

If all artificial variables are equal to zero in the optimal solution, we have found the **optimal solution** to the original problem.

If any artificial variables are positive in the optimal solution, the original problem is **infeasible**!!!

Example 1. Orange Juice

Bevco manufactures an orange flavored soft drink called Orange by combining orange soda and orange juice. Each ounce of orange soda contains 0.5 oz of sugar and 1 mg of vitamin C. Each ounce of orange juice contains oz of sugar and 3 mg of vitamin C. It costs Bevco 2¢ to produce an ounce of orange soda and 3¢ to produce an ounce of orange juice. Marketing department has decided that each 10 oz bottle of Orange must contain at least 20 mg of vitamin C and at most 4 oz of sugar. Use LP to determine how Bevco can meet marketing dept.'s requirements at minimum cost.

LP Model:

Let x1 and x2 be the quantity of ounces of orange soda and orange juice (respectively) in a bottle of Oranj.

min z = 2x1 + 3x2				
s.t.	0.5 x1 + 0.25 x2		≤ 4	(sugar const.)
	x1+	3 x2	≥ 20	(vit. C const.)
	x1+	x2	= 10	(10 oz in bottle)
	x1, x2 ≥ 0			

Solving Orange Example with Big M Method

1. Modify the constraints so that the RHS of each constraint is nonnegative

The RHS of each constraint is nonnegative

2. Convert each inequality constraint to standard form

z – 2x1 –		3x2		= 0
0.5x1	+ 0.25x2		+ s1	= 4
x1 +		3x2	- e2	= 20
x1	+	x2		= 10

all variables nonnegative

3. Add ai to the constraints identified as > or = const.s

z – 2x1 –		3x2		=	0	Row 0
0.5x1	+ 0.25x2		+ s1	=	4	Row 1
x1	+	3x2	- e2 + a2		= 20	Row 2
x1	+	x2		+ a3	= 10	Row 3

all variables nonnegative

4. Add Mai to the objective function (min problem)

min z = 2x1 + 3x2 + Ma2 + Ma3

Row 0 will change to

z – 2x1 – 3x2 – Ma2 – Ma3 =

5. Since each artificial variable are in our starting bfs, they must be eliminated from row 0

New Row 0 = Row 0 + M * Row 2 + M * Row 3

z + (2M–2) x1 + (4M–3) x2 – M e2 = 30M New Row 0

Initial tableau:

z	x1	x2	s1	e2	a2	a3	RHS	BV	Ratio
1	2M-2	**4M-3**	0	-M	0	0	30M	z=30M	
0	0.5	0.25	1	0	0	0	4	s1=4	16
0	1	3	0	-1	1	0	20	a2=20	**20/3***
0	1	1	0	0	0	1	10	a3=10	10

In main problem, entering variable is the variable that has the "most positive" coefficient in row 0!

First tableau:

z	x1	x2	s1	e2	a2	a3	RHS	BV	Ratio
1	**(2M-3)/3**	0	0	(M-3)/3	(3-4M)/3	0	20+3.3Mz		
0	5/12	0	1	1/12	-1/12	0	7/3	s1	28/5
0	1/3	1	0	-1/3	1/3	0	20/3	x2	20
0	2/3	0	0	1/3	-1/3	1	10/3	a3	**5***

Optimal table:

z	x1	x2	s1	e2	a2	a3	RHS	BV
1	0	0	0	-1/2	(1-2M)/2	(3-2M)/2	25	z=25
0	0	0	1	-1/8	1/8	-5/8	1/4	s1=1/4
0	0	1	0	-1/2	1/2	-1/2	5	x2=5
0	1	0	0	1/2	-1/2	3/2	5	x1=5

Report:

In a bottle of Orange, there should be 5 oz orange soda and 5 oz orange juice.

In this case the cost would be 25¢.

Example 2. Modified Orange Juice

Consider Bevco's problem. It is modified so that 36 mg of vitamin C are required.

Related LP model is given as follows:

Let x1 and x2 be the quantity of ounces of orange soda and orange juice (respectively) in a bottle of Oranj.

min z = 2x1 + 3x2			
s.t.	0.5 x1 + 0.25 x2	≤ 4	(sugar const.)

x1+	3 x2 ≥ 36		(vit. C const.)
x1+	x2 = 10		(10 oz in bottle)
x1, x2 ≥ 0			

Solving with Big M method:

Initial tableau:

z	x1	x2	s1	e2	a2	a3	RHS	BV	Ratio
1	2M-2	**4M-3**	0	-M	0	0	46M	z=46M	
0	0.5	0.25	1	0	0	0	4	s1=4	16
0	1	3	0	-1	1	0	36	a2=36	36/3
0	1	**1**	0	0	0	1	10	a3=10	**10**

Optimal tableau:

z	x1	x2	s1	e2	a2	a3	RHS	BV
1	1-2M	0	0	-M	0	3-4M	30+6M	z=30+6M
0	1/4	0	1	0	0	-1/4	3/2	s1=3/2
0	-2	0	0	-1	1	-3	6	a2=6
0	1	1	0	0	0	1	10	x2=10

An artificial variable (a2) is BV so the original LP has no feasible solution

Report:

It is impossible to produce Orange under these conditions.

UNIT-II

INTRODUCTION:

In Optimization Techniques Linear programming is one of the model in mathematical programming, which is very broad and vast. Mathematical programming includes many more optimization models known as Non - linear Programming, Stochastic programming, Integer Programming and Dynamic Programming, each one of them is an efficient optimization technique to solve the problem with a specific structure, which depends on the assumptions made in formulating the model. We can remember that the general linear programming model is based on the assumptions:

(a) Certainty

The resources available and the requirement of resources by competing candidates, the profit Coefficients of each variable are assumed to remain unchanged and they are certain in nature.

(b) Linearity

The objective function and structural constraints are assumed to be linear.

(c) Divisibility

All variables are assumed to be continuous; hence they can assume integer or fractional values.

(d) Single stage

The model is static and constrained to one decision only. And planning period is assumed to be fixed.

(e) Non-negativity

A non-negativity constraint exists in the problem, so that the values of all variables are to be □ 0, i.e. the lower limit is zero and the upper limit may be any positive number.

(f) Fixed technology

Production requirements are assumed to be fixed during the planning period.

(g) Constant profit or cost per unit

Regardless of the production schedules profit or cost remains constant.

Now let us examine the applicability of linear programming model for **transportation** and Assignment models.

The transportation model deals with a special class of linear programming problem in which the objective is to transport a homogeneous commodity from various origins or factories to different destinations or markets at a total minimum cost.

COMPARISON BETWEEN TRANSPORTATION MODEL AND GENERAL LINEAR PROGRAMMING MODEL

Similarities

- Both have objective function.
- Both have linear objective function.
- Both have non - negativity constraints.
- Both can be solved by simplex method. In transportation model it is laborious.

- A general linear programming problem can be reduced to a transportation problem if (a) the a_{ij}'s (coefficients of the structural variables in the constraints) are restricted to the values 0 and/or 1 and (b) There exists homogeneity of units among the constraints.

Differences:

- Transportation model is basically a minimization model; whereas general linear programming model may be of maximization type or minimizationtype.
- The resources, for which, the structural constraints are built up is homogeneous in transportation model; where as in general linear programming model they are different. That is one of the constraint may relate to machine hours and next one may relate to man-hours etc. In transportation problem, all the constraints are related to one particular resource or commodity, which is manufactured by the factories and demanded by the market points.
- The transportation problem is solved by transportation algorithm; where as the general linear programming problem is solved by simplex method.
- The values of structural coefficients (*i.e.* x_{ij}) are not restricted to any value in general linear programming model, where as it is restricted to values either 0 or 1 in transportation problem. Say for example:

Let one of the constraints in general linear programming model is: $2x - 3y + 10z$ □ 20. Here the coefficients of structural variables x, y and z may negative numbers or positive numbers of zeros. Where as in transportation model, say for example $x_{11} + x_{12} + x_{13} + x_{14} = b_i = 20$. Suppose the value of variables x_{11}, and x_{14} are 10 each, then $10 + 0.\ x_{12} + 0.\ x_{13} + 10 = 20$. Hence the coefficients of x_{11} and x_{14} are 1 and that of x_{12} and x_{13} are zero.

APPROACH TO SOLUTION TO A TRANSPORTATION PROBLEM BY USING TRANSPORTATION ALGORITHM

The steps used in getting a solution to a transportation problem are given below:

Initial Basic Feasible Solution

Step I. Balancing the given problem. Balancing means check whether sum of availability constraints must be equals to sum of requirement constraints. That is $ab_i = ad_j$. Once they are equal go to step two. If not by opening a Dummy row or Dummy column balance the problem.

Step II. A .Basic feasible solution can be obtained by three methods, they are

North - west corner method.

Least - cost cell method. (Or Inspection method Or Matrix minimum - row minimum - column minimum method)

Vogel's Approximation Method, generally known as VAM. After getting the basic feasible solution (*b.f.s.*) give **optimality test** to check whether the solution is optimal or not.

There are two methods of giving optimality test:

Stepping Stone Method.

Modified Distribution Method, generally known as **MODI** method.

Properties of a Basic feasible Solution

- The allocation made must satisfy the rim requirements, i.e., it must satisfy availability constraints and requirement constraints.
- should satisfy non negativity constraint.

- Total number of allocations must be equal to (m + n – 1), where 'm' is the number of rows and 'n' is the number of columns. Consider a value of m = 4 and n = 3, i.e. 4 × 3 matrix. This will have four constraints of type and three constraints of □ type. Totally it will have 4 + 3 (i.e m + n) inequalities. If we consider them as equations, for solution purpose, we will have 7 equations. In case, if we use simplex method to solve the problem, only six rather than seen structural constraints need to be specified. In view of the fact that the sum of the origin capacities (availability constraint) equals to the destination requirements (requirement constraint).

Basic Feasible Solution by North - West corner Method

Let us take a numerical example and discuss the process of getting basic feasible solution by various methods.

Example 1:

Four factories, A, B, C and D produce sugar and the capacity of each factory is given below: Factory A produces 10 tons of sugar and B produces 8 tons of sugar, C produces 5 tons of sugar and that of D is 6 tons of sugar. The sugar has demand in three markets X, Y and Z. The demand of market X is 7 tons, that of market Y is 12 tons and the demand of market Z is 4 tons. The following matrix gives the transportation cost of 1 ton of sugar from each factory to the destinations. Find the Optimal Solution for least cost transportation cost.

Factories.	Cost in Rs. per ton (× 100) Markets.			Availability in tons.
	X	Y	Z	
A	4	3	2	10
B	5	6	1	8
C	6	4	3	5
D	3	5	4	6
Requirement in tons.	7	12	4	b = 29, d = 23

Here ab is greater than ad hence we have to open a dummy column whose requirement constraint is 6, so that total of availability will be equal to the total demand. Now let get the basic feasible solution by three different methods and see the advantages and disadvantages of these methods. After this let us give optimality test for the obtained basic feasible solutions.

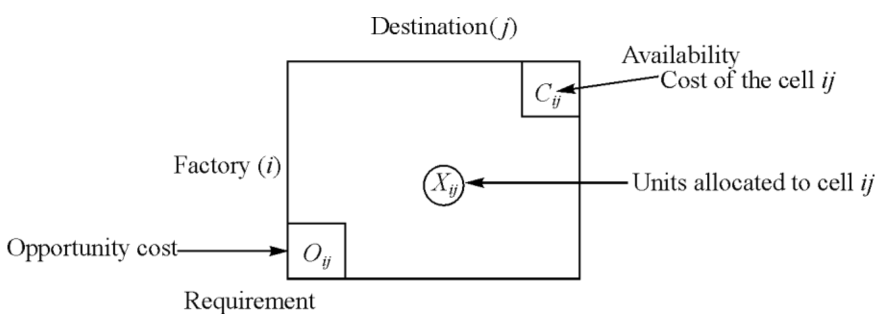
North- west corner method:

- Balance the problem. That is see whether ab_i = ad_j. If not open a dummy column or dummy row as the case may be and balance the problem.
- Start from the left hand side top corner or cell and make allocations depending on the availability and requirement constraint. If the availability constraint is less than the requirement constraint, then for that cell make allocation in units which is equal to the availability constraint. In general, verify which is the smallest among the availability and requirement and allocate the smallest one to the cell under question. Then proceed allocating either sidewise or down- ward to satisfy the rim requirement. Continue this until all the allocations are over.
- Once all the allocations are over, *i.e.*, both rim requirement (column and row *i.e.*, availability and requirement constraints) are satisfied, write allocations and calculate the cost of transportation

Solution by North-west corner method:

	X	Y	Z	Dummy	Availability
A	4 (7)	3 (3)	2	0	10
B	5	6 (8)	1	0	8
C	6	4 (1)	3 (4)	0	5
D	3	5	4 (1)	0 (5)	6
Requirement.	7	12	5	5	29

For cell AX the availability constraint is 10 and the requirement constraint is 7. Hence 7 is smaller than 10, allocate 7 to cell AX. Next 10 – 7 = 3, this is allocated to cell AY to satisfy availability requirement. Proceed in the same way to complete the allocations. Then count the allocations, if it is equals to m + n – 1, then the solution is basic feasible solution. The solution, we got have **7** allocations which is = 4 + 4 – 1 = 7. Hence the solution is basic feasible solution.



Now allocations are:

From	To	Units in tons	Cost in Rs.
A	*X*	7	7 × 4 = 28
A	*Y*	3	3 × 3 = 09
B	*Y*	8	8 × 6 = 48
C	*Y*	1	1 × 4 = 04
C	*Z*	4	4 × 3 = 12
D	*Z*	1	1 × 4 = 04
D	DUMMY	5	5 × 0 = 00

	Total in Rs.		105

Solution by Least cost cell (or inspection) Method: (Matrix Minimum method):

(*i*) Identify the lowest cost cell in the given matrix.

	X	Y	Z	Dummy	Availability
A	4 ×	3 ⑧	2 ②	0 ×	10(8) (0) iv
B	5 ×	6 ×	1 ③	0 ⑤	8(3) (0) ii
C	6 ①	4 ④	3 ×	0 ×	5(4) (0) viii
D	3 ⑥	5 ×	4 ×	0 ×	6 (0) v
Requirement.	7 (1) (0) vi	12 (4) (0) vii	5 (2) (0) iii	5 (0) i	29

Allocations are:

From	*To*	*Units in tons*	*Cost in Rs.*
A	Y	8	8 × 3 = 24
A	*Z*	2	2 × 2 = 04
B	*Z*	3	3 × 1 = 03
B	DUMMY	5	5 × 0 = 00
C	*X*	1	1 × 6 = 06
C	*Y*	4	4 × 4 = 16
D	*X*	6	6 × 3 = 18
		Total in Rs.	**71**

Solution by Vogel's Approximation Method: (Opportunity cost method)

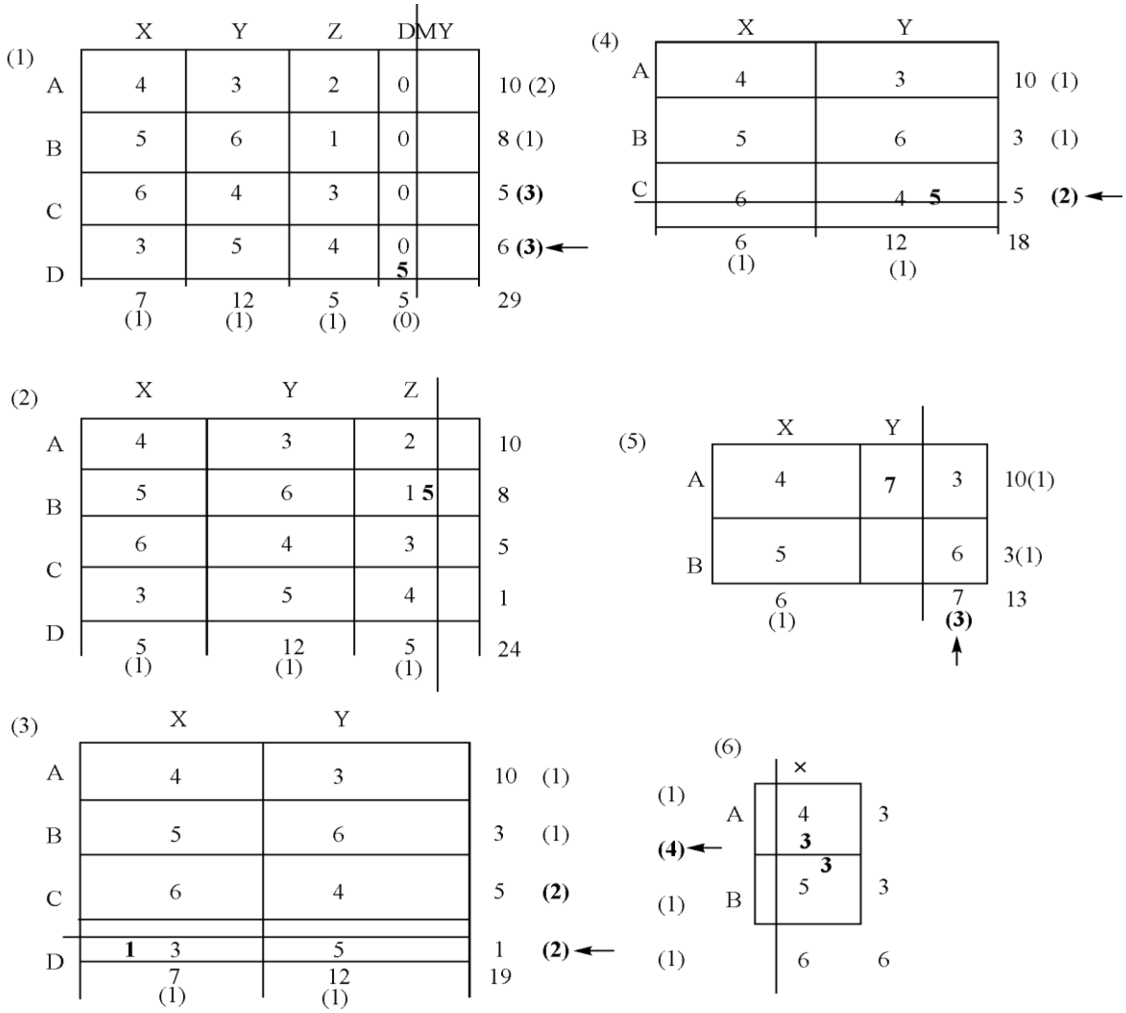
In this method, we use concept of opportunity cost. Opportunity cost is the penalty for not taking correct decision. To find the row opportunity cost in the given matrix deducts the smallest element in the row from the next highest element. Similarly to calculate the column opportunity cost, deduct smallest element in the column from the next highest element. Write row opportunity costs of each row just by the side of availability constraint and similarly write the column opportunity cost of each column just below the requirement constraints. These are known as penalty column and penalty row.

- Write row opportunity costs and column opportunity costs as described above.
- Identify the highest opportunity cost among all the opportunity costs and write a tick mark at that element.
- If there are two or more of the opportunity costs which of same magnitude, then select any one of them, to break the tie. While doing so, see that both availability constraint and requirement constraint are simultaneously satisfied. If this happens, we may not get basic feasible solution *i.e* solution with $m + n - 1$ allocations. As far as possible see that both are not satisfied simultaneously. In case if inevitable, proceed with allocations. We may not get a solution with, $m + n - 1$ allocations. For this we can allocate a small element epsilon (to) any one of the empty cells. This situation in transportation problem is known as degeneracy. (This will be discussed once again when we discuss about optimal solution).

In transportation matrix, all the cells, which have allocation, are known as **loaded cells** and those, which have no allocation, are known as **empty cells**.

(Note: All the allocations shown in matrix 1 to 6 are tabulated in the matrix given below:**)**

	X	Y	Z	Dummy	Availability
A	[4] ③	[3] ⑦	[2]	[0]	10
B	[5] ③	[6]	[1] ⑤	[0]	8
C	[6]	[4] ⑤	[3]	[0]	5
D	[3] ①	[5]	[4]	[0] ⑤	6
Requirement.	7	12	5	5	29



From	*To*	*Load*	*Cost in Rs.*
A	*X*	3	3 × 4 = 12
A	*Y*	7	7 × 3 = 21
B	*X*	3	3 × 5 = 15
B	*Z*	5	5 × 1 = 05
C	*Y*	5	5 × 4 = 20
D	*X*	1	1 × 3 = 03
D	DUMMY	5	5 × 0 = 00
		Total Rs.	**76**

Now let us compare the three methods of getting basic feasible solution:

North – west corner method.	Inspection or least cost cell method	Vogel's Approximation Method.
1. The allocation is made from the left hand side top corner irrespective of the cost of the cell.	The allocations are made depending on the cost of the cell. Lowest cost is first selected and then next highest etc.	The allocations are made depending on the opportunity cost of the cell.
2. As no consideration is given to the cost of the cell, naturally the total transportation cost will be higher than the other methods.	As the cost of the cell is considered while making allocations, the total cost of transportation will be comparatively less.	As the allocations are made depending on the opportunity cost of the cell, the basic feasible solution obtained will be very nearer to optimal solution.
3. It takes less time. This method is suitable to get basic feasible solution quickly.	The basic feasible solution, we get will be very nearer to optimal solution. It takes more time than northwest coroner method.	It takes more time for getting basic Feasible solution. But the solution we get will be very nearer to Optimal solution.
4. When basic feasible solution alone is asked, it is better to go for northwest corner method.	When optimal solution is asked, better to go for inspection method for basic feasible solution and MODI for optimal solution.	VAM and MODI is the best option to get optimal solution.

In the problem given, the total cost of transportation for Northwest corner method is Rs. 101/-. The total cost of transportation for Inspection method is Rs. 71/- and that of VAM is Rs. 76/-. The total cost got by inspection method appears to be less. That of Northwest coroner method is highest. The cost got by VAM is in between

Optimality Test: (Approach to Optimal Solution)

Once, we get the basic feasible solution for a transportation problem, the next duty is to test whether the solution got is an optimal one or not? This can be done by two methods. (*a*) By Stepping Stone Method, and (*b*) By Modified Distribution Method, or MODI method.

1. Put a small '+' mark in the empty cell.
2. Starting from that cell draw a loop moving horizontally and vertically from loaded cell to loaded cell. Remember, there should not be any diagonal movement. We have to take turn only at loaded cells and move to vertically downward or upward or horizontally to reach another loaded cell. In between, if we have a loaded cell, where we cannot take a turn, ignore that and proceed to next loaded cell in that row or column.
3. After completing the loop, mark minus (–) and plus (+) signs alternatively.
4. Identify the lowest load in the cells marked with negative sign.
5. This number is to be added to the cells where plus sign is marked and subtract from the load of the cell where negative sign is marked.
6. Do not alter the loaded cells, which are not in the loop.

7. The process of adding and subtracting at each turn or corner is necessary to see that rim requirements are satisfied.
8. Construct a table of empty cells and work out the cost change for a shift of load from loaded cell to loaded cell.
9. If the cost change is positive, it means that if we include the evaluated cell in the programme, the cost will increase. If the cost change is negative, the total cost will decrease, by including the evaluated cell in the programme.
10. The negative of cost change is the opportunity cost. Hence, in the optimal solution of transportation problem empty cells should not have positive opportunity cost.
11. Once all the empty cells have negative opportunity cost, the solution is said to be optimal.

Let us take the basic feasible solution we got by Vogel's Approximation method and give optimality test to it by stepping stone method.

Basic Feasible Solution obtained by VAM:

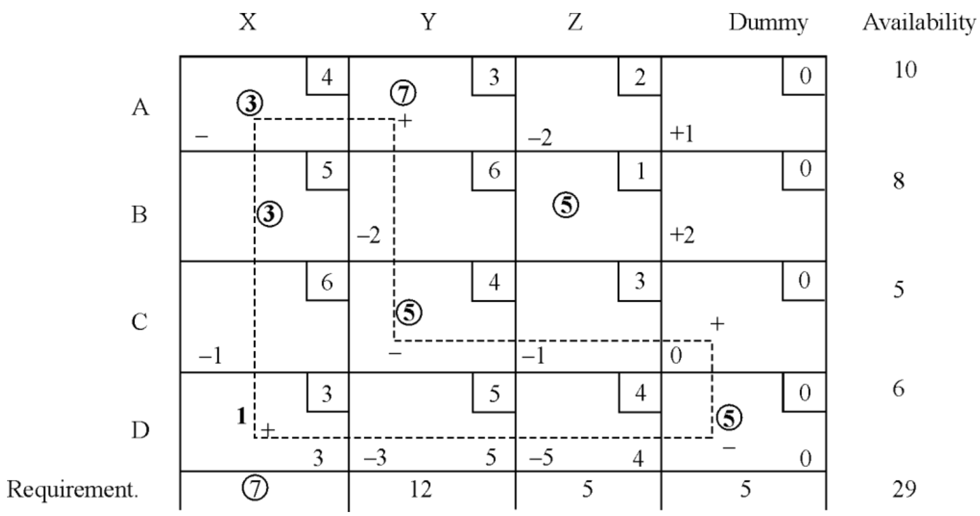
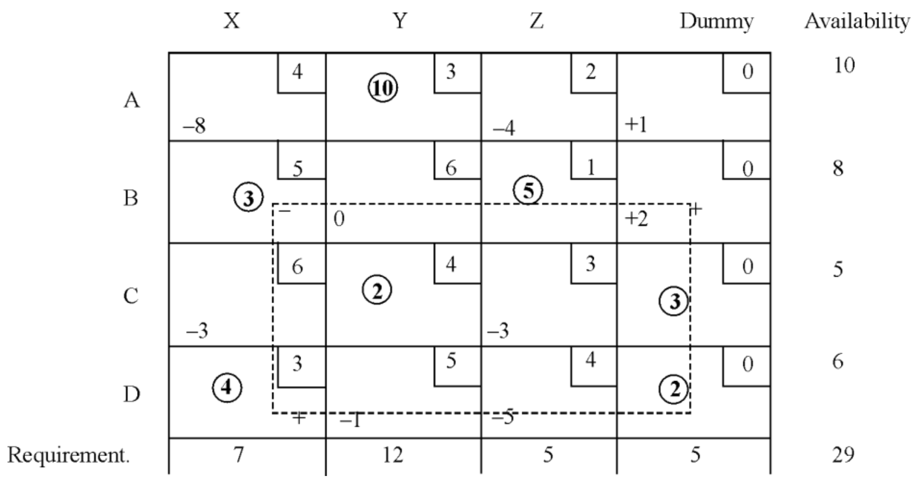


Table showing the cost change and opportunity costs of empty cells:

Table I.

S.No.	Empty Cell	Evaluation Loop formation	Cost change in Rs.	Opportunity cost -(Cost change)
1.	AZ	+AZ – AX + BX – BZ	+2 – 4 + 5 – 1 = + 2	–2
2	A Dummy	+ A DUMMY – AX + BX – B DUMMY	+0 – 4 + 3 – 0 = – 1	+1
3	BY	+ BY – AY + AX – BX	+6 – 3 + 4 – 5 = +2	-2
4	B DUMMY	+ B DUMMY – BX + DX – D DUMMY	+0 – 5 +3 – 0 = –2	+2
5	CX	+CX – CY + AX – AY	6 – 4 + 3 – 4 = +1	– 1
6	CZ	+CZ – BZ + BX –AX + AY – CY	+2 –1 +5 – 4 +5 – 4 =+1	– 1

7	C DUMMY	+ C DUMMY – D DUMMY + DX – AX + AY – CY	+ 0 – 0 +3 – 4 +3 – 4 = –2	+2
8	DY	+DY – DX + AX – AY	+5 – 3 +4 – 3 = +3	– 3
9	DZ	+DZ – DX +BX – BZ	+4 – 3 + 5 – 1 = +5	– 5



In the table 1 cells A DUMMY, B DUMMY, C DUMMY are the cells which are having positive opportunity cost. Between these two cells B DUMMY and C DUMMY are the cells, which are having higher opportunity cost i.e Rs. 2/ - each. Let us select any one of them to include in the improvement of the present programme. Let us select C DUMMY.

Table II.

S.No.	Empty Cell	Evolution Loop formation	Cost change in Rs.	Opportunity Cost
1	AX	+AX –DX + D DUMMY – C DUMMY + CY – AY	+ 4 – 3 + 0 – 0 + 4 – 3 = + 2	–2
2	AX	AZ – AY + CY – C DUMMY + D DUMMY – DX+ BX – BZ	+ 2 – 3 + 4 – 0 + 0 – 3 + 3 – 0 = + 4	–4
3	ADUMMY	+ A DUMMY – AY + DX – D DUMMY	+ 0 – 4 + 3 – 0 = – 1	+1
4	BY	+BY – BX + DX – D DUMMY + C DUMMY – CY	+ 6 –5 + 3 –0 + 0 – 4 = 0	0
5	B DUMMY	+ B DUMMY – BX + DX – D DUMMY	+ 0 – 5 + 3 – 0 = –2	+2

6	CX	+ CX – DX + D DUMMY – C DUMMY	+ 6 –3 + 0 – 0 = +3	–3
7	CZ	+ CZ – C DUMMY + D DUMMY – DX + BX – BZ	+ 2 – 0 + 0 – 3 + 5 – 1 = + 3	–3
8	DY	DY – CY + C DUMMY – D DUMMY	+ 5 – 4 + 0 – 0 = 1	–1
9	DZ	+ DZ – DX + BX – BZ	+ 4 – 3 +5 – 1 = + 5	–5

Cells A DUMMY and B DUMMY are having positive opportunity costs. The cell B DUMMY is having higher opportunity cost. Hence let us include this cell in the next programme to improve the solution.

Table III.

S.No.	Empty Cell	Evaluation Loop formation	Cost change in Rs.	Opportunity Cost
1	AX	+AX – AY + CY – C DUMMY + B DUMMY – BX	+4 – 3 + 4 – 0 + 0 – 5 = 0	0
2	AZ	+ AZ – BZ + B DUMMY – C DUMMY + CX – AX	+2 –1 + 0 – 0 + 4 – 3 =+ 2	–2
3	A DUMMY	+ A DUMMY – C DUMMY + CY – AY	+0 – 0 + 4 – 3 = +1	–1
4	BY	+ BY – B DUMMY + C DUMMY – CY	+6 – 0 + 0 – 4 = + 2	–2
5	CX	+ CX – BX + B DUMMY – C DUMMY	+6 – 5 + 0 – 0 = +1	–1
6	CZ	+ CZ – BZ + B DUMMY – C DUMMY	+2 – 1 + 0 – 0 = +1	–1
7	DY	+DY – CY + C DUMMY – B DUMMY + BX – DX	+5 – 4 + 0 – 0 + 5 – 3 = +3	–3
8	DZ	+ DZ – BZ + BX – DX	+4 – 1 + 5 – 3 = +5	–5
9	D DUMMY	+ D DUMMY – DX + BX – B DUMMY	+ 0 – 3 + 5 – 0 = +2	–2

All the empty cells have negative opportunity cost hence the solution is optimal. The **allocations are:**

S.No	Loaded cell	Load	Cost in Rs.
1	AY	10	10 × 3 = 30
2	BX	01	01 × 5 = 05
3	BZ	05	05 × 1 = 05

4	B DUMMY	02	02 × 0 = 00
5	CY	02	02 × 4 = 08
6	C DUMMY	03	03 × 0 = 00
7	DX	06	06 × 3 = 18
	Total in Rs.		66

Total minimum transportation cost is Rs. 66/-

MAXIMIZATION CASE OF TRANSPORTATION PROBLEM:

Basically, the transportation problem is a minimization problem, as the objective function is to minimize the total cost of transportation. Hence, when we would like to maximize the objective function. There are two methods.

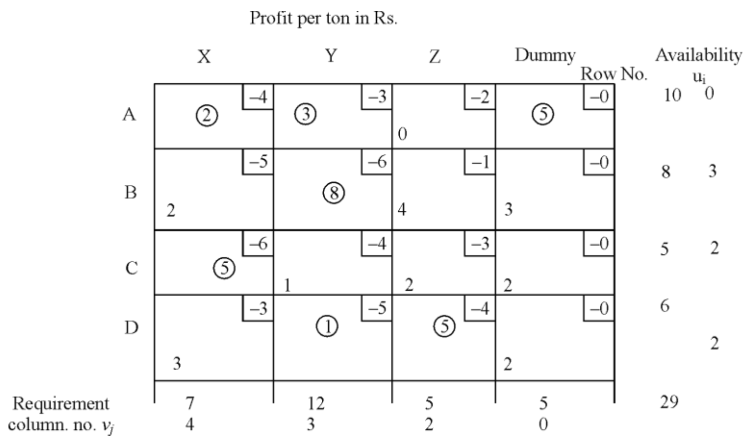
- The given matrix is to be multiplied by –1, so that the problem becomes maximization problem.
- Subtract all the elements in the matrix from the highest element in the matrix. Then the problem becomes maximization problem. Then onwards follow all the steps of maximization problem to get the solution. Let us consider the same problem solved above.

Problem 2.

Four factories, *A*, *B*, *C* and *D* produce sugar and the capacity of each factory is given below: Factory *A* produces 10 tons of sugar and *B* produces 8 tons of sugar, *C* produces 5 tons of sugar and that of *D* is 6 tons of sugar. The sugar has demand in three markets *X*, *Y* and *Z*. The demand of market *X* is 7 tons, that of market *Y* is 12 tons and the demand of market *Z* is 4 tons. The following matrix gives the returns the factory can get, by selling the sugar in each market. Formulate a transportation problem and solve for maximizing the returns.

		Profit in Rs. per ton (× 100) Markets.		Availability in tons.
	X	Y	Z	
Factories.				
A	4	3	2	10
B	5	6	1	8
C	6	4	3	5
D	3	5	4	6
Requirement in tons.	7	12	4	b = 29, d = 23

The balanced matrix of the transportation problem is:

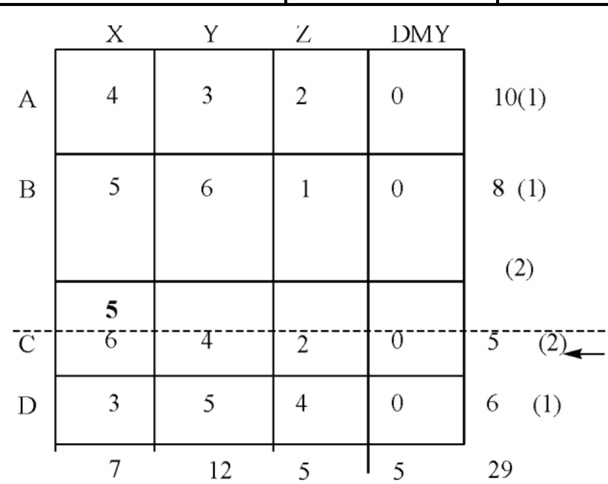


By multiplying the matrix by –1, we can convert it into a maximization problem. Now in VAM we have to find the row opportunity cost and column opportunity costs. In minimization problem, we use to subtract the smallest element in the row from next highest element in that row for finding row opportunity cost. Similarly, we use to subtract smallest element in the column by next highest element.

In the given problem as the opportunity costs of all empty cells are positive, the solution is optimal. And the optimal return to the company is Rs. 125/-.

<u>Allocations:</u>

S.No	Loaded Cell	Load	Cost in Rs.
1.	AX	02	02 × 4 = 08
2.	AY	03	03 × 3 = 09
3.	A Dmy	05	05 × 0 = 00
4.	BY	08	08 × 6 = 48
5.	CX	05	05 × 6 = 30
6.	DY	01	01 × 5 = 05
7.	DZ	05	05 × 4 = 20
	Total returns in Rs.		125



	X	Y	Z	DMY		
A	4	3	2	0	10	1
B	5	6	1	0	8	1
D	3	5	4 **5**	0	6	1
	7	12	5	5	24	
	1	1	2 ↑	0		

	X	Y	DMY		
A	4	3	0	10	1
B	5	6	0	8	1
D	3	5 **1**	0	1	2 ←
	2	12	5	19	
	1	1	0		

	X	Y	DMY		
A	4	3	0	10	1
B	5	6 **8**	0	8	1
	2	11	5	18	
	1	3 ↑	0		

	X	Y	DMY	
A	4 **2**	3 **3**	0 **5**	10
	2	3	5	10

DEGENERACY IN TRANSPORTATION PROBLEM:

Degeneracy in transportation problem can develop in two ways. First, the problem becomes degenerate when the initial programme is designed by northwest corner or inspection or VAM, *i.e.* at the stage of initial allocation only.

To solve degeneracy at this stage, we can allocate extremely small amount of goods (very close to zero) to one or more of the empty cells depending on the shortage, so that the total occupied cells becomes $m + n - 1$. The cell to which small element (load) is allocated is considered to be an occupied cell. In transportation problems, Greek letter '□' represents the small amount. One must be careful enough to see that the smallest element epsilon is added to such an empty cell, which will enable us to write row number 'u_i' and column number 'v_j' without any difficulty while giving optimality test to the basic feasible solution by MODI method. That is care must be taken to see that the epsilon is added to such a cell, which will not make a **closed loop**, when we move horizontally and vertically from loaded cell to loaded cell. (Note: Epsilon is so small so that if it is added or subtracted from any number, it does not change the numerical value of the number for which it added or from which it is subtracted.).

Secondly, the transportation problem may become degenerate during the solution stages. This happens when the inclusion of a most favorable empty cell *i.e.* cell having highest opportunity cost results in simultaneous vacating of two or more of the currently occupied cells. Here also, to solve degeneracy, add epsilon to one or more of the empty cells to make the number of occupied cells equals to $(m + n - 1)$

Problem 3. Solve the transportation problem given below

(Cost in Rs. per unit)
Destinations.

Origins	A	B	C	Available capacity	Row number u_i
X	2	1	2	20	
Y	3	4	1	40	
Requirement	20	15	25	60	
Column element v_j					

Solution by Northwest corner method:

Initial allocation show that the solution is not having $(m+n-1)$ allocations. Hence degeneracy occurs.

(Cost in Rs. per unit)
Destinations.

Origins	A	B	C	Available capacity	Row number u_i
X	[2] 20	[1]	[2]	20	
Y	[3]	[4] 15	[1] 25	40	
Requirement	20	15	25	60	

(Cost in Rs. per unit)
Destinations.

Origins	A	B	C	Available capacity	Row number u_i
X	[2] 20 −	[1] ε +	[2]	20	
Y	[3] +	[4] 15 −	[1] 25	40	
Requirement	20	15	25	60	
Column element v_j					

(Cost in Rs. per unit)
Destinations.

Origins	A	B	C	Available capacity	Row number u_i
X	[2] 20	[1] ε	[2]	20	
Y	[3]	[4] 15	[1] 25	40	
Requirement	20	15	25	60	
Column element v_j					

Shifting of load by drawing loops to cell YA.

(Cost in Rs. per unit)
Destinations.

Origins	A	B	C	Available capacity	Row number u_i
X	[2] (5)	[1] (15)	[2]	20	
Y	[3] (15)	[4]	[1] (25)	40	
Requirement	20	15	25	60	
Column element v_j					

The basic feasible solution is having four loaded cells. As the number of columns is 3 and number of rows is 2 the total number of allocations must be 2 + 3 – 1 = 4. The solution got has four allocations. Hence the basic feasible solution. Now let us give optimality test by MODI method.

(Cost in Rs. per unit)
Destinations.

Origins	A	B	C	Available capacity	Row number u_i
X	[2] (5)	[1] (15)	[2] –2	20	0
Y	[3] (15)	[4] –2	[1] (25)	40	1
Requirement	20	15	25	60	
Column element v_j	2	1	0		

Row numbers u_i s and column numbers v_j s are written in the matrix and opportunity cost of empty cells are evaluated. As the opportunity cost of all empty cells is negative, the solution is optimal. The allocations and the total cost of transportation is:

S.No	Loaded Cell	Load	Cost in Rs.
1.	XA	05	05 × 2 = 50
2.	XB	15	15 × 1 = 15
3.	YA	15	15 × 3 = 45
4.	YC	25	25 × 1 = 25
	Total cost in Rs.		135

TIME MINIMISATION MODEL OR LEAST TIME MODEL OF TRANSPORTATION TIME:

It is well known fact that the transportation problem is cost minimization model, *i.e* we have to find the least cost transportation schedule for the given problem. Some times the cost will become secondary factor when the time required for transportation is considered. This type of situation we see in military operation. When the army want to send weapons or food packets or medicine to the war front, then the time is important than the money. They have to think of what is the least time required to transport the goods than the least cost of transportation. Here the given matrix gives the time elements, *i.e*. time required to reach from one origin to a destination than the cost of transportation of one unit from one origin to a destination. A usual, we can get the basic feasible solution by Northwest corner method or by least time method or by VAM. To optimize the basic feasible solution, we have to identify the highest time element in the allocated cells, and try to eliminate it from the schedule by drawing loops and encouraging to take the cell, which is having the time element less than the highest one. Let us take a problem and work out the solution. Many a time, when we use VAM for basic feasible solution, the chance of getting an optimal solution is more. Hence, the basic feasible solution is obtained by Northwest corner method.

Problem 4: The matrix given below shows the time required to shift a load from origins to destinations. Formulate a least time schedule. Time given in hours.
Roc: Row opportunity cost, Coc: Column opportunity cost, Avail: Availability, Req: Requirement.

Destinations (Time in hours)

Origins	D_1	D_2	D_3	D_4	Avail
O_1	7	8	4	5	5
O_2	8	10	2	3	7
O_3	7	6	17	8	8
O_4	19	10	11	3	10
Req	10	5	10	5	

- Initial assignment by Northwest corner method: The Maximum time of allocated cell is 17 hours. Any cell having time element greater than 17 hours is cancelled, so that it will not in the programme.

Destinations (Time in hours)

Origins	D_1	D_2	D_3	D_4	Avail
O_1	7 (5)	8	4	5	5
O_2	8 (5)	10 (2) −	2 +	3	7
O_3	7	6 (3) +	17 (5)	8	8
O_4	19 (X)	10	11 (5)	3 (5)	10
Req	10	5	10	5	

By drawing loops, let us try to avoid 17 hours cell and include a cell, which is having time element less than 17 hours. The basic feasible solution is having $m + n - 1$ allocation.

Destinations (Time in hours)

Origins	D_1	D_2	D_3	D_4	Avail	Roc
O_1	7 (5)	8	4	5	5	
O_2	− 8 (5)	10	+ 2 (2)	3	7	
O_3	+ 7	6 (5)	− 17 (3)	8	8	
O_4	19	10	11 (5)	3 (5)	10	
Req	10	5	10	5		
Coc.						

Here also the maximum time of transport is 17 hours.

Destinations (Time in hours)

Origins	D_1	D_2	D_3	D_4	Avail	Roc
O_1	7 (5)	8	4	5	5	
O_2	− 8 (2)	10	+ 2 (5)	3	7	
O_3	+ 7 (3)	− 6 (5)	17	8	8	
O_4	19	+ 10	− 11 (5)	3 (5)	10	
Req	10	5	10	5		
Coc.						

In this allocation highest time element is 11 hours. Let us try to reduce the same.

Destinations (Time in hours)

Origins	D_1	D_2	D_3	D_4	Avail	Roc
O_1	− 7 (5)	8	+ 4	5	5	
O_2	8	10	2 (7)	3	7	
O_3	+ 7 (5)	− 6 (3)	17	8	8	
O_4	19	+ 10 (2)	− 11 (3)	3 (5)	10	
Req	10	5	10	5		
Coc.						

In this allocation also the maximum time element is 11 hours. Let us try to avoid this cell.

Destinations (Time in hours)

Origins

	D_1	D_2	D_3	D_4	Avail
O_1	(2) 7	8	(3) 4	5	5
O_2	8	10	(7) 2	3	7
O_3	(8) 7	6	~~17~~	8	8
O_4	~~19~~	(5) 10	~~11~~	(5) 3	10
Req	10	5	10	5	

No more reduction of time is possible. Hence the solution is optimal and the time required for completing the transportation is 10 Hours. T_{max} = 10 hours.

PURCHASE AND SELL PROBLEM: (TRADER PROBLEM):

Problem. 5:

M/S Epsilon traders purchase a certain type of product from three manufacturing units in different places and sell the same to five market segments. The cost of purchasing and the cost of transport from the traders place to market centers in Rs. per 100 units is given below:

Place of Manufacture.	Availability In units x 10000.	Manufacturing cost in Rs. per unit	Market Segments. (Transportation cost in Rs.per 100 units).				
			1	2	3	4	5
Bangalore (B)	10	40	40	30	20	25	35
Chennai (C)	15	50	30	50	70	25	40
Hyderabad (H)	5	30	50	30	60	55	40
	Requirement in units × 10000		6	6	8	8	4

The trader wants to decide which manufacturer should be asked to supply how many to which market segment so that the total cost of transportation and purchase is minimized.

Solution:

Here availability is 300000 units and the total requirement is 320000 units. Hence a dummy row (*D*) is to be opened. The following matrix shows the cost of transportation and purchase per unit in Rs. from manufacturer to the market centers directly.

	1	2	3	4	5	Availability
B	4040	4030	4020	4025	4035	10
C	5030	5050	5070	5025	5040	15
H	3050	3030	3060	3055	3040	5
D	0	0	0	0	0	2
Requirement.	6	6	8	8	4	32

Let us multiply the matrix by 100 to avoid decimal numbers and get the basic feasible solution by VAM. Table. Avail: Availability. Req: Requirement, Roc: Row opportunity cost, Coc: Column opportunity cost.

Tableau. I Cost of transportation and purchase Market segments.

	1	2	3	4	5	Avail.	Roc
B	4040	4030	4020	4025	4035	10	
C	5030	5050	5070	5025	5040	15	
H	3050	3030	3060	3055	3040	5	
D	0	0	0	0	0	2	
Req.	6	6	8	8	4	32	
Coc.							

Tableau. II Cost of transportation and purchase Market segments.

	1	2	3	4	5	Avail	Roc
B	4040	4030	4020	4025	4035	10	5
C	5030	5050	5070	5025	5040	15	5
H	3050	3030	3060	3055	3040	5	10
D	0	0	② 0	0	0	2	0
Req	6	6	8	8	4		
Coc	3050	3030	3060 ↑	3055	3040		

Tableau. III Cost of transportation and purchase Market segments.

	1	2	3	4	5	Avail.	Roc
B	4040	4030	4020	4025	4035	10	5
C	5030	5050	5070	5025	5040	15	5
H	3050	(5) 3030	3060	3055	3040	5	10
D	0	0	(2) 0	0	0	2	
Req.	6	6	6	8	4	30	
Coc.	990	1000	960	970	995		

Tableau. IV Cost of transportation and purchase Market segments.

	1	2	3	4	5	Avail.	Roc
B	4040	4030	(5) 4020	4025	4035	10	5
C	5030	5050	5070	5025	5040	15	5
H	3050	(5) 3030	3060	3055	3040	5	5
D	0	0	(2) 0	0	0	2	0
Req.	6	1	6	8	4	27	
Coc.	990	1020	1050	1000	1005		

Tableau. V Cost of transportation and purchase Market segments.

	1	2	3	4	5	Avail.	Roc
B	4040	(1) 4030	(6) 4020	4025	4035	4	5
C	5030	5050	5070	5025	5040	15	5
H	3050	(5) 3030	3060	3055	3040		
D	0	0	(2) 0	0	0		
Req.	6	1		8	4	19	
Coc.	990	1020		1000	1005		

Tableau. VI Cost of transportation and purchase Market segments.

	1	2	3	4	5	Avail.	Roc
B	4040	(1) 4030	(6) 4020	4025	3 4035	3	5
C	5030	5050	5070	5025	5040	15	5
H	3050	(5) 3030	3060	3055	3040		5
D	0	0	2 0	0	0		0
Req.	6			8	4	18	
Coc.	990			1000	1005 ↑		

Tableau.VII Cost of transportation and purchase Market segments.

	1	2	3	4	5	Avail.
B	4040	(1) 4030	(6) 4020	4025	(3) 4035	10
C	(6) 5030	5050	5070	(8) 5025	(1) 5040	15
H	3050	(5) 3030	3060	3055	3040	5
D	0	0	2 0	0	0	2
Req.	6	6	8	8	4	32

Final Allocation by MODI method.

Tableau. VIII Cost of transportation and purchase Market segments.

	1	2	3	4	5	Avail.
B	4040	(1) 4030	(8) 4020	4025	(1) 4035	10
C	(6) 5030	5050	5070	(8) 5025	(1) 5040	15
H	3050	(5) 3030	3060	3055	3040	5
D	0	0	0	0	(2) 0	2
Req.	6	6	8	8	4	32

Allocation:	From	To	Load	Cost in Rs.
	Bangalore	2	10,000	4,03,000

	Bangalore	3	80,000	32, 16,000
	Bangalore	5	10,000	4, 03,000
	Chennai	1	60,000	30, 18,000
	Chennai	4	80,000	40, 20,000
	Chennai	5	10,000	5, 04,000
	Hyderabad	2	50,000	15, 15,000
	Total cost in Rs.			1,30, 79,000

MAXIMISATION PROBLEM: (PRODUCTION AND TRANSPORTATION SCHEDULE FOR MAXIMIZATION):

This type of problems will arise when a company having many units manufacturing the same product and wants to satisfy the needs of various market centers. The production manager has to work out for transport of goods to various market centers to cater the needs. Depending on the production schedules and transportation costs, he can arrange for transport of goods from manufacturing units to the market centers, so that his costs will be kept at minimum. At the same time, this problem also helps him to prepare schedules to aim at maximizing his returns.

Problem 6:

A company has three manufacturing units at *X*, *Y* and *Z* which are manufacturing certain product and the company supplies warehouses at *A*, *B*, *C*, *D*, and *E*. Monthly regular capacities for regular production are 300, 400 and 600 units respectively for *X*, *Y* and *Z* units. The cost of production per unit being Rs.40, Rs.30 and Rs. 40 respectively at units *X*, *Y* and *Z*. By working overtime it is possible to have additional production of 100, 150 and 200 units, with incremental cost of Rs.5, Rs.9 and Rs.8 respectively. If the cost of transportation per unit in rupees as given in table below, find the allocation for the total minimum production cum transportation cost. Under what circumstances one factory may have to work overtime while another may work at under capacity?

To

Transportation cost in Rs.

From	*A*	*B*	*C*	*D*	*E*
X	12	14	18	13	16
Y	11	16	15	11	12
Z	16	17	19	16	14
REQ	400	400	200	200	300

- If the sales price per unit at all warehouses is Rs. 70/- what would be the allocation for maximum profit? Is it necessary to obtain a new solution or the solution obtained above holds valid?
- If the sales prices are Rs.70/-, Rs. 80/-, Rs. 72/-, Rs. 68/- and Rs. 65/- at A, B, C, D and E respectively what should be the allocation for maximum profit?

Solution: Total production including the overtime production is 1750 units and the total requirement by warehouses is 1500 units. Hence the problem is unbalanced. This can be balance by opening a Dummy Row (*DR*), with cost coefficients equal to zero and the requirement of units is 250. The cost coefficients of all other cells are got by adding production and transportation costs. The production cum transportation matrix is given below:

	A	**B**	**C**	**D**	**E**	**DC**	**Availability**
X	52	54	58	53	56	0	300
Y	41	46	45	41	42	0	400
Z	56	57	59	56	54	0	600
XOT	57	59	63	58	61	0	100
YOT	50	55	54	50	51	0	150
ZOT	64	65	67	64	62	0	200
Requirement:	400	400	200	200	300	250	1750

Initial Basic feasible solution by VAM:

	A	B	C	D	E	DC	Avail.	u_i
X	[52] (300)	[54] 0	[58] –2	[53] 0	[56] –4	[0] –7	300	52
Y	[41] –1	[46] –4	[45] –1	[41] (100)	[42] (300)	[0] –17	400	40
Z	[56] –1	[57] (400)	[59] (100)	[56] (100)	[54] –1	[0] –2	600	55
XOT	[57] (50)	[59] 0	[63] –2	[58] 0	[61] –4	[0] **50**	100	57
YOT	[50] (50)	[55] –3	[54] (100)	[50] –1	[51] –1	[0] –7	150	50
ZOT	[64] –7	[65] –6	[67] –6	[64] –6	[62] –5	[0] (200)	200	57
REQ. v_i	400 0	400 2	200 4	200 1	300 0	250 –57	1750	

As we have *m* +*n* – 1 (= 11) allocations, the solution is feasible and all the opportunity costs of empty cells are negative, the solution is optimal.

<u>Allocations:</u>

Cell	Load	Cost in Rs.
XA	300	300 × 52 = 15,600
YD	100	100 × 41 = 4,100
YE	300	300 × 40 = 12,000
ZB	400	400 × 54 = 21,000
ZC	100	100 × 59 = 5,900
ZD	100	100 × 56 = 5,600
XOT A	50	50 × 57 = 2,850
XOT DR	50	50 × 0 = 0

YOT A	50	50 × 50 = 5,500
YOT C	100	100 × 54 = 5,400
ZOT DR	50	50 × 0 =0
Total Cost in Rs.		75, 550

Allocation by VAM:

1.

	A	*B*	*C*	*D*	*E*	*DC*	*AVAIL*	*ROC*
X	52	54	58	53	56	0	300	52
Y	41	46	45	41	42	0	400	41
Z	56	57	59	56	54	0	600	54
XOT	57	59	63	58	61	0	100	50
YOT	50	55	54	50	51	0	150	50
ZOT	64	65	67	64	62	0 **(200)**	2 00	62
REQ	400	400	2 00	2 00	300	2 50	1750	
COC	9	8	9	9	9	0		

As for one allocation a row and column are getting eliminated. Hence, the degeneracy occurs

2.

	A	*B*	*C*	*D*	*E*	*DC*	*AVAIL*	*ROC*
X	52	54	58	53	56	0	300	52
Y	41	46	45	41	42	0	400	41
Z	56	57	59	56	54	0	600	54
XOT	57	59	63	58	61	0 **(50)**	100	57
YOT	50	55	54	50	51	0	150	50
REQ	400	400	2 00	2 00	300	2 50	15 50	
COC	9	8	9	9	9	0		

3.

	A	*B*	*C*	*D*	*E*	*AVAIL*	*ROC*
X	52	54	58	53	56	300	1
Y	41	46	45	41	42 **(300)**	400	0
Z	56	57	59	56	54	600	2
XOT	57	59	63	58	61	50	2
YOT	50	55	54	50	51	150	0
REQ	400	400	2 00	2 00	300	1500	

COC	9	8	9	9	9		

Here also for one allocation, a row and a column are getting eliminated. Degeneracy will occur. In all we may have to allocate two's to two empty cells.

4.

	A	*B*	*C*	*D*	*AVAIL*	*ROC*
X	52	54	58	53	300	1
Y	41	46	45	41 **(100)**	100	0
Z	56	57	59	56	600	0
XOT	57	59	63	58	50	1
YOT	50	55	54	50	150	0
REQ	40 0	400	2 00	2 00	1200	
COC	9	8	9	9		

5.

	A	*B*	*C*	*D*	*Avail*	*Ro c*
X	52	54	58	53	300	1
Z	56	57	59	56	600	0
XOT	57	59	63	58	50	1
YOT	50	55	54 **(150)**	50	150	0
Req	400	400	200	100	110 0	
Coc	2	1	4	3		

6

	A			B	C	D	Avail	Roc
X	52	**(30 0)**		54	58	53	300	1
Z	56			57	59	56	600	0
XOT	57			59	63	58	50	1
Req	400			400	50	100	950	
Coc	4			3	1	3		
		↑						
	A	B		C		D	Avail	Roc
Z	56	57		59	**(50)**	56	550	0
XOT	57	59		63		58	50	1
Req	100	400		50		100	600	
Coc	1	2		4		2		

7

	A	B	D	Avail	Roc
Z	56	57	56 **(100)**	550	0
XOT	57	59	58	50	1
Req	100	400	100	600	
Coc	1	2	2		

8.

	A	*B*	*Avail*	*Roc*
Z	56	57 **(400)**	450	1
XOT	57	59	50	2
Req	100	400	500	
Coc	1	2		

9.

	A	*Avail*
Z	56 **(50)**	50
XOT	57 **(50)**	50
	100	

In the table showing optimal solution, we can understand that the company *X* has to work 50% of its over time capacity, and company *Y* has to work 100% of its overtime capacity and company Z will not utilize its overtime capacity.

Here the total profit or return that the trading company gets is equals to Sales revenue – total expenses, which include manufacturing cost and transportation cost. Hence,

Profit = (Total Sales Revenue) – (Manufacturing cost + transportation cost).

In the question given the sales price is same in all market segments, hence, the profit calculated is independent of sales price. Hence the programme, which minimizes the total cost will, maximizes the total profit. Hence the same solution will hold good. We need not work a separate schedule for maximization of profit.

Here sales price in market segments will differ. Hence we have to calculate the total profit by the formula given above for all the markets and work for solution to maximize the profit.

The matrix showing the total profit earned by the company:

	A	B	C	D	E	DC	Avail.	u_j
X	18 3	26 (300)	14 5	15 3	9 8	0 6	900	6
Y	29 (300)	34 14	27 (ε)	27 1	25 0	0 14	400	14
Z	14 1 ↑	23 ∈	13 (200)	12 (50)	11 (300)	0 (50)	600	0
XOT	13 3	21 (100)	9 5	10 3	4 8	0 1	100	1
YOT	20 1	25 1	18 1	18 (150)	14 3	0 6	150	6
ZOT	6 9	15 5	5 8	4 8	3 8	0 (200)	200	0
Req.	400	400	200	200	300	250	1750	
Coc.	15	23	13	12	11	0		

As all the opportunity cost of empty cells are positive (maximization problem), the solution is optimal. The allocations are:

Cell	Load	Cost in Rs.
XB	300	300 × 26 = 7, 800
YA	400	400 × 29 = 11, 600
ZC	200	200 × 13 = 2, 600
ZD	50	50 × 12 = 600
ZE	300	300 × 11 = 3, 300
Z DR	50	50 × 0 = 0
XOT B	100	100 × 21 = 2, 100
YOT D	150	150 × 18 = 2, 700
ZOT DR	200	200 × 0 = 0
Profit in Rs.		= 30, 700

	A	*B*	*C*	*D*	*E*	*DC*	*Avail*	*Coc*
X	18	26	14	15	9	0	300	8
Y	29 **400**	34	27	27	25	0	400	5
Z	14	23	13	12	11	0	600	9
XOT	13	21	9	10	4	0	100	8
YOT	20	25	18	18	14	0	150	5
ZOT	6	15	5	4	3	0	200	9
Req	400	400	200	200	300	250	1750	
Coc	11	8	9	9	9	0		

As for one allocation a row and column are getting eliminated. Hence, the degeneracy occurs.

	B	*C*	*D*	*E*	*DC*	*Avail*	*Coc*
X	26 **300**	14	15	9	0	300	11
Z	23	13	12	11	0	600	10
XOT	21	9	10	4	0	100	11
YOT	25	18	18	14	0	150	7
ZOT	15	5	4	3	0	200	10
Req	400	200	200	300	250	1350	
Coc	1	4	3	5	0		

	B	*C*	*D*	*E*	*DC*	*Avail*	*Coc*
Z	23	13	12	11	0	600	10
XOT	21	9	10	4	0	100	11
	100						
YOT	25	18	18	14	0	150	7
ZOT	15	5	4	3	0	200	10
Req	100	200	200	300	250	1050	
Coc	2	5	6	3	0		

Here also for one allocation, a row and a column are getting eliminated. Degeneracy will occur. In all we may have to allocate two s to two empty cells.

	C	D	E	DC	Avail	Coc
Z	13	12	11	0	600	1
YOT	18	18 **150**	14	0	150	0
ZOT	5	4	3	0	200	1
Req	200	200	300	250	950	
Coc	5	6	3	0		

	C	D	E	DC	Avail	Coc
Z	13 **200**	12	11	0	600	1
ZOT	5	4	3	0	200	1
Req	200	50	300	250	800	
Coc	8	8	8	0		

	D	E	DC	Avail	Coc
Z	12	11 **300**	0	400	1
ZOT	4	3	0	200	1
Req	50	300	250	600	
Coc	8	8	0		

	D	DC		
Z	12 **50**	0	100	12 ←
ZOT	4	0	200	4
	50	250	300	
	8	0		

	DC	
Z	**50** 0	50
ZOT	**200** 0	200
	250	250

<u>Problem:</u>

A company has booked the orders for its consignment for the months of April, May, June and July as given below:

April: 900 units, May: 800 units, June: 900 units and July: 600 units. The company can produce 750 units per month in regular shift, at a cost of Rs. 80/- per unit and can produce 300 units per month by overtime production at a cost of Rs. 100/- per unit. Decide how much the company has to produce in which shift for minimizing the cost of production. It is given that there is no holding cost of inventory.

<u>Solution:</u> Remember here the production of April is available to meet the orders of April and subsequent months. But the production of May cannot be available to meet the demand of April. Similarly, the production of June is not available to meet the demand of April, May, but it can meet the demand of June and subsequent months and so on. Hence very high cost of production is allocated to the cells (Infinity or any highest number greater than the costs given in the problem), which cannot meet the demands of previous months (*i.e.* back ordering is not allowed). Here total availability is 4200 units and the total demand is for 3200 units. Hence we have to open a dummy column (DC), with cost coefficients equal to zero. The balanced matrix is shown below. Let us find the initial basic feasible solution by Northwest corner method and apply optimality test by MODI method.

A: April, M: May, J: June, Jl: July, AT: April Over time, MT: May overtime, JT: June overtime, JLT: July over time. DC : Dummy column.

Month of Demand (Cost in Rs)

	A	M	J	JI	DC	Avail.	u_j
A	80 (750)	80 0	80 0	80 0	0 0	750	0
AT	100 (150)	100 (150)	100 0	100 0	0 20	300	20
M	X	80 (650)	80 (100)	80 0	0 0	750	0
MT	X	100	100 (300)	100 0	0 20	300	20
J	X	X	80 (500)	80 (250)	0 0	750	0
JT	X	X	100 0	100 – (300)	0 20 +	300	20
JL	X	X	X	80 (50) +	0 – (700)	750	0
JLT	X	X	X	100 –20	0 (300)	300	0
Req.	900	800	900	600	1000	4200	
v_j	80	80	80	80	0		

Tableau 1.

Here the cell JT DC is having highest opportunity cost. Hence let us include the cell in the revised programme. To find the opportunity costs of empty cells, the row number ui and column number vj are shown. The cells marked with (X) are avoided from the programme. We can also allocate very high cost for these cells, so that they will not enter into the programme.

Month of Demand (Cost in Rs)

	A	M	J	Jl	DC	Avail.	u_i
A	80 (750)	80 0	80 0	80 0	0 0	750	0
AT	100 (150)	100 (150)	100 0	100 0	0 0	300	20
M	X	80 (650)	80 (100)	80 0	0 0	750	0
MT	X	100	100 (300)	100 0	0 0	300	20
J	X	X	80 (500)	80 (250)	0 0	750	0
JT	X	X	100 –20	100 **–20**	0 (300)	300	20
JL	X	X	X	80 (350)	0 (400)	750	0
JLT	X	X	X	100 –20	0 (300)	300	0
Req.	900	800	900	600	1000	4200	
v_j	80	80	80	80	0		

<u>Tableau II</u>. Revised programme.

As the opportunity costs of all empty cells are either zeros or negative elements, the solution is optimal. As many empty cells are having zero as the opportunity cost, they can be included in the solution and get alternate solution.

<u>Allocations:</u>

Demand month.	Production of the month	Load	Cost in Rs.
April	April regular	750	750 × 80 = 60,000
April	April over time	150	150 × 100 = 15,000
May	April over time	150	150 × 100 = 15,000
May	May regular	650	650 × 80 = 52,000
June	May regular	100	100 × 80 = 8,000
June	May over time	300	300 × 100 = 30,000
June	June Regular	500	500 × 80 = 40,000
July	June regular	250	250 × 80 = 20,000
July	July regular	350	350 × 80 = 28,000
Dummy column	June over time	300	300 × 0
Dummy Column	July regular	300	300 × 0
Dummy column	July over time	300	300 × 0
	Total cost in Rs.:		2,68,000

TRANSSHIPMENT PROBLEM:

When a company stocks its goods in warehouse and then sends the goods from warehouse to the market, the problem is known as Transshipment problem.

REDUNDANCY IN TRANSPORTATION PROBLEMS:

Sometimes, it may very rarely happen or while writing the alternate solution it may happen or during modifying the basic feasible solution it may happen that the number of occupied cells of basic feasible solution or sometimes the optimal solution may be greater than m + n – 1. This is called redundancy in transportation problem. This type of situation is very helpful to the manager who is looking about shipping of available loads to various destinations. This is as good as having more number of independent simultaneous equations than the number of unknowns. It may fail to give unique values of unknowns as far as mathematical principles are concerned. But for a transportation manager, it enables him to plan for more than one orthogonal path for an or several cells to evaluate penalty costs, which obviously will be different for different paths.

SENSITIVITY ANALYSIS:

- **Non - basic variables**

While discussing MODI method for getting optimal solution, we have discussed significance of implied cost, which fixes the upper limit of cost of the empty cell to entertain the cell in the next programme. Now let us discuss the influence of variations in present parameters on the optimum solution i.e sensitivity of optimal solution for the variations in the costs of empty cells and loaded cells. If unit cost of transportation of a particular non-basic variable changes, at what value of the cost of present optimum will no longer remain optimum? To answer this question, in the first instance, it is obvious that as the empty cell is not in the solution, any increase in its unit transportation cost will to qualify it for entering variable. But if the unit cost of empty cells is reduced the chances of changing the optimum value may be examined. Let us take an optimum solution and examine the abovestatement.

	A	B	C	D	E	Avail.	u_i
X	[10] –3	[15] –3	[17] –1	[19] –27	[16] 50	50	0
Y	[20] –23	[12] (40)	[16] (30)	[18] –26	[20] –4	70	0
Z	[9] (40)	[14] (10)	☒	[10] 30	[18] 0	80	2
DR	[0] –9	[0] –4	[0] (30)	[0] –24	[0] (20)	50	–16
Req.	40	50	60	30	70	250	
v_j	7	12	16	–8	16		

In the solution shown above as all the opportunity costs of empty cells are negative. Consider empty cell *XA*. Its opportunity cost is Rs. -3/- This means to say that the units cost of transportation of cell *XA* decreases by Rs.3/- or more *i.e* Rs.10/- the unit cost of transportation of the empty cell *XA* minus 3 = 7, or less than 7 the optimal solution changes, *i.e.* the cell *XA* will become eligible for entering into solution. Hence this cost, which shows the limit of the unit cost of empty cell, is known as implied cost in transportation problems. We can see that the opportunity cost of empty cell *ZE* is zero. This shows that the cell ZE is as good as a loaded cell and hence we can write alternate solutions by taking the cell *ZE* into consideration. **(Note: No unit cost of transportation is given for the cell *ZC*. Hence that cell should not be included in the programme. For this purpose, we can cross the cell or allocate very high unit cost of transportation for the cell. In case zero or any negative element is given as the unit cost of transportation for a cell, the value can be taken for further treatment.)**

- **Basic variables**

If unit cost of loaded cell *i.e.* basic variable is changed, it affects the opportunity costs of several cells. Now let us take the same solution shown above for our discussion. In case the unit cost of transportation for the cell *XE* is instead of 16, and other values remaining unchanged. Now let us workout the opportunity costs of other cells.

	A	B	C	D	E	Avail.	u_i
X	[10] θ – 19	[15] θ – 19	[17] θ – 17	[19] θ – 27	[θ] (50)	50	0
Y	[20] –13	[12] (40)	[16] (30)	[18] –10	[20]	70	θ + 16
Z	[9] (40)	[14] 10	X	[10] (30)	[18] 0	80	θ + 18
DR	[0] –9	[0] –4	[0] (30)	[0] –8	[0] (20)	50	–θ
Req.	40	50	60	30	70	250	
v_j	θ.9	θ.4	θ	θ–8	θ		

Cells *XA* and *XB* is positive when XA is > than 19. Cell *XC* is positive when XA is > 17 and cell *XD* is positive when XA is > 27. Other cells are not influenced by □ .

If unit cost of transportation increase and becomes 17, the present optimum may change. In case the unit cost of transportation of the cell *XA* is reduced, the solution will still remain optimum, as our objective is to minimize the total transportation cost.

A point to note here is we have used Northwest corner method and Vogel's approximation method to get basic feasible solution. Also we have discussed the least cost method and there are some methods such as row minimum and column minimum methods. These methods attempt to optimize the sub- system and do not consider marginal trade-offs. Therefore, such methods have no merit to serve useful purpose.

UNIT-III

INTRODUCTION:

A sequence is the order in which the jobs are processed. Sequence problems arise when we are concerned with situations where there is a choice in which a number of tasks can be performed. A sequencing problem could involve:

- Jobs in a manufacturing plant.
- Aircraft waiting for landing and clearance.
- Maintenance scheduling in a factory.
- Programmes to be run on a computer.
- Customers in a bank & so-on.

Terms used:

Job : The jobs or items or customers or orders are the primary stimulus for sequencing. There should be a certain number of jobs say ‘n’ to be processed or sequenced.

Number of Machines : A machine is characterized by a certain processing capability or facilities through which a job must pass before it is completed in the shop. It may not be necessarily a mechanical device. Even human being assigned jobs may be taken as machines. There must be certain number of machines say ‘k’ to be used for processing the jobs.

Processing Time : Every operation requires certain time at each of machine. If the time is certain then the determination of schedule is easy. When the processing times are uncertain then the schedule is complex.

Total Elapsed Time : It is the time between starting the first job and completing the last one.

Idle time : It is the time the machine remains idle during the total elapsed time.

Technological order : Different jobs may have different technological order. It refers to the order in which various machines are required for completing the jobs.

Types of sequencing problems:

There can be many types of sequencing problems which are as follows:

- Problem with ‘n’ jobs through one machine.
- Problem with ‘n’ jobs through two machines.
- Problem with ‘n’ jobs through three machines.

- Here the objective is to find out the optimum sequence of the jobs to be processed and starting and finishing time of various jobs through all the machines.
- No passing rule: it implies that passing is not allowed i.e. the same order of jobs is maintained over each machine
- Static arrival pattern. If all the jobs arrive simultaneously.
- Dynamic arrival pattern. Where the jobs arrive continuously.

Basic assumptions:

Following are the basic assumptions underlying a sequencing problem:

- No machine can process more than one job at a time.
- The processing times on different machines are independent of the order in which they are processed.
- The time involved in moving a job from one machine to another is negligibly small.
- Each job once started on a machine is to be performed up to completion on that machine.
- All machines are of different types.
- All jobs are completely known and are ready for processing.
- A job is processed as soon as possible but only in the order specified.

Flow-Shop Sequencing:

Job shop scheduling is basically an optimization process in which ideal jobs are assigned to resource at particular times.

Let there be 'n' jobs J1, J2, J3…..Jn. Let there be 'm' machines M1, M2, M3…..Mn. If below conditions are met, then it is said to be a flow shop sequencing problem.

- The order of sequencing jobs remains same on every machine.
- The order of machine processing every job remains same.
- All the job requires processing on every machine.

The very purpose of flow shop sequencing is to assign jobs to each assigned machines in a manner that the every machines are engaged all the time without being left ideal.

Benefits of flow shop sequencing in organization:

- Improved process efficiency.
- Improved machine utilization.
- Increased production rate.
- Reduced total processing time.
- Minimum or Zero Ideal Time.
- Potential increase in profits and decrease in costs.

n jobs through two machines:

- Let there be 'n' jobs each of which is to be processed through two machines say A & B, in the order AB. That is each job will go to machine A first and then to B in other words passing off is not allowed.
- All 'n' jobs are to be processed on A without any idle time. On the other hand the machine B is subject to its remaining idle at various stages.
- Let A1 A2…………An & B1 B2……..Bn be the expected processing time of n jobs on these two machines.

Steps for n jobs through two machines:

Step 1: Select the smallest processing time occurring in list Ai or Bi, if there is a tie select either of the smallest processing time.
Step 2: If the smallest time is on machine A, then place it at first place if it is for the B machine place the corresponding job at last. Cross off that job.
Step 3: If there is a tie for minimum time on both the machines then select machine A first & machine B last and if there is tie for minimum on machine A (same machine) then select any one of these jobs first and if there is tie for minimum on machine B among and select any of these job in the last.
Step 4: Repeat step 2 & 3 to the reduced set of processing times obtained by deleting the processing time for both the machines corresponding to the jobs already assigned
Step 5: Continue the process placing the job next to the last and so on till all jobs have been placed and it is called optimum sequence.
Step 6: after finding the optimum sequence we can find the followings

- Total elapsed time = Total time between starting the first job of the optimum sequence on machine A and completing the last job on machine B.
- Idle time in machine A = Time when the last job in the optimum sequence is completed on Machine B – Time when the last job in the optimum sequence is completed on Machine A.

Problem:
In a factory, there are six jobs to process, each of which should go to machines A & B in the order AB. The processing timings in minutes are given, determine the optimal sequencing & total elapsed time.

Jobs	**1**	**2**	**3**	**4**	**5**	**6**
Machine A	**7**	**4**	**2**	**5**	**9**	**8**
Machine B	**3**	**8**	**6**	**6**	**4**	**1**

Solution:

Step 1: the least of all the times given in for job 6 in machine B. so perform job 6 in the end. It is last in the sequences. Now delete this job from the given data.

Step 2: Of the remaining timings now the minimum is for job 3 on machine A. so do the job 3 first. Now delete this job 3 also.
Step 3: Now the smallest time is 3 minutes for job first on machine B. thus perform job 1 at the second last before job 6.

Example 2:

Suppose we have five jobs, each of which has to be processed on two machines A & B in the order AB. Processing times are given in the following table:

Job	Machine A	Machine B
1	6	3
2	2	7
3	10	8
4	4	9
5	11	5

Determine an order in which these jobs should be processed so as to minimize the total processing time.

Solution:

The minimum time in the above table is 2, which corresponds to job 2 on machine A.

2				

Now we eliminate job 2 from further consideration. The reduced set of processing times are as follows:

Job	Machine A	Machine B
1	6	3
3	10	8
4	4	9
5	11	5

The minimum time is 3 for job 1 on machine B. Therefore, this job would be done in last. The allocation of jobs till this stage would be

2				1

After deletion of job 1, the reduced set of processing times are as follows:

Job	Machine A	Machine B
3	10	8
4	4	9
5	11	5

Similarly, by repeating the above steps, the optimal sequence is as follows:

2	4	3	5	1

Once the optimal sequence is obtained, the minimum elapsed time may be calculated as follows:

Job	Machine A		Machine B	
	Time in	Time out	Time in	Time out
2	0	2	2	9
4	2	6	9	18
3	6	16	18	26
5	16	27	27	32
1	27	33	33	36

- Idle time for machine A = total elapsed time - time when the last job is out of machine A 36-33=3hours.
- Idle time for machine B = 2 + (9 - 9) + (18 - 18) + (27 - 26) + (33 - 32) = 4 hours.

Example3

:Strong Book Binder has one printing machine, one binding machine, and the manuscripts of a number of different books. Processing times are given in the following table:

Book	Time In Hours	
	Printing	Binding
A	5	2
B	1	6
C	9	7
D	3	8
E	10	4

We wish to determine the order in which books should be processed on the machines, in order to minimize the total time required.

Solution:

The minimum time in the above table is 1, which corresponds to the book B on printing machine.

B				

Now book B is eliminated. The reduced set of processing times is as follows:

Book	**Time In Hours**	
	Printing	**Binding**
A	5	2
C	9	7
D	3	8
E	10	4

The minimum time is 2 for book A on binding machine. Therefore, this job should be done in last. The allocation of jobs till this stage is:

B				A

The reduced set of processing times is as follows:

Book	**Time In Hours**	
	Printing	**Binding**
C	9	7
D	3	8
E	10	4

Similarly, by repeating the above steps, the optimal sequence is as follows:

B	D	C	E	A

Once the optimal sequence is obtained, the minimum elapsed time may be calculated as follows:

Book	Printing		Binding	
	Time in	**Time out**	**Time in**	**Time out**
B	0	1	1	7
D	1	4	7	15
C	4	13	15	22
E	13	23	23	27
A	23	28	28	30

- Idle time for printing process = total elapsed time - time when the last job is out of machine A 30-28=2hours.

Idle time for binding process = 1 + (7 - 7) + (15 - 15) + (23 - 22) + (28 - 27) = 3 hours.

<u>Processing n jobs on 3 Machines:</u>

There is no solution available for the general sequencing problems of n jobs through 3 machines. However we do have a method under the circumstance that no passing of jobs is permissible and if either or both the following conditions are satisfied.
1)The minimum time on machine A is greater than or equal to the maximum time on machine B.
2)The minimum time on machine C is greater than or equal to the maximum time on machine B

Or both are satisfied that the following method can be applied

<u>Method of Procedure:</u>

Step1: First of all, the given problem is replaced with an equivalent problem involving n jobs and 2 fictitious machines G and H .define the corresponding processing times Gi and Hi by

Gi=AI+BI
Hi=Bi+Ci

Step2: to the problem obtained step1 above ,the method for processing n jobs through 2 machines is applied .The optimal sequence resulting this shall also be optimal for the given problem.

Example 1:
There are five jobs which must go through these machines A,B and C n th order ABC .Processing times of the jobs on different machines given below.

Jobs	A	B	C
1	7	5	6
2	8	5	8
3	6	4	7
4	5	2	4
5	6	1	3

Determine a sequence for 5 jobs which will minimize elapsed time(T) .

Solution: according to given information

Min .Ai=5
Max .Bi=5
Min .Ci=3

Here since Min .Ai=Max .Ci , the first of the conditions is satisfied.

We shall now determines Gi and Hi and from them find the optimal sequence.

In accordance with the rules for determining optimal sequence in respect of n jobs processing on 2 machines , the sequence for above shall be:

3 2 1 4 5

Table : Calculation of Total Elapsed Time(T) .

Jobs	Machine A In	Machine A out	Machine B In	Machine B out	Machine C In	Machine C out
3	0	6	6	10	10	17
2	6	14	14	19	19	27
1	14	21	21	26	27	33
4	21	26	21	28	33	37
5	26	32	32	33	37	40

Total elapsed time (T) =40 hours.

<u>Example 2:</u>

The MDH Masala Company has to process five items on three machines: - A, B & C. Processing times are given in the following table:

Item	A_i	B_i	C_i
1	4	4	6
2	9	5	9
3	8	3	11
4	6	2	8
5	3	6	7

Find the sequence that minimizes the total elapsed time.

<u>Solution:</u>

Here, Min. $(A_i) = 3$, Max. $(B_i) = 6$ and Min. $(C_i) = 6$. Since the condition of Max. $(B_i) \leq$ Min. (C_i) is satisfied, the problem can be solved by the above procedure. The processing times for the new problem are given below.

Item	$G_i = A_i + B_i$	$H_i = B_i + C_i$
1	8	10
2	14	14
3	11	14
4	8	10
5	9	13

The optimal sequence is

1	4	5	3	2

Item	Machine A		Machine B		Machine C	
	Time in	Time out	Time in	Time out	Time in	Time out
1	0	4	4	8	8	14
4	4	10	10	12	14	22
5	10	13	13	19	22	29
3	13	21	21	24	29	40
2	21	30	30	35	40	49

Total elapsed time = 49.

Idle time for machine A = 49 – 30 = 19 hours.
Idle time for machine B = 4 + (10 - 8) + (13 - 12) + (21 - 19)+ (30 - 24) + (49 - 35) = 29 hours.
Idle time for machine C = 8 + (14 - 14) + (22 - 22) + (29 - 29)+ (40 - 40) = 8 hours.

Example 3:

Shahi Export House has to process five items through three stages of production, viz, cutting, sewing & pressing. Processing times are given in the following table:

Item	Cutting (A_i)	Sewing (B_i)	Pressing (C_i)

1	3	3	5
2	8	4	8
3	7	2	10
4	5	1	7
5	2	5	6

Determing an order in which these items should be processed so as to minimize the total processing time.

Solution:

The processing times for the new problem are given below.

Item	$G_i = A_i + B_i$	$H_i = B_i + C_i$
1	6	8
2	12	12
3	9	12
4	6	8
5	7	11

Thus, the optimal sequence may be formed in any of the two ways.

1	4	5	3	2

4	1	5	3	2

Item	Cutting		Sewing		Pressing	
	Time in	Time out	Time in	Time out	Time in	Time out
1	0	3	3	6	6	11
4	3	8	8	9	11	18
5	8	10	10	15	18	24

3	10	17	17	19	24	34
2	17	25	25	29	34	42

Total elapsed time = 42
Idle time for cutting process = 42 – 25 = 17 hours.
Idle time for sewing process = 3 + (8 - 6) + (10 - 9) + (17 - 15)+ (25 - 19) + (42 - 29) = 27 hours.
Idle time for pressing process = 6 + (11 - 11) + (18 - 18) + (24 - 24)+ (34 - 34) = 6 hours.

Processing n jobs through m machines:

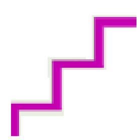
This section focuses on the sequencing problem of processing two jobs through m machines. Problems under this category can be solved with the help of graphical method. The graphical method below is explained with the help of the following example.

Two jobs are to be performed on five machines A, B, C, D, and E. Processing times are given in the following table.

			Machine				
Job 1	**Sequence**	**:**	A	B	C	D	E
	Time	**:**	3	4	2	6	2
Job 2	**Sequence**	**:**	B	C	A	D	E
	Time	**:**	5	4	3	2	6

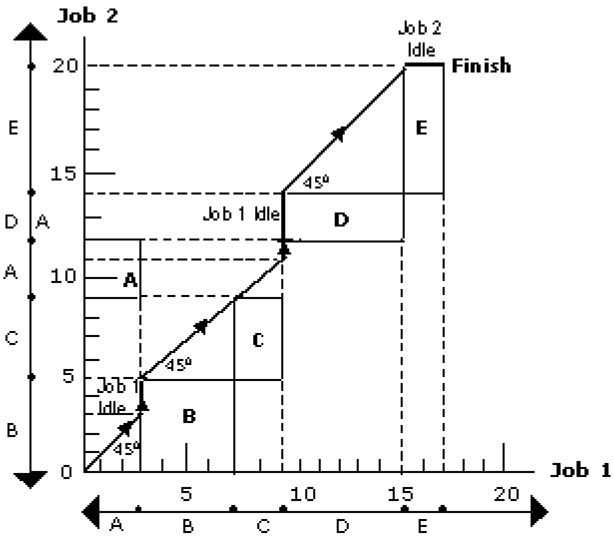
Use graphical method to obtain the total minimum elapsed time.

Solution:



Steps

- Mark the processing times of job 1 & job 2 on X-axis & Y-axis respectively.
- Draw the rectangular blocks by pairing the same machines as shown in the following figure.



- Starting from origin O, move through the 45^0 line until a point marked finish is obtained.
- The elapsed time can be calculated by adding the idle time for either job to the processing time for that job. In this illustration ,

Idle time for **job 1** is 5 hours.
Elapsed time = processing time for job 1+Idle tome of job 1
= (3+4+2+6+2)+5=22 hours.
Likewise **idle time** of **job 2** is 2 hours.
Elapsed time =processing time of job 2+Idle time of job 2
= (5+4+3+2+6)+2=22 hours.

Example 2:There are 4 job ABCD are required to be processed on four machine M1, M2,M3, M4 in that order. Determine optimal sequence and total elapsed time.

Job	M1	M2	M3	M4
A	13	8	7	14
B	12	6	8	19
C	9	7	5	15
D	8	5	6	15

Given

Job	M1	M2	M3	M4
A	13	8	7	14
B	12	6	8	19
C	9	7	5	15
D	8	5	6	15

Step 1- 1st we have to convert this problem into two machine problem. For that we have to check following condition:

Min M1 or Min M4 >= Max M2 or Max M3
here Min M1=8, Min M4=14, Max M2=7, Max M3=8.
therefore 8=8
Min M1=Max M3

Consolidation or Conversion Table:

JOB	MACHINES 5		MACHINES 6	
	P(M1+M2+M3)	**NEW TIME**	**P(M2+M3+M4)**	**NEW TIME**
A	13+8+7	28	8+7+14	29
B	12+6+8	26	6+8+19	33
C	9+7+5	21	7+5+15	27
D	8+5+6	19	5+6+15	26

New job timing According to Consolidation Table:

Job	A	B	C	D
New M/c 5	28	26	21	19
New M/c 6	29	33	27	26

Sequencing According to consolidation Table:

Consolidated table:

Job	A	B	C	D
New M/c 5	28	26	21	19
New M/c 6	29	33	27	26

Job sequence:

JOB	D	B	A	C

	Machine 1			Machine 2			Machine 3			Machine 4		
S.Q	ST	TT	ET	ST	TT	ET	ST	TT	ET	ST	TT	ET
D	0	8	8	8	5	13	13	6	19	19	15	34
B	8	12	20	20	6	26	26	8	34	34	19	53
A	20	13	33	33	8	41	41	7	48	53	14	67
C	33	9	42	42	7	49	49	5	54	67	15	82
T.T		49			26			26			63	

START TIME – ST, TIME TAKEN –TT,END TIME –ET

Total Elapsed time= 82 hrs.

Idle Time for M/c 1=Total Elapsed Time- Total time of M/c 1
=82-42= 40hrs.
Idle Time for M/c 2=Total Elapsed Time- Total time of M/c 2
=82-26= 56hrs.
Idle Time for M/c 3=Total Elapsed Time- Total time of M/c 3
= 82-26= 56hrs.
Idle Time for M/c 4=Total Elapsed Time- Total time of M/c 4
=82-63=19hrs.

Introduction to Game Theory:

Game theory is a kind of decision theory in which one's alternative action is determined after taking into consideration all possible alternatives available to an opponent playing the similar game, rather than just by the possibilities of various outcome results. Game theory does not insist on how a game must be played but tells the process and principles by which a particular action should be chosen. Therefore it is a decision theory helpful in competitive conditions.

Properties of a Game

1. There are finite number of competitors known as 'players'
2. All the strategies and their impacts are specified to the players but player does not know which strategy is to be selected.
3. Each player has a limited number of possible courses of action known as 'strategies'
4. A game is played when every player selects one of his strategies. The strategies are supposed to be prepared simultaneously with an outcome such that no player recognizes his opponent's strategy until he chooses his own strategy.
5. The figures present as the outcomes of strategies in a matrix form are known as 'pay-off matrix'.
6. The game is a blend of the strategies and in certain units which finds out the gain or loss.

7. The player playing the game always attempts to select the best course of action which results in optimal pay off known as 'optimal strategy'.
8. The expected pay off when all the players of the game go after their optimal strategies is called as 'value of the game'. The main aim of a problem of a game is to determine the value of the game.
9. The game is said to be 'fair' if the value of the game is zero or else it s known as 'unfair'.

Characteristics of Game Theory:

1. Competitive game:

A competitive situation is known as **competitive game** if it has the four properties

1. There are limited number of competitors such that $n \geq 2$. In the case of $n = 2$, it is known as **two-person game** and in case of $n > 2$, it is known as **n-person game**.
2. Each player has a record of finite number of possible actions.
3. A play is said to takes place when each player selects one of his activities. The choices are supposed to be made simultaneously i.e. no player knows the selection of the other until he has chosen on his own.
4. Every combination of activities finds out an outcome which results in a gain of payments to every player, provided each player is playing openly to get as much as possible. Negative gain means the loss of same amount.

2. Strategy

The strategy of a player is the determined rule by which player chooses his strategy from his own list during the game. The two types of strategy are

1. Pure strategy
2. Mixed strategy

Pure Strategy

If a player knows precisely what another player is going to do, a deterministic condition is achieved and objective function is to maximize the profit. Thus, the pure strategy is a decision rule always to choose a particular strategy.

Mixed Strategy

If a player is guessing as to which action is to be chosen by the other on any particular instance, a probabilistic condition is achieved and objective function is to maximize the expected profit. Hence the mixed strategy is a choice among pure strategies with fixed probabilities.

Repeated Game Strategies

- In repeated games, the chronological nature of the relationship permits for the acceptance of strategies that are dependent on the actions chosen in previous plays of the game.
- Most contingent strategies are of the kind called as "trigger" strategies.

For Example trigger strategies

In prisoners' dilemma: At start, play doesn't confess. If your opponent plays Confess, then you need to play Confess in the next round. If your opponent plays don't confess, then go for doesn't confess in the subsequent round. This is called as the "tit for tat" strategy.

In the investment game, if you are sender: At start play Send. Play Send providing the receiver plays Return. If the receiver plays keep, then never go for Send again. This is called as the "grim trigger" strategy.

3. Number of persons

When the number of persons playing is 'n' then the game is known as 'n' person game. The person here means an individual or a group aims at a particular objective.

Two-person, zero-sum game

A game with just two players (player A and player B) is known as 'two-person, zero-sum game', if the losses of one player are equal to the gains of the other one so that the sum total of their net gains or profits is zero.

Two-person, zero-sum games are also known as rectangular games as these are generally presented through a payoff matrix in a rectangular form.

3. Number of activities

The activities can be finite or infinite.

4. Payoff

Payoff is referred to as the quantitative measure of satisfaction a person obtains at the end of each play

6. Payoff matrix

Assume the player A has 'm' activities and the player B has 'n' activities. Then a payoff matrix can be made by accepting the following rules

- Row designations for every matrix are the activities or actions available to player A
- Column designations for every matrix are the activities or actions available to player B
- Cell entry V_{ij} is the payment to player A in A's payoff matrix when A selects the activity i and B selects the activity j.
- In a zero-sum, two-person game, the cell entry in the player B's payoff matrix will be negative of the related cell entry V_{ij} in the player A's payoff matrix in order that total sum of payoff matrices for player A and player B is finally zero.

7. Value of the game

Value of the game is the maximum guaranteed game to player A (maximizing player) when both the players utilizes their best strategies. It is usually signifies with 'V' and it is unique.

Classification of Games:

Simultaneous vs. Sequential Move Games

- Games where players select activities simultaneously are simultaneous move games.
  - Examples: Sealed-Bid Auctions, Prisoners' Dilemma.
  - Must forecast what your opponent will do at this point, finding that your opponent is also doing the same.

- Games where players select activities in a particular series or sequence are sequential move games.
  - Examples: Bargaining/Negotiations, Chess.
  - Must look forward so as to know what action to select now.
  - Many sequential move games have deadlines on moves.

- Many strategic situations include both sequential and simultaneous moves.

One-Shot versus Repeated Games:

- One-shot: play of the game takes place once.
  - Players likely not know much about each another.
  - Example - tipping on vacation

- Repeated: play of the game is recurring with the same players.
  - Finitely versus Indefinitely repeated games
  - Reputational concerns do matter; opportunities for cooperative behavior may emerge.

- Advise: If you plan to follow an *aggressive* strategy, ask yourself whether you are in a one-shot game or in repeated game. If a repeated game then *think again*.

Usually games are divided into:

- Pure strategy games
- Mixed strategy games

The technique for solving these two types changes. By solving a game, we require to determine best strategies for both the players and also to get the value of the game. **Saddle point method** can be used to solve pure strategy games.
The diverse methods for solving a mixed strategy game are

- Dominance rule
- Analytical method
- Graphical method
- Simplex method

Basic Game Theory Terms:

- Game : Description of the situation includes the rules of the game.
- Players : Decision makers in the game.
- Payoffs : Expected rewards enjoyed at the end of the game.
- Actions : Possible choices made by the player.
- Strategies : Specified plan of action for every contingency played by other players.

Classification of Games:
Simultaneous vs. Sequential Move Games

- Games where players select activities simultaneously are simultaneous move games.
  - Examples: Sealed-Bid Auctions, Prisoners' Dilemma.
  - Must forecast what your opponent will do at this point, finding that your opponent is also
  - doing the same.

- Games where players select activities in a particular series or sequence are sequential move games.

- Examples: Bargaining/Negotiations, Chess.
- Must look forward so as to know what action to select now.
- Many sequential move games have deadlines on moves.

- Many strategic situations include both sequential and simultaneous moves.

One-Shot versus Repeated Games:

- One-shot: play of the game takes place once.
  - Players likely not know much about each another.
  - Example - tipping on vacation

- Repeated: play of the game is recurring with the same players.
  - Finitely versus Indefinitely repeated games
  - Reputational concerns do matter; opportunities for cooperative behavior may emerge.

- Advise: If you plan to follow an *aggressive* strategy, ask yourself whether you are in a one-shot game or in repeated game. If a repeated game then *think again*.

Usually games are divided into:

- Pure strategy games
- Mixed strategy games

The technique for solving these two types changes. By solving a game, we require to determine best strategies for both the players and also to get the value of the game.**Saddle point method** can be used to solve pure strategy games.
The diverse methods for solving a mixed strategy game are

- Dominance rule
- Analytical method
- Graphical method
- Simplex method

Basic Game Theory Terms:

- Game : Description of the situation includes the rules of the game.
- Players : Decision makers in the game.
- Payoffs : Expected rewards enjoyed at the end of the game.
- Actions : Possible choices made by the player.
- Strategies : Specified plan of action for every contingency played by other players.

Pure Strategies (with saddle points):

- In a zero-sum game**,** the pure strategies of two players constitute a saddle point if the corresponding entry of the payoff matrix is simultaneously a maximum of row minima and a minimum of column maxima. This decision-making is referred to as the **minimax-maximin principle** to obtain the best possible selection of a strategy for the players.

- In a pay-off matrix, the minimum value in each row represents the minimum gain for player A. Player A will select the strategy that gives him the maximum gain among the row minimum values. The selection of strategy by player A is based on maximin principle. Similarly, the same pay-off is a loss for player B. The maximum value in each column represents the maximum loss for Player B. Player B will select the strategy that gives him the minimum loss among the column maximum values.
- The selection of strategy by player B is based on minimax principle. If the maximin value is equal to minimax value, the game has a saddle point (i.e., equilibrium point). Thus the strategy selected by player A and player B are optimal

Example 1: Consider the example to solve the game whose pay-off matrix is given in the following table as follows:

Game Problem

		Player B	
		1	2
	1	1	3
Player A	2	–1	6

The game is worked out using minimax procedure. Find the smallest value in each row and select the largest value of these values. Next, find the largest value in each column and select the smallest of these numbers. The procedure is shown in the following table.

Minimax Procedure

		Player B		
		1	**2**	**Row Min**
Player A →	**1**	1	3	(1)
	2	–1	6	–1
Col Max →		(1)	6	

- If Maximum value in row is equal to the minimum value in column, then saddle point exists.

  Max Min = Min Max

  1 = 1
- Therefore, there is a saddle point.

  The strategies are,

  Player A plays Strategy A1, (A A1).

  Player B plays Strategy B1, (B B1).

- Value of game = 1.

Example 2: Check whether the following game is given in Table, determinable and fair.

Game Problem

		Player B 1	Player B 2
Player A	1	7	0
	2	0	8

Solution: The game is solved using maximin criteria as shown in Table.

Maximin Procedure

		Player B 1	Player B 2	Row Min
Player A	1	7	0	0
	2	0	8	0
Column Max		7	8	

The game is strictly neither determinable nor fair.

Example 3: Identify the optimal strategies for player A and player B for the game, given below in Table. Also find if the game is strictly determinable and fair.

Game Problem

		Player B 1	Player B 2	Row Min
Player A	1	4	0	(0)
	2	1	–3	–3
Col Max		4	(0)	

The game is strictly determinable and fair. The saddle point exists and the game has a pure strategy. The optimal strategies are given in the following table.

Optimal Strategies

$$(a)\quad S_A \begin{pmatrix} 1 & 2 \\ p_1 & p_2 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad (b)\quad S_B \begin{pmatrix} 1 & 2 \\ q_1 & q_2 \\ 0 & 1 \end{pmatrix}$$

Analytical Method:[No saddle point exists so using analytical method]

A 2 x2 payoff matrix where there is no saddle point can be solved by analytical method.
Given the matrix

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Value of the game is

$$V = \frac{a_{11}\,a_{22} - a_{21}\,a_{12}}{(a_{11}+a_{22}) - (a_{12}+a_{21})}$$

With the coordinates

$$x_1 = \frac{a_{22} - a_{21}}{(a_{11}+a_{22}) - (a_{12}+a_{21})} \quad , \quad x_2 = \frac{a_{11} - a_{12}}{(a_{11}+a_{22}) - (a_{12}+a_{21})}$$

$$y_1 = \frac{a_{22} - a_{12}}{(a_{11}+a_{22}) - (a_{12}+a_{21})} \quad , \quad y_2 = \frac{a_{11} - a_{21}}{(a_{11}+a_{22}) - (a_{12}+a_{21})}$$

Alternative procedure to solve the strategy

- Find the difference of two numbers in column 1 and enter the resultant under column 2. Neglect the negative sign if it occurs.
- Find the difference of two numbers in column 2 and enter the resultant under column 1. Neglect the negative sign if it occurs.
- Repeat the same procedure for the two rows

Example 1:

$$\begin{array}{cc} & B \\ A & \begin{bmatrix} 5 & 1 \\ 3 & 4 \end{bmatrix} \end{array}$$

Solution:

It is a 2 x 2 matrix and no saddle point exists. We can solve by analytical method

$$
\begin{array}{cc}
 & B \\
A & \begin{bmatrix} 5 & 1 \\ 3 & 4 \end{bmatrix} \begin{matrix} 1 \\ 4 \end{matrix} \\
 & \begin{matrix} 3 & 2 \end{matrix}
\end{array}
$$

$$V = \frac{a_{11}\,a_{22} - a_{21}\,a_{12}}{(a_{11} + a_{22}) - (a_{12} + a_{21})} = \frac{20 - 3}{9 - 4}$$

$V = 17 / 5$

$S_A = (x_1, x_2) = (1/5, 4/5)$

$S_B = (y_1, y_2) = (3/5, 2/5)$

Example 2:

$$
\begin{array}{cc}
 & B \\
A & \begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix}
\end{array}
$$

Solution:

Example 1: Solve the game with the pay-off matrix for player A as given in table.

Game Problem

		Player B		
		B_1	B_2	B_3
	A_1	–4	0	4
Player A	A_2	1	4	2
	A_3	–1	5	–3

Solution: Find the smallest element in rows and largest elements in columns as shown in table.

Minimax Procedure

		Player B			
		B_1	B_2	B_3	Row min
Player A	A_1	–4	0	4	–4
	A_2	1	4	2	(1)
	A_3	–1	5	–3	–3
	Column Max	(1)	5	4	

Select the largest element in row and smallest element in column. Check for the minimax criterion,

Max Min = Min Max

1 = 1

Therefore, there is a saddle point and it is a pure strategy.

Optimum Strategy:

Player A A_2 Strategy

Player B B_1 Strategy

The value of the game is 1.

Example 2: Solve the game with the payoff matrix given in table and determine the best strategies for the companies A and B and find the value of the game for them.

Game Problem

Company B

$$\text{Company A} \begin{pmatrix} 2 & 4 & 2 \\ 1 & -5 & -4 \\ 2 & 6 & -2 \end{pmatrix}$$

Solution: The matrix is solved using maximin criteria, as shown in table below.

Maximin Procedure

		Company B			
		1	2	3	Row Min
	1	2	4	2	(2)
Company A	2	1	–5	–4	–5
	3	2	6	–2	–2
Column Max		(2)	6	(2)	

$$\text{Max Min} = \text{Min Max}$$

$$2 = 2$$

Therefore, there is a saddle point.

Optimum strategy for company A is A_1 and

Optimum strategy for company B is B_1 or B_3.

$$A \overset{B}{\begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix}} \begin{matrix} 1 \\ 3 \end{matrix}$$
$$\quad\; 1 \quad\; 3$$

$$V = \frac{a_{11} a_{22} - a_{21} a_{12}}{(a_{11} + a_{22}) - (a_{12} + a_{21})} = \frac{0 - 1}{2 + 2}$$

$V = - 1 / 4$

$S_A = (x_1, x_2) = (1/4, 3/4)$

$S_B = (y_1, y_2) = (1/4, 3/4)$

Example 1:

A and B play a game in which each has three coins, *a* 5 paisa, 10 paisa and 20 paisa coins. Each player selects a coin without the knowledge of the other's choice. If the sum of the coins is an odd amount, *A* wins *B*'s coins. If the sum is even, *B* wins *A*'s coins. Find the optimal strategies for the players and the value of the game.

Solution:

The pay of matrix for the given game is: Assume 5 paisa as the I strategy, 10 paisa as the II strategy and the 20 paisa as the III strategy.

			5 I	10 II	20 III
	5	I	–10	15	25
A	10	II	15	–20	–30
	20	III	25	–30	–40

In the problem it is given when the sum is odd, *A* wins *B*'s coins and when the sum is even, *B* will win *A*'s coins. Hence the actual pay of matrix is:

			5 I	10 II	20 III	Row minimum
	5	I	–5	10	20	–5
A	10	II	5	–10	–10	–10
	20	III	5	–20	–20	–20
Column maximum.			5	10	20	

The problem has no saddle point. Column I and II are dominating the column III. Hence it is removed from the game. The reduced matrix is:

			5 I	10 II	Row minimum
	5	I	–5	10	–5
A	10	II	5	–10	–10
	20	III	5	–20	–20
Column maximum.			5	10	

The problem has no saddle point. Considering *A*, row III is dominated by row II, hence row III is eliminated from the matrix. The reduced matrix is:

		I	II	Row minimum
A	I	–5	10	–5
	II	5	–10	–10
Colum maximum.		5	10	

No saddle point. By application of formulae:

$x_1 = (a_{22} – a_{21}) / (a_{11} + a_{22}) – (a_{12} + a_{21})$ **or = 1 –** x_2 = (–10 – 5) / [–5 + (–10)] – (10 – 5)
= –15 / (–15 – 5) = (–15 / –20) = (15 / 20) = 3 / 4, hence x_2 **= 1 – (3 / 4) = 1 / 4**
$y_1 = (a_{22} – a_{12}) / (a_{11} + a_{22}) – (a_{12} + a_{21})$ **or = 1 –** y_2 = (–10 – 10) / –20 = 20 / 20 = **1** and

$y_2 = 0$

Value of the game = $\boldsymbol{v = (a_{11}\, a_{22} - a_{12}\, a_{21}) / (a_{11} + a_{22}) - (a_{12} + a_{21})}$ = (50 – 50) / – 20 = **0 Answer is** ***A*** **(3/ 4, 1 / 4, 0), B (1, 0, 0) ,** $\boldsymbol{v = 0}$**.**

2*n game problem:

When we can reduce the given payoff matrix to 2 × 3 or 3 × 2 we can get the solution by method of **sub games**. If we can reduce the given matrix to 2 × *n* or *m* × 2 sizes, then we can get the solution by **graphical method**. A game in which one of the players has two strategies and other player has number of strategies is known as 2 × *n* or *m* × 2 games. If the game has saddle point it is solved. If no saddle point, if it can be reduced to 2 × 2 by method of dominance, it can be solved. When no more reduction by dominance is possible, we can go for Method of Sub games or Graphical method. We have to identify 2 × 2 sub games within 2 × *n* or *m* × 2 games and solve the game.

Problem 1:

Solve the game whose payoff matrix is:

B

	I	II	III
I	–4	3	–1
A II	6	–4	–2

Column maximum.

No saddle point.

The sub games are:

Sub game I:

B

		I	II	Row minimum
A	I	–4	3	–4
	II	6	–4	–4
Column Maximum.		6	3	

No saddle point. First let us find the value of the sub games by applying the formula. Then compare the values of the sub games; whichever is favorable for the candidate, that sub game is to be selected.NowhereasAhasonlytwostrategiesandBhasthreestrategies,thegame,whichisfavorable to *B*, is to be selected.

Value of the game = $\boldsymbol{v_1 = (a_{11}\, a_{22} - a_{12}\, a_{21}) / (a_{11} + a_{22}) - (a_{12} + a_{21})}$

= (–4 × –4) – (3 × 6) / [(–4 + –4] – (3 + 6) = 2 / 17

m*2 game problem:

When we can reduce the given payoff matrix to 2 × 3 or 3 × 2 we can get the solution by method of **sub games**. If we can reduce the given matrix to 2 × *n* or *m* × 2 sizes, then we can get the solution by **graphical method**. A game in which one of the players has two strategies and other player has number of strategies is known as 2 × *n* or *m* × 2 games. If the game has saddle point it is solved. If no saddle point, if it can be reduced to 2 × 2 by method of dominance, it can be solved. When no more reduction by dominance is possible, we can go for Method of Sub games or Graphical method. We have to identify 2 × 2 sub games within 2 × *n* or *m* × 2 games and solve the game.

Problem 1:

Solve the following 2 × n sub game:

B

		I	II
	I	1	8
A	II	3	5
	III	11	2

Solution:

The given game is *m* × 2 game.

B

		I	II	Row minimum
	I	1	8	1
A	II	3	5	3
	III			2
Column maximum.		11	8	

No saddle point. Hence *A*'s Sub games are:

A's sub game No.1.

		B		
		I	II	Row minimum
A	I	1	8	1
	II	3	5	3
Column Maximum.		3	8	

The game has saddle point and hence value of the game is $v_1 = 3$ A's sub game No.2.

		I	II	Row minimum
A	I	1	8	1
	III	11	2	2
Column maximum.		11	8	

No saddle point. Hence the value of the game $v_2 = (a_{11}\, a_{22} - a_{12}\, a_{21}) / (a_{11} + a_{22}) - (a_{12} + a_{21})$

$= [(1) \times (2) - (8) \times (11)] / (3) - (19) = (3 - 88) / (- 16) = (85 / 16)$

A's Sub game No. 3:

		I	II	Row minimum
	II	3	5	3
A	III	11	2	2
Column maximum.		11	5	

No saddle point. Hence the value of the game $v_3 = \mathbf{(a_{11}\, a_{22} - a_{12}\, a_{21}) / (a_{11} + a_{22}) - (a_{12} + a_{21})}$

$(3 \times 2) - (11 \times 5) / (3 + 2) - (5 + 11) = (6 - 55) / (5 - 16) = - (49 / 11)$

Now $v_1 = 3$, $v_2 = 85 / 16 = 5.31$, and $v_3 = 49 / 11 = 4.45$. Comparing the values, as far as *A* is concerned, v_2 gives him good returns. Hence *A* prefers to play the sub game No. 2. For this game we have to find out the probabilities of playing the strategies. For sub game No.2:

$\mathbf{x_1 = (a_{22} - a_{21}) / (a_{11} + a_{22}) - (a_{12} + a_{21})}$ **or** $\mathbf{= 1 - x_2}$ $= (2 - 11) / (- 16) = (9 / 16)$,

$x_2 = 1 - (9 / 16) = (7/16)$

$\mathbf{y_1 = (a_{22} - a_{12}) / (a_{11} + a_{22}) - (a_{12} + a_{21})}$ **or** $\mathbf{= 1 - y_2}$ $= (2 - 8) / (- 16) = (6 / 16)$

$y_2 = 1 - (6 / 16) = (10 / 16)$.

Therefore optimal strategies for *A* and *B* are:

A (9 / 16, 0, 7 / 16), B (6 / 16, 10 / 16) and value of the game v = (85 / 16) = 5.31

Graphical method:

The graphical method is used to solve the games whose payoff matrix has

- Two rows and n columns (2 x n)
- m rows and two columns (m x 2)

Algorithm for solving 2 x n matrix games:

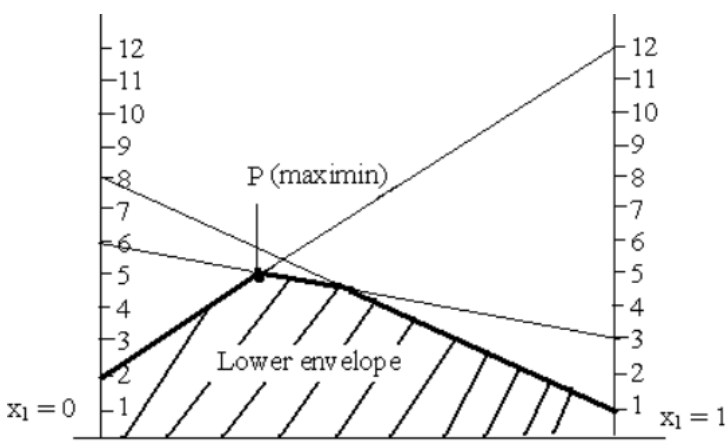
- Draw two vertical axes 1 unit apart. The two lines are x1= 0, x1= 1
- Take the points of the first row in the payoff matrix on the vertical line x1= 1 and the points of the second row in the payoff matrix on the vertical line x1= 0.
- The point a1jon axis x1= 1 is then joined to the point a2jon the axis x1= 0 to give a straight line. Draw 'n' straight lines for j=1, 2... n and determine the highest point of the lower envelope obtained. This will be the maximin point.
- The two or more lines passing through the maximin point determines the required 2 x 2 payoff matrix. This in turn gives the optimum solution by making use of analytical method.

Example 1:

Solve by graphical method

	B1	B2	B3
A1	1	3	12
A2	8	6	2

Solution:



	B2	B3	
A1	3	12	4
A2	6	2	9
	10	3	

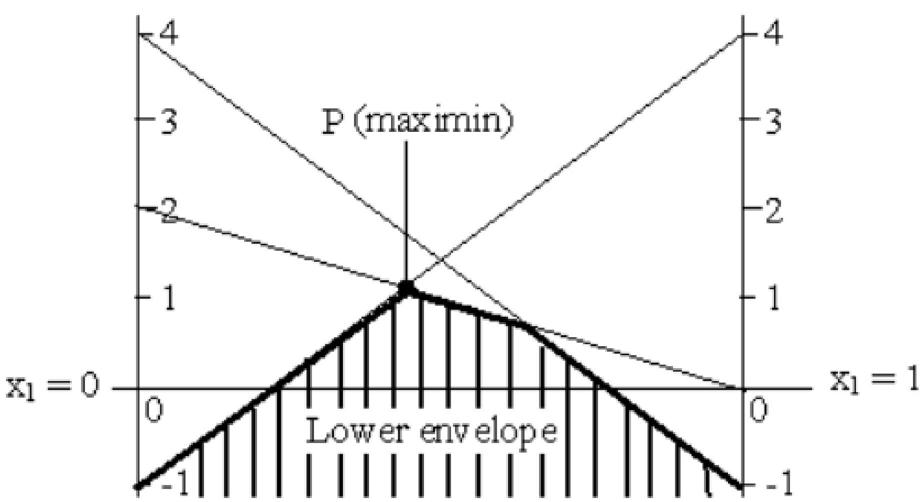
$$V = \frac{a_{11}\,a_{22} - a_{21}\,a_{12}}{(a_{11} + a_{22}) - (a_{12} + a_{21})} = \frac{6 - 72}{5 - 18}$$

V = 66/13
S_A = (4/13, 9 /13)
S_B = (0, 10/13, 3 /13)

Example 2:

$$\begin{array}{c} \\ A1 \\ A2 \end{array} \begin{array}{c} \begin{array}{ccc} B1 & B2 & B3 \end{array} \\ \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & 2 \end{bmatrix} \end{array}$$

Solution:



$$\begin{array}{c} \\ A1 \\ A2 \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} B1 & B3 \end{array} \\ \begin{bmatrix} 4 & 0 \\ -1 & 2 \end{bmatrix} \\ \begin{array}{cc} 2 & 5 \end{array} \end{array} \begin{array}{c} \\ 3 \\ 4 \\ \\ \end{array}$$

$$V = \frac{a_{11}\,a_{22} - a_{21}\,a_{12}}{(a_{11} + a_{22}) - (a_{12} + a_{21})} = \frac{8 - 0}{6 + 1}$$

V = 8/7
SA= (3/7, 4 /7)
SB= (2/7, 0, 5 /7)

Sub game II:

		B		
		I	III	Row minimum
A	I	−4	−1	−4
	II	6	−2	−2
Column maximum.		6	1	

No saddle point, hence value of the game = $v_2 = (a_{11}\, a_{22} - a_{12}\, a_{21}) / (a_{11} + a_{22}) - (a_{12} + a_{21})$

$= [(-4) \times (-2)] - [(-1) \times 6] / [(-4) + (-2) - (6 - 1) = -(14 / 11)$

The game has saddle point (1,3), the element is (–1). Hence the value of the game v_3 =-1.

Comparing the two values v_1 and v_2, v_2, and v_3, both v_2 and v_3 have negative values, which are favorable to player *B*. But v_2 is more preferred by *B* as it gives him good returns. Hence *B* prefers to play strategies I and III. Hence sub game II is selected. For this game we have to find the probabilities of strategies. For sub game II the probabilities of strategies are:

$\mathbf{x_1 = (a_{22} - a_{21}) / (a_{11} + a_{22}) - (a_{12} + a_{21})}$ **or = 1 –** $\mathbf{x_2}$ = [(–2) – 6] / (– 11) = (8 / 11), hence

x_2= 1 – (8 / 11) = 3 / 11

$\mathbf{y_1 = (a_{22} - a_{12}) / (a_{11} + a_{22}) - (a_{12} + a_{21})}$ **or = 1 –** $\mathbf{y_2}$ = [(–2) – (–1)] / –11 = (1 / 11), Hence

y_2 = 1 / (1 / 11) = (10 / 11).

Hence optimal strategies for the players are:

A (8/11, 3 / 11), B (1 / 11, 0, 10 / 11) and the value of the game is – (14 / 11).

Algorithm for solving m x 2 matrix games:

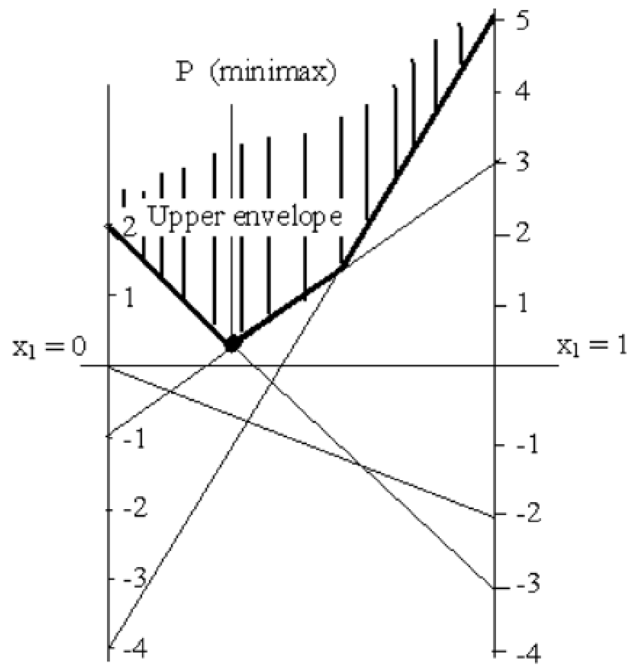
- Draw two vertical axes 1 unit apart. The two lines are x1=0, x1= 14
- Take the points of the first row in the payoff matrix on the vertical line x1= 1 and the points of the second row in the payoff matrix on the vertical line x1= 0.
- The point a1jon axis x1= 1 is then joined to the point a2jon the axis x1= 0 to give a Straight line. Draw ‘n’ straight lines for j=1, 2... n and determine the lowest point of the upper envelope obtained. This will be the minimax point.
- The two or more lines passing through the minimax point determines the required 2 x 2 payoff matrix. This in turn gives the optimum solution by making use of analytical method.

Example 1:

Solve by graphical method

$$\begin{array}{c} \\ A1 \\ A2 \\ A3 \\ A4 \end{array}\begin{array}{c} \begin{array}{cc} B1 & B2 \end{array} \\ \begin{bmatrix} -2 & 0 \\ 3 & -1 \\ -3 & 2 \\ 5 & -4 \end{bmatrix} \end{array}$$

Solution:



$$\begin{array}{c|cc|c} & B1 & B2 & \\ \hline A2 & 3 & -1 & 5 \\ A3 & -3 & 2 & 4 \\ \hline & 3 & 6 & \end{array}$$

$$V = \frac{a_{11}\,a_{22} - a_{21}\,a_{12}}{(a_{11} + a_{22}) - (a_{12} + a_{21})} = \frac{6-3}{5+4}$$

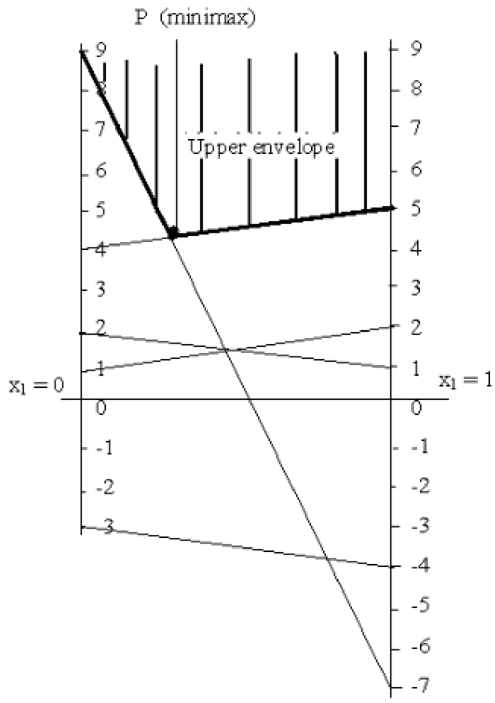
V = 3/9 = 1/3
SA= (0, 5 /9, 4/9, 0)
SB= (3/9, 6 /9)

Example 2

Solve by graphical method

	B1	B2
A1	1	2
A2	5	4
A3	-7	9
A4	-4	-3
A5	2	1

Solution:



	B1	B2	
A2	5	4	16
A3	-7	9	1
	5	12	

$$V = \frac{a_{11}\,a_{22} - a_{21}\,a_{12}}{(a_{11}+a_{22}) - (a_{12}+a_{21})} = \frac{45+28}{14+3}$$

V = 73/17

S_A = (0, 16/17, 1/17, 0, 0)

S_B = (5/17, 12 /17)

UNIT -IV

INTRODUCTION:

In previous chapters, we have seen how to solve the problems, where decision is made in single stage, one time period. But we may come across situations, where we may have to make decision in multistage, i.e. optimization of multistage decision problems. Dynamic programming is a technique for getting solutions for multistage decision problems. A problem, in which the decision has to be made at successive stages, is called a multistage decision problem. In this case, the problem solver will take decision at every stage, so that the total effectiveness defined over all the stages is optimal. Here the original problem is broken down or decomposed into small problems, which are known as sub problems or stages which is much convenient to handle and to find the optimal stage. For example, consider the problem of a sales manager, who wants to start from his head office and tour various branches of the company and reach the last branch. He has to plan his tour in such a way that he has to visit number of branches and cover less distance as far as possible. He has to divide the network of the route connecting all the branches into various stages and workout, which is the best route, which will help him to cover more branches and less distance. We can give plenty of business examples, which are multistage decision problems. The technique of Dynamic programming was developed by Richard Bellman in the early 1950.

The computational technique used is known as **Dynamic Programming** or **Recursive Optimization**. We do not have a standard mathematical formulation of the Dynamic Programming Problem (D.P.P). For each problem, depending on the variables given, and objective of the problem, one has to develop a particular equation to fit for situation. Though we have quite good number of dynamic programming problems, sometimes to take advantage of dynamic programming, we introduce multistage nature in the problem and solve it by dynamic programming technique. Nowadays, application of Dynamic Programming is done in almost all day to day managerial problems, such as, inventory problems, waiting line problems, resource allocation problems etc. Dynamic programming problem may be classified depending on the following conditions.

Dynamic programming problems may be classified depending on the nature of data available as Deterministic and Stochastic or Probabilistic models. In deterministic models, the outcome at any decision stage is unique, determined and known. In Probabilistic models, there is a set of possible outcomes with some probability distribution.

The possible decisions at any stage, from which we are to choose one, are called '**states**'. These may be finite or infinite. States are the possible situations in which the system may be at any stage.

Total number of stages in the process may be finite or infinite and may be known or unknown. Now let us try to understand certain terms, which we come across very often in this chapter.

Stage: A stage signifies a portion of the total problem for which a decision can be taken. At each stage there are a number of alternatives, and the best out of those is called **stage decision**, which may be optimal for that stage, but contributes to obtain the optimal decision policy.

State: The condition of the decision process at a stage is called its state. The variables, which specify the condition of the decision process, *i.e.* describes the **status** of the system at a particular stage are called **state variables**. The number of state variables should be as small as possible, since larger the number of the state variables, more complicated is the decision process.

Policy: A rule, which determines the decision at each stage, is known as Policy. A policy is optimal one, if the decision is made at each stage in a way that the result of the decision is optimal over all the stages and not only for the current stage.

Principle of Optimality: Bellman's Principle of optimality states that ''An optimal policy (a sequence of decisions) has the property that whatever the initial state and decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.''

This principle implies that a wrong decision taken at a stage does not prevent from taking optimal decision for the reaming stages. This principle is the firm base for dynamic programming technique. In the light of this, we can write a recurrence relation, which enables us to take the optimal decision at each stage.

Steps in getting the solution for dynamic programming problem:

- Mathematical formulation of the problem and to write the recursive equation (recursive relation connecting the optimal decision function for the 'n' stage problem with the optimal decision function for the $(n – 1)$ stage subproblem).
- To write the relation giving the optimal decision function for one stage subproblem and solve it.
- To solve the optimal decision function for 2-stage, 3-stage .... $(n – 1)$ stage and then n-stage problem.

COMPUTATIONAL PROCEDURE IN DYNAMIC PROGRAMMING:

Discrete or Continuous systems: There are two ways of solving (computational procedure) recursive equations depending on the type of the system. If the system is continuous one the procedure is different and if the system is discrete, we use a different method of computation. If the system is discrete, a tabular computational scheme is followed at each stage. The number of rows in each table is equal to the number of corresponding feasible state values and the number of columns is equal to the number of possible decisions. In case of continuous system, the optimal decision at each stage is obtained by using the usual classical technique such as differentiation etc.

- **Forward and Backward Equations:** If there are 'n' stages, and recursive equations for each stage is f_1, f_2 ……f_n and if they are solved in the order f_1 to f_n and optimal return for f_1 is r_1 and that of f_2 is r_2 and so on, then the method of calculation is known as **forward computational procedure**.
- On the other hand, if they are solved in the order from f_n, f_{n-1}, ….. f_1, then the method is termed as **backward computational procedure**. (e.g. Solution to L.P.P. by dynamic programming).

The Algorithm

- Identify the decision variables and specify objective function to be optimized under certain limitations, if any.
- Decompose or divide the given problem into a number of smaller sub-problems or stages. Identify the state variables at each stage and write down the transformation function as a function of the state variable and decision variables at the next stage.
- Write down the general recursive relationship for computing the optimal policy. Decide whether forward or backward method is to follow to solve the problem.
- Construct appropriate stage to show the required values of the return function at each stage.
- Determine the overall optimal policy or decisions and its value at each stage. There may be more than one such optimal policy.

CHARACTERISTICS OF DYNAMIC PROGRAMMING:

The basic features, which characterize the dynamic programming problem, are as follows:

- Problem can be sub-divided into stages with a policy decision required at each stage. A stage is a device to sequence the decisions. That is, it decomposes a problem into sub-problems such that an optimal solution to the problem can be obtained from the optimal solution to the sub-problem.
- Every stage consists of a number of states associated with it. The states are the different possible conditions in which the system may find itself at that stage of the problem.
- Decision at each stage converts the current stage into state associated with the next stage.
- The state of the system at a stage is described by a set of variables, called **state variables**.
- When the current state is known, an optimal policy for the remaining stages is independent of the policy of the previous ones.
- To identify the optimum policy for each state of the system, a recursive equation is formulated with 'n' stages remaining, given the optimal policy for each stage with $(n - 1)$ stages left.
- Using recursive equation approach each time the solution procedure moves backward, stage by stage for obtaining the optimum policy of each stage for that particular stage, still it attains the optimum policy beginning at the initial stage.

PROBLEMS

Problem 1. (Product allocation problem)

A company has 8 salesmen, who have to be allocated to four marketing zones. The return of profit from each zone depends upon the number of salesmen working that zone. The expected returns for different number of salesmen in different zones, as estimated from the past records, are given below. Determine the optimal allocation policy.

	SALES	*MARKETING IN*	*ZONES Rs. X 000*	
No. of . Salesmen	*Zone 1*	*Zone 2*	*Zone 3*	*Zone 4*
0	45	30	35	42
1	58	45	45	54
2	70	60	52	60
3	82	70	64	70
4	93	79	72	82
5	101	90	82	95
6	108	98	93	102
7	113	105	98	110
8	118	110	100	110

Solution:

Maximize $Z = f_1(x_1) + f_2(x_2) + f_3(x_3) + f_4(x_4)$

Subject to: $x_1 + x_2 + x_3 + x_4 \leq 8$ and x_1, x_2, x_3 and x_4 are non-negative integers.

4 4

No. of Salesmen in zone 1.	0	1	2	3	4	5	6	7	8
No. of Salesmen in zone 2.	8	7	6	5	4	3	2	1	0

Construct a table to calculate the return from the above combination.

Zone 1 → Salesmen		0	1	2	3	4	5	6	7	8
Return		45	58	70	82	93	101	108	113	118
Zone 2 ↓ Salesmen	Return									
0	30	**75**	88	100	112	123	141	138	143	148
1	45	**90**	103	115	127	138	146	153	158	
2	60	**105**	**118**	**130**	**142**	**153**	161	168		
3	70	115	128	140	152	**163**	171			
4	79	124	137	149	161	**172**				
5	90	135	148	160	**172**					
6	98	143	156	168						
7	105	150	163							
8	110	155								

Number of salesmen.	0	1	2	3	4	5	6	7	8	
Zone 1	0	0	0	1	2	3	4	4	4	3
Zone 2	0	1	2	2	2	2	2`	3	4	5
Outcome in Rs. × 1000	75	90	105	118	130	142	15 3	162	172	172

Now in the second stage, let us combine zone 3 and zone 4 and get the total market returns.
Combination of zone 3 and zone 4.

Zone 3 → Salesmen		0	1	2	3	4	5	6	7	8
Return.		35	45	52	64	72	82	93	98	100
Zone 4 Salesmen	Return.									
0	42	**77**	**97**	94	**106**	114	124	136	140	142
1	54	89	**99**	**106**	**118**	126	136	**147**	152	
2	60	95	105	112	124	132	142	153		

3	70	105	115	122	134	142	152
4	82	117	127	134	146	154	
5	95	**130**	**140**	**147**	**159**		
6	102	137	**147**	154			
7	110	145	155				
8	110	145					

Now the table below shows the allocation and the outcomes for zone 3 and zone 4.

Number of Salesmen	0	1	2	3	4	5	6	7		8
Zone 3	0	1	1	2	3	5	1	1	2	3
Zone 4	0	0	1	1	1	0	5	6	5	5
Return in Rs. × 1000	77	97	99	106	118	130	140	147	147	159

In third stage we combine both zones 1 & 2 outcomes and zones 3 and 4 outcomes.
Zones 1 and 2 and zones 3 and 4 combined.

Zones 1& 2 Salesmen		(0, 0) 0	(0, 1) 1	(0, 2) 2	(1, 2) 3	(2, 2) 4	(3, 2) 5	(4, 2) 6	(4, 3) 7	(4, 4)(3, 5) 8
Return.		75	90	105	118	130	142	153	163	172
Zones 3 & 4 Salesmen ▼	Return.									
0(0,0)	77	**152**	**187**	182	195	207	219	230	240	247
1(1,0)	97	172	**187**	**202**	**215**	**302**	**239**	**243**	**260**	
2(1,1)	99	174	189	204	217	229	241	252		
3(2,1)	106	181	196	211	224	236	248			
4 (3,1)	118	193	208	223	236	248				
5(5,0)	130	205	220	235	248					
6(1,5)	140	215	230	245						
7(1,6) (2,5)	147	222	237							
8(3,5)	159	234								

<u>Optimal allocation is:</u>

Salesmen	0	1	2	3	4	5	6	7	8
Zone 1	0	0	1	0	1	2	3	4	4
Zone 2	0	0	0	2	2	2	2	2	3
Zone 3	0	1	1	1	1	1	1	1	1

Zone 4	0	0	0	0	0	0	0	0	0
Total return in Rs. × 1000	152	187	187	202	215	302	239	243	26 0

The above table shows that how salesmen are allocated to various zones and the optimal outcome for the allocation. **Maximum outcome is Rs. 260 × 1000.**

Problem 2.

The owner of a chain of four grocery stores has purchased six crates of fresh strawberries. The estimated probability distribution of potential sales of the strawberries before spoilage differs among the four stores. The following table gives the estimated total expected profit at each store, when it is allocated various numbers of crates:

Number of Crates	1	2	3	4
0	0	0	0	0
1	4	2	6	2
2	6	4	8	3
3	7	6	8	4
4	7	8	8	4
5	7	9	8	4
6	1	10	8	4

For administrative reasons, the owner does not wish to split crates between stores. However he is willing to distribute zero crates to any of his stores.

Solution

Let the four stores be considered as four stages in dynamic programming formulation. The decision variables x_i (i = 1, 2, 3 and 4) denote the number of crates allocated to the i th stage. Let f (x_i) be the expected profit from allocation of x_i crates to the store 'i', then the problem is:

Maximize $Z = f_1 (x_1) + f_2 (x_2) + f_3 (x_3) + f_4 (x_4)$ subject to $x_1 + x_2 + x_3 + x_4 = 6$ and all $x_i = 0$

No. of crates		0	1	2	3	4	5	6
Profit.		0	6	8	8	8	8	8
No. of crates	Profit.							
0	0	0	**6**	**8**	8	8	8	8
1	2	2	3	**10**	10	10	10	
2	3	3	9	**11**	11	11		
3	4	4	10	**12**	**12**			

4	4	4	10	**12**				
5	4	4	10					
6	4	4						

No. of crates	0	1	2	3	4	5	6
Store 3	0	1	2	2	2	2	3 2
Store 4	0	0	0	1	2	3	3 4
Profit	0	6	8	10	11	12	12

No. of crates		0	1	2	3	4	5	6
Profit		0	4	6	7	7	7	7
No. of crates	Profit							
0	0	**0**	**4**	**6**	7	7	7	7
1	2	2	**6**	**8**	9	9	9	
2	4	4	**8**	**10**	11	11		
3	6	6	**10**	**12**	12			
4	8	8	**12**	**14**				
5	9	9	13					
6	10	10						

No. of crates	0	1	2		3		4		5		6
Store 1	0	1	2	1	2	1	2	1	2	1	2
Store 2	0	0	0	1	1	2	2	3	3	4	4
Profit.	0	4	6	6	8	8	10	10	12	12	14

No. of crates		0 0, 0	1 1, 0	2 2, 0	3 2, 1	4 2,2	5 2, 3	6 3 2
Profit		0	6	8	10	11	12	12
No. of crates	Profit							
0 (0, 0)	0	0	6	8	10	11	12	12
1 (1, 0)	4	4	10	12	14	15	16	
2 (2, 0), (1, 1)	6	6	12	14	16	17		
3 (2, 1), (1, 2)	8	8	14	16	18			
4 (2,2), (1, 3)	10	10	16	18				

5(2, 3), (1, 4)	12	12	18					
6 (2, 4)	14	14						

All the four stores combined at 3 rd stage

No. of crates	0	1	2	3			4					5						6					
Store 1				2	1	1	2	1	2	1	1	2	1	2	1	2	1	2	1	2	1	2	1
Store 2				0	1	0	1	2	0	1	0	2	3	1	2	0	1	3	4	2	3	1	2
Store 3				1	1	2	1	1	2	2	2	1	1	2	2	2	2	1	1	2	2	2	2
Store 4				0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	1
Profit.	0	6	10	12			14					16						18					

Maximum profit is Rs. 18/-

Problem 3 (Cargo load problem)

A vessel is to be loaded with stocks of 3 items. Each item 'i' has a weight of w_i and a value of v_i. The maximum cargo weight the vessel can take is 5 and the details of the three items are as follows:

j	w_j	v_j
1	1	30
2	3	80
3	2	65

Develop the recursive equation for the above case and find the most valuable cargo load without exceeding the maximum cargo weight by using dynamic programming.

Solution:

Let us represent the three items as x_j (j = 1, 2, 3) and we have to take decision how much of each item is to be loaded into the vessel to fulfill the objective. Let f_j (x_j) is the value of optimal allocation for the three items, and if f (s, x) is the value associated with the optimum solution f^* (s) for (j = 1, 2

Now as the weight value of item number 1 is 1 (= w_1) only and the maximum load (W) that can be loaded is 5 the largest value of item number one that can be loaded is = W/w_1 = 5/1 = 5. The tabular computation for stage 1 is:

x_1	0	1	2	3	4	5	$f_1^*(s)$	x_1^*
s	-	-	-	-	-	-	-	-
0	0						0	0
1	0	30					30	1
2	0	30	60				60	2
3	0	30	60	90			90	3
4	0	30	60	90	120		120	4
5	0	30	60	90	120	150	150	5

The entries in the above table are obtained as follows: As the five items can be loaded as $W/w_1 = 5$, when load is zero the value is 30 × 0 = 0, when load is 1, value 30 × 1 = 30 and so on. The maximum in the row is written in 8th column, *i.e.* 0, 30, 60… 150. And the load for that weight is written in the last column. Similarly we can write for item number 2.

Specimen calculations:

For zero load: $[80 \times 0 + f_1 (0 + 3 \times 0) = 0$.

For load 1: [80 + 0] = 80.

The load of item 2 that can be loaded is $W/w_2 = 5/3 = 1$. Hence in the table for x_2 only 0 and 1 are shown.

Value of $80\, x_2 + f^* (s - 3x_2)$.

x_2	0	1					$f_2(s)$	x^*_2
s	-	-	-	-	-	-	-	-
0	0 + 0 = 0						0	0
1	0 + 30 = 30						30	0
2	0 + 60 = 60						60	0
3	0 + 90 = 90	80 + 0 = 80					90	0
4	0 + 120 = 120	80 + 30 = 110					120	0
5	0 = 150 = 150	80 + 60 = 140					150	0

As all the maximum values are due to item number 1, the item number 2 is not loaded into the cargo. Hence here $x_2 = 0$.

For stage 3, the items that can be loaded into cargo is $S/w_3 = 5/2 = 2$. Hence 0, 1, 2 are shown in the table.

x_3	0*f*	1	2				$f_3(s)$	x^*_3
s	-	-	-	-	-	-	-	-
0	0 + 0 = 0						0	0
1	0 + 30 = 30						30	0
2	0 + 60 = 60	65 + 0 = 65					65	1
3	0 + 90 = 90	65 + 35 = 95					95	1
4	0 + 120 = 120	65 + 60 = 125	130 + 0 = 130				130	2
5	0 + 150 = 150	65 + 90 = 155	130 + 30 = 160				160	2

In the above table, $x_1 = 1$, $x_2 = 0$ and $x_3 = 2$ and the maximum value is 160, therefore answer is:

$\mathbf{x^*_1 = 1, x^*_3 = 2}$ **and** $\mathbf{f^*_3 = 160}$.

Problem 4:

In a cargo-loading problem, there are four items of different weight per unit and value as shown below. The maximum cargo load is restricted to 17 units. How many units of each item is loaded to maximize the value?

Item (i)	Weight (w_1)	Value (v_1)
1	1	1
2	3	5
3	4	7
4	6	11

Solution

Let x_1, x_2, x_3 and x_4 be the items loaded then we have to maximize sum of $x_i\ v_i$, *i.e.*

Maximize Z = $a_1\ x_1 + a_2\ x_2 + a_2\ x_3 + a_4\ x_4$ s.t.

$a_1\ x_1 + a_2\ x_2 + a_2\ x_3 + a_4\ x_4$ □ 17 and all a_i are = 0.

For item number 1$f_1\ (x_1)$ = Max [$a_1\ v_1$] where the value of a_1 may be anything between maximum weight (W) allowed divided by item weight (w). Here W/w = 17/1 = 17.

For item number 2, $f_2\ (x_2)$ = Max [$a_2\ v_2 + f_1\ (x_2 – a_2\ w_2)$] For item number 3, $f_3\ (x_3)$ = Max [$a_3\ v\ _3 + f_2\ (\ x_3 – a_3\ w_3)$] For item number 4, $f_4\ (x_4)$ = Max [$a_4\ v_4 + f_3\ (\ x_4 – a_4\ w_4)$]

In general, for item '*i*' $f_i\ (x_i)$ = Max. [$av + f_{i\,–\,1}\ (x_i – a_i\ w_i)$ is the recursive equation.

In the given problem the recursive equations are:

1. $x_1 v_1$

2. $5x_2 + f_1\ (x_2 – 3x_2)$

3. $7x_3 + f_2\ (x_3 – 4x_3\ )$

4. $11x_4 + f_3\ (x_4 – 6x_4)$

Substituting the value stage by stage, the values are tabulated in the table given below: (Remember as there are 4 items, this is a 4-stage problem.

x_i	Stage 1	= x_1v	Stage 2	= 5 x_2 + $f_1(x_2 – 3x_2)$	Stage 3	= 7x_3 + $f_2(x_3 – 4x_3)$	Stage 4	= 11x_4 + $f_3(x_4 – 6x_4)$	
	w_1 = 1	v_1 = 1	w_2 = 3	v_2 = 5	w_3 = 4	v_3 = 7	w_4 = 6	v_4 = 11	$F^*_I\ (x_I)$
	x_1	$f_1\ (x_1)$	x_2	$f_2\ (x_2)$	x_3	$f\ (x_3)$	x_4	$f_4\ (x_4)$	
0	0	**0**	0		0		0		0
1	1	**1**	0		0		0		1
2	2	**2**	0		0		0		2
3	3	3	1	**5 + 0 = 5**	0		0		5

4	4	4	1	5 + 1 = 6	1	**7 + 0 = 7**	0		7
5	5	5	1`	5 + 2 = 7	1	**7 + 1 = 8**	0		8
6	6	6	2	10 + 0 = 10	1	**7 + 2 = 9**	1	**11 + 0 = 11**	11
7	7	7	2	1 0 + 1 = 11	1	**7 + 5 = 12**	1	**11 + 1 = 12**	12
8	8	8	2	10 + 2 = 12	2	**14 + 0 = 14**	1	11 + 2 = 13	14
9	9	9	3	15 + 0 = 15	2	14 = 1 = 15	1	**11 + 5 = 16**	16
10	10	10	3	15 + 1 = 16	2	14 + 2 = 16	1	**11 + 7 = 18**	18
11	11	11	3	15 + 2 = 17	2	**14 + 5 = 19**	1	**11 + 8 = 19**	19
12	12	12	4	20 + 0 = 20	3	21 + 0 = 21	2	**22 + 0 = 22**	22
13	13	13	4	20 + 1 = 21	3	21 + 1 = 22	2	**22 + 1 = 23**	23
14	14	14	4	20 + 2 = 22	3	21 + 2 = 23	2	**22 + 2 = 24**	24
15	15	15	5	25 + 0 = 25	3	21 + 5 = 26	2	**22 + 5 = 27**	27
16	16	16	5	25 + 1 = 26	4	28 + 0 = 28	2	**22 + 7 = 29**	29
17	17	17	5	25 + 2 = 27	4	28 + 1 = 29	2	**22 + 8 = 30**	30

Answer: To maximize the value of the cargo load 1 unit of item 1, 1 unit of item 3 and 2 units of item 4. The maximum value of the cargo is 30.

Problem 5:

In a cargo-loading problem, there are four items of different unit weight and value. The maximum cargo load is 6 units. How many units of each item are loaded to maximize the value?

Item	Weight (w_i)	Value per unit.
1	1	1
2	3	3
3	4	5
4	4	4

Solution:

The model is: Maximize $Z = 1a + 3b + 5c + 4d$ subject to $1a + 3b + 4c + 4d = 6$ units. And a, b, c, d all = 0.

Number of units of 'a' = W /w = 6 / 1 = 6

'b' = 6 / 3 = 2

'c' = 6 / 4 = 1

‘*d*’ = 6 / 4 = 1

Let us combine weights *c* and *d* first.

			W	0	4	8	12	16
			C	0	1	2	3	4
W	d	Z	Z	0	5	10	15	20
0	0	0		**0**	**0, 5**	8, 10		
4	1	4		4, 4	**4, 5**			
8	2	8		8, 8				

Here for *c* = 1 and *d* = 1 the element (4, 5) has selected instead of (8, 8) and (8, 10) because it is within the given limit of maximum load 6 units.

C	0	1	1
D	0	0	1
W	0	4	4
Z	0	5	5

For Table 2 let us combine ‘*a*’ and ‘*b*’

			W	0	1	2	3	4	5	6
			a	0	1	2	3	4	5	6
W	*b*	*Z*	*Z*	0	1	2	3	4	5	6
0	0	0		**0**	1, 1	2, 2	3, 3	4, 4	5, 5	6, 6
3	1	3		**3, 3**	**3, 3**					
6	2	6		**6, 6**						

A	0	0	1	0	6
B	0	1	0	2	0
W	0	3	3	6	6
Z	0	3	3	6	6

Now combining, *a* and *b* with *c* and *d* we get.

				W	0	4	4
				c	0	1	1
				d	0	0	1
W	*a*	*b*	*Z*	*Z*	0	5	5
0	0	0	0		**0**	**4, 5**(0,0,1,0)	**4, 5**(0, 0, 1, 1)

3	0	1	3		3, 3(0,1,0,0)	1, 5(0, 1, 1, 0)	
3	1	0	3		(1, 0,0,0,)	(1, 0, 1, 0)	
6	0	2	6		**6, 6**(0, 2, 0, 0)		
6	6	0	6		(6, 1,0,0)		

Maximum weight = 6 units, $a = 6$, $b = 0$, $c = 0$ and $d = 0$ or $a = 0$, $b = 2$, $c = 0$ and $d = 0$ Substituting the values in the model we get, Maximize $Z = 1a + 3b + 5c + 4d$ and

$1a + 3b + 4c + 4d = 6$

$1 \times 6 + 0 \times 0 + 0 \times 0 + 0 \times 0 = 6$ or $0 \times 0 + 3 \times 2 + 0 \times 0 + 0 \times 0 = 6$ and

$1 \times 6 + 3 \times 0 + 4 \times 0 + 4 \times 0 = 6$.

Problem 6:

Determine the value of u_1 , u_2, and u_3 so as to Maximize $u_1 . u_2 . u_3$ subject to $u_1 + u_2 + u_3 = 10$ and u_1, u_2 and u_3 all = 0.

Solution

This can be treated as a 3-stage problem, with the state variable x_i and the return f_i (x_i), such that

At stage 3,$x_3 = u_1 + u_2 + u_3 = 10$,

at stage 2,$x_2 = x_3 – u_3 = u_1 + u_2$,

at stage 1$x_1 = x_2 – u_2 = u_1$.

and the returns are:

$f_3 (x_3) = \max [u_3 f_2 (x_2)]$

$f_2 (x_2) = \max [u_2 f_1(x_1)]$

Since $u_1 = (x_2 – u_2)$

$f_1 (x_1) = u_1$

$f_2 (x_2) = \max [u_2 (x_2 – u_2)] = \max [u_2 x_2 – u_2^2]$

Differentiating $[u_2 x_2 – u_2^2]$ w.r.t. u_2 and equating to zero (to find the maximum value)

$u_2 – 2 u_2 = 0$ or $u_2 = (x_2/2)$, therefore,

$f_2 (x_2) = (x_2/2) . x_2 – (x_2/2)^2 = (x_2^2/4)$

Differentiating $[u_3 . (x_3 – u_3)^2 / 4]$ w.r.t. u_3 and equating to zero, $(1/4) [u_3 . 2 (x_3 – u_2) (–1) + (x_3 – u_3)^2] = 0$ or $(x_3 – u_3) (– 2 u_3 + x_3 – u_3) = 0$ or $(x_3 – u_3) (x_3 – 3u_3) = 0$.

Now, either $u_3 = x_3$, which is trivial as $u_1 + u_2 + u_3 = x_3$ or $u_3 = (x_3/3) = (10/3)$ Therefore, $u_2 (x_2/2) = (x_3 – u_3)/2 = (1/2) [10 – (10/3)] = (10/3)u_1 = x_2 – u_2 = (20/3) – (10/3) = (10/3)$

Therefore, $u_1 = u_2 = u_3 = (10/3)$ and maximum $(u_1 . u_2 . u_3) = (1000/27)$.

UNIT-V

Introduction to quadratic approximation:-

Quadratic approximation is an extension of linear approximation, where we are adding one more term, which is related to the second derivative. The formula fo r the quadratic approximation of a function f(x) for values of x near x"0" is :

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 \quad (x \approx x_0)$$

Compare this to the old formula for the linear approximation of f:

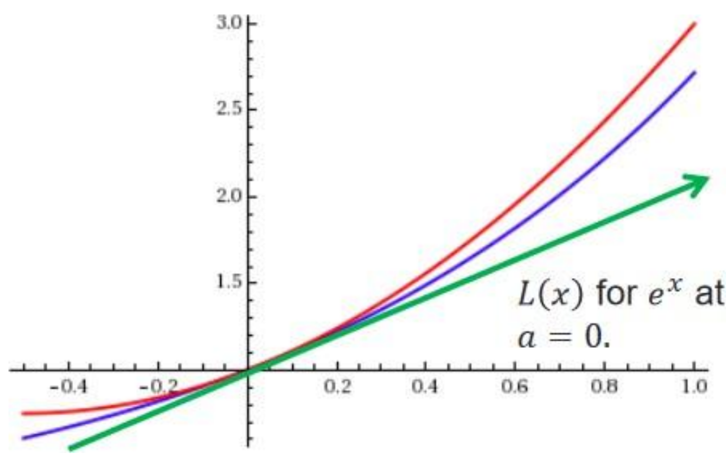
$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \quad (x \approx x_0).$$

We got from the linear approximation to the quadratic one by adding one more term that is related to the second derivatives:

$$f(x) \approx \underbrace{f(x_0) + f'(x_0)(x - x_0)}_{\text{Linear Part}} + \underbrace{\frac{f''(x_0)}{2}(x - x_0)^2}_{\text{Quadratic Part}} \quad (x \approx x_0)$$

These are more complicated and so are only used when higher accuracy is needed.

The quadratic approximation also uses the point x=a to approximate nearby values, but uses a parabola instead of just a tangent line to do so.



This gives a closer approximation because the parabola stays closer to the actual function.

Algorithm to Convert a CFG into Greibach Normal Form

Step 1 − If the start symbol S occurs on some right side, create a new start symbol S' and a new production S' → S.

Step 2 − Remove Null productions. (Using the Null production removal algorithm discussed earlier)

Step 3 − Remove unit productions. (Using the Unit production removal algorithm discussed earlier)

Step 4 − Remove all direct and indirect left-recursion.

Step 5 − Do proper substitutions of productions to convert it into the proper form of GNF.

Example: Convert the following grammar G into Greibach Normal Form (GNF).

S→ X A|BB

B → b|SB

X → b

A → a

To write the above grammar G into GNF, we shall follow the following steps:

1. Rewrite G in Chomsky Normal Form (CNF) It is already in CNF.
2. Re-label the variables

S with A_1

X with A_2

A with A_3

B with A_4

After re-labeling the grammar looks like:

A1 → A2 A3|A4A4

A4 → b|A1 A4

A2 → b

A3 → a

Identify all productions which do not conform to any of the types listed below:

$A_i \rightarrow A_j x_k$ such that j > I

$Z_i \rightarrow A_j x_k$ such that $j \leq n$

$A_i \rightarrow a x_k$ such that $x_k \in V*$ and $a \in T$

$A_4 \rightarrow A_1 A_4$..................identified

$A_4 \rightarrow A_1 A_4$ |b.

To eliminate A_1 we will use the substitution rule $A_1 \rightarrow A_2 A_3|A_4A_4$.

Therefore, we have $A_4 \rightarrow A_2 A_3A_4 |A_4A_4 A_4|b$

The above two productions still do not conform to any of the types in step 3.

Substituting for $A_2 \rightarrow b$

$A_4 \rightarrow bA_3 A_4 | A_4A_4A_4 | b$

Now we have to remove left recursive production

$A_4 \rightarrow A_4A_4 A_4$

$A_4 \rightarrow bA_3 A_4 | b | bA_3 A_4Z | bZ$

$Z \rightarrow A_4 A_4 | A_4A_4Z$

At this stage our grammar now looks like

A1 → A2 A3|A4A4

A4 → bA3 A4 |b|bA3 A4Z|bZ

Z → A4 A4 |A4A4Z

A2 → b

A3 → a

All rules now conform to one of the types in step 3.

But the grammar is still not in Greibach Normal Form!

All productions for A_2 , A_3 and A_4 are in GNF

for $A_1 \rightarrow A_2 A_3 | A_4A_4$

Substitute for A_2 and A_4 to convert it to GNF

$A_1 \rightarrow bA_3 | bA_3 A_4A_4 | bA_4 | bA_3A_4 Z A_4 | bZ A_4$

for $Z \rightarrow A_4 A_4 | A_4A_4Z$

Substitute for A_4 to convert it to GNF

$Z \rightarrow bA_3 A_4 A_4 | bA_4 | bA_3 A_4 Z A_4 | bZ A_4 | bA_3A_4 A_4 Z | bA_4Z | bA_3 A_4Z A_4 Z | bZ A_4Z$ 8. Finally the grammar in GNF is

$A_1 \rightarrow bA_3 | bA_3 A_4A_4 | bA_4 | bA_3A_4 Z A_4 | bZ A_4$

$A_4 \rightarrow bA_3 A_4 | b | bA_3 A_4Z | bZ$

$Z \rightarrow bA_3 A_4 A_4 | bA_4 | bA_3 A_4 Z A_4 | bZ A_4 | bA_3A_4 A_4 Z | bA_4Z | bA_3 A_4Z A_4 Z | bZ A_4Z$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

Convert the following CFG into CNF

S → XY | Xn | p

X → mX | m

Y → Xn | o

Solution

Here, **S** does not appear on the right side of any production and there are no unit or null productions in the production rule set. So, we can skip Step 1 to Step 3.

Step 4

Now after replacing

X in S → XY | Xo | p

with

mX | m

We obtain

S → mXY | mY | mXo | mo | p.

And after replacing

X in Y → X_n | o with the right side of

X → mX | m

We obtain

Y → mXn | mn | o.

Two new productions O → o and P → p are added to the production set and then we came to the final GNF as the following −

S → mXY | mY | mXC | mC | p

X → mX | m

Y → mXD | mD | o

O → o

P → p

<u>Types of Non-Linear Programming Problems:-</u>

In the preceding two chapters we considered a number of alternative strategies for exploiting linear approximations to nonlinear problem functions. In general we found that, depending upon the strategy employed, linearization's would either lead to vertex points of the linearized constraint sets or generate descent directions for search. In either case, some type of line search was required in order to approach the solution of non-corner-point constrained problems. Based upon our experience with unconstrained problems, it is reasonable to consider the use of higher order approximating functions since these could lead directly to better approximations of non-corner-point solutions. For instance, in the single-variable case we found that a quadratic approximating function could be used to predict the location of optima lying in the interior of the search interval. In the multivariable case, the use of a quadratic approximation (e.g., in Newton's method) would yield good estimates of unconstrained minimum points. Furthermore, the family of QuasiNewton methods allowed us to reap some of the benefits of a quadratic approximation without explicitly developing a full second-order approximating function at each iteration. In fact, in the previous chapter we did to some extent exploit the acceleration capabilities of quasi-Newton methods by introducing their use within the direction generation mechanics of the reduced gradient and gradient projection methods.

Thus, much of the discussion of the previous chapters does point to the considerable algorithmic potential of using higher order, specifically quadratic, approximating functions for solving constrained problems. In this chapter we examine in some detail various strategies for using quadratic approximations. We begin by briefly considering the consequence of direct quadratic approximation, the analog of the successive LP strategy.

Then we investigate the use of the second derivatives and Legrangian constructions to formulate quadratic programming (QP) sub problems, the analog to Newton's method. Finally, we discuss the application of quasi-Newton formulas to generate updates of quadratic terms.

We will find that the general NLP problem can be solved very efficiently via a series of sub problems consisting of a quadratic objective function and linear constraints, provided a suitable line search is carried out from the solution of each such subproblem. The resulting class of exterior point algorithms can be viewed as a natural extension of quasi-Newton methods to constrained problems.

Direct Quadratic Approximation:-

The solution of the general NLP problem is by simply replacing each nonlinear function by its local quadratic approximation at the solution estimate x0 and solving the resulting series of approximating sub problems. If each function $f(x)$ is replaced by its quadratic approximation

$$q(x; x^0) = f(x^0) + \nabla f(x^0)^T (x - x^0) + \tfrac{1}{2}(x - x^0)^T \nabla^2 f(x^0)(x - x^0)$$

then the subproblem becomes one of minimizing a quadratic function subject to quadratic equality and inequality constraints. While it seems that this subproblem structure ought to be amendable to efficient solution, in fact, it is

not. To be sure, the previously discussed strategies for constrained problems can solve this subproblem but at no real gain over direct solution of the original problem. For a sequential strategy using subproblem solutions to be Effective, the subproblem solutions must be substantially easier to obtain than the solution of the original problem. Recall from the discussion in Section 9.2.3 that problems with a quadratic positive-definite objective function and linear constraints can be solved in a finite number of reduced gradient iterations provided that quasi-Newton or conjugate gradient enhancement of the reduced gradient direction vector is used. Of course, while the number of iterations is finite, each iteration requires a line search, which strictly speaking is itself not a finite procedure. However, as we will discover in Chapter 11, there are specialized methods for these so-called QP problems that will obtain a solution in a finite number of iterations without line searching, using instead simplex like pivot operations. Given that QP problems can be solved efficiently with truly finite procedures, it appears to be desirable to formulate our approximating sub problems as quadratic programs. Thus, assuming that the objective function is twice continuously differentiable, a plausible solution strategy would consist of the following steps:

Direct Successive Quadratic Programming Solution

Given x 0, an initial solution estimate, and a suitable method for solving QP sub problems.

Step 1. Formulate the QP problem

$$\begin{aligned} &\text{Minimize} && \nabla f(x^{(t)})^T d + \tfrac{1}{2} d^T \nabla^2 f(x^{(t)}) d \\ &\text{Subject to} && h_k(x^{(t)}) + \nabla h_k(x^{(t)})^T d = 0 \qquad k = 1, \dots, K \\ & && g_j(x^{(t)}) + \nabla g_j(x^{(t)})^T d \geq 0 \qquad j = j = 1, \dots, J \end{aligned}$$

Step 2. Solve the QP problem and.

Step 3. Check for convergence. If not converged, repeat step 1.

As shown in the following example, this approach can be quite effective.

Example 1

Solve the problem

$$\begin{aligned} &\text{Minimize} && f(x) = 6x_1 x_2^{-1} + x_2 x_1^{-2} \\ &\text{Subject to} && h(x) = x_1 x_2 - 2 = 0 \\ & && g(x) = x_1 + x_2 - 1 \geq 0 \end{aligned}$$

From the initial feasible estimate x 0 = (2, 1) using the direct successive QP strategy.

At x0, f(x 0) = 12.25, h(x0) = 0, and g(x 0) = 2 > 0. The derivatives required to construct the QP subproblem are

$$\begin{aligned} \nabla f(x) &= ((6x_2^{-1} - 2x_2 x_1^{-3}), (-6x_1 x_2^{-2} + x_1^{-2}))^T \\ \nabla^2 f &= \begin{pmatrix} (6x_2 x_1^{-4}) & (-6x_2^{-2} - 2x_1^{-3}) \\ (-6x_2^{-2} - 2x_1^{-3}) & (12x_1 x_2^{-3}) \end{pmatrix} \\ \nabla h(x) &= (x_2, x_1)^T \end{aligned}$$

Thus, the first QP subproblem will be

$$\begin{aligned} &\text{Minimize} && (\tfrac{23}{4}, -\tfrac{47}{4}) d + \tfrac{1}{2} d^T \begin{pmatrix} \tfrac{3}{8} & -\tfrac{25}{4} \\ -\tfrac{25}{4} & 24 \end{pmatrix} d \\ &\text{Subject to} && (1, 2) d = 0 \\ & && (1, 1) d + 2 \geq 0 \end{aligned}$$

Since the first constraint can be used to eliminate one of the variables, that is,

d1 = - 2d2

the resulting single-variable problem can be solved easily analytically to give

$$d^0 = (-0.92079, 0.4604)$$

Thus, the new point becomes

$$f(x^{(1)}) = 5.68779$$
$$h(x^{(1)}) = -0.42393$$
$$g(x^{(1)}) > 0$$

At which point

Note that the objective function value has improved substantially but that the equality constraint is violated. Suppose we continue with the solution procedure. The next subproblem is

$$\text{Minimize} \quad (1.78475, -2.17750)d + \tfrac{1}{2}d^T\begin{pmatrix} 6.4595 & -4.4044 \\ -4.4044 & 4.1579 \end{pmatrix}d$$
$$\text{Subject to} \quad (1.4604, 1.07921)d - 0.42393 = 0$$
$$(1, 1)d + 1.5396 \geq 0$$

The solution is $d^{(1)} = (-0.03043, 0.43400)$, resulting in the new point, with

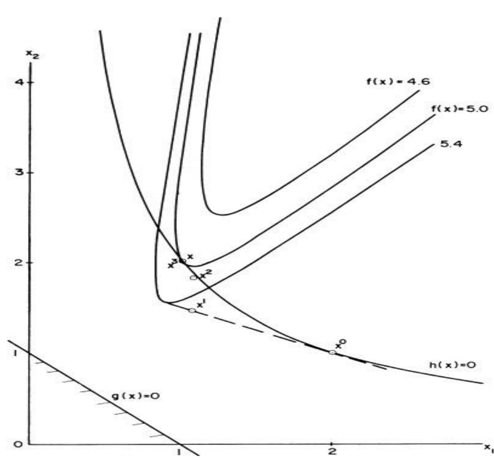
$$x^{(2)} = (1.04878, 1.89440)$$
$$f(x^{(2)}) = 5.04401$$
$$h(x^{(2)}) = -0.013208$$
$$g(x^{(2)}) > 0$$

Note that both the objective function value and the equality constraint violation are reduced. The next two iterations produce the results

x(3) = (1.00108, 1.99313) f(x(3)) = 5.00457 h(x(3)) = - 4.7 * 10-3

x(4) = (1.00014, 1.99971) f(x(4)) = 5.00003 h(x(4)) = -6.2 * 10-6

The exact optimum is x* (1, 2) with f(x*) 5.0; a very accurate solution has been obtained in four iterations. As is evident from Figure 10.1, the constraint linearization's help to define the search directions, while the quadratic objective function approximation effectively fixes the step length along that direction. Thus, the use of a nonlinear objective function approximation does lead to non-vertex solutions. However, as shown in the next example, the self-bounding feature of the quadratic approximation does not always hold.



Direct Quadratic Approximation:-

Example 2

Suppose the objective function and equality constraint of Example 10.1 are interchanged. Thus, consider the problem

$$\text{Minimize} \quad f(x) = x_1 x_2$$

$$\text{Subject to} \quad h(x) = 6x_1 x_2^{-1} + x_2 x_1^{-2} - 5 = 0$$

$$g(x) = x_1 + x_2 - 1 \geq 0$$

The optimal solution to this problem is identical to that of Example 10.1. With the starting point $x^0 = (2, 1)$ the first subproblem becomes

$$\text{Minimize} \quad (1, 2)d + \tfrac{1}{2}d^T \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} d$$

$$\text{Subject to} \quad (\tfrac{23}{4}, -\tfrac{47}{4})d + \tfrac{29}{4} = 0$$

$$(1, 1)d + 2 \geq 0$$

The solution to this subproblem is $d^0 = (1.7571, 0.24286)$, at which both constraints are tight. Thus, a subproblem corner point is reached. The resulting intermediate solution is

$$x^1 = x^0 + d^0 = (0.24286, 0.75714)$$

With

$$f(x^{(1)}) = 0.18388 \qquad h(x^{(1)}) = 9.7619 \qquad g(x(1)) = 0$$

Although the objective function value decreases, the equality constraint vio- lation is worse. The next subproblem becomes

$$\text{Minimize} \quad (0.75714, 0.24286)d + \tfrac{1}{2}d^T \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} d$$

$$\text{Subject to} \quad -97.795d_1 + 14.413d_2 + 9.7619 = 0$$

$$d_1 + d_2 \geq 0$$

The resulting new point is

$$x^{(2)} = (0.32986, 0.67015)$$

$$f(x^{(2)}) = 0.2211 \qquad h(x^{(2)}) = 4.1125 \qquad g(x^{(2)}) = 0$$

Again, g(x) is tight. The next few points obtained in this fashion are

$$x^{(3)} = (0.45383, 0.54618)$$

$$x^{(4)} = (-0.28459, 1.28459)$$

$$x^{(5)} = (-0.19183, 1.19183)$$

These and all subsequent iterates all lie on the constraint g(x) = 0. Since the objective function decreases toward the origin, all iterates will be given by the intersection of the local linearization with g(x) = 0. Since the slope of the linearization becomes larger or smaller than -1 depending upon which side of the constraint ''elbow'' the linearization point lies, the successive iterates simply follow an oscillatory path up and down the surface g(x)=0.

Evidently the problem arises because the linearization cannot take into account the sharp curvature of the constraint h(x) = 0 in the vicinity of the optimum. Since both the constraint and the objective function shapes serve to define the location of the optimum, both really ought to be taking into ac-count.

Quadratic Approximation of the Legrangian Function:-

The examples of the previous section suggest that it is desirable to incorporate into the subproblem definition not only the curvature of the objective function but also that of the constraints. However, based on computational considerations, we also noted that it is preferable to deal with linearly constrained rather than quadratically constrained sub problems. Fortunately, this can be accomplished by making use of the Legrangian function, as will be shown below. For purposes of this discussion, we first consider only the equality- constrained problem. The extension to inequality constraints will follow in a straightforward fashion. Consider the problem

$$\text{Minimize} \quad f(x)$$

$$\text{Subject to } h(x) = 0$$

Recall that the necessary conditions for a point x^* to be a local minimum are that there exist a multiplier v^* such that

$$Q_x L(x^*, v^*) = Qf^* - v^{*T} \; Qh^* = 0 \quad \text{and} \quad h(x^*) = 0 \quad (10.1)$$

Sufficient conditions for x* to be a local minimum (see Theorem 5.7 of Chapter 5) are that conditions (10.1) hold and that the Hessian of the La- grangian function,

$$Q_x^2 L(x^*, v^*) = Q^2 f^* - (v)^{*T} Q^2 h^*$$

Satisfies

$$d^T Q_x^2 L d > 0$$

for all d such that (Qh)*Td = 0 (10.2)

Given some point $(\bar{x}, \bar{v})$, we construct the following subproblem expressed in terms of the variables d:

$$\text{Minimize} \quad \text{Q}f(\bar{x})^{\text{T}}d + \tfrac{1}{2}d^{\text{T}}\,\text{Q}_x^2 L(\bar{x}, \bar{v})d \qquad (10.3)$$

$$\text{Subject to} \quad h(\bar{x}) + \text{Q}h(\bar{x})^{\text{T}}d = 0 \qquad (10.4)$$

We now observe that if d* = 0 is the solution to the problem consisting of (10.3) and (10.4), then x must satisfy the necessary conditions for a local minimum of the original problem. First note that if d* = 0 solves the sub-problem, then form (10.4) it follows that h(x) = 0; in other words, x is a feasible point. Next, there must exist some v* such that the subproblem functions satisfy the Legrangian necessary conditions at d* = 0. Thus, since the gradient of the objective function (10.3) with respect to d at d* = 0 is Qf(x) and that of (10.4) is Qh(x), it follows that

$$d\{\text{Q}_x^2 L(\bar{x}, v^*) - v^*(0)\}d > 0 \qquad \text{for all } d \text{ such that } \text{Q}h(x)^{\text{T}}d = 0$$

Note that the second derivative with respect to d of (10.4) is zero, since it is a linear function in d. Consequently, the above inequality implies that d T Qx2L $\bar{x}$ (, v*)d is positive also. Therefore, the pair $\bar{x}$ (, v*) satisfies the sufficient conditions for a local minimum of the original problem. This demonstration indicates that the subproblem consisting of (10.3) and (10.4) has the following very interesting features:

1. If no further corrections can be found, that is, d = 0, then the local minimum of the original problem will have been obtained.
2. The Lagrange multipliers of the subproblem can be used conveniently as estimates of the multipliers used to formulate the next subproblem.
3. For points sufficiently close to the solution of the original problem the quadratic objective function is likely to be positive definite, and thus the solution of the QP subproblem will be well behaved.

By making use of the sufficient conditions stated for both equality and in- equality constraints, it is easy to arrive at a QP subproblem formulation for the general case involving K equality and J inequality constraints. If we let

$$L(x, u, v) = f(x) - 3\, v_k h_k(x) - 3\, u_j g_j(x) \qquad (10.6)$$

then at some point $(\bar{x}, \bar{u}, \bar{v})$ the subproblem becomes

$$\text{Minimize} \quad q(d; \bar{x}) \div \text{Q}f(\bar{x})^{\text{T}}d + \tfrac{1}{2}d^{\text{T}}\,\text{Q}_x L(\bar{x}, \bar{u}, \bar{v})d \qquad (10.7)$$

$$\text{Subject to} \quad \tilde{h}_k(d; \bar{x}) \div h_k(\bar{x}) + \text{Q}h_k(\bar{x})^{\text{T}}d = 0 \qquad k = 1, \ldots, K \quad (10.8a)$$

$$\tilde{g}_j(d; \bar{x}) \div g_j(\bar{x}) + \text{Q}g_j(\bar{x})^{\text{T}}d \;``\; 0 \qquad j = 1, \ldots, J \quad (10.8b)$$

The algorithm retains the basic steps outlined for the direct QP case. Namely, given an initial estimate x^0 as well as u^0 and v^0 (the latter could be set equal to zero, we formulate the subproblem [Eqs. (10.7), (10.8a), and

(10.8b)]; solve it; set x(t+1) = x(t) + d; check for convergence; and repeat, using as next estimates of u and v the corresponding multipliers obtained at the solution of the sub problem.

Quadratic Approximation of the Legrangian Function:-

Example 3

Repeat the solution of the problem of Example 10.1 using the Legrangian QP subproblem with initial estimates $x^0 = (2, 1)T$, $u^0 = 0$, and $v^0 = 0$. The first subproblem becomes

$$\text{Minimize} \quad \left(\tfrac{23}{4}, -\tfrac{47}{4}\right)d + \tfrac{1}{2}d^T\begin{pmatrix} \tfrac{3}{8} & -\tfrac{25}{4} \\ -\tfrac{25}{4} & 24 \end{pmatrix}d$$

$$\text{Subject to} \quad (1, 2)d = 0$$

$$(1, 1)d + 2 \geqslant 0$$

This is exactly the same as the fist subproblem of Example because with the initial zero estimates of the multipliers of the constraint terms of the Legrangian will vanish. The subproblem solution is thus, as before,

$$d^0 = (—0.92079, 0.4604)^T$$

Since the inequality constraint is loose at this solution, u(1) must equal zero. The equality constraint multiplier can be found from the solution of the Legrangian necessity conditions for the subproblem. Namely,

$$\nabla q(d^0; x^0) = v \, \nabla\tilde{h}(d^0; x^0)$$

or

$$\left\{\left(\tfrac{23}{4}, -\tfrac{47}{4}\right) + d^T\begin{pmatrix} -\tfrac{3}{8} & -\tfrac{25}{4} \\ -\tfrac{25}{4} & 24 \end{pmatrix}\right\} = v(1, 2)^T$$

Thus, $v^{(1)} = 2.52723$. Finally, the new estimate of the problem solution will be $x^{(1)} = x^0 + d^0$, or

$x^{(1)} = (1.07921, 1.4604)^T$ $\qquad f(x^{(1)}) = 5.68779$ $\qquad h(x^{(1)}) = —0.42393$

as was the case before.

The second subproblem requires the gradients

$$\nabla f(x^{(1)}) = (1.78475, -2.17750)^T \qquad \nabla h(x^{(1)}) = (1.4604, 1.07921)^T$$

$$\nabla^2 f(x^{(1)}) = \begin{pmatrix} 6.45954 & -4.40442 \\ -4.40442 & 4.15790 \end{pmatrix}$$

$$\nabla^2 h(x^{(1)}) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

The quadratic term is therefore equal to

$$\nabla^2 L = \nabla^2 f - v\, \nabla^2 h$$

$$= \nabla^2 f - 2.52723\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 6.45924 & -6.93165 \\ -6.93165 & 4.15790 \end{pmatrix}$$

The complete problem becomes

$$\text{Minimize} \quad (1.78475, -2.17750)d + \tfrac{1}{2}d^T\begin{pmatrix} 6.45924 & -6.93165 \\ -6.93165 & 4.15790 \end{pmatrix}d$$

$$\text{Subject to} \quad 1.4604d_1 + 1.07921d_2 = 0.42393$$

$$d_1 + d_2 + 1.539604 \geqslant 0$$

The solution is $d^{(1)} = (0.00614, 0.38450)$. Again, since $\tilde{g}(d^{(1)}; x^{(1)}) > 0$, $u^{(2)} = 0$, and the remaining equality constraint multiplier can be obtained from

$$\nabla q(d^{(1)}; x^{(1)}) = v^T\, \nabla h(d^{(1)}; x^{(1)})$$

or

$$\begin{pmatrix} -0.84081 \\ -0.62135 \end{pmatrix} = v\begin{pmatrix} 1.46040 \\ 1.07921 \end{pmatrix}$$

Thus,

$$v^{(2)} = -0.57574 \qquad \text{and} \qquad x^{(2)} = (1.08535, 1.84490)^T$$

with

$$f(x^{(2)}) = 5.09594 \qquad \text{and} \qquad h(x^{(2)}) = 2.36 \times 10^{-3}$$

Continuing the calculations for a few more iterations, the results obtained are

$$x^{(3)} = (0.99266, 2.00463)^T \qquad v^{(3)} = -0.44046$$

$$f^{(3)} = 4.99056 \qquad h^{(3)} = -1.008 \times 10^{-2}$$

and

$$x^{(4)} = (0.99990, 2.00017)^T \qquad v^{(4)} = -0.49997$$

$$f^{(4)} = 5.00002 \qquad h^{(4)} = -3.23 \times 10^{-5}$$

It is interesting to note that these results are essentially comparable to those obtained in Example 10.1 without the inclusion of the constraint second derivative terms. This might well be expected, because at the optimal solution

(1, 2) the constraint contribution to the second derivative of the Legrangian is small:

$$\nabla^2 f^* - v^* \nabla^2 h^* = \begin{pmatrix} 12 & -3.5 \\ -3.5 & 1.5 \end{pmatrix} - \left(-\frac{1}{2}\right)\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 12 & -3.0 \\ -3.0 & 1.5 \end{pmatrix}$$

The basic algorithm illustrated in the preceding example can be viewed as an extension of Newton's method to accommodate constraints. Specifically, if no constraints are present, the subproblem reduces to

$$\text{Minimize} \quad \nabla f(\bar{x})^T d + \tfrac{1}{2} d^T \nabla^2 f(\bar{x}) d$$

Quadratic Approximation by Legrangian Function Problems:-

Example 4

Consider the problem of Example 10.2 with the initial estimate $x^0 = (2, 2.789)$ and u = v = 0. The first subproblem will be given by

$$\text{Minimize} \quad (2.789, 2)d + \tfrac{1}{2} d^T \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} d$$

$$\text{Subject to} \quad 1.4540 d_1 - 1.2927 d_2 = 1.3 \times 10^{-4}$$

$$d_1 + d_2 + 3.789 \geq 0$$

The solution is $d^0 = (-1.78316, -2.00583)$. The inequality constraint is tight, so both constraint multipliers must be computed. The result of solving the system

$$\nabla \tilde{q}(d^0; x^0) = v \nabla \tilde{h}(d^0; x^0) + u \nabla \tilde{g}(d^0; x^0)$$

$$\begin{pmatrix} 0.78317 \\ 0.21683 \end{pmatrix} = v \begin{pmatrix} -145.98 \\ 19.148 \end{pmatrix} + u \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

is $v^{(1)} = -0.00343$ and $u^{(1)} = 0.28251$.

At the corresponding intermediate point,

$$x^{(1)} = (0.21683, 0.78317)^T$$

we have

$$f(x^{(1)}) = 0.1698 \qquad h(x^{(1)}) = 13.318 \qquad g(x^{(1)}) = 0$$

Note that the objective function decreases substantially, but the equality constraint violation becomes very large. The next subproblem constructed at $x^{(1)}$ with multiplier estimates $u^{(1)}$ and $v^{(1)}$ is

$$\text{Minimize} \quad (0.78317, 0.21683)d + \tfrac{1}{2}d^T\left\{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} - v^{(1)}\begin{pmatrix} 2125.68 & -205.95 \\ -205.95 & 5.4168 \end{pmatrix} - u^{(1)}(0)\right\}d$$

$$\text{Subject to} \quad -145.98d_1 + 19.148d_2 = -13.318$$

$$d_1 + d_2 \geq 0$$

The subproblem solution is $d^{(1)} = (+0.10434,-0.10434)^T$, and the multipliers are $v^{(2)} = -0.02497$, $u^{(2)} = 0.38822$. At the new point $x^{(2)} = (0.40183,0.59817)^T$, the objective function value is 0.24036 and the constraint value is 2.7352.

The results of the next iteration,

$$x^{(3)} = (0.74969, 0.25031)^T \qquad f(x^{(3)}) = 0.18766 \qquad h(x^{(3)}) = 13.416$$

Indicate that the constraint violation has increased considerably while the objective function value has decreased somewhat. Comparing the status at $x^{(1)}$ and $x^{(3)}$, it is evident that the iterations show no real improvement. In fact, both the objective function value and the equality constraint violation have increased in proceeding from $x^{(1)}$ to $x^{(3)}$.

The solution to the problem of unsatisfactory convergence is, as in the unconstrained case, to perform a line search from the previous solution estimate in the direction obtained from the current QP subproblem solution. However, since in the constrained case both objective function improvement and reduction of the constraint infeasibilities need to be taken into account, the line search must be carried using some type of penalty function. For instance, as in the case of the SLP strategy advanced by Palacios-Gomez, an exterior penalty function of the form

$$P(x, R) = f(x) + R\left\{\sum_{k=1}^{K} [h_k(x)]^2 + \sum_{j=1}^{J} [\min(0, g_j(x))]^2\right\}$$

could be used along with some strategy for adjusting the penalty parameter R. This approach is illustrated in the next example.

<u>Example 5</u>

Consider the application of the penalty function line search to the problem of Example 10.4 beginning with the point $x^{(2)}$ and the direction vector $d^{(2)} = (0.34786, -0.34786)^T$ which was previously used to compute the point x(3) directly.

Suppose we use the penalty function

$$P(x, R) = f(x) + 10\{h(x)^2 + [\min(0, g(x)]^2 \}$$

and minimize it along the line

$$x = x^{(2)} + \alpha d^{(2)} = \begin{pmatrix} 0.40183 \\ 0.59817 \end{pmatrix} + \begin{pmatrix} 0.34786 \\ -0.34786 \end{pmatrix} \alpha$$

Note that at $\alpha = 0,$ P 75.05, while at 1, P 1800.0. Therefore, a minimum ought to be found in the range 0 1. Using any convenient line search method, the approximate minimum value P 68.11 can be found with 0.1. The resulting point will be

$$x^{(3)} = (0.43662, 0.56338)^T$$

with $f(x^{(3)}) = 0.24682$ and $h(x^{(3)}) = 2.6053$.

To continue the iterations, updated estimates of the multipliers are required. Since $\bar{d} = x^{(3)} - x^{(2)}$ is no longer the optimum solution of the previous subproblem, this value of d cannot be used to estimate the multipliers. The only available updated multiplier values are those associated with $d^{(2)}$, namely $v^{(3)} = 0.005382$ and $u^{(3)} = 0.37291$.

The results of the next four iterations obtained using line searches of the penalty function after each subproblem solution are shown in Table 10.1. As is evident from the table, the use of the line search is successful in forcing convergence to the optimum from poor initial estimates. The use of the quadratic approximation to the Legrangian was proposed by Wilson. Although the idea was pursued by Beale and by Bard and Greestadt, it has not been widely adopted in its direct form.

As with Newton's method, the barriers to the adoption of this approach in engineering applications have been two fold: first the need to provide second derivative values for all model functions and, second, the sensitivity solution estimates. Far from the optimal solution, the function second derivatives and especially the multipler estimates will really have little

Table 10.1 Results for Example 10.5

Iteration	x_1	x_2	f	h	v
3	0.43662	0.56338	0.24682	2.6055	−0.005382
4	0.48569	0.56825	0.27599	2.5372	−0.3584
5	1.07687	1.8666	2.0101	0.07108	−0.8044
6	0.96637	1.8652	1.8025	0.10589	−1.6435
7	0.99752	1.99503	1.9901	0.00498	−1.9755
∞	1.0	2.0	2.0	0.0	−2.0

relevance to defining a good search direction. (For instance, Table 10.1, $v^{(3)} = -5.38 \times 10^{-3}$, while $v^* = -2$.) Thus, during the initial block of iterations, the considerable computational burden of evaluating all second derivatives may be entirely wasted. A further untidy feature of the above algorithm involves the strategies required to adjust the penalty parameter of the line search penalty function. First, a good initial estimate of the parameter must somehow be supplied; second, to guarantee convergence, the penalty parameter must in principle be increased to large values.

Variable metric methods for Constrained optimization:-

The desirable improved convergence rate of Newton's method could be approached by using suitable update formulas to approximate the matrix of second derivatives. Thus, with the wisdom of hindsight, it is not surprising that, as first shown by Garcia Palomares and Mangasarian, similar constructions can be applied to approximate the quadratic portion of our Legrangian sub problems. The idea of approximating $\nabla^2 L$ using quasi-Newton update formulas that only require differences of gradients of the Legrangian function was further developed by Han and Powell. The basic variable metric strategy proceeds as follows.

Constrained Variable Metric Method:-

Given initial estimates x^0, u^0, v^0 and a symmetric positive-definite matrix H^0.

Step 1. Solve the problem

$$\begin{aligned} \text{Minimize} \quad & \nabla f(x^{(t)})^T d + \tfrac{1}{2} d^T \mathbf{H}^{(t)} d \\ \text{Subject to} \quad & h_k(x^{(t)}) + \nabla h_k(x^{(t)})^T d = 0 \qquad k = 1, \ldots, K \\ & g_j(x^{(t)}) + \nabla g_j(x^{(t)})^T d \geq 0 \qquad j = 1, \ldots, J \end{aligned}$$

Step 2. Select the step size α along $d^{(t)}$ and set $x^{(t+1)} = x^{(t)} + \alpha d^{(t)}$.

Step 3. Check for convergence.

Step 4. Update $H^{(t)}$ using the gradient difference

$$\nabla_x L(x^{(t+1)}, u^{(t+1)}, v^{(t+1)}) - \nabla_x L(x^{(t)}, u^{(t+1)}, v^{(t+1)})$$

in such a way that $H^{(t+1)}$ remains positive definite.

The key choices in the above procedure involve the update formula for $H^{(t)}$ and the manner of selecting. Han considered the use of several well known update formulas, particularly DFP. He also showed that if the initial point is sufficiently close, then convergence will be achieved at a super linear rate without a step-size procedure or line search by setting $\alpha =$ 1. However, to assure convergence from arbitrary points, a line search is required. Specifically, Han recommends the use of the penalty function

$$P(x, R) = f(x) + R\left\{\sum_{k=1}^{K} |h_k(x)| - \sum_{j=1}^{J} \min[0, g_j(x)]\right\}$$

to select α^* so that

$$P(x(\alpha^*)) = \min_{0 \leq \alpha \leq \delta} P(x^{(t)} + \alpha\, d^{(t)}, R)$$

where R and δ are suitably selected positive numbers.

Powell, on the other hand, suggests the use of the BFGS formula together with a conservative check that ensures that $H^{(t)}$ remains positive definite. Thus, if

$$z = x^{(t+1)} - x^{(t)}$$

and

$$y = \nabla_x L(x^{(t+1)}, u^{(t+1)}, v^{(t+1)}) - \nabla_x L(x^{(t)}, u^{(t+1)} v^{(t+1)})$$

Then define

$$\theta = \begin{cases} 1 & \text{if } z^T y \geqslant 0.2 z^T \mathbf{H}^{(t)} z \\ \dfrac{0.8 z^T \mathbf{H}^{(t)} z}{z^T \mathbf{H}^{(t)} z - z^T y} & \text{otherwise} \end{cases} \tag{10.9}$$

and calculate

$$w = \theta y + (1 - \theta)\mathbf{H}^{(t)} z \tag{10.10}$$

Finally, this value of w is used in the BFGS updating formula,

$$\mathbf{H}^{(t+1)} = \mathbf{H}^{(t)} - \frac{\mathbf{H}^{(t)} z z^T \mathbf{H}^{(t)}}{z^T \mathbf{H}^{(t)} z^T} + \frac{w w^T}{z^T w} \tag{10.11}$$

Note that the numerical value 0.2 is selected empirically and that the normal BFGS update is usually stated in terms of y rather than w.

On the basis of empirical testing, Powell proposed that the step-size procedure be carried out using the penalty function

$$P(x, \mu, \sigma) = f(x) + \sum_{k=1}^{K} \mu_k |h_k(x)| - \sum_{j=1}^{J} \sigma_j \min(0, g_j(x)) \tag{10.12}$$

where for the first iteration

$$\mu_k = |v_k| \qquad \sigma_j = |u_j|$$

and for all subsequent iterations t

$$\mu_k^{(t)} = \max\{|v_k^{(t)}|, \tfrac{1}{2}(\mu_k^{(t-1)} + |v_k^{(t)}|)\} \tag{10.13}$$

$$\sigma_j^{(t)} = \max\{|u_j^{(t)}|, \tfrac{1}{2}(\sigma_j^{(t+1)} + |u_j^{(t)}|)\} \tag{10.14}$$

The line search could be carried out by selecting the largest value of α , 0 <= α <=1, such that

$$P(x(\alpha)) < P(x(0)) \tag{10.15}$$

However, Powell prefers the use of quadratic interpolation to generate a sequence of values of k until the more conservative condition

$$P(x(\alpha_k)) \leq P(x(0)) + 0.1\alpha_k \frac{dP}{d\alpha}(x(0)) \qquad (10.16)$$

is met. It is interesting to note, however, that examples have been found for which the use of Powell's heuristics can lead to failure to converge. Further refinements of the step-size procedure have been reported, but these details are beyond the scope of the present treatment. We illustrate the use of a variant of the constrained variable metric (CVM) method using update (10.11), penalty function (10.12), and a simple quadratic interpolation-based step-size procedure.

<u>Variable metric methods for Constrained optimization:-</u>

<u>Example</u>

Solve the problem

$$\begin{aligned} \text{Minimize} \quad & f(x) = 6x_1x_2^{-1} + x_2x_1^{-2} \\ \text{Subject to} \quad & h(x) = x_1x_2 - 2 = 0 \\ & g(x) = x_1 + x_2 - 1 \geq 0 \end{aligned}$$

using the CVM method with initial metric $H^0 = I$.

At the initial point (2, 1), the function gradients are

$$\nabla f = (\tfrac{23}{4}, -\tfrac{47}{4})^T \qquad \nabla h = (1, 2)^T \qquad \nabla g = (1, 1)^T$$

Therefore, the first subproblem will take the form

$$\begin{aligned} \text{Minimize} \quad & (\tfrac{23}{4}, -\tfrac{47}{4})d + \tfrac{1}{2}d^T\mathbf{I}d \\ \text{Subject to} \quad & (1, 2)d = 0 \\ & (1, 1)d + 2 \geq 0 \end{aligned}$$

It is easy to show that the problem solution lies at the intersection of the two constraints. Thus, $d^0 = (-4, 2)^T$, and the multipliers at this point are solutions of the system

$$\left\{\begin{pmatrix} \frac{23}{4} \\ -\frac{47}{4} \end{pmatrix} + \begin{pmatrix} -4 \\ 2 \end{pmatrix}\right\} = v\begin{pmatrix} 1 \\ 2 \end{pmatrix} + u\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

or

$$v^{(1)} = -\tfrac{46}{4},\ u^{(1)} = \tfrac{53}{4}.$$

For the first iteration, we use the penalty parameters

$$\mu^{(1)} = |-\tfrac{46}{4}| \qquad \text{and} \qquad \sigma^{(1)} = |\tfrac{53}{4}|$$

The penalty function thus take the form

$$P = 6x_1x_2^{-1} + x_2x_1^{-2} + \tfrac{46}{4}|x_1x_2 - 2| - \tfrac{53}{4}\min(0, x_1 + x_2 - 1)$$

We now conduct a one-parameter search of P on the line $x = (2, 1)^T + \alpha(-4, 2)^T$. At $a = 0$, $P(0) = 12.25$. Suppose we conduct a bracketing search with $\Delta = 0.1$. Then

P(0 + 0.1) = 9.38875 and p(0.1 + 2(0.1)) = 13.78

Clearly, the minimum on the line has been bounded. Using quadratic interpolation on the three trial points of α = 0, 0.1, 0.3, we obtain α dash = 0.1348 with P(α) = 9.1702. Since this is a reasonable improvement over P(0), the search is terminated with this value of α. The new point is

$$X^{(1)} = (2, 1)^T + (0.1348)(-4, 2) = (1.46051, 1.26974)$$

We now must proceed to update the matrix H. Following Powell, we calculate

$$z = x^{(1)} - x^0 \ (-0.53949, 0.26974)^T$$

Then

$$\nabla_x L(x^0, u^{(1)}, v^{(1)}) = \begin{pmatrix} \frac{23}{4} \\ -\frac{47}{4} \end{pmatrix} - \left(-\frac{46}{4}\right)\begin{pmatrix} 1 \\ 2 \end{pmatrix} - \left(\frac{53}{4}\right)\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ -2 \end{pmatrix}$$

$$\nabla_x L(x^{(1)}, u^{(1)}, v^{(1)}) = \begin{pmatrix} 3.91022 \\ -4.96650 \end{pmatrix} - \left(-\frac{46}{4}\right)\begin{pmatrix} 1.26974 \\ 1.46051 \end{pmatrix} - \frac{53}{4}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 5.26228 \\ 4.81563 \end{pmatrix}$$

Note that both gradients are calculated using the same multiplier values $u^{(1)}$,$v^{(1)}$. By definition,

$$y = \nabla_x L(x^{(1)}) - \nabla_x L(x^0) = (1.26228, 6.81563)^T$$

Next, we check condition (10.9):

$$z^T y = 1.15749 > 0.2z^T \mathbf{I} z = 0.2(0.3638)$$

Therefore, $\theta = 1$ and $w = y$. Using (10.11), the update $\mathbf{H}^{(1)}$ is

$$\mathbf{H}^{(1)} = \mathbf{I} - \frac{zz^{\mathrm{T}}}{\|z\|^2} + \frac{yy^{\mathrm{T}}}{z^{\mathrm{T}}y}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - (0.3638)^{-1}\begin{pmatrix} -0.53949 \\ +0.26974 \end{pmatrix}(-0.53949,\ 0.26974)$$

$$+ (1.15749)^{-1}\begin{pmatrix} 1.26228 \\ 1.26228 \end{pmatrix}(1.26228,\ 1.26228)$$

$$= \begin{pmatrix} 1.57656 & 7.83267 \\ 7.83267 & 40.9324 \end{pmatrix}$$

Note that $H^{(1)}$ is positive definite.

This completes on iteration. We will carry out the second in abbreviated form only.

The subproblem at x(1) is

$$\text{Minimize} \quad (3.91022,\ -4.96650)d + \tfrac{1}{2}d^{\mathrm{T}}\begin{pmatrix} 1.57656 & 7.83267 \\ 7.83267 & 40.9324 \end{pmatrix}d$$

$$\text{Subject to} \quad (1.26974,\ 1.46051)d - 0.14552 = 0$$

$$d_1 + d_2 + 1.73026 \geqslant 0$$

The solution of the quadratic program is

$$d^{(1)} = (\ -0.28911,\ 0.35098)$$

At this solution, the inequality is loose, and hence $u^{(2)} = 0$. The other multiplier value is $v^{(2)} = 4.8857$.

The penalty function multipliers are updated using (10.13) and (10.14):

$$\mu^{(2)} = \max\left(|4.8857|,\ \frac{46}{4} + \frac{4.8857}{2}\right) = 8.19284$$

$$\sigma^{(2)} = \max\left(|0|,\ \frac{\frac{53}{4} + 0}{2}\right) = 6.625$$

The penalty function now becomes

$$P(x(\alpha)) = f(x) + 8.19284|x_1x_2 - 2| - 6.625\min(0,\ x_1 + x_2 - 1)$$

where

$$x(\alpha) = x^{(1)} + \alpha d^{(1)} \qquad 0 \leqslant \alpha \leqslant 1$$

$$x(\alpha) = x^{(1)} + \alpha d^{(1)} \qquad 0 \leqslant \alpha \leqslant 1$$

At $\alpha = 0$, $P(0) = 8.68896$, and the minimum occurs at $\alpha = 1$, $P(1) = 6.34906$. The new point is

$$x^{(2)} = (1.17141, 1.62073)$$

with

$$f(x^{(2)}) = 5.5177 \qquad \text{and} \qquad h(x^{(2)}) = -0.10147$$

The iterations continue with an update of $\mathbf{H}^{(1)}$ .The details will not be elaborated since they are repetitious. The results of the next four iterations are summarized below.

Iteration	x_1	x_2	f	h	v
3	1.14602	1.74619	5.2674	0.001271	−0.13036
4	1.04158	1.90479	5.03668	−0.01603	−0.17090
5	0.99886	1.99828	5.00200	−0.003994	−0.45151
6	1.00007	1.99986	5.00000	-1.9×10^{-6}	−0.50128

Recall that in Example 10.3, in which analytical second derivatives were used to formulate the QP subproblem, comparable solution accuracy was attained in four iterations. Thus, the quasi-Newton result obtained using only first derivatives is quite satisfactory, especially in view of the fact that the line searches were all carried out only approximately.

It should be reemphasized that the available convergence results (super linear rate) [6, 11] assume that the penalty function parameters remain unchanged and that exact line searches are used. Powell's modifications (10.13) and (10.14) and the use of approximate searches thus amount to useful heuristics justified solely by numerical experimentation.

Finally, it is noteworthy that an alternative formulation of the QP subproblem has been reported by Biggs as early as 1972. The primary differences of that approach lie in the use of an active constraint strategy to select the inequality constraints that are linearized and the fact that the quadratic approximation appearing in the subproblem is that of a penalty function. In view of the overall similarity of that approach to the Legrangian-based construction, we offer no elaboration here, but instead invite the interested reader to study the recent exposition of this approach offered in reference 13 and the references cited therein. It is of interest that as reported by Bartholomew Biggs, a code implementing this approach (OPRQ) has been quite successful in solving a number of practical problems, including one with as many as 79 variables and 190 constraints. This gives further support to the still sparse but growing body of empirical evidence suggesting the power of CVM approaches.