# MS002: Management Science and Business Modelling

**PUBLISHED BY**

Open Education

Dublin City University

**COURSE WRITERS**

Professor Harold Harrison M.A. M.A.I. Ph.D., F.I.S

Meriel Huggard B.Sc, M.Sc.

Ms Mary Sharp B.Sc., M.Sc.

Mr Donncha Ryan, BSs MSc; MA; MSc.

Dr Alexander O'Connor, B.A.I., M.A., M.Sc., Pg.Dip., PhD

**EDITOR**

Adrian Redmond B.Sc., M.Sc. (Mathematics – U.C.D.);
M.Sc. (Statistics & Operational Research – T.C.D.)



**PUBLICATION DATE:**

2018

# CONTENTS: MS002 - Management Science and Business Modelling

## Unit 10: Decision Analysis



## Unit 11: Markov Processes and Applications



## Unit 12: Introduction to Machine Learning



## Unit 13: Supervised Learning



## Unit 14: Unsupervised Learning

# UNIT 1
## INTRODUCTION TO LINEAR PROGRAMMING AND ASSOCIATED BUSINESS MODELS

### AIM
The aim of this unit is to introduce the concept of mathematical modelling. In particular, it will show how linear mathematical models can be applied in industrial, commercial and economic situations.

### OBJECTIVES
At the end of this unit you will:

- Understand the concept of a mathematical model.
- Be able to construct linear mathematical models of industrial and other systems.
- Appreciate the different ways that these models may be solved.

### REQUIRED READING
Students should read the following section of the mandatory text book (David R. Anderson, Dennis J. Sweeney, Thomas A. Williams and Kipp Martin, *An Introduction to Management Science: Quantitative Approaches to Decision Making*, 13th International Edition, Cengage Learning, ISBN-13: 9780538475655) in conjunction with this unit: Chapter 1, paying particular attention to the solution of problems by computer using both the software that comes with the book and also Excel. Excel is more powerful and is readily available to most people so is the preferred choice throughout the course.

### 1.1 THE ORIGINS OF MATHEMATICAL MODELS IN DECISION MAKING
Since ancient times mathematics has been used to provide a better understanding of the forces and workings of nature. The grandfather of applied mathematics, Archimedes, used the famous phrase "Eureka: I have found it" when he discovered the *Principle of Flotation*. From this he deduced that the crown of King Heron II of Syracuse was not made of pure gold as had been claimed by its maker. Archimedes also used his *Principle of the Lever* to construct huge catapults. These were used to launch enormous rocks at the invading warships of the Roman General Marcellus**.** Indeed, it is claimed that Archimedes discovered Integral Calculus some two thousand years before Newton – but is unclear whether he used this powerful tool in his investigations of nature.

The Irishman Robert Boyle discovered the law of gases in the 17th Century. This is named after him as Boyle's Law and is usually given as:

$$PV = C$$

Where P = Pressure; V = Volume; C = Constant.

In the early 17th Century Isaac Newton discovered the Universal Law of Gravity. This is usually formulated as the relationship:

$$F = G\frac{m_1 \times m_2}{d^2}$$

Where F is the force of attraction; $m_1$ and $m_2$ are the masses of two objects and d is the distance between them. The constant of proportionality, G, is known as the universal gravitational constant. It is termed a "universal constant" because it is thought to be the same at all places and all times, and thus universally characterizes the intrinsic strength of the gravitational force. Hence, the Universal Law of Gravity tells us that the force of attraction between two objects is proportional to the product of their masses divided by the square of the distance between them. With this "**mathematical model**" scientists could compute the orbits of planets, the paths of comets and indeed the critical velocity or the velocity of escape from the surface of the earth. This escape velocity can be expressed as:

$$v > \sqrt{2gr}$$

Where:

r is the earth's radius
g is the acceleration at earth's surface

Thus $v$ can be computed to be $\geq$ 11.2 kilometres per second

or

40032 kilometres per hour.

Coming closer to our own time, in the early 1900s Albert Einstein postulated his *Special Theory of Relativity* and then the *General Theory of Relativity.* These theories yielded the famous relationship:

$E = mc^2$

where E is the energy available from mass m and c is the speed of light. Through this equation, scientists realized the enormous energy contained in the atom and the atomic age was born.

Whilst scientists and mathematicians were using *Quantitative Tools* to investigate nature, others, including engineers and scientists, began in the early 1900s, to apply the quantitative scientific method to industrial management. People such as F.W Taylor and Frank and Lillian Gilbreath were pioneers in this field; creating the concepts of *Work Standards*, *Job Design* and *Time and Motion Study*. H.L Gantt produced the Gantt Chart which was the forerunner for some of today's most widely used scheduling techniques.

The rate of application of the quantitative methods of science to industrial and management problems has increased dramatically with the passage of time. As in many other fields, World War II acted as a strong catalyst to the use of the scientific method through the development of *Operational Research* in England. The British Government brought together groups of scientists from various disciplines to apply the scientific method to military problems. Two such problems were the U-boat threat to allied shipping in the north Atlantic and the deployment of radar installations that resulted in the victory of the Battle of Britain. The American armed forces also successfully adopted the scientific approach to tackle some of the problems they were facing in World War II

## 1.2 THE EMERGENCE OF LINEAR MATHEMATICAL MODELS IN BUSINESS, COMMERCE AND ECONOMICS

Once the Second World War was over, the Military Operational Research teams were disbanded and their members entered peace-time business, industry and government. These people, trained in the scientific method, began to analyse the problems they faced in a logical way and identified a structured Problem and Solving Process.

This **Problem and Solving Process** has the following steps:

1. Identify and define the problem.
2. Identify the alternatives.
3. Determine the criteria that will be used to evaluate the alternatives.
4. Evaluate the alternatives.
5. Select an alternative.
6. Implement this selected alternative.
7. Evaluate the results to determine if a satisfactory solution has been achieved.

As problems were being addressed and solved by this structured approach two further important developments were taking place. Firstly, the increasing use of **quantitative methods** in the problem-solving process led to increased research into the ways that such systems of equations (usually linear equations) could be solved in an efficient manner. In 1947, George B. Dantzig discovered the ***Simplex Method*** which addressed the problem of maximising or minimising a linear equation subjected to linear constraints. Dantzig's approach was more efficient then the method of Lagrangian Multipliers, a calculus based procedure, developed by the French Mathematician, Lagrange (1736–1813). At the same time, the continued development of the digital computer resulted in a massive increase in computing power. Dantzig's Simplex Method was soon programmed for these (1950s) digital computers, allowing 'Operational Researchers' to successfully solve many types of industrial, commercial and economic problems. The advance in computing power continues to this day: desktop computers are now more powerful than the mainframe computers of the 1970s and 1980s and are being used to solve problems larger than those that existed 20 and 30 years ago.

## 1.3 TYPICAL APPLICATIONS OF LINEAR MODELS IN THE MANAGEMENT PROCESS

The formulation and solution of linear models to assist in the Management Process has become one of the most successful of the quantitative approaches to decision making. Very many applications have been reported in a wide variety of industrial and commercial activities and include:

- Blending problems in the animal food, fertilizer and chemical industries
- Scheduling problems in the airline, manufacturing and financial service industries.
- Logistical and routing problems in the transportation industry.
- Planning and management control problems in the building and construction industries.

A complete book could be written on all of these applications. The **objective** of all of these linear models is either to **maximise** or **minimize** such measures as profitability or cost. Therefore, we will limit the description (at this point) of applications to two, namely a profit maximization and a cost minimization model.

## 1.4 Fundamental Properties of Linear Programming Models

Irrespective of whether the model is to maximize (profit) or minimize (cost), the models considered will have the following characteristics:

- **An Objective**
  To maximize or minimize a linear equation (which, in our case, may describe profit or cost).

- **Constraints**
  Linear equations which describe inputs to the process and which constrain the "value" of the objective.

Whereas the structure of the **Objective** function (or equation) is relatively straightforward, setting up the **Constraint** equations requires careful thought.

Constraint equations have two distinct parts, the **left-hand side** (LHS) and the **right-hand side** (RHS). Whereas the LHS is comprised of variables ($x_1$, $x_2$, $x_3$ etc) and their coefficients ($a_1$, $a_2$, $a_3$ etc) the RHS are composed of **numbers** typically describing available amounts of consumables.

In mathematical terms these consumables usually exhibit one of the three characteristics namely:

1. **At Most** or **Upper Bound** $\leq$

2. **At Least** or **Lower Bound** $\geq$

3. **Exactly Equal to** $=$

These relationships can be expressed mathematically as inequalities or equalities:

1. **At Most:** $\mathbf{g}_1(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m) \leq \mathbf{U}$

2. **At Least:** $\mathbf{g}_2(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m) \geq \mathbf{L}$

3. **Exactly Equal to:** $\mathbf{g}_3(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m) = \mathbf{E}$

Where U, L and E are finite numbers and $g_1(x_1, x_2, \ldots, x_m)$, $g_2(x_1, x_2, \ldots, x_m)$ and $g_3(x_1, x_2, \ldots, x_m)$ are **linear functions**. Linear functions have the form

$g_i(x_1, x_2, \ldots, x_m) = c_1x_1 + c_2x_2 + \ldots + c_mx_m$

where $x_1, x_2, \ldots, x_m$ are a set of *m* distinct variables and $c_1, c_2, \ldots, c_m$ are constants.

Note that:

i. A linear function of *m* variables consists of *m* terms, one for each variable.

ii. Each term has a coefficient which can have a positive, negative or zero value (the latter would, in fact, reduce the number of terms in the function since any variable with a zero coefficient could be omitted).

iii. Each variable is not raised to a power higher than 1.

iv. A variable cannot be multiplied by another variable.

A linear model representing real world situations will typically contain combinations of inequalities and equalities. It is important to make sure that the correct relationships are used. The simplest mistake, and one often made, is to assume that an **Equality** relationship is the one that best describes a given constraint. The importance of the form of the relationships will be explored in Unit 2.

Finally, the overall structure of a linear model can be generally described as:

Max / Min: $Z = f(x_1, x_2, \ldots, x_m)$

Subject to: $g_i(x_1, x_2, \ldots, x_m) \leq$ (or =, or $\geq$) $B_i$

$(x_1, x_2, \ldots, x_m) \geq 0$

where i = 1,2, …, n (n is the number of constraints on the system) and $B_i$ is the boundary value.

## 1.5 MODEL FORMULATION - MAXIMIZATION PROBLEMS

To illustrate the formulation of a maximization problem let us consider the following example:

The Superb Football Company makes two different types of football: the Standard Model and the Superior Model. The company has 1800 hours of production time available in its cutting and sewing department, 600 hours in its polishing and finishing section and 200 hours in its packaging and distribution division. Production time per unit and profitability per football type are given in the following table:

| FOOTBALL TYPE | UNIT PRODUCTION TIME (HRS) | | | UNIT PROFIT PER TYPE |
|---|---|---|---|---|
| | CUTTING AND SEWING | POLISHING AND FINISHING | PACKAGING AND DISTRIBUTION | |
| STANDARD MODEL | 2 | 1 | 1/4 | €8 |
| SUPERIOR MODEL | 3 | 2/3 | 1/2 | €12 |
| HOURS AVAILABLE | 1800 | 600 | 200 | |

What is the linear programming model whose solution will result in the production plan that maximises company profitability?

To answer the above question, we proceed as follows:

**Step I**

Let $x_1$ = the number of Standard Models produced.

Let $x_2$ = the number of Superior Models produced

**Step II**

The objective equation or function can be stated as:

Max Profit: $Z = 8x_1 + 12x_2$

**Step III**

The constraints are constructed as follows:

CUTTING and SEWING

$2x_1 + 3x_2 \leq 1800$

POLISHING and FINISHING

$1x_1 + 2/3x_2 \leq 600$

PACKAGING and DISTRIBUTION

$¼\, x_1 + ½\, x_2 \leq 200$

Notice that in each case the relationship between LHS and RHS is $\leq$, indicating an **upper bound** as there are only 1800, 600 and 200 hours available respectively in each production department.

**Step IV**

It is a trivial but important point to realize that we cannot produce **negative** numbers of footballs. Therefore, we impose what are known as the non-negativity conditions:

$x_1 \geq 0, x_2 \geq 0$

**Step V**
The complete model of the production process is stated as:

Max Profit: $Z = 8x_1 + 12x_2$

Subject to:

$$2x_1 + 3x_2 \leq 1800$$
$$1\,x_1 + 2/3x_2 \leq 600$$
$$¼\,x_1 + ½x_2 \leq 200$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

Notice that this **linear programming model** has only two variables. This means that it can be displayed and solved graphically. The methodology for doing this will be described in detail in Unit 2.

▶ See video **V01_LPForm_1**, which presents a scenario involving the manufacture of loudspeakers and then formulates it as a linear programming problem. https://goo.gl/iZ0rN4

▶ See video **V02_LPForm_2**, which provides a linear programming formulation for a farmer planting barley and wheat. https://goo.gl/XEd8HZ

## *SAQ 1*

*A manufacturer of garden furniture wants to determine what quantities of the two lines that they manufacture will optimise daily profitability from their industrial standards division. The managing director received the following information:*

| *Product Line* | *Time Required to Process Each Line (HRS)* | | *Unit Profit Contribution* |
|---|---|---|---|
| | *Assembly* | *Finishing* | |
| *Reclining Chairs* | *4* | *5* | *€25* |
| *Cocktail Table* | *7* | *3* | *€30* |
| *Daily Hours Available* | *60* | *40* | |

*The managing director wants you to construct the appropriate mathematical model of this situation.*

## *SAQ 2*

*An engineering company produces three different products. The labour resources required and materials consumed together with the project per unit for each of the products are shown in the following table:*

| | Resources Usec | | Profit |
|---|---|---|---|
| Product | (hrs/ Unit) | (kg/Unit) | (€/Unit) |
| A | 5 | 4 | 30 |
| B | 2 | 6 | 50 |
| C | 4 | 3 | 20 |

*There are 240 hours of daily labour available for production materials supply is limited to 400 kg per day. What quantity of each product should the company produce in order to maximise total daily profit?*

*Construct an appropriate mathematical model.*

## *SAQ 3*

*A company produces two types of hats. Each hat of the first type requires twice as much labour time as does each hat of the second type. If all hats are of the second type only, the company can produce a total of 500 hats a day. The market limits daily sales of the first and second types to 150 and 200 hats. Assume that the profit per hat is €8 for type 1 and €5 for type 2.*

*Formulate a linear program to determine the number of hats of each type the company should produce and sell each day so as to maximise daily revenue.*

## *SAQ 4*

*A company can advertise its product by using local radio and TV stations. Its budget limits the advertising expenditure to €1000 a month. Each minute of radio advertising costs €5, and each minute of TV advertising costs €100. The company would like to use the radio at least twice as much as TV. Past experience shows that each minute of TV advertising will generate 25 times as many sales as each minute of radio advertising.*

*Formulate a linear program to determine how the company should optimally allocate its monthly budget to radio and TV advertising.*

## 1.6 MODEL FORMULATION - MINIMIZATION PROBLEMS

We will now illustrate the formulation of a minimization problem through the medium of an example:

Cats and Co mix together ingredients for a top-selling cat food. One of the selling points of the cat food is its "excellent balance between protein and fat at an affordable price".

The firm's chemists have recommended that the cat food must contain at least 0.3 kg of fat per day and at least 0.5 kg of dairy protein. The costs and chemical analysis of the raw materials mixed to form the cat food are as follows:

| RAW MATERIALS | FAT (%) | PROTEIN (%) | COST PER kg |
| --- | --- | --- | --- |
| FISH CUBES | 15 | 30 | €6 |
| AVIAN CONCENTRATE | 30 | 20 | €5 |

Again, we want to formulate the linear programming model whose solution will result in the cat food that meets the fat/protein requirements at minimum cost to Cats and Co.

**Step I**

Let $y_1$ = the weight in kilograms of Fish Cubes in the mix, and

$y_2$ = the weight in kilograms of Avian Concentrate in the mix.

**Step II**

The objective function can be stated as:

Minimise Cost $Z = 6y_1 + 5y_2$

**Step III**

The constraints can be constructed as follows:

FAT: $0.15y_1 + 0.3y_2 \geq 0.3$

PROTEIN: $0.3y_1 + 0.2y_2 \geq 0.5$

Notice that in each case the relationship between LHS and RHS is $\geq$, indicating a **lower bound** since at least 0.3 kg and 0.5kg of daily fat and protein are required.

As these constraints involve proportions (i.e. 0.3 and 0.5) it is best to multiply each equation by a factor of 100 to make them easier to work with. The constraint equations then become:

FAT: $15y_1 + 30y_2 \geq 30$

PROTEIN: $30y_1 + 20y_2 \geq 50$

**Step IV**
Again we require the non-negativity conditions

$$y_1 \geq 0, \qquad y_2 \geq 0$$

**Step V**
The complete model for the blending of the new materials of the cat food problem can now be stated as:

Minimise Cost: $Z = 6y_1 + 5y_2$

Subject to: $15y_1 + 30y_2 \geq 30$

$30y_1 + 20y_2 \geq 50$

$y_1 \geq 0$

$y_2 \geq 0$

This **Linear Programming Model** has only two variables and so it can be displayed and solved graphically. The procedures for doing this will be described in Unit 2.

▶ See video **V03_LPForm_3**, which provides a linear programming formulation for minimising the cost of animal feed mix. https://goo.gl/b6vmdg

## *SAQ 5*

*The Animal Feed Company (AFC) makes feed for farm animals by using two types of grain as the summary ingredients in its feed mixers. By mixing these grains together in various proportions feeds are produced that meet required nutritional requirements. Whilst wishing to meet these requirements, AFC requires to do so as cheaply as possible. The following table presents the necessary data:*

| *Primary Ingredient* | *Nutrient Type* | | | *Purchase Price / Unit* |
|---|---|---|---|---|
| | *A* | *B* | *C* | |
| *Grain I* | *1* | *1* | *1/3* | *€12* |
| *Grain II* | *½* | *1* | *1* | *€8* |
| *Nutrient Requirements* | *AT LEAST 10 UNITS* | *NOT FEWER THAN 14 UNITS* | *A LOWER BOUND OF 6 UNITS* | |

*What is the appropriate mathematical model for AFC?*

## *SAQ 6*

*A coffee shop, which also serves breakfast, wishes to ensure that minimum daily requirements (mdr) for vitamins A and B are met whilst their breakfasts are prepared at the lowest possible cost. The breakfast ingredients providing the vitamins are eggs, bacon and cereal. The vitamin data and cost of the breakfast ingredients are:*

| | | VITAMIN CONTRIBUTION | | |
|---|---|---|---|---|
| VITAMIN | mg/egg | mg/Bacon Strip | mg/cereal | mdr (mg) |
| A | 2 | 4 | 1 | 16 |
| B | 3 | 2 | 1 | 12 |
| Cost | 40c/egg | 30c/strip | 20c/cup | |

*How much of each ingredient should be served to meet the minimum daily requirements whilst minimising the cost of the breakfast?*

*Construct an appropriate mathematical model.*

## *SAQ 7*

*A company manufactures two types of radios. The only scarce resource that is needed to produce the radios is labour. At present, the company employs two technicians. Technician I is willing to work up to 40 hours per week and is paid €5 per hour. Technician II is willing to work up to 50 hours per week and is paid €6 per hour. The selling price received by the company for each radio type as well as the resources required to build each type of radio are given in the following table:*

| | ***Radio 1*** | ***Radio 2*** |
|---|---|---|
| *Selling Price per unit* | *€25* | *€22* |
| *Required Labour Hours of Technician I for each unit of production* | *1 hour* | *2 hours* |
| *Required Labour Hours of Technician II for each unit of production* | *2 hours* | *1 hour* |
| *Raw material cost for each unit of production* | *€5* | *€4* |

*The company wishes to determine the weekly production schedule that will maximise weekly profit. It can be assumed that the company can sell all of its radio production at the selling prices quoted above.*

*Formulate the above problem as a linear program.*

## *SAQ 8*

This type of problem will be formulated in a later unit as a transportation type problem.
*A national beer company has breweries in three cities that can supply the following quantities of draft beer to its distributors each month:*

| *Brewery* | *Monthly Supply (barrels)* |
|---|---|
| *A* | *2,000* |
| *B* | *1,600* |
| *C* | *2,400* |

*These breweries supply draft beer to distributors in 4 regional centres, which have the following monthly demand:*

| *Distributor* | *Monthly Demand (barrels)* |
|---|---|
| *I* | *800* |
| *II* | *2,200* |
| *III* | *1,800* |
| *IV700* | |

*The following distribution costs per barrel have been determined:*

| | | *TO* | | | |
|---|---|---|---|---|---|
| | | *I* | *II* | *III* | *IV* |
| | *A* | *50p* | *40* | *28* | *30* |
| *FROM* | *B* | *26* | *16* | *12* | *18* |
| | *C* | *22* | *27* | *31* | *24* |

*The company wishes to determine the optimal distribution plan from breweries to distributors that will minimise the monthly transport cost.*

*Formulate the above problem as a linear programming problem.*

### *SAQ 9*

*Four products are produced on two machines. The manufacturing times in hours per unit of each product are tabulated for the two machines:*

*Production Time (Hours) per Unit*

| *Machine* | *Product 1* | *Product 2* | *Product 3* | *Product 4* |
|---|---|---|---|---|
| *1* | *2* | *3* | *4* | *2* |
| *2* | *3* | *2* | *1* | *2* |

*The total cost of producing 1 unit of each product is based directly on machine time. Assume that the cost per hour of machines 1 and 2 is €10 and €15, respectively. Over the planning horizon, the total hours available on machine 1 are 500 hours. 380 hours are available for machine 2. The sales price per unit for products 1, 2, 3 and 4 are €73, €70, €70 and €56. Management wishes to determine the production schedule that will maximise total net profit over the planning horizon.*

*Formulate the above problem as a linear programming problem.*

### *SAQ 10*

*The Outdoor Furniture Corporation manufactures two products, benches and picnic tables, for use in yards and parks. The firm has two main resources: its carpenters (labour force) and a supply of redwood for use in the furniture. During the next production cycle, 1,200 hours of man power are available under a union agreement. The firm also has a stock of 3,500lbs of quality redwood. Each bench that Outdoor Furniture produces requires 4 labour hours and 10lbs of redwood; each picnic table takes 6 labour hours and 35lbs of redwood. Completed benches will yield a profit of €9 each, and tables will result in a profit of €20 each. How many benches and tables should Outdoor Furniture produce in order to obtain the largest possible profit?*

*Formulate and solve the above problem as a linear programming problem.*

### SUMMARY

In this unit we are introduced to the notion that many real world problems can be formulated as mathematical models and subsequently solved using various solution techniques ranging from specialised algorithms to graphical and software solutions. A wide class of such problems is that of **linear programming** models where **three main ingredients** are identified. These are:

- **A set of decision variables** (unknowns needing to be solved for) which represent quantities whose values are typically under our control and whose values may be varied to achieve 'best results'.
- **An objective function** which encapsulates/represents mathematically the single objective/criterion being optimised. This may represent for example a profit function which we wish to maximise or a cost function which we seek to minimise. Its defining feature however is that it is a **linear** function/expression of the decision variables.
- **A set of constraints** (equations representing bounded resources or limitations under which we must operate such as resource availability). Their defining feature however is that they are also **linear**

**functions**/expressions of the decision variables. These constraints will if properly expressed and accounted for, dictate where the solution may be found. This region bounded by the constraints is called the **solution space** and the optimal solution will be at one or more of the **corner**/intersection points.

### *SUMMARY OF STEPS*

Before a problem can be solved it must be formulated or represented in a solvable/mathematical format. The following steps are required:

- ❒ Identify the nature of the problem and assess if it is amenable and desirable to model its features for subsequent solution.
- ❒ Identify and label the various decision variables.
- ❒ Identify the criterion being addressed. This will be the objective function and an expression for it needs to be developed and it will be linear in the variables. Identify whether we are facing a maximisation or minimisation problem.
- ❒ Identify all the constraints being careful not to omit any and not have contradictory or infeasible combinations. A linear expression needs to be developed for each one and the correct form of the relevant inequality/equality is present. Usually non negativity conditions are also imposed on the decision variables.
- ❒ Before solution check that the formulated model is complete, makes sense and is properly annotated/explained so that other stakeholders can understand and communicate the structure and sense of the model to each other.
- ❒ Proceed to identify the solution approach which can be graphical if only two variables are involved or using a spreadsheet, specialised software or various algorithms if several variables are present.
- ❒ Once the solution is developed it can be interpreted, implemented, reviewed and various sensitivity/scenario analyses can be conducted. Solution reports can be generated and archived as required.

## FINAL COMMENTS

A problem cannot be solved if not firstly formulated properly. Recognising and formulating linear programming problems requires practise and care. As such the student should study some of the generic problem types to be found in most management science/Operational Research textbooks and web resources, including the reading list. The wealth of applications ranging from finance, portfolio analysis, capital rationing, product mix to logistics to project management confers a uniquely important position to linear and indeed non-linear programming problems. These problem types and their solvability with the aid of software make for a very powerful analysis platform for addressing real world and practical applications.

## ANSWERS TO SAQS

### *SAQ 1*

***Step I***

*Let $x_1$ = the number of reclining chairs produced.*

*Let $x_2$ = the number of cocktail tables produced.*

***Step II***

*The objective equation or function can be stated as:*

*Maximise Profit: $Z = 25x_1 + 30x_2$*

***Step III***

*The constraints are constructed as follows:*

***Assembly***

$4x_1 + 7x_2 \leq 60$

***Finishing***

$5x_1 + 3x_2 \leq 40$

*Notice that in each case the relationship between LHS and RHS is indicating between LHS and RHS is $\leq$, indicating an **upper bound** as there are only 60 and 40 hours available respectively in each production department.*

***Step IV***

*It is a trivial but important point to realize that we cannot produce **negative** numbers of footballs. We impose therefore what are known as the non-negativity conditions*

$$x_1 \geq 0, x_2 \geq 0$$

***Step V***

*The complete model of the production process is therefore stated as:*

*MAX $Z = 25x_1 + 30x_2$*

*SUBJECT TO:*

$4x_1 + 7x_2 \leq 60$

$5x_1 + 3x_2 \leq 40$

$x_1 \geq 0$

$x_2 \geq 0$

## *SAQ2*

***Step I***
*Let*
$x_1$ *= the quantity of produce A produced.*
$x_2$ *= the quantity of product B produced.*
$x_3$ *= the quantity of product C produced.*

***Step II***
*The objective equation or function can be stated as:*

*Maximise Profit:* $Z = 30x_1 + 50x_2 + 20x_3$

***Step III***
*The constraints are constructed as follows:*

***Labour***
$5x, + 2x_2 + 4x_3 \leq 240$

***Production materials***
$4x, + 6x_2 + 3x_3 \leq 400$

***Step IV***
*Cannot produce negative numbers of A, B or C, hence:*
$x_1 \geq 0$, $x_2 \geq 0$ *and* $x_3 \geq 0$

***Step V***
*The complete model of the production process is therefore stated as:*

*Maximise* $Z = 30x_1 + 50x_2 + 20x_3$

*Subject to:*
$5x, + 2x_2 + 4x_3 \leq 240$
$4x, + 6x_2 + 3x_3 \leq 400$
$x_1, x_2, x_3 \geq 0$

## *SAQ3*

***Step I***
*Let*
$x_1$ = *Daily production of hats of type 1.*
$x_2$ = *Daily production of hats of type 2.*

***Step II***
*The objective equation or function can be stated as:*
*Maximise Daily Revenue:* $Z = 8x_1 + 5x_2$

***Step III***
*The constraints are constructed as follows:*

$$2x_1 + x_2 \leq 500$$
$$x_1 \leq 150$$
$$x_2 \leq 200$$

***Step IV***
*Cannot produce negative numbers of hats hence:*
$x_1 \geq 0$ *and* $x_2 \geq 0$

***Step V***
*The complete model of the hat production process is therefore stated as:*

*Maximise Daily revenue:* $Z = 8x_1 + 5x_2$

*Subject to:*
$$2x_1 + x_2 \leq 500$$
$$x_1 \leq 150$$
$$x_2 \leq 200$$

$$x_1, x_2 \geq 0$$

## *SAQ 4*

*Let*
$x_1$ = *Minutes of radio per month*
$x_2$ = *Minutes of TV per month.*

*Maximise Advertising Impact:* $Z = x_1 + 25 x_2$

*Constraints:*
$5 x_1 + 100 x_2 \leq 1000$

$x_1 \geq 2 x_2$

*or*

$x_1 - 2 x_2 \geq 0$

*Full Advertising Budget Model:*

*Maximise* $Z = x_1 + 25 x_2$

*Subject to:*
$5 x_1 + 100 x_2 \leq 1000$
$x_1 - 2 x_2 \geq 0$

$x_1, x_2 \geq 0$

**Step I**
Let
$y_1$ = the number of units of grain 1 in the mix
$y_2$ = the number of units of grain 2 in the mix

**Step II**
The objective function can be stated as:

Minimum Cost: $Z = 12y_1 + 8y_2$

**Step III**
The constraints can be constructed as follows:

Nutrient type A: $y_1 + ½ y_2 \geq 10$

Nutrient type B: $y_1 + y_2 \geq 14$

Nutrient type C: $\frac{1}{3} y_1 + y_2 \geq 6$

**Step IV**
Again we require the non-negativity conditions
$y_1 \geq 0$ and $y_2 \geq 0$

**Step V**
The complete model for the production of farm animal feed problem can now be stated as:

Minimise Cost $Z = 12 y_1 + 8 y_2$

Subject to:

$y_1 + ½ y_2 \geq 10$

$y_1 + y_2 \geq 14$

$\frac{1}{3} y_1 + y_2 \geq 6$

$y_1 + y_2 \geq 6$

$y_1 \geq 0$

$y_2 \geq 0$

## *SAQ 6*

*Let*
$x_1$ *= the number of eggs served*
$x_2$ *= the number of bacon strips served*
$x_3$ *= the number of cup of cereal served.*

*Thus the model is:*

*Minimise Cost:* $Z = 40\ x_1 + 30x_2 + 20x_3$

*Subject to:*

$2\ x_1 + 4\ x_2 + 1\ x_3 \geq 16$
$3\ x, + 2\ x_2 + 1\ x_3 \geq 12$
$x_1, x_2, x_3 \geq 0$

## *SAQ 7*

*Let x represent the number of type 1 radios to manufacture and y the number of type 2 radios.*
*Labour cost for Radio 1 = 1 x 5 + 2 x 6 = 17.*
*Profit for Radio 1 = 25 – 17 – 5 = 3.*
*Labour cost for Radio 2 = 2 x 5 + 1 x 6 + 16.*
*Profit for Radio 2 = 22 – 16 – 4 = 2.*
*Objective is to maximise profit:*
*Maximise Z = 3x + 2y*

*Subject to:*

*x + 2y <= 40 Labour restriction for Technician 1*
*2x + y <= 50 Labour restriction for Technician 2*
*x, y >=0*

## *SAQ 8*

*Let Xi,j be the variables from Distributor A, B and C to Regional Centres I, II, III and IV*

*Objective is to minimise costs:*
$Min\ Z = 50x_{A,I} + 40x_{A,II} + 28\ x_{A,III} + 30x_{A,IV} + 26x_{B,I} + 16x_{B,II} + 12x_{B,III} + 18x_{B,IV} + 22x_{C,I} + 27x_{C,II} + 31x_{C,III} + 24x_{C,IV}$

*Subject to:*
*Constraints for supply*
$X_{A,I} + X_{A,II} + X_{A,III} + X_{A,IV} = 2{,}000$
$X_{B,I} + X_{B,II} + X_{B,III} + X_{B,IV} = 1{,}600$
$X_{C,I} + X_{C,II} + X_{C,III} + X_{C,IV} = 2{,}400$

*Constraints for demand:*
$X_{A,I} + X_{B,I} + X_{C,I} = 800$
$X_{A,II} + X_{B,III} + X_{C,II} = 2{,}200$
$X_{A,III} + X_{B,III} + X_{C,III} = 1{,}800$
$X_{A,IV} + X_{B,IV} + X_{C,IV} = 700$

*All Variables >=0*

## *SAQ 9*

*Let A, B, C, and D represent the numbers of units of the four products to be manufactured.*

*Profit = Sales price – Cost of production*

*Profit A = 73 – 20- 45 = 8*

*Profit B = 70 – 30 – 30 = 10*

*Profit C = 70 – 40 – 15 = 15*

*Profit D = 56 – 20 – 30 = 6*

*Objective is to maximise profit.*

$Max\ Z = 8A + 10B + 15C + 6D$

*Subject to constraints on hours available on machines.*

$2A + 3B + 4C + 2D <= 500$

$3A + 2B + 1C + 2D <= 380$

*All variables >= 0*

## *SAQ 10*

*Let A = number of benches to manufacture and B = number of tables to manufacture.*

*Objective is to maximise profit.*

*Max Z = 9A + 20B*

*Subject to the constraints which are labour and wood:*

*4A + 6B <= 1,200*

*10A + 35B <= 3,500*

*All variables >= 0*

# UNIT 2

## GRAPHICAL METHODS OF SOLUTION

### AIM
This unit describes how linear programmes can be depicted, and solved, using the graphical method.

### OBJECTIVES
At the end of this unit you will:

- Understand how to solve linear programmes by graphical methods.
- Be able to construct a graphical representation of a linear programme.
- Be able to construct the solution space and to appreciate its significance.
- Appreciate how the optimal solution can vary.
- Recognise the limitations of the graphical solution procedure.

### REQUIRED READING
Students should read the following sections of the mandatory text book in conjunction with this unit.

Chapter 2, 3 and 4.

### 2.1 A TYPICAL MAXIMIZATION PROBLEM
You will recall that in Unit 1 we constructed the following linear programming model:

$\text{Max } Z = 8x_1 + 12x_2$

Subject to:

$2x_1 + 3x_2 \leq 1800$

$x_1 + 2/3x_2 \leq 600$

$¼\, x_1 + ½\, x_2 \leq 200$

$x_1 \geq 0$

$x_2 \geq 0$

This is a typical maximization problem, which has the added "advantage" that all of the right hand sides are upper bounds. The significance of this "advantage" will become evident in section 2.6.

Since the model has only two variables ($x_1$ and $x_2$) it can be solved graphically. Note that if the model contained three variables ($x_1$, $x_2$, $x_3$) it could still be solved graphically but with greater difficulty, as will be demonstrated in section 2.10

## 2.2 CONSTRUCTION OF THE SOLUTION SPACE

The "Subject to" conditions give us the **solution space** - that is that area within which the solution to the model (if it exists) will be found. To create this solution space, we begin by constructing the $x_1$ and $x_2$ axes as depicted in Figure 2.1.

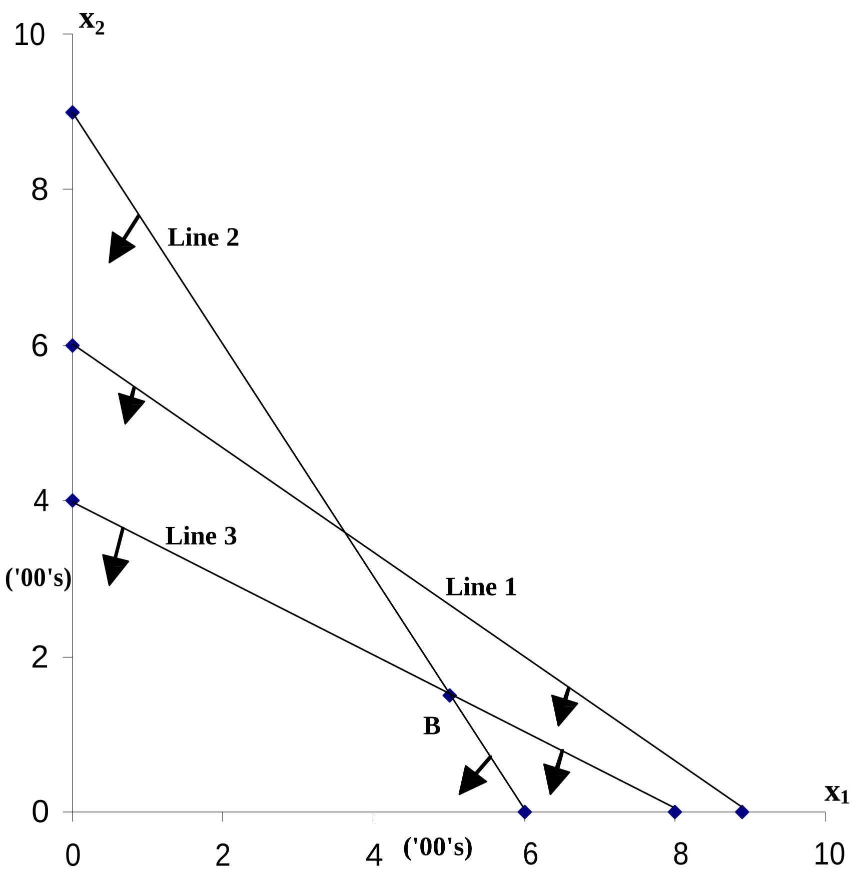



**Figure 2.1**
**Constructing the Solution Space**

Next we take the first constraint and replace $\leq$ with = to get:

**Step I** $2x_1 + 3x_2 = 1800$

We then find where this line crosses the $x_1$ and $x_2$ axes:

**Step II** Let $x_1 = 0$: Whence $x_2 = 600$

**Step III** Let $x_2 = 0$: Whence $x_1 = 900$

Finally, we use the two points found to construct the appropriate line on the graph:

**Step IV** With $x_1 = 900$ and $x_2 = 600$ construct **Line 1** as shown in **Figure 2.1**

We then repeat this procedure for the second constraint yielding **Line 2**, and for the third constraint, yielding **Line 3** as also shown in **Figure 2.1**.

Each of these lines represents the strict **equality (=)** relationship with the right-hand sides (RHS). It is important to realize that values along each line and underneath it satisfy the corresponding **inequality (≤)** relationship. The *arrows* under the lines represent the < part of the relationship. You will note that **lines** 1, 2, 3 intersect each other. We say that line 1 **dominates** line 2 up to their intersection point, since the line 1 arrow is below the line 2 arrow, but does not dominate line 2 after their intersection point. Note that line 3 dominates both line 1 and line 2 but does not dominate line 2 after their intersection at point B.

These rules of dominance result in the area OABC depicted in **Figure 2.2** which is known as the **Feasible Region or Solution Space.** That is, the area from which the solution to the model will be determined. It can be proved, mathematically, that the **Optimal Solution** (should it exist) will be found at the **Extreme Points** of the solution space OABC, in fact at either A, B or C since O is the origin where $x_1 = 0$ and $x_2 = 0$.

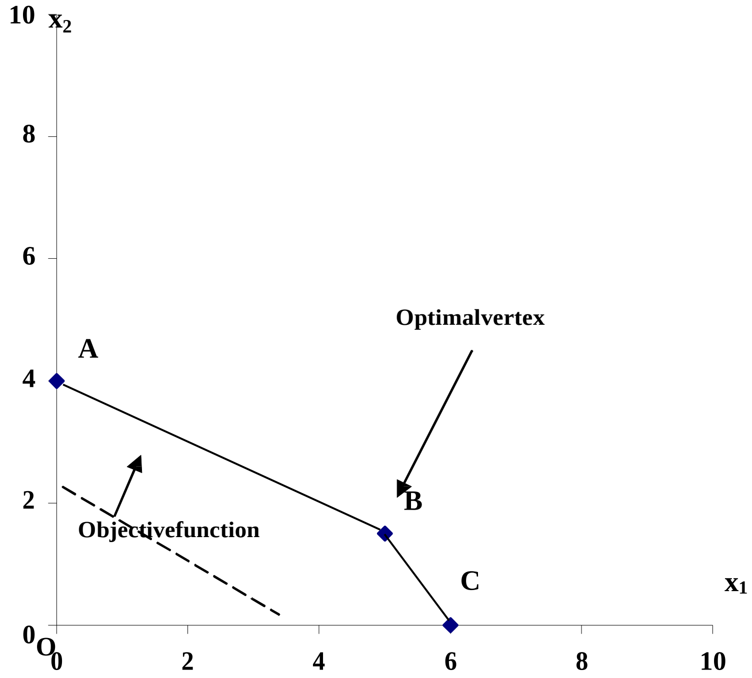



**Figure 2.2**
**The Solution Space**

## 2.3 CONSTRUCTION OF THE OBJECTIVE FUNCTION

From section 2.1 we see that the objective function equation is:

$8x_1 + 12x_2$

Since this equation has not got a RHS we allocate one to it in order to be able to depict it on a graph. Ideally we need to locate the objective function inside the solution space since this function is to be maximized. The coefficients of the objective function equation are 8 and 12, thus we can set the RHS to be 24 (divisible by both 8 and 12). By choosing a number that is divisible by both 8 and 12 we ensure that the points we use to draw this objective function line have integer (whole number) values.

We thus have

| | | |
|---|---|---|
| **Step I** | $8x_1 + 12x_2 = 24$ | |
| **Step II** | Let $x_1 = 0$: | Whence $x_2 = 2$ |
| **Step III** | Let $x_2 = 0$: | Whence $x_1 = 3$ |

Thus we can construct the **objective function line** shown in dashed format in **Figure 2.2**. It should be noted that had we chosen the RHS to be 32 then the objective function line would have been parallel to our existing one but further away from the origin (O). You should verify this yourself as an exercise. We call these parallel lines representing increasing values of the objective function **Iso-Function lines**. These iso-function lines are usually represented as dashed lines. If we move the line out until the entire feasible region except one point lies between the function line and the origin, we have the **optimum point**.

You will notice in **Figure 2.2** that the objective function line has an arrow pointing **away** from the origin. This indicates that to maximise the value of the objective function the line should be as far from the origin as it permitted by the solution space.

## 2.4 THE CHARACTERISTICS OF THE SOLUTION SPACE

In Section 2.2 it was stated that the optimal solution (should it exist) to the linear programming model will be found at the extreme points of the solution space, that is, vertices A, B or C. There are two ways of determining which vertex maximizes the value of the objective function namely.

- Complete Enumeration
- Maximum Distance

**Complete Enumeration** requires the objective function to the evaluated at each of the vertices and the maximum value (and vertex) selected. For instance:

| | |
|---|---|
| VERTEX A: $x_1 = 0$; $x_2 = 400$: | $8x_1 + 12x_2 = 4800$ |
| VERTEX C: $x_1 = 600$; $x_2 = 0$: | $8x_1 + 12x_2 = 4800$ |
| VERTEX B: $x_1 = 500$; $x_2 = 150$: | $8x_1 + 12x_2 = 5800$ |

Thus Vertex B maximizes the value of the objective function, and also therefore is the vertex furthest from the origin.

**Maximum Distance** achieves the same result but is easier and quicker. Simply place a straight object (pencil, pen, ruler etc) along the objective function line and slide it at that slope towards A, B and C. That vertex which it touches LAST before leaving the solution space is the vertex at maximum distance from the origin. On accurate graph paper it should be clear at what actual point the two constraints meet. However, because your graph may not always be sufficiently clear it is best to identify the appropriate intersecting constraints from the graph and then calculate the exact point by solving the corresponding simultaneous equations.

Vertex B therefore maximises the objective function and has the co-ordinates

$x_1$ =500;   $x_2$ = 150

and

Max P = 8(500) + 12(150)
= 5800

▶ See video **V04_LPGraph_1**, which uses the graphical method to solve the loudspeaker linear programming problem (presented in the video V01_LPForm_1). https://goo.gl/HqJoAT

▶ See video **V05_LPGraph_2**, which uses the graphical method to solve the farm linear programming problem (presented in video V02_LPForm_2). https://goo.gl/aKdD7c

## 2.5 Solving the Maximization Problem and the Interpretation of the Solution

Section 2.4 showed that the optimal vertex B maximized the value of the objective function and yielded the values of $x_1$ and $x_2$ as 500 and 150 respectively. If we substitute these values into the maximization model, there arises:

Max Z = $8x_1 + 12x_2 \Rightarrow 8(500) + 150\ (12) = 5800$

Subject to:

$2x_1 + 3x_2 \leq 1800 \Rightarrow 2(500) + 3(150) = 1450 < 1800$

$x_1 + 2/3x_2 \leq 600 \Rightarrow (500) + 2/3(150) = 600 \equiv 600$

$¼x_1 + ½x_2 \leq 200 \Rightarrow 1/4(500) + ½(150) = 200 \equiv 200$

A fundamental feature of linear programming is **Sensitivity Analysis**. By examining the solution, we can see if it is sensitive to changes in the constraints or in the objective function used.

Constraints that have no part in defining the solution space are said to be **redundant constraints.** Such constraints have no role to play in the solution. Moreover, adding in such constraints does not affect an existing solution to a linear programme.

Constraints that define the solution space may be **binding** or **non-binding**. In the example above the first constraint is redundant: Such a constraint will limit the feasible region, but will not form part of the optimal solution with the current objective function.

In the problem above the optimal solution is at point B. This is at the intersection of the last two constraints lines. These constraints are said to be **binding constraints**. Changing a binding constraint will immediately affect the optimal solution.

The last two equations show that the value of the left hand sides (LHS) are exactly equal to the value of the right hand sides, (RHS) whereas in the first equation there is a difference of 350 (1800–1450) between the LHS and RHS. This difference is known as **slack** and will be the object of much attention in the next chapter. This slack variable typically represents the amount of a resource that remains unused. If the constraint is of the ≥ type, the variable is known as a **surplus** variable.

The possibility exists of the slack variables of more than two constraints being zero at the optimal solution. This would occur where the optimal solution falls at the junction of more than two constraints. This situation is known as **degeneracy** and it implies that one or more of the constraints are redundant. This situation is important in sensitivity analysis however as it means that if the slack variable of a constraint is zero then even if one of the binding constraints is relaxed (i.e. changed so that it becomes non-binding) the optimal solution may not change at all. In Unit 3 we will see that this can cause certain problems in solving other linear programs using the simplex algorithm.

Suppose that the first inequality above was re-written as:

$2x_1 + 3x_2 \leq 1450$

Figure 2.3 shows the effect on the solution space.

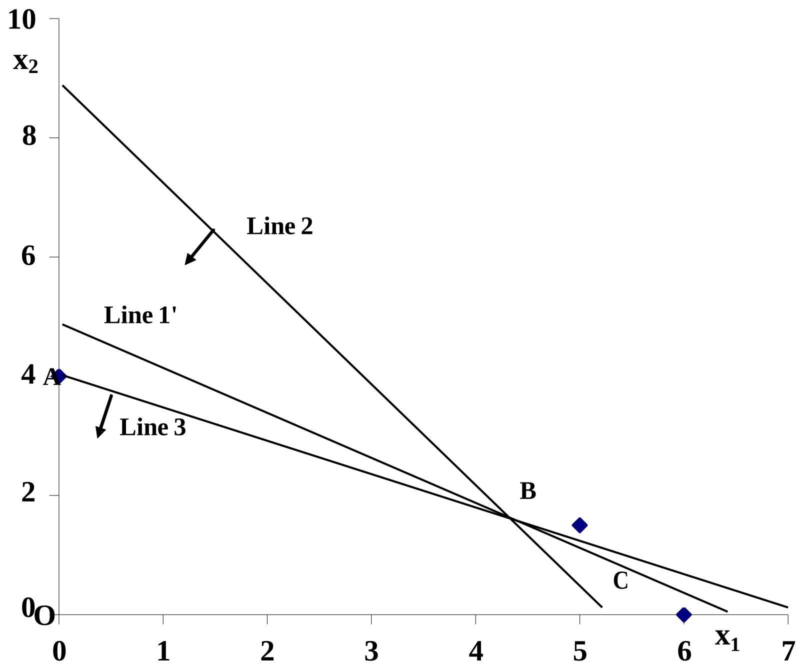



**Figure 2.3**
**The effect on the solution space**

Observe that line 1′ (originally line 1) now goes through the optimal vertex B but **does not** change the shape of the solution space. Thus, as long as the RHS of the first inequality is in the range

$$1450 < (RHS)_1 < 1800,$$

the inequality has no effect on the solution to the model, and hence to production strategy. If, however, $(RHS)_1$ becomes less than 1450 the shape of the solution space will change and also the solution of the model. The same observation also applies if $(RHS)_2 < 600$ and $(RHS)_3 < 200$.

As an exercise you should reduce the value of the RHS and observe how the solution to the model changes.

Another important tool in sensitivity analysis is the calculation of a **shadow price**. The shadow price of a resource at the optimal solution is defined as the change in the optimal objective function if the availability of the resource is increased by one unit. In the problem above the constraints are:

$2x_1 + 3x_2 \leq 1800$

$x_1 + 2/3x_2 \leq 600$

$¼\ x_1 + ½\ x_2 \leq 200$

$x_1 \geq 0$

$x_2 \geq 0$

We have shown that constraint 1 is non-binding. Therefore increasing its availability by 1 unit will not impact on the optimal solution to the problem. A change in the amount of resources available will not affect the objective function if all the resources are not already being used.

Suppose that we increase the availability of the resource described by constraint 2 by one unit so that it becomes:

$x_1 + 2/3x_2 \leq 601$

This is a binding constraint, and as it has changed we must now re-calculate the optimal point. This is got by solving the simultaneous equations:

$x_1 + 2/3x_2 = 601$

$¼x_1 + ½x_2 = 200$

We find $x_1 = 501½$ and $x_2 = 149¼$, which gives Z = 5803. Hence the shadow price for constraint 2 is 3, as we see there has been an improvement in the objective function because of the change. The result can be interpreted in the following way: If one extra unit of resource 2 becomes available it should be possible to increase the profit (i.e. the value of the objective function) by 3 units.

While shadow prices allow us to measure the effect of changing resource availability, they are only valid to the extent that the same constraints remain the binding ones at the optimal solution. You should now try and calculate the shadow price for constraint 3 yourself. Note that it is only the Right Hand Side of the constraint 3 resource constraint that is increased by 1 unit for these calculations.

▶ See video **V06_LPGraph_3**, which calculates shadow prices for the loudspeaker linear programming problem (solved in the video V04_LPGraph_1).
https://goo.gl/ZNGgqm

Having calculated the shadow price for a constraint, we may then wish to examine the range in which this shadow price is valid. Let us consider constraint 2. If we increase the right hand side value of this constraint, then line 2 moves upwards. When it reaches the stage where it crosses the x1 axis at 8, it becomes non-binding (and also redundant), at which stage the shadow price suddenly drops to zero. Similarly, if we decrease the right hand side value of constraint 2, then line 2 moves downwards until it reaches the stage where it crosses the x2 axis at 4, at which stage the shadow price again changes.

We may also consider the range for which the shadow price of 0 for a non-binding constraint is valid. Let us consider constraint 1. It is a non-binding constraint. However, if we decrease the right hand side of constraint 2 then line 2 moves downwards. As we continue to decrease this value this line will eventually pass through point B and become a binding constraint.

These factors that affect the solution of a linear programming model will be discussed in some detail in Chapter 3.

▶ See video **V07_LPGraph_4**, which calculates right hand side ranges for the loudspeaker linear programming problem (solved in the video V04_LPGraph_1). https://goo.gl/SOvdy9

We will now consider what effect changing the value of the coefficients in the objective function will have on the solution.

When we were drawing the iso-function lines we saw that they were parallel to each other. This is due to the linear nature of the equation of the objective function. We can also see that if we change either of the coefficients of the objective function the equation will remain linear but will not be parallel to the original function. This is as a result of changing the slope of the objective function.

e.g. above we had Max Z = 8x1 + 12x2

If we put this equation into its slope – intercept format of y = mx + c we get the following:

Letting x2 = y

12x2 = Z – 8x1

X2 =Z/12 – 8x1/12 = Z/12 – 2x1/3

The slope is equal to -2/3.

If we now change the coefficient of x1 to be 6 we get Z = 6x1 + 12x2.

In slope intercept format this gives

X2 = Z/12 – x1/2.

The slope is –½ and so has changed.

From this we can see that the direction of the line can change and can lead to a different solution. This will be discussed in more detail in Unit 3.

## *SAQ 1*

*Solve graphically the following linear programming problem:*

*Maximise* $Z = 10\ x_1 + 50\ x_2$

*Subject to:*
$5\ x_1 + 5\ x_2 \leq 250$
$2\ x_1 + 4x_2 \leq 160$
$x_1, x_2, \geq 0$

## *SAQ 2*

*A joinery workshop can make wooden tables and chairs using 4 separate processes. The usage time in hours of the processes by each of the products and the maximum time available in each of the departments is given below.*

| | *Planing* | *Cutting* | *Turning* | *Painting* |
|---|---|---|---|---|
| *Chair* | *1* | *1* | *1* | *0* |
| *Table* | *2* | *1* | *0* | *1* |
| *Max hours* | *250* | *150* | *125* | *140* |

*Given that each chair makes a profit of €15 when sold and each table makes a profit of €20, formulate this problem as a linear program and solve using the graphical method, clearly identifying the feasible region. What is the value of the objective function at the optimal solution?*

## *SAQ 3*

*Mac's Furniture Ltd. Manufactures two products: tables and benches. Each table sells for €64 and uses €19 of wood and €8 of fittings. It costs Mac €20 in labour and general overheads to make a table. Each bench sells for €52, uses €8 of wood and €11 of fittings. The cost of labour and overheads is €21 for each bench. There are three stages in manufacturing both tables and benches: cutting, assembly and finishing. A table requires a half hour cutting, two hours of assembly and six hours of finishing. A bench requires one hour cutting, three hours assembly and one and a half hours finishing. Mac has an unlimited supply of wood and fittings. However, in any week, he is limited to forty hours of cutting, sixty hours of assembly and eighty hours of finishing.*

*(a) Construct a linear programming model to maximise Mac's weekly profit.*

*(b) Plot the constraints and identify the feasible area on a graph.*

*(c) Identify the optimum solution and calculate the value of the objective function at this point.*

## *SAQ 4*

*For the problem of Mac's Furniture Ltd used in SAQ 3:*

*Identify the binding and non-binding constraints.*

*Calculate the slack variables at the optimal solution.*

*Augment the problem with the following constraints:*

*$x_1 \geq 7$ and $x_2 \leq 10$*

*Identify the new optimum solution if one exists and calculate the new slack values for the constraints.*

## *SAQ 5*

*For the problem in SAQ 2 calculate the slack variables, the shadow prices and the ranges in which the shadow prices remain valid for all constraints.*

## 2.6 MORE DIFFICULT MAXIMIZATION PROBLEMS

In Section 2.1 it was noted that the maximization model under examination had an "advantage" in that all of the RHS are upper bounds. As has been explained the **form** of the RHS determines the shape of the solution space. Thus changes in the type of RHS alters the **shape** of the solution space and hence (usually) the solution to the model. The simplest illustration of this is to change the $x_1$ condition from $\geq 0$ to $\leq 5$.

Thus the model now becomes:

$$\text{Max } Z = 8x_1 + 12x_2$$

$$\begin{aligned} \text{Subject to} \quad & 2x_1 + 3x_2 \leq 1800 \\ & x_1 + 2/3x_2 \leq 600 \\ & ¼x_1 + ½x_2 \leq 200 \\ & x_1 \leq 500 \\ & x_2 \geq 0 \end{aligned}$$

Figure 2.4 illustrates the effect of the new condition on the solution space.

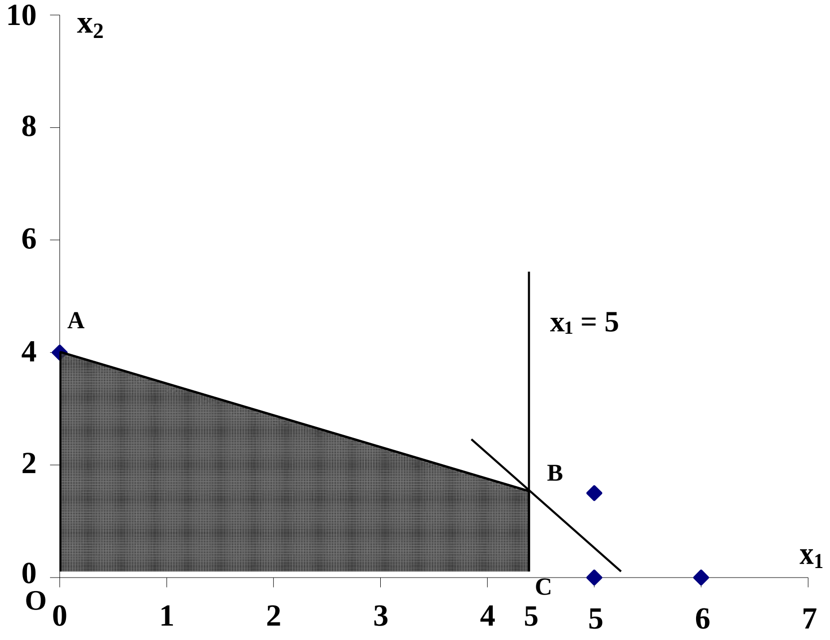



**Figure 2.4**

**The effect on the solution space where $x_1 \leq 5$**

The solution space has now become smaller but the optimal vertex is still at B. Satisfy yourself that this is the case. Notice that if $x_1 \leq 4$, 3 or 2 the optimal vertex moves nearer to the origin and the value of the objective function changes. In fact, the complexity of the solution space is a function of ≤, ≥ or = and it is perfectly possible to have mixtures of all three conditions in a single model.

## *SAQ 6*

*Given the following linear program:*

*Maximise:* $Z = 4x_1 + 3x_2$

*Subject to:*

$5x_1 + 2x_2 \leq 60$

$2x_1 + 4x_2 \leq 48$

$3x_1 \geq 15$

$5x_1 - 4x_2 \leq 40$

*(i) Graphically show the feasible solution area.*

*(ii) Compute the co-ordinates of all the intersecting feasible corners.*

*(iii) Find the optimal solution.*

*(iv) Find the optimal value of the objective function.*

## 2.7 A TYPICAL MINIMIZATION PROBLEM

Again, recall from Unit 1 we also constructed the linear programming model.

Min Z = $6y_1 + 5y_2$

Subject to:

$15y_1 + 30y_2 \geq 30$ EQN I

$30y_1 + 20y_2 \geq 50$ EQN II

$y_1 \geq 0$

$y_2 \geq 0$

This is a typical minimization problem where all of the constraints are lower bounds. Again, since only two variables ($y_1$ and $y_2$) exist the model can be solved graphically.

## 2.8 CONSTRUCTION OF THE SOLUTION SPACE AND OBJECTIVE FUNCTION

Following the steps outlined in Section 2.2 the solution space is as shown in Figure 2.5

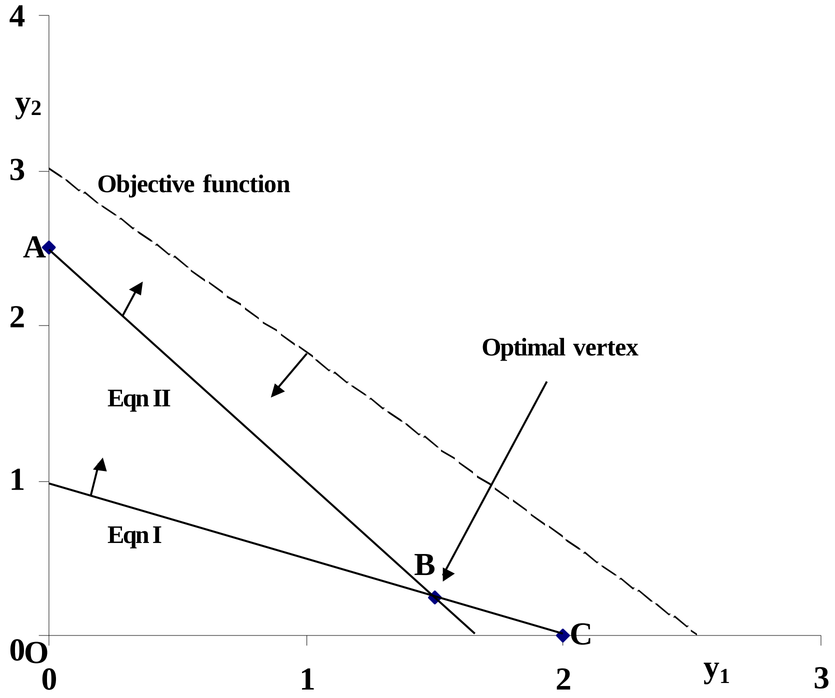



**Figure 2.5**
**Solution space of minimization problem**

Since each of the equations are lower bounds the solution spaces of interest are along the lines and to *the right*, hence the directional arrows are pointing *away* from the origin. Using the principle of dominance results in the solution space ABC and to the right.

Assigning the arbitrary value of 15 to the objective function allows us to plot the line representing this function and since it has to be minimized its arrow points in the direction of the origin.

## 2.9 THE CHARACTERISTICS OF THE SOLUTION SPACE AND THE SOLUTION TO THE MINIMIZATION PROBLEM

The edges of the solution space are AB and BC and the optimal solution to the model will be found at vertices A, B or C. Note that *non-optimal solutions* are to be found to the right of the edges and that there are an infinite number of them. You should satisfy yourself that ($y_1$ = 2, $y_2$ = 2) and ($y_1$ = 3; $y_2$ = 3) are feasible solutions to the model, however they are non-optimal.

By sliding the objective function towards the origin the vertex B is the closest it can get to the origin before leaving the solution space. Thus, B is the optimal vertex, and its co-ordinates are:

$y_1 = 1.5$ i.e. 1.5 kg

and

$y_2 = 0.25$ i.e. 0.25 kg

Thus the value of the objective function can be computed to be

Min $Z = 6y_1 + 5y_2 = 6(1.5) + 5(0.25) = 10.25$

Note that $y_1 = 3$ i.e. 3 kg and $y_2 = 3$ i.e. 3 kg yields the value of the objective function as:

Minimise $Z = 6(3) + 5(3) = 33$, i.e. 33 kg

This is clearly non-optimal.

If we substitute the optimal values of $y_1$ and $y_2$ into the complete minimization model, we obtain the following:

Min Z = $6y_1 + 5y_2 \Rightarrow 6(1.5) + 5(0.25) = 10.25$

Subject to: $15y_1 + 30y_2 \geq 3 \Rightarrow 15(1.5) + 30(0.25) = 30 \geq 30$

$30y_1 + 20y_2 \geq 5 \Rightarrow 30(1.5) + 20(0.25) = 50 \geq 50$

Thus all the conditions of the model are clearly satisfied and at minimum cost.

## *SAQ 7*

*Solve graphically the following linear programming problem:*

*Minimise* $Z = 200\,y_1 + 160\,y_2$

*Subject to:*

$3\,y_1 + y_2 \geq 60$

$y_1 + y_2 \geq 40$

$2\,y_1 + 6\,y_2 \geq 120$

$y_1, y_2 \geq 0$

## 2.10 SOLUTION OF A THREE VARIABLE PROBLEM

Suppose we have a model in three variables of the form:

Max Z = $6x_1 + 3x_2 + 8x_3$

Subject to: $x_1 + 2x_2 + x_3 \leq 400$

$2x_1 + 4x_2 + x_3 \leq 600$

$\frac{1}{4}x_1 + 1/3x_2 + 1/3x_3 \leq 100$

$x_1 \geq 0$

$x_2 \geq 0$

$x_3 \geq 0$

Since three variables can be represented graphically, it is therefore possible to attempt to solve the model in this way. However, the geometrical format of a three variable problem is challenging and the conceptualisation of the solution space can be difficult. **Figure 2.6** illustrates the problem.

Visualizing which extreme point maximizes the plane FGH is hard. Complete enumeration is of course, possible but laborious. As an exercise you might like to "have a go". The solution will be revealed in Unit 3.

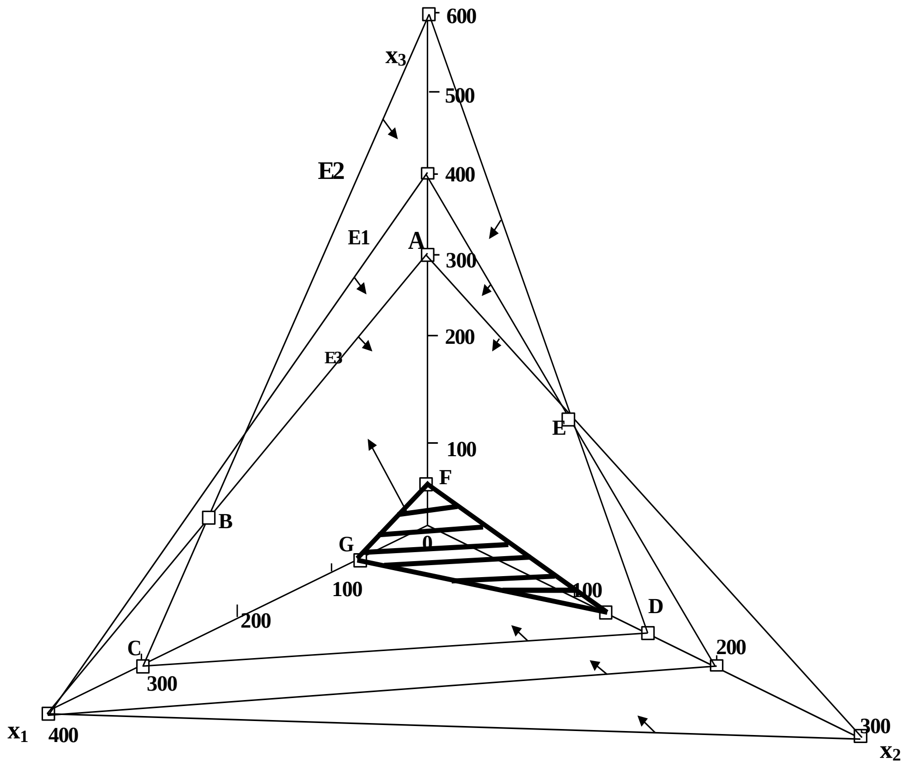



**Figure 2.6**
**Solution of a 3-Variable Problem**

## SUMMARY

In this unit formulated linear programming problems involving two decision variables were examined and solved using graphical methods. Given that we are dealing with linear programming problems the method simply consisted of drawing the two positive axes corresponding to the two decision variables and a series of lines representing each constraint in turn. The final line drawn (often as a dotted line) is a typical iso-function line representing the objective function. In maximisation problems we will be looking for the corner point in the solution space where this iso-function last touches the feasible region. This can be judged by moving this trial function parallel to itself away from the origin. In the case of a minimisation problem move the trial function (dotted line) towards the origin parallel to itself. When the corner point (intersection of at least two of the constraints) is identified determine the coordinates which are then the optimal values of the decision variables. The value of these coordinates is best found by solving the simultaneous equations pertaining to the intersecting lines forming the corner in question. The optimal value of the objective function can be then determined and each constraint equation examined in turn to determine the values of any slack or surplus variables. It may happen that one of these constraints is parallel to the iso-function in which case the solution will lie on that line between two sets of corner points and hence there is an infinity of solutions.

Realizing that the solution may change if some of the coefficients are changed is an important feature of a solution. In the real world values may change or be unknown and we often have estimates only. For example, the unit profit contribution for a product mix problem might differ from what we assumed in our model and solution or less of a resource was available than thought. Consequently, sensitivity analysis is integral to any solution. Knowing which inputs when changed will impact on the solution and by how much can be crucial to developing a solution that is understood, plausible, robust and implementable. The model and solution should be more than a simple prescription of what to do as regards the decision variables and are a tacit acknowledgement that we do not necessarily know the value of all the model parameters with certainty.

Various economic type insights are afforded by determining the 'shadow prices' which allow for a value to be assigned for additional units of some of the resources (the binding constraints RHS values) if used optimally.
The graph allows for these insights and scenario analyses to be conducted relatively easily in the case of two variables. The transition to three or more decision variables calls for other approaches which will be introduced in the next units. The Simplex algorithm has enjoyed great success in this regard and more latterly spreadsheets with optimiser tools such as Excel with the Solver add-in are widely used.

When the solution is found it is good practice to generate a simple report on the solution and the implications for not only the objective function but also each resource involved and their shadow prices and ranges for which they can be expected to hold true.

### *SUMMARY OF STEPS*

1. **Construct the coordinate axes** (positive quadrant) and label them.
2. **Scale the axes values** so as to achieve a clear and uncluttered graph. This will involve checking how big the two variables can be by looking at each constraint and letting each variable be zero in turn and see how big the other variable is when the line cuts the axis. For example, in SAQ 6 we had;

   Maximise: $4x_1 + 3x_2$

   Subject to:

   | | | |
   |---|---|---|
   | (i) | $5x_1 + 2x_2$ | $\leq 60$ |
   | (ii) | $2x_1 + 4x_2$ | $\leq 48$ |
   | (iii) | $3x_1 \geq 15$ | |
   | (iv) | $5x_1 - 4x_2$ | $\leq 40$ |

   And the following points were on the lines (constraints):

   point (0,30) lies on the line
   point (12,0) lies on the line
   point (0, 12) lies on the line
   point (24,0) lies on the line
   point (8,0) lies on the line

   Hence the horizontal axis had to extend to at least 24 and the vertical axis to at least 30.

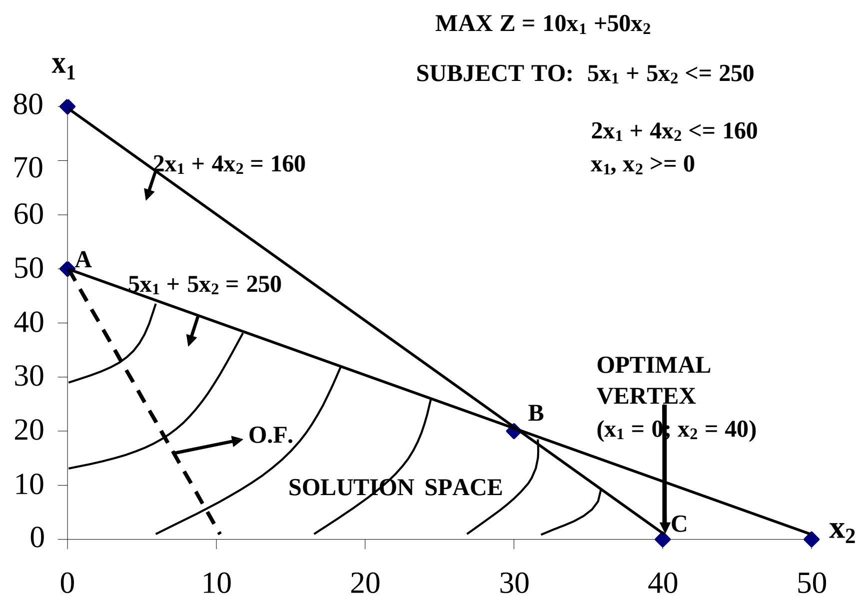
3. **Draw each constraint line** (treating them as equality conditions) using where they intersect the two axes as a means of drawing each line. Note the direction of the inequality using an arrow on your graph.
4. Collectively the constraints will determine a **well-defined region** where the solution may be found.
5. **Draw a typical iso-function** and move it parallel to itself away from the origin for a maximisation problem and towards it for a minimisation problem.
6. **Determine which corner point** determines the optimal solution by noting where the iso-function last touches the solution space.
7. **Calculate** the coordinates at this point by solving the relevant simultaneous equations (intersecting constraints).
8. Calculate **all values of interest** such as the objective function optimal value, all slack and surplus variables and noting which constraints are binding, shadow prices and ranges of validity of the shadow prices. Other scenarios can be investigated also.
9. **Generate** a solution report that is clear to stakeholders (typically non-technical) and implement and review the findings as needed.

## FINAL COMMENTS

A linear programming problem if formulated properly and involving only two variables can be easily solved graphically. Solving realistic problems will often require including many variables and constraints and hence more sophisticated approaches will be required. Despite its simplicity a two variable problem solved graphically allows for surprisingly deep insight and appreciation of many of the pertinent factors involved in more general problems. Linear programming problem solution requires practise and care and the effort expended in this unit will form a strong basis for understanding the issues central to these problems. As such the student should again study some of the generic problem types to be found in most management science/Operational Research textbooks and web resources, including the reading list. These problems and their solvability particularly with the aid of software make for a very powerful analysis capability when applied to practical applications.

## ANSWERS TO SAQS

### *SAQ 1*




*O.F. = Objective Function*

*Optimal value of Z = Z* = 10(0) + 50(40) = 2000*

## *SAQ 2*

*The problem can be formulated as follows:*

*Let* $x_1$ = *number of chairs produced*
$x_2$ = *number of tables produced*

*Maximise:* $15x_1 + 20x_2$

*Subject to:*

| | | | |
|---|---|---|---|
| *(i)* | $x_1 + 2x_2$ | $\leq$ | *250 (planing time constraint)* |
| *(ii)* | $x_1 + x_2$ | $\leq$ | *150 (cutting time constraint)* |
| *(iii)* | $x_1$ | $\leq$ | *125 (turning time constraint)* |
| *(iv)* | $x_2$ | $\leq$ | *140 (painting time constraint)* |

$x_1 \geq 0$ *and* $x_2 \geq 0$

*To draw constraint lines on the graph we must identify more than one point on each line. For constraints (iii) and (iv) this is easy, for the other constraints this can be done by identifying the point where each constraint cuts the* $x_1$ *and* $x_2$ *axes i.e.*

$x_1 = 0$ *and* $x_2 = 0$ *respectively.*

*Constraint (i)* $x_1 + 2x_2 \leq 250$

*Putting* $x_1 = 0 \Rightarrow x_2 = 125$

*point (0,125) lies on the line*

*Putting* $x_2 = 0 \Rightarrow x_1 = 250$

*point (250,0) lies on the line*

*Constraint (ii)* $x_1 + x_2 \leq 150$

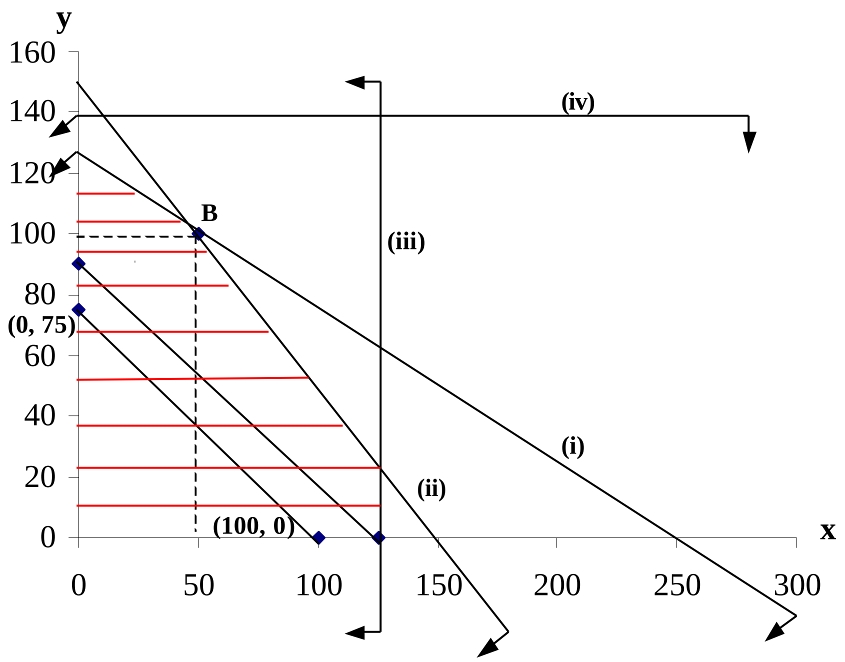
*Putting* $x_1 = 0 \Rightarrow x_2 = 150$

*point (0,150) lies on the line*

*Putting* $x_2 = 0 \Rightarrow x_1 = 150$

*point (150,0) lies on the line*

*../*




*To draw the objective function on the graph we locate an objective function line using some convenient values. If we draw the line $15x_1 + 20x_2 = 1500$ we can locate the points where the line cuts the axes. We can then sketch lines parallel to this one to locate the optimal point.*

$x_1 = 0 \Rightarrow x_2 = 75$

*point (0,75) lies on this objective function line*

$x_2 = 0 \Rightarrow x_1 = 100$

*point (100,0) lies on this objective function line*

*From the graph the optimal solution is $x_1 = 50$, $x_2 = 100$ at point B on the graph.*

*Objective Function value*

$15x_1 + 20x_2 = 15(50) + 20(100) = 2750$

## *SAQ 3*

*The model for Mac's Furniture Ltd. is as follows:*

*Let*
*x1 = Number of tables produced each week*
*x2 = Number of benches produced each week.*

*Maximise:* $17x_1 + 12x_2$
*Subject to:*

(i) $0.5x_1 + x_2 \leq 40$

(ii) $2x_1 + 3x_2 \leq 60$

(iii) $6x_1 + 1.5x_2 \leq 80$

$x_1 \geq 0$ *and* $x_2 \geq 0$

*To draw constraint lines on the graph we must identify more than one point on each line. This can be done by identifying the point where each constraint cuts the x and Y axes.*

*Constraint (i)* $0.5x_1 + x_2 \leq 40$

$x_1 = 0 \Rightarrow x_2 = 40$ *point (0,40) lies on the line*

$x_2 = 0 \Rightarrow x_1 = 80$ *point (80,0) lies on the line*

*Constraint (ii)* $2x_1 + 3x_2 \leq 60$

$x_1 = 0 \Rightarrow x_2 = 20$ *point (0,20) lies on the line*

$x_2 = 0 \Rightarrow x_1 = 30$ *point (30,0) lies on the line*

*Constraint (iii)* $6x_1 + 1.5x_2 \leq 80$

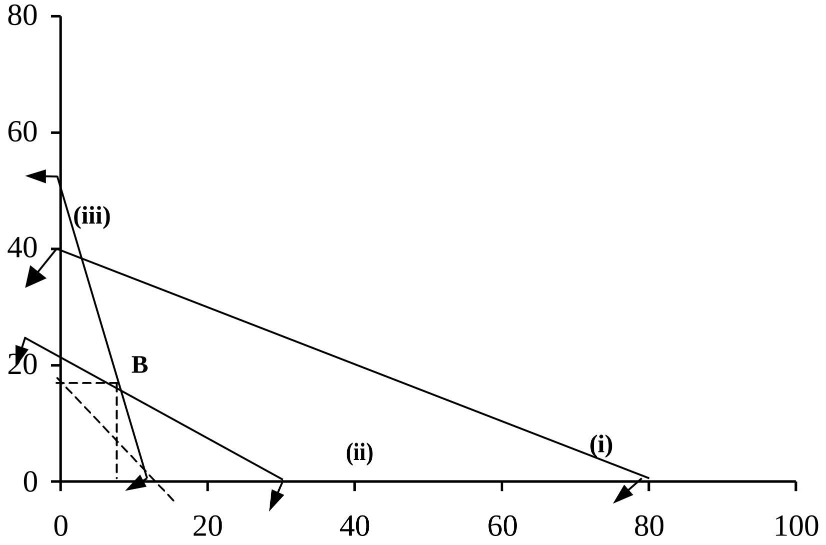
$x_1 = 0 \Rightarrow x_2 = 53.333$ *point (0, 53.333) lies on the line*

$x_2 = 0 \Rightarrow x_1 = 13.333$ *point (13.333, 0) lies on the line*

*Draw objective function line* $17x_1 + 12x_2 = 204$

$x_1 = 0 \Rightarrow x_2 = 17$ *point (0,17) lies on this objective function line*

$x_2 = 0 \Rightarrow x_1 = 12$ *point (12,0) lies on this objective function line*




*From the graph the optimum solution is $x_1$ = 10, $x_2$ = 13.333 at point on graph.*

*Objective function value:*

$$17x_1 + 12x_2 = 17(10) + 12(13.333) = 330$$

## *SAQ 4*

*Binding constraints (ii) and (iii).*

*To calculate slack variables for optimum solution (10, 13.33).*

*Constraint (i):*

$$0.5\ x_1 + x_2 + s_1 = 40$$

$$0.5\ (10) + 13.33 + s_1 = 40$$

$$s_1 = 21.67$$

*Constraint (ii):*

$$2\ x_1 + 3\ x_2 + s_2 = 60$$

$$2\ (10) + 3(13.33) + s_2 = 60$$

$$s_2 = 0 \quad \text{(binding constraint)}$$

*Constraint (iii):*

$$6\ x_1 + 1.5\ x_2 + s_3 = 80$$

$$6\ (10) + 1.5(13.33) + s_3 = 80$$

$$s_3 = 0 \quad \text{(binding constraint)}$$

*New constraints:*

*(iv)* $x_1 \geq 7$
*(v)* $x_2 \leq 10$

*New optimal solution* $x_1 = 10.833$, $x_2 = 10$ *(obtained graphically).*

| *Constraint* | | | | *Slack or Surplus Variable* |
|---|---|---|---|---|
| 1 | $0.5(10.833)$ | $+ 10$ | $+ s_1 = 40$ | 24.58 |
| 2 | $2(10.833)$ | $+3(10)$ | $+ s_2 = 60$ | 8.33 |
| 3 | $6(10.833)$ | $+ 1.5(10)$ | $+ s_3 = 80$ | 0 |
| 4 | $10.833$ | | $+ s_4 = 7$ ($s_4$=surplus) | 3.83 |
| 5 | | $10$ | $+ s_5 = 10$ | 0 |

## *SAQ 5*

*From SAQ 2 the optimal solution is $x = 50$, $y = 100$*
*Constraint (i) $1(50) + 2(100) + s_1 = 250$, and therefore $s_1 = 0$.*

*To calculate the shadow price, we increase the planing time by one unit*

$$\text{(i) } x + 2y = 251$$
$$\text{(ii) } x + y = 150$$

*From the graph it can be seen that constraints (i) and (ii) remain binding, hence the optimum solution remains where the constraints cross. Solving the two equations above as a set of simultaneous equations gives $x = 49$ and $y = 101$.*

*The new objective function value is $15(49) + 20(101) = 2755$*
*The change in the objective function is $2755 - 2750 = 5$*

*Hence, the shadow price for planing time is 5.*

*From the graph the shadow price remains valid until the planing time increases to the extent that the line on the graph passes through the point where constraint (ii) and (iv) cross.*

$$\text{(ii) } x + y \leq 150$$
$$\text{(iv) } y \leq 140$$

*Solving these gives $x=10$ and $y=140$.*

*To pass through the point (10, 140) the planing time constraint would need to be*

$$10 + 2(140) = 10 + 280 = 290$$

*If the planing time exceeds 290, the shadow price will become invalid.*

*From the graph the shadow price remains valid until the planing time decreases to the extent that the line on the graph passes through the point where constraints (ii) and (iii) cross.*

$$\text{(ii) } x + y \leq 150$$
$$\text{(iii) } x \leq 125$$

*Solving these gives $x=125$ and $y=25$.*

*To pass through the point (125, 25) the planing time constraint would need to be*

$$125 + 2(25) = 125 + 50 = 175$$

*If the planing time is less than 175, the shadow price will become invalid.*

*Constraint (ii) $1(50) + 1(100) + s_2 = 150 \Rightarrow s_2 = 0$.*

*To calculate the shadow price, we increase the cutting time by one unit.*

$$\text{(i) } x + 2y = 250$$
$$\text{(ii) } x + y = 151$$

../

*Solving these gives $x$=52 and $y$=99.*

*New objective function value is 15(52) + 20 (99) = 2760*

*Change in objective function is 2760-2750 = 10*

*Shadow price for planing time is 10.*

*From the graph it can be seen that the shadow price remains valid until the cutting time increases to the extent that the line on the graph passes through the point where constraints (i) and (iii) cross.*

(i) $x + 2y \leq 250$
(ii) $x \leq 125$

*Solving these gives $x = 125$ and $y = 62.5$*

*To pass through the point (125, 62.5) the cutting time constraint would need to be*

$$x + y = 125 + 62.5 = 187.5$$

*If the cutting time exceeds 187.5, the shadow price will become invalid.*

*From the graph the shadow price remains valid until the cutting time decreases to the extent that the line on the graph passes through the point where constraint (iii) meets the x-axis.*

(iii) $x \leq 125$
$y \geq 0$

*Solving these gives $x$=125 and $y$=0*

*To pass through the point (125,0) the cutting time constraint would need to be*

$x + y = 125 + 0 = 125.$

*If the cutting time is less than 125, the shadow price will become invalid.*

*Constraint (iii) $1(50) + 0(100) + s_3 = 125 \Rightarrow s_3 = 75$*

*This constraint is non-binding, and therefore the shadow price is 0.*

*From the graph, an increase in the constraint will never affect the solution, so an infinite increase is possible without changing the shadow price.*

*From the graph the shadow price remains valid until the turning time decreases to the extent that the line on the graph passes through the point where constraints (i) and (ii) cross.*

(i) $x + 2y \leq 250$
(ii) $x \leq 125$

*This gives optimal point $x$=50 and $y$=100.*

*To pass through the point (50, 100) the turning time constraint would need to be*

$x = 50 + 0 = 50.$

../

*If the turning time is less than 50 the shadow price will become invalid.*

*Constraint (iv) 0(50) + 1(100) +* $s_4$ *=100* $\Rightarrow$ $s_4$ *=40*

*This constraint is non-binding, therefore the shadow price is 0.*

*From the graph, an increase in the constraint will never affect the solution, so an infinite increase is possible without changing the shadow price.*

*From the graph the shadow price remains valid until the painting time decreases to the extent that the line on the graph passes through the point where constraints (i) and (ii) cross.*

(i) $x + 2y \leq 250$
(ii) $x \leq 125$

*This gives optimal point x=50 and y=100.*

*To pass through the point (50, 100) the painting time constraint would need to be*

$y = 0 + 100 = 100.$

*If the painting time is less than 100, the shadow price will become invalid.*

## *SAQ 6*

*Maximise:* $4x_1 + 3x_2$

*Subject to:*

| | | |
|---|---|---|
| *(i)* | $5x_1 + 2x_2$ | $\leq 60$ |
| *(ii)* | $2x_1 + 4x_2$ | $\leq 48$ |
| *(iii)* | $3x_1$ | $\geq 15$ |
| *(iv)* | $5x_1 - 4x_2$ | $\leq 40$ |

*To draw constraint lines on the graph we must identify more than one point on each line. This can be done by identifying the point where each constraint cuts the* $x_1$ *and* $x_2$ *axes.*

*Constraint 1* $\quad 5x_1 + 2x_2 \leq 60$

$x_1 = 0 \Rightarrow x_2 = 30$ *point (0,30) lies on the line*

$x_2 = 0 \Rightarrow x_1 = 12$ *point (12,0) lies on the line*

*Constraint 2* $\quad 2x_1 + 4x_2 \leq 48$

$x_1 = 0 \Rightarrow x_2 = 12$ *point (0, 12) lies on the line*

$x_2 = 0 \Rightarrow x_1 = 24$ *point (24,0) lies on the line*

*Constraint 3* $\quad 3x_1 \geq 15$

$x_1 = 5$

*Constraint 4* $\quad 5x_1 - 4x_2 \leq 40$

$x_2 = 0 \Rightarrow x_1 = 8$ *point (8,0) lies on the line*

*In this case the intercept with the* $x_2$ *axis is in the negative part of the graph so it is more use for graph sketching purposes to locate another point in the positive region of the graph.*

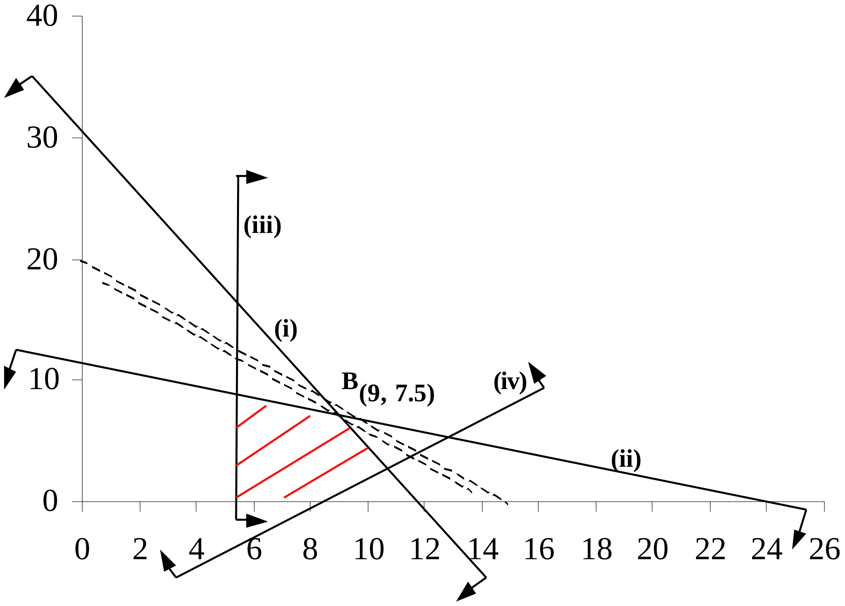
$x_1 = 20 \Rightarrow x_2 = 15$ *point (20,15) lies on the line*

*To draw the objective function on the graph we locate an objective function line using some convenient values. If we draw the line* $4x_1 + 3x_2 = 60$ *we can locate the points where the line cuts the axes.*

$x_1 = 0 \Rightarrow x_2 = 20$ *point (0,20) lies on the line*

$x_2 = 0 \Rightarrow x_1 = 15$ *point (15,0) lies on the line*

*../*

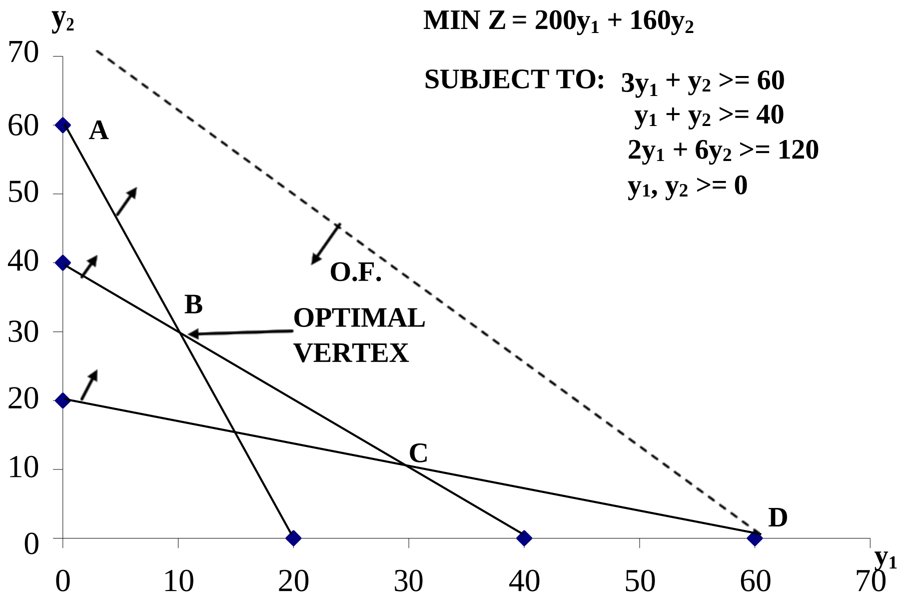



*(ii) point x1 = 5 x2 = 9.5 (constraints 2 & 3)*
*point x1 = 9 x2 = 7.5 (constraints 1 & 2)*
*point x1 = 10.66 x2 = 3.33 (constraints 1 & 4)*
*point x1 = 8 x2 = 0 (constraint 4 and x2 > 0)*
*point x1 = 5 x2 = 0 (constraints 3 and x2 > 0)*

*(iii) optimum solution x1 = 9 x2 = 7.5*

*(iv) optimum value of objective function*
*4x1 + 3x2*
*4(9) + 3(7.5) = 36 + 22.5 = 58.5*

## *SAQ 7*




*Points used to draw the constraint lines are:*
*(0, 60) and (20, 0) for* $3y_1 + y_2 \geqslant 60$
*(0, 40) and (40, 0) for* $y_1 + y_2 \geqslant 40$
*(0, 20) and (60, 0) for* $2y_1 + 6y_2 \geqslant 120$

*The objective function line (OF) is got from setting* $200y_1 + 160y_2 = 120,000$

*The feasible region lies above the points A,B,C,D.*
*The objective function is minimized at the point B* $y_1 = 10$; $y_2 = 30$

*Optimal value of* $Z = Z^* = 200(10) + 160(30) = 6,800$

# UNIT 3

# INTRODUCTION TO THE SIMPLEX ALGORITHM

## AIM

This unit aims to show how linear programming models, of any size, can be solved using the simplex algorithm.

## OBJECTIVES

At the end of this unit you will:

- Have an understanding of how linear programming models are prepared for solution by the simplex method.
- Appreciate the underlying mathematics of the algorithm.
- Be able to use the mathematical rules of the simplex algorithm to solve both maximization and minimization linear programmes.
- Be able to interpret the characteristics of sub-optimal and optimal simplex tableaux.
- Appreciate the role of computer software in the solution of linear programmes.

## REQUIRED READING

Students should read the following chapters of the mandatory text book (David R. Anderson, Dennis J. Sweeney, Thomas A. Williams and Kipp Martin, *An Introduction to Management Science: Quantitative Approaches to Decision Making*, 13th International Edition, Cengage Learning, ISBN-13: 9780538475655)

Chapter 17 (On CD), Chapter 18.1 (On CD)

## 3.1 LINEAR MODELS IN THREE AND MORE VARIABLES

Section 2.10 proposed a linear model in three variables and Figure 2.6 indicated how it could be solved by graphical methods. The illustration of the method showed how difficult it is to visualize a three dimensional solution space and hence discover the extreme point of the solution space which maximizes the objective function. The graphical solution of a 3-variable minimization problem, whose right hand sides (RHS) are all lower bounds, is even more difficult to achieve; and a maximization or a minimization problem with *mixed* upper and lower bounds is yet again more difficult to solve graphically.

Suppose we add one more variable, $x_4$, to the maximization problem in section 2.10 to give:

Max $\quad Z = 6x_1 + 3x_2 + 8x_3 + x_4$

Subject to:

$x_1 + 2x_2 + x_3 + x_4 \leq 400$

$2x_1 + 4x_2 + x_3 + x_4 \leq 600$

$\frac{1}{4} x_1 + 1/3\ x_2 + 1/3\ x_3 + x_4 \leq 100$

$x_1 \geq 0$

$x_2 \geq 0$

$x_3 \geq 0$

$x_4 \geq 0$

This model obviously cannot be solved graphically. Neither can models containing 40, 100, 200 or many more variables. Some other methodology must be used if we are to succeed in our quest. Before we move on to describing the methodology that is used for models of such dimensionality let us present the general statement of the maximization linear programming model.

Maximize: $c_1 x_1 + c_2\ x_2 + \ldots\ldots\ldots\ldots\ldots + c_n\ x_n$

Subject to:

$a_{11}\ x_1 + a_{12}\ x_2 + \ldots\ldots\ldots\ldots\ldots\ldots + a_{1\,n}\ x_n \leq b_1$

$a_{2\,1}\ x_1 + a_{2\,2}\ x_2 \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots + a_{2\,n}\ x_n \leq b_2$

. . . . . . .

. . . . . . .

. . . . . . .

. . . . . . .

. . . . . . .

$a_{m\,1}\ x_1 + a_{m\,2}\ x_2 + \ldots\ldots\ldots\ldots\ldots\ldots a_{m\,n}\ x_n \leq b_m$

and

$x_j \geq\ 0$ for all j  j = 1,2,….n

Where

n = the number of variables

m = the number of constraints

The shorthand version of the above is expressed as:

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

Subject to:

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

$x_j \geq 0$

where j=1, 2, 3……..n and i= 1, 2, 3…….m

## 3.2 Preparing Maximization Models for the Simplex Algorithm

George B. Dantzig worked for the U.S. Air Force during World War II. Dantzig pioneered mathematical techniques to solve military logistics problems. He called his technique the *programming of interdependent activities in a linear structure*. The first paper on a workable methodology was published in 1947. The technique is now known as the Simplex Method and is applied to linear programming models of all shapes and sizes. Since those early days the Simplex Algorithm has been extended to cope with other programming models such as integer linear programming (where some or all of the variables in the solution must have integer values), multi-objective linear programming and certain non-linear programming formulations. Mathematical Programming has been found to be a powerful problem-solving approach to many problems in business, industry, economics and finance. The number of applications has vastly increased in recent years as computers have enabled researchers to solve larger and more complex linear programming models.

Let us return to the simple two variable maximization problem which was solved graphically in section 2.5:

Max $\quad Z = 8x_1 + 12x_2$

Subject to:

$2x_1 + 3x_2 \leq 1800$

$x_1 + 2/3\, x_2 \leq 600$

$¼x_1 + ½x_2 \leq 200$

$x_1, x_2 \geq 0$

The simplex method does not like inequalities, preferring to work with equalities. The inequalities in the above model are less than (or equal to) their corresponding RHS. Therefore, making both sides equal is simply achieved by adding a variable to each LHS as follows:

$2x_1 + 3x_2 + s_1 = 1800$

$x_1 + 2/3x_2 + s_2 = 600$

$¼x_1 + ½x_2 + s_3 = 200$

Each $s_i$ (i = 1, 2, 3) takes up the slack between each LHS and RHS. Each slack variable will have a value of zero or greater than zero in the optimal solution depending on the optimal values of the decision variables $x_1$ and $x_2$. If you refer back to section 2.5 you will see that, in fact, $s^*_1 = 350$, whilst $s^*_2 = s^*_3 = 0$. The * indicates the optimal value of a variable. Hence the optimal solution can be stated as:

$Z^* = 5800$

$x^*_1 = 500$

$x^*_2 = 150$

$s^*_1 = 350$

$s^*_2 = 0$

$s^*_3 = 0$

To use the simplex algorithm, we must note that each variable in the constraint equations also appears in the objective function. Thus we must augment the objective function with the three slack variables. We give each slack variable a zero coefficient. More will be said about the coefficients of slack variables in later sections of this unit.

The full model is therefore as follows:

Max $Z = 8x_1 + 12\,x_2 + 0s_1 + 0s_2 + 0s_3$

Subject to:

$2x_1 + 3x_2 + s_1 = 1800$

$x_1 + 2/3\,x_2 + s_2 = 600$

$\frac{1}{4}x_1 + \frac{1}{2}x_2 + s_3 = 200$

$x_j \geq 0$

$s_i \geq 0$

Having converted the constraints to equalities, we can now attempt to find a solution that satisfies all of the equations that form the model. In this attempt to find the initial solution, we ignore the objective function and solely concentrate on finding a solution that is feasible. As we will see later, if all of the original constraints are of the form ≤, then the initial solution will be found by assuming that all of the decision variables are zero, and that the slack variables have a value equal to the RHS resource value. Clearly this initial solution is sub-optimal from the objective function point of view, but it is feasible and does provide a starting point for the algorithm. This initial solution satisfies the equations, and the original problem constraints. Thus, this solution is called the *Initial Feasible Solution.*

The Variables that are solved for are called *Basic* variables. While, the variables set to zero are called *non-Basic*.

Let us analyse what the simplex algorithm must achieve and how it might achieve it. We know at this stage that the solution to any linear programme occurs at a corner (or vertex) of the feasible region. We also know that this feasible region is defined by the constraints of the model. Basic geometry guarantees us that if we treat the set of constraints as a set of simultaneous equations, we can determine the co-ordinates of the intersecting points i.e. the values of the variables at the various vertices.

The requirement of any proposed solution procedure is to identify the vertex at which the objective function has an optimal value (maximum or minimum value, depending on the nature of the problem). We could propose the following procedure for general problems of n variables:

- Determine the co-ordinates of the vertices.
- Substitute the co-ordinate values into the objective function and identify the vertex giving the optimal value.

The disadvantage with this method is that the amount of time it takes to obtain the solution is directly dependent on the number of vertices v: for any value of v, we determine the optimal solution by finding and eliminating (v - 1) sub-optimal solutions.

Let us consider again the feasible region for a 3 variable problem. We know that this is a 'shape' with, in general, v corners, formed by the intersection of the surfaces of the shape. Imagine that you are standing at one of these corners, and you are going to walk along the edges to the optimal vertex (an edge is the line of intersection between two surfaces). Let's say it is a maximization problem.

Because we are dealing with a linear model (i.e. the objective function and constraints are linear equations), the following properties hold:

- the edges are straight
- the edges have a constant slope
- there are no local maxima/minima.

In the case of polynomial or non-linear equations, as illustrated in Figure 3.1, a local maximum occurs at F, whereas the global maximum is at G.

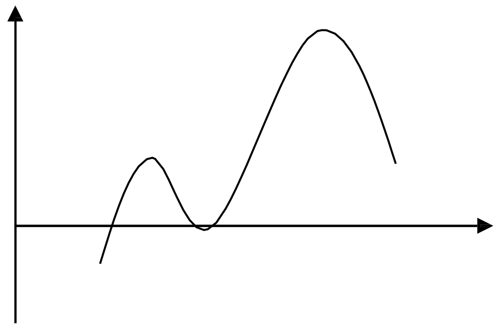


**Figure 3.1: A Local Maximum and a Global Maximum for a polynomial function**

So, in terms of walking from an arbitrary vertex to the optimal vertex, what path would you take?

At any corner there will be a choice of directions: some will lead you 'up' (higher objective function value), some will lead 'down' (lower objective function value). If we want to find the highest point (maximum objective value) as quickly as possible, we should always choose the edge with the steepest ascent. Eventually we will reach a vertex from which all edges lead downwards, and we will have found the optimal vertex.

This is in fact the procedure of the simplex algorithm. What we must do next is convert the above conceptual procedure into a mathematical algorithm.

Don't forget that the inequalities should be converted into equalities. The reason for this is that inequalities are difficult to solve simultaneously. In fact, there is a further theoretical reason for working with equality constraints in that strict equality prevents us from 'sinking' into the feasible region while walking along the edges towards the optimal vertex!

## 3.3 Setting Up the Simplex Array for a Maximization Problem

We now must embark on an iterative process of finding a better solution to the Initial Feasible solution and checking if it is optimal. This will involve a lot of arithmetic operations. To provide a simple and systematic method of doing this, the linear program formulation is converted into tabular format. There are many variations on the format of the simplex tableau. We shall use one of the standard formats. Check that this format is consistent with the text that you are using, and that the output from your computer package corresponds to this format.

For the general linear program formulation with n variables and m constraints, the tableau will have the following format.

Dimensions: m + 3 rows
n + m + 3 columns.

Row 1:

| $c_j$ / $c_b$ | | | $c_1$ | $c_2$ | | $c_n$ | $c_{n+1}$ | | $c_{n+m}$ |
|---|---|---|---|---|---|---|---|---|---|
| | Basis | $b_i$* | $x_1$ | $x_2$ | | $x_n$ | $s_1$ | | $s_m$ |

where $c_j$ is the coefficient of the jth variable ($x_j$ or $s_{j-n}$) in the objective function for j = 1, ..., n + m and where each of the other entries are column headings for columns containing the following:

In rows 2, ..., m + 1,

Basis : the names of the m variables which are currently basic, i.e. the variables which are currently in the solution and have a non-zero value.

$b_i^*$ : The value (no. of units) of the $i^{th}$ basic variable in the current solution for i = 1 to m.

$x_1$ .. $x_n$ : The coefficients of the variables $x_1$, ..., $x_n$ in the constraints.

$s_1$ .. $s_m$ : The coefficients of the variables $s_1$, ..., $s_m$ in the constraints.

In addition: in column 1 we have $c_b$, the per unit contribution of each variable in the basis.

In **row m + 2** are listed

the current objective function value, under $b_i$*

$Z_j$ the current per unit contribution of the jth variable to the objective function, for j = 1 to n + m.

In **row m + 3**, the possible net increase in contribution of each jth variable to the objective function if this variable were to be made basic (i.e. the difference between the potential per unit contribution $C_j$, and the current per unit contribution $Z_j$).

(We have to be careful when considering contribution: when dealing with a maximization problem, contribution is in terms of profit and when dealing with a minimzation problem will be in terms of cost. Obviously the concept of 'an increase in contribution' is desirable in the former case, but not so in the latter).

We are now ready to set up the initial tableau for our problem. First, draw up the template, as follows.

| $c_j$ / $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ | ratio |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| $Z_j$ | | | | | | | | |
| $C_j$ - $Z_j$ | | | | | | | | |

Notice there is now a column on the right hand side of the tableau - we will use this to store the ratio while determining the variable to leave the basis.

Next we insert the appropriate values into the template.

| $c_j$ \ $c_b$ | Basis | $b_i$* | | 8 | 12 | 0 | 0 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ | |
| | | | | 2 | 3 | 1 | 0 | 0 | |
| | | | | 1 | 2/3 | 0 | 1 | 0 | |
| | | | | 1/4 | 1/2 | 0 | 0 | 1 | |
| $Z_j$ | | | | | | | | | |
| $C_j$ - $Z_j$ | | | | | | | | | |

Obj. Function Coeffients

Coefficients from constraint equations

To complete our initial tableau, we must choose some initial solution. As previously discussed, we set the resource variables to zero, and make the slack variables basic, with a value equal to the RHS.

Coefficients of current basis variables in the objective function

| $c_j$ \ $c_b$ | Basis | $b_i$* | 8 | 12 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| | | | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
| 0 | $s_1$ | 1800 | 2 | 3 | 1 | 0 | 0 |
| 0 | $s_2$ | 600 | 1 | 2/3 | 0 | 1 | 0 |
| 0 | $s_3$ | 00 | 1/4 | ½ | 0 | 0 | 1 |
| $Z_j$ | | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_j$ - $Z_j$ | | | 8 | 12 | 0 | 0 | 0 |

Values for slack variables when $x_1$ and $x_2$ are set to zero.

Finally, to help us evaluate the initial solution we compute the Zj and Cj-Zj values, where

$$Z = \sum_{i=1}^{m} c_{bi} \times a_{ij}$$ where $a_{ij}$ is the entry in row i, column j

Notice that the cell entries corresponding to basic variables have a value of 1 in the simplex tableau.

To calculate the elements in the $Z_j$ row, we proceed as follows:

$b_i$* Column

Z = (0) 1800 + (0) 600 + (0)200= 0

$x_1$ Column

Take each element in the column and multiply it by its opposite number in the $C_b$ column. Add the products together. Subtract this quantity from the x coefficient in the objective function ($C_j$) row i.e. 8, thus:

[2 × 0 + 1 × 0 + ¼ × 0] - 8 = 0, hence the $C_j$ - $Z_j$ value is 8 – 0 = 8

$x_2$ Column

The same set of rules apply, thus we have:

[3 × 0 + 2/3 × 0 + 1/2 × 0] -12 = 0, hence the $C_j$ - $Z_j$ value is 12 – 0 = 12

Similarly, for the $s_1$, $s_2$, $s_3$ columns.
This gives us the completed initial simplex tableau given in Table 3.1

| $c_j$ / $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
|---|---|---|---|---|---|---|---|
| | | | 8 | 12 | 0 | 0 | 0 |
| 0 | $s_1$ | 1800 | 2 | 3 | 1 | 0 | 0 |
| 0 | $s_2$ | 600 | 1 | 2/3 | 0 | 1 | 0 |
| 0 | $s_3$ | 200 | 1/4 | ½ | 0 | 0 | 1 |
| $Z_j$ | | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_j - Z_j$ | | | 8 | 12 | 0 | 0 | 0 |

**Table 3.1 Initial Simplex Tableau**

We can read off the initial solution to be:

$s_1 = 1800$ $\quad x_1 = 0$

$s_2 = 600$ $\quad x_2 = 0$

$s_3 = 200$

The value of Z is given by:
Z = (0) 1800 + (0) 600 + (0)200= 0

Now, since $x_1 = 0$ and $x_2 = 0$, the initial feasible solution is at the origin. Obviously, this is not the best solution and we use the computational rules of the Simplex Algorithm to seek out the best solution (if it exists).

## *SAQ 1*

*Consider the Linear Program:*

*MAX* $Z = 3x_1 + 2x_2$

*Subject to*

$x_1 + 2x_2 \leq 6$

$2x_1 + x_2 \leq 8$

$-x_1 + x_2 \leq 1$

$x_2 \leq 2$

$x_1, x_2 \geq 0$

*Set up the initial simplex tableau for this problem.*

▶ See video **V08_Simplex_1**, which sets up the initial tableau for the loudspeaker linear programming problem (presented in the video V01_LPForm_1).
https://goo.gl/0tGUKl

## 3.4 EVALUATION OF THE SIMPLEX TABLEAU

Returning to the Initial Tableau we notice that in the $x_1$ and $x_2$ columns there are the numbers 8 and 12 respectively in the $C_j$ - $Z_j$ row. The existence of positive values in the $C_j$ - $Z_j$ row of a maximization problem indicates that the current solution is not optimal.

The "economic" interpretation of these numbers is that for each unit of $x_1$ or $x_2$ that is introduced into the solution the value of the objective function will increase by eight or twelve units i.e.:

| $x_1$ | | | $x_2$ | | |
|---|---|---|---|---|---|
| $x\ B_{1B} = 1$ | : | $Z = 8$ | $x\ B_{2B} = 1$ | : | $Z = 12$ |
| $x\ B_{1B} = 2$ | : | $Z = 16$ | $x\ B_{2B} = 2$ | : | $Z = 24$ |
| $x\ B_{1B} = 3$ | : | $Z = 24$ | $x\ B_{2B} = 3$ | : | $Z = 36$ |
| etc. | | | etc. | | |

It is usual to introduce only one decision variable at a time into the solution and since $x_2$ results in the fastest rate of profit growth, it will be introduced first into the solution.

Another glance at Table 3.1 shows that if $x_2$ is introduced into the solution either $s_1$, $s_2$ or $s_3$ must leave the solution. How do we determine which one should leave? Note that, if $x_2$ is in the solution and $x_1$ is not, $x_1 = 0$.

Looking anew at our original constraints under this scenario we have:

$0 + 3\ x_2 \leq 1800$ *or* $x_2 \leq 600$

$0 + 2/3\ x_2 \leq 600$ *or* $x_2 \leq 900$

$0 + ½\ x_2 \leq 200$ *or* $x_2 \leq 400$

Logically therefore for no constraint to be violated:

$x_2 \leq 400$

This is known as the **Minimum Ratio** value.

This value of $x_2$ is deduced from the third constraint, which contains the (slack) variable $s_3$. Therefore $s_3$ exits the solution in favour of $x_2$.

At this juncture, we have enunciated the following procedures of the Simplex Algorithm for the pure maximization problems.

***Step I:***
Set up the simplex tableau using the constraint coefficients with the initial solution located geometrically at the origin i.e,

$x_1, x_2, \ldots, x_n = 0$

and with each of the slack variables equal to its corresponding RHS value,

$s_1 = b_1,\ s_2 = b_2, \ldots, s_m = b_m.$

This is the initial solution.

***Step II:***
Compute the values of the $C_j$ - $Z_j$ Row. Then elect that decision variable with the largest positive value as the incoming variable to the (improved) solution. If one or more variables have the same positive value, select one at random.

***Step III:***
For the incoming decision variable divide each of its elements into the corresponding value in the RHS or $b_i$* column. Select the minimum positive value (that is, the minimum ratio) and hence determine the outgoing variable.

The application of these Simplex Rules results in the *second tableau* depicted in outline form in Table 3.2.

| $c_j$ | | | 8 | 12 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
| 0 | $s_1$ | | | | | | |
| 0 | $s_2$ | | | | | | |
| 12 | $x_2$ | | | | | | |
| $Z_j$ | | | | | | | |
| $C_j$ - $Z_j$ | | | | | | | |

Outgoing Variable and Pivot Row

The Pivot Element

Incoming Variable and Pivot Column

**Table 3.2: The Second (Simplex) Tableau.**

The decision variable $x_2$ now takes the $s_3$ cell in the Basis column and its objective function coefficient (12) is inserted into the corresponding cell in the $c_b$ column. We say that $x_2$ enters the solution and $s_3$ leaves.

## 3.5 COMPUTATIONAL RULES FOR THE SIMPLEX ALGORITHM

Table 3.2 portrays the second simplex tableau where $x_2$ has taken the space of $s_3$ in the variable column and its objective function coefficient (12) is located in the appropriate cell of the $c_b$ column. As you can see all of the other cells are blank. Before we compute their values it is instructive to pause and consider the implications of the entries in the Basis column, namely:

$s_1$; $s_2$ ; $x_2$

Since $x_1$ is not in the Basis column it is not in the current solution and therefore $x_1 = 0$. Geometrically this means that we are proceeding up the $x_2$ axis and away from the origin in the solution space. Now we know that optimal solutions (if they exist) can only exist at the vertices of the solution space. Therefore the second feasible solution must be at the Vertex A of Figure 2.2, a modified version of which is given in Figure 3.2.

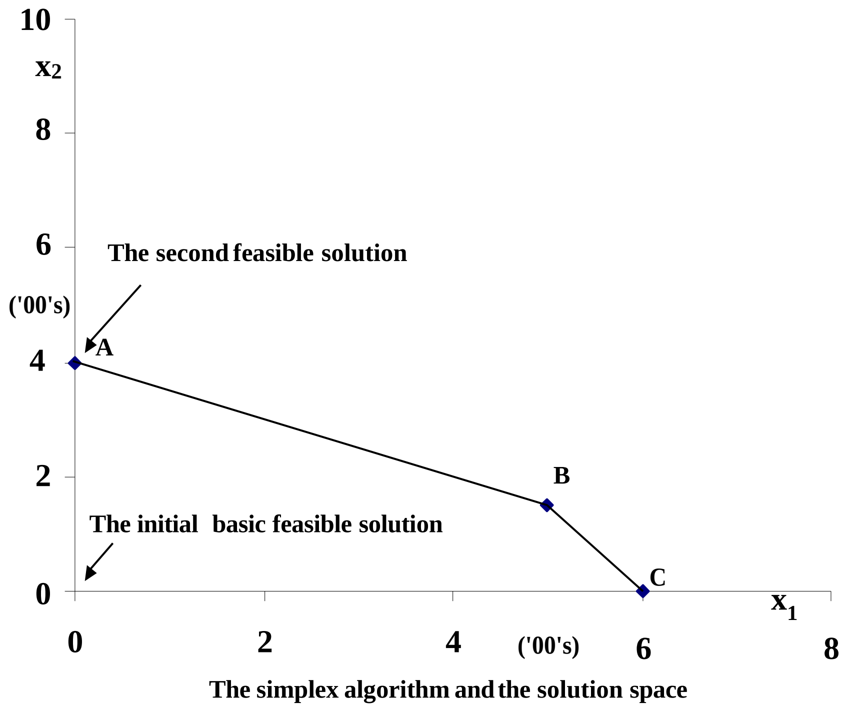



**Figure 3.2: The Simplex Algorithm and The Solution Space.**

To re-express the inequalities at the Vertex A (rather than at the origin) the Simplex Rules are:

Rule (I): Divide the pivot row by the pivot element
Rule (II): Compute all remaining elements using the formula:

New Element = Old Element - (Corresponding Element in Pivot Column × Corresponding Element in Pivot Row ÷ Pivot Element)

Since we are progressing from Table 3.1 to the completed version of Table 3.2,
Rule (I) requires us to divide the third ($s_3$) row of Table 3.1 by ½, the Pivot Element. On replacing $s_3$ by $x_2$ (for Table 3.2) we get:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 12 | $x_2$ | 400 | ½ | 1 | 0 | 0 | 2 |

Note that in Figure 3.2 the co-ordinates of the vertex A are:

$x_1 = 0$ ; $x_2 = 400$

Thus Rule I has re-expressed the third inequality in the necessary format for vertex A.

Table 3.2 is now:

| $c_j$ / $c_b$ | Basis | $b_i$* | 8<br>$x_1$ | 12<br>$x_2$ | 0<br>$s_1$ | 0<br>$s_2$ | 0<br>$s_3$ |
|---|---|---|---|---|---|---|---|
| 0 | $s_1$ | | | | | | |
| 0 | $s_2$ | | | | | | |
| 12 | $x_2$ | 400 | ½ | 1 | 0 | 0 | 2 |
| $Z_j$ | | | | | | | |
| $C_j - Z_j$ | | | | | | | |

We now have to re-express the first two inequalities in the necessary format for Vertex A, and to do so we employ Rule II as follows:

$b_i$*Column

[$s_1$, $b_i$* ] cell: New Element = 1800 - ((3 x 200) ÷ ½)

Note that 3 is the element corresponding to 1800 in the Pivot Column.
Note that 200 is the element corresponding to 1800 in the Pivot Row.

Thus we have:

[$s_1$, $b_i$*] cell: New element = 1800 – 1200 = 600

[$s_2$, $b_i$*] cell: New element = 600 - ((2/3 × 200)÷1/2)

Note that 2/3 is the *element corresponding* to 600 in the *Pivot Column*
Note that 200 is the *element corresponding* to 600 in the *Pivot Row*

Thus we have:

[$s_2$, $b_i$* ] cell: New element = 600 - 800/3 = 1000/3

[$Z_j$, $b_i$*] cell: $Z_j = 600 \times 0 + (1000/3) \times 0 + 400 \times 12$

$= 0 + 0 + 4800$

= 4800

Table 3.2 is now:

| $c_j$ / $c_b$ | Basis | $b_i$* | 8<br>$x_1$ | 12<br>$x_2$ | 0<br>$s_1$ | 0<br>$s_2$ | 0<br>$s_3$ |
|---|---|---|---|---|---|---|---|
| 0 | $s_1$ | 600 | | | | | |
| 0 | $s_2$ | 1000/3 | | | | | |
| 12 | $x_2$ | 400 | ½ | 1 | 0 | 0 | 2 |
| $Z_j$ | | 4800 | | | | | |
| $C_j - Z_j$ | | | | | | | |

It is always wise to compute the $b_i$* column first since, in a maximization problem, each successive iteration of the simplex algorithm should, in general, increase the value of the objective function up to its maximum value.

We now re-employ Rule (II) to fill in the remaining blank cells. However, before we do so let us have a further look at rule (II) and make some pertinent observations:

Rule (II)

$$\text{New Element} = \text{Old Element} - \left(\text{Corresponding Element in Pivot Column} \times \text{Corresponding Element in Pivot Row} \div \text{Pivot Element}\right)$$

Observation (i)
If either of the corresponding elements in the Pivot Column or Pivot Row are zero then:

New Element = Old Element

Observation(ii)
In computing the new elements for the *Pivot Column* note that the corresponding element in the Pivot Row is the Pivot Element. Thus we have:

$$\text{New Element} = \text{Old Element} - \left(\text{Corresponding Element in Pivot Column} \times \text{Pivot Element} \div \text{Pivot Element}\right)$$

i.e

New Element = Old Element - Corresponding Element in Pivot Column

New Element = Old Element - Old Element

New Element = 0

Observation (iii)
In the Initial Tableau, current solution variables in the Basis column in Table 3.1 i.e. $s_1$; $s_2$; $s_3$ have configurations of:

$$\begin{bmatrix}1\\0\\0\end{bmatrix} \quad \begin{bmatrix}0\\1\\0\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}0\\0\\1\end{bmatrix}$$

That is 1 at their intercept and 0 elsewhere.
Applying these observations Table 3.2 now becomes:

| $C_j$ / $C_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
|---|---|---|---|---|---|---|---|
| | | | 8 | 12 | 0 | 0 | 0 |
| 0 | $s_1$ | 600 | | 0 | 1 | 0 | |
| 0 | $s_2$ | 1000/3 | | 0 | 0 | 1 | |
| 12 | $x_2$ | 400 | ½ | 1 | 0 | 0 | 2 |
| $Z_j$ | | 4800 | | | | | |
| $C_j$ - $Z_j$ | | | | | | | |

Thus all we now have to do is use RULE (II) for the $x_1$ and $s_3$ columns and then re-evaluate the $C_j$ - $Z_j$ Row. See if you can do this without looking at the final version of Table 3.2 re-numbered 3.3 and make your comparisons.

| $c_j$ / $c_b$ | Basis | $b_i$* | 8<br>$x_1$ | 12<br>$x_2$ | 0<br>$s_1$ | 0<br>$s_2$ | 0<br>$s_3$ |
|---|---|---|---|---|---|---|---|
| 0 | $s_1$ | 600 | ½ | 0 | 1 | 0 | -6 |
| 0 | $s_2$ | 1000/3 | 2/3 | 0 | 0 | 1 | -4/3 |
| 12 | $x_2$ | 400 | ½ | 1 | 0 | 0 | 2 |
| $Z_j$ | | 4800 | 6 | 12 | 0 | 0 | 24 |
| $C_j$ - $Z_j$ | | | 2 | 0 | 0 | 0 | -24 |

Pivot Row

Pivot Element

Pivot Column

**Table 3.3 The Second Feasible (Simplex) Solution**

## 3.6 SOLVING THE MAXIMIZATION PROBLEM AND INTERPRETING THE SOLUTION

The second feasible solution to the maximization problem displayed in Table 3.3 is:

$s_1 = 600$
$s_2 = 1000/3$
$x_2 = 400$
$Z = 4800$

However there is a 2 in the $C_j$ - $Z_j$ Row of the $x_1$ variable indicating that the current solution is not optimal and that $x_1$ should enter the solution. Thus $x_1$ constitutes the *Pivot Column*.

Dividing each of the $x_1$ elements into its opposite number in the $b_i$* column yields:

1200 500 800

If desired an extra column (the ratio column) may be added to the simplex tableau to store these ratio values.

Thus 500 is the minimum ratio and the variable $s_2$ makes way for $x_1$. The $s_2$ row is thus the *Pivot Row* and 2/3 is the *Pivot Element*. Employing Simplex RULE (I) together with *Observations (i), (ii)* and *(iii).* We can construct the partially completed Table 3.4.

| $c_j$ / $c_b$ | Basis | $b_i$* | 8<br>$x_1$ | 12<br>$x_2$ | 0<br>$s_1$ | 0<br>$s_2$ | 0<br>$s_3$ |
|---|---|---|---|---|---|---|---|
| 0 | $s_1$ | | 0 | 0 | 1 | | |
| 8 | $x_1$ | 500 | 1 | 0 | 0 | 3/2 | -2 |
| 12 | $x_2$ | | 0 | 1 | 0 | | |
| $Z_j$ | | | | | | | |
| $C_j$ - $Z_j$ | | | | | | | |

**Table 3.4 The Third Feasible (Simplex) Solution**

However, the $b_i$*, $s_2$ and $s_3$ columns are incomplete since Simplex Rule II has not been applied. The application of RULE II results in Table 3.5. See if you can do it yourself without reference to the finished table.

| $C_j$ / $C_b$ | Basis | $b_i$* | 8<br>$x_1$ | 12<br>$x_2$ | 0<br>$s_1$ | 0<br>$s_2$ | 0<br>$s_3$ |
|---|---|---|---|---|---|---|---|
| 0 | $s_1$ | 350 | 0 | 0 | 1 | -3/4 | -5 |
| 8 | $x_1$ | 500 | 1 | 0 | 0 | 3/2 | -2 |
| 12 | $x_2$ | 150 | 0 | 1 | 0 | - 3/4 | 3 |
| $Z_j$ | | 5800 | 8 | 12 | 0 | 3 | 20 |
| $C_j$ - $Z_j$ | | | 0 | 0 | 0 | -3 | -20 |

**Table 3.5: The Optimal Solution to the Maximization Problem**

Since no positive entries are in the $C_j$ - $Z_j$ Row we have reached the *Optimal Solution* which is:

$Z^* = 5800$; $x_1^* = 500$; $x_2^* = 150$; $s_1^* = 350$

▶ See video **V09_Simplex_2**, which goes through the first iteration of the simplex method. (This follows from the initial tableau in the previous video, V08_Simplex_1.) https://goo.gl/hN7iHZ

▶ See video **V10_Simplex_3**, which proceeds to a second iteration to reach the optimum solution. (This follows from the tableau in the previous video, V09_Simplex_2.) https://goo.gl/uruc0S

This is identical to the solution achieved by graphical methods in Unit 2, Figure 2.2. By using Simplex Calculations we have also confirmed that $b_i$* is the Optimal Vertex in Figure 3.2. The $C_j$ - $Z_j$ Row, however, contains further information of interest. You will note that $x_1$, $x_2$ and $s_1$ are in the optimal solution and that they have zero (0) entries in the $C_j$ - $Z_j$ row. This is to be expected since none of these variables can contribute further to the value of the objective function. However under $s_2$ and $s_3$ we have -3 and -20 respectively. It should be understood that, whereas the optimal solution has been achieved, we can, if we wish, force $s_2$ and /or $s_3$ into the solution.

With regard to $s_2$, the outgoing variable would be $x_1$ *as minimum ratios are never formed using negative elements.* The use of negative elements induces a "looping procedure" into the simplex method, which is to be avoided at all costs. However, following the Simplex Rules, if we wished to introduce $s_2$ into the solution in place of $x_1$, we would divide the Pivot Row by 3/2 which would yield a value for $s_2$ of $500 \times 2/3$ or 1000/3. The consequence of doing so however would be to *reduce* the value of the objective function by $(1000/3) \times 3$ or 1000. This is easily proven by using the Simplex Rules and results in:

| $c_b$ | Basis | $b_i$* |
|---|---|---|
| 0 | $s_1$ | 600 |
| 0 | $s_2$ | 1000/3 |
| 12 | $x_2$ | 400 |
| Zj | | 4800 |

You should test your skill by re-producing the above result.

Therefore, the existence of negative values under $s_2$ and $s_3$ indicates that, for each unit of $s_2$ and $s_3$ introduced into the current (Optimal) solution, its value will be reduced by 3 and 20 units respectively.

- ▶ See video **V11_Simplex_4**, which provides an interpretation of the final tableau. (This follows from the previous video, V10_Simplex_3.) https://goo.gl/NpyNUl
- ▶ See video **V12_Simplex_5**, which involves the interpretation of a final simplex tableau. https://goo.gl/HCGUjO

## *SAQ 2*

*Solve the following Linear Program using the SIMPLEX Algorithm:*

*MAX* $Z = 50x_1 + 40x_2$

*Subject to:*

$3x_1 + 5x_2 \leq 150$

$x_2 \leq 20$

$8x_1 + 5x_2 \leq 300$

$x_1, x_2 \geq 0$

## *SAQ 3*

*Solve the following Linear Program using the SIMPLEX Algorithm:*

*MAX* $Z = 100x_1 + 80x_2$

*Subject to:*

$2x_1 + x_2 \leq 12$

$x_2 \leq 8$

$x_1, x_2 \geq 0$

## *SAQ 4*

*Solve the following problem using the simplex algorithm:*

*Max Z = 60 $x_1$ + 50 $x_2$ + 45 $x_3$ + 50 $x_4$*

*Subject to:*
$x_2 \leq 20$
$x_4 \leq 15$
$10 x_1 + 5 x_2 \leq 120$
$8 x_3 + 6 x_4 \leq 135$
$x_1, x_2, x_3, x_4 \geq 0$

## *SAQ 5*

*A small company specialises in the assembly of watsits. Basically, the company assembles two types of watsit, and the assembly process has three resource restrictions – assembly time, inspection time and storage space. The owner uses part-time assembly staff who work a maximum of 100 hours between them and she employs a part-time inspector/tester who can work a maximum of 22 hours per week. There is a total of 39 units of storage space available for the finished product. Each watsit of type one takes 4 hours to assemble, 2 hours to inspect/test and requires 3 units of storage space. Each watsit of type two takes 10 hours to assemble, 1 hour to inspect/test and requires 3 units of storage space. The profit on type 1 is €60 and the profit on type 2 is €50.*

*The owner wishes to maximize her total profit from the process.*

*Construct an appropriate linear programme model and solve using the Simplex Algorithm.*

### 3.7 Preparing a Minimizaton Model for the Simplex Algorithm

In Unit 2, section 2.7, we graphically solved the minimization problem:

Min Z = $6 y_1 + 5 y_2$

Subject to:
$15 y_1 + 30 y_2 \geq 30$
$30 y_1 + 20 y_2 \geq 50$
$y_1 \geq 0$
$y_2 \geq 0$

In this case the constraints are *Lower Bounds* and thus the LHS is greater than the RHS. In order to achieve equality, we must therefore subtract the slack variables:

$15 y_1 + 30 y_2 - s_1 = 30$

$30 y_1 + 20 y_2 - s_2 = 50$

However, the Simplex Algorithm requires that there be + 1 down the main diagonal and zeroes (0) elsewhere at the right-most extremity of the detached coefficient array, (see Table 3.1). Such an entity is known as the

*Identity Matrix* (I). For the above problem, therefore we need the construct,

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

which obviously can't be achieved. Recognising this problem, Dantzig created the concept of the A*rtificial Variable* as follows:

$15\ y_1 + 30\ y_2 - s_1 + A_1 = 30$

$30\ y_1 + 20\ y_2 - s_2 + A_2 = 50$

Where $A_1$ and $A_2$ are *Artificial Variables*. This then gives the Identity Matrix at the right of the coefficient array. However, unlike Slack Variables, Artificial Variables have no meaning other than they are "mathematical ploys". As you have seen, Slack Variables can appear in Optimal Solutions and can be interpreted as "unused resources". However, Artificial Variables have no meaning and cannot appear in optimal solutions. Again, Dantzig recognised this fact and gave them very large coefficients in the objective function, namely 1,000,000 or M for compactness. Thus, since the objective is to minimize, the Artificial Variables will not appear in the optimal solution. The minimization model thus takes the form:

Min C = $6\ y_1 + 5\ y_2 + 0\ s_1 + 0\ s_2 + M\ A_1 + M\ A_2$

Subject to:

$15\ y_1 + 30\ y_2 - s_1 + A_1 = 30$
$30\ y_1 + 20\ y_2 - s_2 + A_2 = 50$
$y_1 \geq 0$
$y_2 \geq 0$
$s_1 \geq 0$
$s_2 \geq 0$
$A_1 \geq 0$
$A_2 \geq 0$

The resulting detached coefficient array is depicted in Table 3.6

Initial Tableau

| $C_j$ / $C_b$ | Basis | $b_i$* | 6 | 5 | 0 | 0 | M | M |
|---|---|---|---|---|---|---|---|---|
| | | | $y_1$ | $y_2$ | $s_1$ | $s_2$ | $A_1$ | $A_2$ |
| M | $A_1$ | 30 | 15 | 30 | -1 | 0 | 1 | 0 |
| M | $A_2$ | 50 | 30 | 20 | 0 | -1 | 0 | 1 |
| $Z_j$ | | 80M | 45M | 50M | -M | -M | M | M |
| $C_j - Z_j$ | | | 6-45M | 5-50M | M | M | 0 | 0 |

**Table 3.6: The Detached Coefficient Array for The Minimization Problem**

In this Tableau the following should be noted:

(i) The Artificial Variables are in the Basis column, and thus represents the first "solution".
(ii) The Cj - Zj Row contains a number of M's i.e. $10^6$s.
(iii) In the *minimization problem* the existence of *negative numbers* in the Cj - Zj Row indicates the solution can be improved.

(iv) The Simplex calculations therefore continue until *no negative numbers* exist in the Cj - Zj Row.

Thus, since the largest negative number -50M, or "minus 50 million" is at the foot of the $y_2$ column, this is designated as the *Pivot Column.* The minimum ratio is 30/30 - thus the $A_1$ row is designated as the *Pivot Row* and 30 as the *Pivot Element*. Table 3.7 presents the Tableau after the application of Simplex Rule (I).

| $c_j$ / $c_b$ | Basis | $b_i$* | 6<br>$y_1$ | 5<br>$y_2$ | 0<br>$s_1$ | 0<br>$s_2$ | M<br>$A_1$ | M<br>$A_2$ | |
|---|---|---|---|---|---|---|---|---|---|
| 5 | $y_2$ | 30/30 | 15/30 | 1 | -1/30 | 0 | 1/30 | 0 | ← Pivot Row |
| M | $A_2$ | | | | | | | | |
| $Z_j$ | | | | | | | | | |
| $C_j$ - $Z_j$ | | | | ↑ Pivot Column | | | | | |

**Table 3.7: The Initial Format of the second Simplex Tableau.**

The blank cells can have their values computed using simple RULE (II) and is as shown in Table 3.8.

| $c_j$ / $c_b$ | Basis | $b_i$* | 6<br>$y_1$ | 5<br>$y_2$ | 0<br>$s_1$ | 0<br>$s_2$ | M<br>$A_1$ | M<br>$A_2$ | |
|---|---|---|---|---|---|---|---|---|---|
| 5 | $y_2$ | 30/30 | 15/30 | 1 | -1/30 | 0 | 1/30 | 0 | |
| M | $A_2$ | 30 | 20 | 0 | 2/3 | -1 | -2/3 | 1 | ← Pivot Row |
| $Z_j$ | | 30M+5 | 20M+5/2 | 5 | 2/3M-1/6 | -M | 1/6-2/3M | M | |
| $C_j$ - $Z_j$ | | | 7/2-20M | 0 | 1/6-2/3M | M | -1/6 +5/3M | 0 | |
| | | | ↑ Pivot Column | | | | | | |

**Table 3.8: The Second Simplex Tableau**

See if you can reproduce this tableau.

Note
(i) $y_1$ is the *Pivot Column*
(ii) $A_2$ is the *Pivot Row*
(iii) 20 is the *Pivot Element*
(iv) The first two interactions of the simplex Algorithm result in both artificial variables being removed from the Basis column.

The application of RULES (I) AND (II) results in Table 3.9.

| $c_j$ / $c_b$ | Basis | $b_i$* | 6<br>$y_1$ | 5<br>$y_2$ | 0<br>$s_1$ | 0<br>$s_2$ | M<br>$A_1$ | M<br>$A_2$ |
|---|---|---|---|---|---|---|---|---|
| 5 | $y_2$ | 10/40 | 0 | 1 | -1/20 | 1/40 | 1/20 | -1/40 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 6 | $y_1$ | 30/20 | 1 | 0 | 1/30 | -1/20 | -1/30 | 1/20 |
| $Z_j$ | | 410/40 | 6 | 5 | -1/20 | -7/40 | 1/20 | 7/40 |
| $C_j - Z_j$ | | | 0 | 0 | 1/20 | 7/40 | M-1/20 | M-7/40 |

**Table 3.9: The Third Simplex Tableau**

Since no negative numbers are in the Cj - Zj Row, the third Simplex Tableau represents the optimal Solution and is:

$Z^* = 410/40 = 10.25$

$y_1^* = 30/20 = 1.5$ kg

$y_2^* = 10/40 = 0.25$ kg

This agrees with the solution obtained graphically in Unit 2.9.

## *SAQ 6*

*A coffee shop, which also serves breakfast, wishes to ensure that minimum daily requirements (mdr) for vitamins A & B are met whilst their breakfasts are prepared at the lowest possible cost. The breakfast ingredients providing the vitamins are eggs, bacon & cereal. The vitamin data and cost of the breakfast ingredients are:*

*VITAMIN CONTRIBUTION*

| *VITAMIN* | *mg/egg* | *mg/Bacon Strip* | *mg/cereal* | *mdr (kg)* |
|---|---|---|---|---|
| *A* | *2* | *4* | *1* | *16* |
| *B* | *3* | *2* | *1* | *12* |
| *Cost* | *40c/egg* | *30c/strip* | *20c/cup* | |

*How much of each ingredient should be served to meet the minimum daily requirements whilst minimising the cost of the breakfast?*

## 3.8 SOME EXAMPLES OF MIXED CONSTRAINTS

Very few problems in the "real world" can be modelled with all constrains of the same type i.e all ≤ or ≥ or =. Most commonly the constraints are a mixture of the three types.

We have seen how inequalities are transformed into equalities. But supposing, at the outset, the relationship is a strict equality, what then? Suppose, for instance, we have

$4x_1 + 7x_2 + 3x_3 = 120$

The problem here is that although the Simplex requirement for an equality already exists, there is no variable to act as " a pivot" from which the simplex algorithm can generate an improved solution from the initial basic feasible solution at the origin. The problem is simply solved by inserting an artificial variable:

$4x_1 + 7x_2 + 3x_3 + A_1 = 120$

and, of course, having its coefficient as M in the objective function. Remember that, if the problem is to minimize, + M is the coefficient whereas – M is used for maximization. Suppose we have a maximization model of the form:

$$\text{Max } Z = 5x_1 + x_2 + 3x_3$$

Subject to:

$x_1 + x_2 \geq 40$
$2x_1 + 3x_2 + x_3 \leq 50$
$3x_1 + 2x_2 + 2x_3 = 25$
$x_1 + x_3 \geq 10$
$x_1, x_2, x_3 \geq 0$

The correct format for the simplex Algorithm is thus:

$$\text{Max } Z = 5x_1 + x_2 + 3x_3 + 0s_1 - MA_1 + 0s_2 - MA_2 + 0s_3 - MA_3$$

Subject to:

$x_1 + x_2 - s_1 + A_1 = 40$
$2x_1 + 3x_2 + x_3 + s_2 = 50$
$3x_1 + 2x_2 + 2x_3 + A_2 = 25$
$x_1 + x_3 - s_3 + A_3 = 10$

The first simplex Tableau (and first feasible solution) is shown in Table 3.10

| $c_j$ | | | 5 | 1 | 3 | 0 | -M | 0 | -M | 0 | -M |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $x_3$ | $s_1$ | $A_1$ | $s_2$ | $A_2$ | $s_3$ | $A_3$ |
| -M | $A_1$ | 40 | 1 | 1 | 0 | -1 | 1 | 0 | 0 | 0 | 0 |
| 0 | $s_2$ | 50 | 2 | 3 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| -M | $A_2$ | 25 | 3 | 2 | 2 | 0 | 0 | 0 | 1 | 0 | 0 |
| -M | $A_3$ | 10 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | -1 | 1 |
| $Z_j$ | | -75M | -5M | -3M | -3M | -M | -M | 0 | -M | M | -M |
| $C_j - Z_j$ | | | 5+5M | 1+3M | 3+3M | M | 0 | 0 | 0 | -M | 0 |

Pivot row

Pivot Column

Pivot Element

---

**Table 3.10 The Simplex Tableau for a Mixed Constraint Problem.**

---

Note:

(i) The normal rules of the simplex apply – but it is a lengthy calculation.
(ii) Computer packages automatically adjust the problem to the required format and (for the above sized problem) reach the solution in seconds, if it exists.
(iii) If an optimal solution does not exist the computer program should establish this fact and indicate the reason why.

(iv) In terms or the simplex Algorithm, we recognize infeasibility as the case where at least one of the Artificial Variables remains in the final tableau at a positive value.

(v) This, in effect, means that there is no solution which satisfies all the constraints simultaneously.

## *SAQ 7*

*Use the simplex algorithm to solve the following mixed constraint problem:*

*Max* $Z = 200x_1 + 100x_2$

*Subject to:*
$x_1 + x_2 = 1500$
$x_1 \leq 400$
$x_2 \geq 200$
$x_1, x_2 \geq 0$

## 3.9 SHADOW PRICES

In Unit 2 the idea of a shadow price was first introduced. This was defined as the effect on the objective function of adding an additional unit of a resource. In Unit 2 we calculated the shadow price by calculating what the new optimal solution would be if the resource availability was changed by one unit. The difference between the new and old optimal objective functions is the shadow price. In this section we will examine the idea of shadow prices using the simplex tableau and it will be useful to you to refer back to the graphical solutions in Unit 2.

In section 3.6 above it was explained that the index row (i.e. the last row) of the simplex tableau represents the potential change to the objective function of introducing a unit of a variable into the solution. When the optimal solution is reached all values in the index row will be zero or negative, indicating that no further improvement in the objective function is possible. In the index row the columns associated with the basic variables will inevitably be zero, but some of the columns representing slack variables will have negative values. The interpretation that we put on these values is that bringing a slack variable into the basis would disimprove the objective function.

Since the slack variables are associated with constraints it would seem to make sense that these index row values would offer some information about the effect of changing the resource availability. In fact, we can interpret an increase in the value of a slack variable as decreasing the amount of the resource being used. Therefore, if the index row tells us the effect of increasing the value of the slack variable, it also tells us the effect of reducing the availability of the resource associated with that constraint.

Since the problem is, by definition, linear; if we know the effect on the objective function of decreasing resource availability then we also know the effect of increasing that resource. In an optimal tableau for a maximization problem the entries in the index row in the columns representing the slack variables will be zero or negative. Therefore, increasing the slack variable (or decreasing the resource) will reduce the objective function that we are trying to maximize. The value in the index row for a column representing a slack variable is the amount by which the objective function will decrease if a resource is decreased by one unit. Since the problem is linear, this figure also represents the increase in the objective function if a resource is increased by one unit. It is this effect on the objective function of increasing the scarce resource by one unit that was called the shadow price in Unit 2.

Consider the problem presented in SAQ 2 and SAQ5 of Unit 2.
Let $x_1$ = number of chairs produced
$x_2$ = number of tables produced.

Maximize: $15 x_1 + 20 x_2$

Subject to:

(i) $x_1 + 2x_2 \leq 250$ (planing time constraint)

(ii) $x_1 + x_2 \leq 150$ (cutting time constraint)

(iii) $x_1 \leq 125$ (turning time constraint)

(iv) $x_2 \leq 140$ (painting time constraint)

In this case we solved the problem graphically and calculated the shadow prices. The tableau below is the optimal tableau of this problem solved using the simplex method. In the simplex solution the variables $x_1$ and $x_2$ are used in place of x and y.

| $c_j$ | | | 15 | 20 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
| 20 | $x_2$ | 100 | 0 | 1 | 1 | -1 | 0 | 0 |
| 15 | $x_1$ | 50 | 1 | 0 | -1 | 2 | 0 | 0 |
| 0 | $s_3$ | 75 | 0 | 0 | 1 | -2 | 1 | 0 |
| 0 | $s_4$ | 40 | 0 | 0 | -1 | 1 | 0 | 0 |
| $Z_j$ | | 2750 | 15 | 20 | 5 | 10 | 0 | 0 |
| $C_j - Z_j$ | | | 0 | 0 | -5 | -10 | 0 | 0 |

Optimal Answer: $x_1 = 50$, $x_2 = 100$ and P = 2750.

## *SAQ 8*

*Solve the above problem using the simplex method and verify that this is in fact the optimal tableau. Identify on the graph the point through which the simplex method passes.*

From the tableau we see that constraints (i) and (ii) are binding while constraints (iii) and (iv) are not since the slacks $s_3$ and $s_4$ are in the basis. If we look at the index row of the optimal tableau we find that increasing slack $s_1$ will reduce the optimal solution by 5. This is equivalent to saying that reducing the planing time available will reduce the profit by €5. An increase in planing time of one hour would increase profit by €5, therefore we can say that the shadow price of planing time is €5. This is the value calculated in Unit 2.

In a similar way the shadow price of constraint 2, cutting time, can be identified as €10 from the tableau. Since the objective in this problem is to maximize profit we can assume that the coefficients of the variables in the objective function are the difference between the sales price of the product and its cost. Therefore, we can interpret the shadow price as an additional price we would be willing to pay for more of each resource beyond any cost that is already included in the equations.

The useful nature of shadow prices can be seen if we once again look at the problems of Mac's Furniture Ltd which was first introduced in SAQ 3 of Unit 2. The problem is as follows:

Maximize: $17x_1 + 12x_2$
Subject to:

(i) $0.5x_1 + x_2 \leq 40$

(ii) $2x_1 + 3x_2 \leq 60$

(iii) $6x_1 + 1.5x_2 \leq 80$

$x_1 \geq 0$ and $x_2 \geq 0$

The optimal tableau is given below:

| $c_b$ \ $c_j$ | | | 17 | 12 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
| 0 | $s_1$ | 21.67 | 0 | 0 | 1 | -0.35 | 0.03 |
| 12 | $x_2$ | 13.33 | 0 | 1 | 0 | 0.40 | -0.13 |
| 17 | $x_1$ | 10 | 1 | 0 | 0 | -0.10 | 0.2 |
| $Z_j$ | | 330 | 17 | 12 | 0 | 3.1 | 1.8 |
| $C_j - Z_j$ | | | 0 | 0 | 0 | -3.1 | -1.8 |

Optimal Answer: $x_1 = 10$, $x_2 = 13.33$ and $P = 330$, which we already know from the graphical solution.

The cutting time constraint is not binding, therefore has a shadow price of zero. From the tableau we can find the shadow prices for assembly time and finishing time.

Shadow price for each hour of assembly time = 3.10
Shadow price for each hour of finishing time = 1.80

How do we interpret these shadow price values? If an extra hour of assembly becomes available, profit will increase by €3.10. An extra hour in the finishing department would increase profit by only €1.80.

## SAQ 9

*A company specialised in the assembly of PCs. The production manager has developed the following linear program to help him decide on the optimal assembly strategy. Basically, the company assembles two types of PC, and the assembly process has three resource restrictions. The production manager wishes to maximize his total profit from the process. The linear program is as follows:*

$x_1$ = *quantity of Model 1 PC*
$x_2$ = *quantity of Model 2 PC*

$MAX\ Z = 60\ x_1 + 50\ x_2$

*Subject to*

$4x_1 + 10\ x_2 \leq 100$ *(Assembly Time Constraint)*
$2x_1 + x_2 \leq 22$ *(Inspection Time Constraint)*
$3x_1 + 3x_2 \leq 39$ *(Storage Space Constraint)*
$x_1 \geq 0$ *and* $x_2 \geqslant 0$

*Solve the above problem using the Simplex Algorithm. Calculate,*

*and interpret, the Shadow prices for the 3 constraints.*

### 3.10 Right Hand Side Ranges

In Unit 2 the concept was introduced of a range of the right hand side values of the constraints within which the shadow prices remain valid. If we change resource availability outside this range the optimal solution will change and the shadow prices of the old solution will become invalid. In Unit 2 we were able to identify this range using the graphical approach, but for larger problems we need to be able to find this range from the simplex tableau.

Let us assume that we are going to make a change $\Delta_i$ to the right hand side of constraint i. This change is small enough for the optimal basis to remain unchanged. Given the nature of the simplex operations any change in the right hand side of one of the constraints will only affect the $b_i$* column in the simplex tableau. If we again consider the problem of the carpentry workshop which was used in SAQ 2 of Unit 2.

Simplex Tableau: Iteration 0:

| $c_j$ / $c_b$ | Basis | $b_i$* | 15<br>$x_1$ | 20<br>$x_2$ | 0<br>$s_1$ | 0<br>$s_2$ | 0<br>$s_3$ | 0<br>$s_4$ |
|---|---|---|---|---|---|---|---|---|
| 0 | $s_1$ | $250+1\Delta_1$ | 1 | 2 | 1 | 0 | 0 | 0 |
| 0 | $s_2$ | 150 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | $s_3$ | 125 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | $s_4$ | 140 | 0 | 1 | 0 | 0 | 0 | 1 |
| $Z_j$ | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_j$ - $Z_j$ | | | 15 | 20 | 0 | 0 | 0 | 0 |

Simplex Tableau: Iteration 1:

| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
|---|---|---|---|---|---|---|---|---|
| | | $c_j$ | 15 | 20 | 0 | 0 | 0 | 0 |
| 20 | $x_2$ | $125 + 0.5\Delta_1$ | 0.5 | 1 | 0.5 | 0 | 0 | 0 |
| 0 | $s_2$ | $25 - 0.5\Delta_1$ | 0.5 | 0 | -0.5 | 1 | 0 | 0 |
| 0 | $s_3$ | 125 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | $s_4$ | $15 - 0.5\ \Delta_1$ | -0.5 | 0 | -0.5 | 0 | 0 | 1 |
| $Z_j$ | | 2500 | 10 | 20 | 10 | 0 | 0 | 0 |
| $C_j - Z_j$ | | | 5 | 0 | -10 | 0 | 0 | 0 |

Simplex Tableau: Iteration 2:

| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
|---|---|---|---|---|---|---|---|---|
| | | $c_j$ | 15 | 20 | 0 | 0 | 0 | 0 |
| 20 | $x_2$ | $100 + 1\Delta_1$ | 0 | 1 | 1 | -1 | 0 | 0 |
| 15 | $x_1$ | $50 – 1\Delta_1$ | 1 | 0 | -1 | 2 | 0 | 0 |
| 0 | $s_3$ | $75 + 1\Delta_1$ | 0 | 0 | 1 | -2 | 1 | 0 |
| 0 | $s_4$ | $40 - 1\ \Delta_1$ | 0 | 0 | -1 | 1 | 0 | 1 |
| $Z_j$ | | 2750 | 15 | 20 | 5 | 10 | 0 | 0 |
| $C_j - Z_j$ | | | 0 | 0 | -5 | -10 | 0 | 0 |

In order for the basis to remain unchanged all the basic variables must remain non-negative. So the value of $\Delta_1$ must not cause any entry in the $b_i$* column to become negative. In the final tableau above we have calculated the effect of the change by entering $250 + \Delta_1$ in the initial tableau and performing the iterations using this value. If we follow the iterations through the simplex process we find that the coefficients of the amount $\Delta_1$ that is added to the $b_i$* column are exactly the same as the values in the $s_1$ column. This means that we don't have to perform all the iterations again, but we can find the values of $\Delta_1$ simply by looking at the $s_1$ column in the final tableau of the problem.

Therefore, the change in the above problem must be such that all values in the $b_i$* column are non-negative. From the above tableau we can see that for a positive $\Delta_1$, $x_2$ and $s_3$ must remain in the basis since $\Delta_1$ is added to the $b_i$* value. However, we must calculate the value of $\Delta_1$ which would make $x_1$ or $s_4$ leave the basis.

$x_1$: $\quad 50 - 1\Delta_1 \geq 0 \quad \Rightarrow \quad \Delta_1 \leq 50$
$s_4$: $\quad 40 - 1\Delta_1 \geq 0 \quad \Rightarrow \quad \Delta_1 \leq 40$

The maximum value of $\Delta_1$ is given by the smaller of these values so the right hand side of constraint 1 can have a value $\Delta_1$ added to it provided $\Delta_1$ is less than 40. If we want to find by how much the right hand side of equation 1 can decrease, we can use the $\Delta_1$ entries with positive coefficients. In this case $x_1$ and $s_4$ will remain in the basis and we must calculate by how much $\Delta_1$ can decrease to keep $x_2$ and $s_3$ in the basis

$x_2$: $\quad 100 + 1\Delta_1 \geq 0 \quad \Rightarrow \quad \Delta_1 \geq -100$
$s_3$: $\quad 75 + 1\Delta_1 \geq 0 \quad \Rightarrow \quad \Delta_1 \geq -75$

Therefore, the right hand side of constraint 1 may not decrease by more than

Minimum value: 250 - 75 = 175
Maximum value: 250 + 40 = 290

Hence the shadow price for planing time remains valid if planing time availability remains within the range 175 hours to 290 hours as we previously found using the graphical approach.

Using the $s_2$ column together with the $b_i$* column we can find from the tableau the range within which the cutting time may vary without changing the basis.

Maximum increase ($\Delta_2$ positive)

$x_2$: $100 - 1\Delta_2 \geq 0 \Rightarrow \Delta_2 \leq 100$
$s_3$: $75 - 2\Delta_2 \geq 0 \Rightarrow \Delta_2 \leq 37.5$

Maximum decrease ($\Delta_2$ negative)

$x_1$: $50 + 2\Delta_2 \geq 0 \Rightarrow \Delta_2 \geq -25$
$s_4$: $40 + 1\Delta_2 \geq 0 \Rightarrow \Delta_2 \geq -40$

The range within which cutting time may vary is from

Minimum value:150 - 25 = 125
Maximum value: 150 + 37.5 = 187.5

Hence the shadow price for cutting time remains valid if cutting time availability remains within the range 125 hours to 187.5 hours as we previously found using the graphical approach in Unit 2.

▶ See video **V13_Simplex_6**, which shows how to use the final simplex tableau to find the right hand side range of a constraint.
https://goo.gl/Sn7gKc

## *SAQ 10*

*The following tableau is the optimal tableau for the problem:*

*MAX* $Z = 4x_1 + 5x_2$

*Subject to*
$4x_1 + 1x_2 \leq 56$
$5x_1 + 3x_2 \leq 105$
$x_1 + 2x_2 \leq 56$
$x_1 \geq 0$ *and* $x_2 \geq 0$

| $c_j$ / $c_b$ | Basis | bi* | 4 | 5 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| | | | x1 | x2 | s1 | s2 | s3 |
| 0 | s1 | 7 | 0 | 0 | 1 | -1 | 1 |
| 4 | X1 | 6 | 1 | 0 | 0 | 0.29 | -0.43 |
| 5 | x2 | 25 | 0 | 1 | 0 | -0.14 | 0.71 |
| Zj | | 140 | 4 | 5 | 0 | 0.43 | 1.86 |
| Cj - Zj | | | 0 | 0 | 0 | -0.43 | -1.86 |

*From the tableau:*
*(i) Identify the shadow prices of the resources*
*(ii) Calculate the ranges for the right hand sides of the constraints within which the solution remains optimal.*
*(iii) Verify your answers using the graphical approach.*

## 3.11 CHANGES IN COEFFICIENTS OF BASIC VARIABLES

Consider the following problem to determine the impact of changes in the objective function coefficients of basic variables. Reviewing the problem of Mac's furniture which we have used previously, let us assume that Mac feels that he might charge more for tables and therefore increase his profit margin, but he wants to know by how much the profit margin might change without changing his present production plan. The objective function now becomes

MAX $Z = (17 + \Delta)\, x_1 + 12\, x_2$

This gives the initial simplex tableau:

Simplex tableau: iteration 0

| $c_j$ / $c_b$ | Basis | $b_i$* | $17 + \Delta$ | 12 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
| 0 | $s_1$ | 40 | 0.5 | 1 | 1 | 0 | 0 |
| 0 | $s_2$ | 60 | 2 | 3 | 0 | 1 | 0 |
| 0 | $s_3$ | 80 | 6 | 1.5 | 0 | 0 | 1 |
| $Z_j$ | | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_j - Z_j$ | | | $17 + \Delta$ | 12 | 0 | 0 | 0 |

Given the nature of the way in which the index row (the last row) in the simplex tableau is calculated, we can calculate the revised index row from the changes in the $c_b$ column. In this case, the second and third tableaux would be as follows:

Simplex tableau: iteration 1

| $c_j$ / $c_b$ | Basis | $b_i$* | $17 + \Delta$ | 12 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
| 0 | $s_1$ | 33.33 | 0 | 0.88 | 1 | 0 | -0.08 |
| 0 | $s_2$ | 33.33 | 0 | 2.5 | 0 | 1 | -0.33 |
| $17 + \Delta$ | $x_1$ | 13.33 | 1 | 0.25 | 0 | 0 | 0.17 |
| $Z_{ja}$ | | $226.67+13.33\Delta$ | $17 + \Delta$ | $4.25 + 0.25\Delta$ | 0 | 0 | $2.83 + 0.17\Delta$ |
| $C_j - Z_j$ | | | 0 | $7.75 - 0.25\,\Delta$ | 0 | 0 | $-2.83 - 0.17\,\Delta$ |

Simplex tableau: iteration 2

| $c_j$ / $c_b$ | Basis | $b_i$* | $17 + \Delta$ | 12 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
| 0 | $s_1$ | 21.67 | 0 | 0 | 1 | -0.35 | 0.03 |
| 12 | $x_2$ | 13.33 | 0 | 0 | 0 | 0.40 | -0.13 |
| $17 + \Delta$ | $x_1$ | 10 | 1 | 1 | 0 | -0.1 | 0.2 |
| $Z_j$ | | $330 + 10\,\Delta$ | $17 + \Delta$ | 12 | 0 | $3.1 - 0.1\Delta$ | $1.8 + 0.2\Delta$ |
| $C_j - Z_j$ | | | 0 | 0 | 0 | $-3.1+0.1\,\Delta$ | $-1.8-0.2\Delta$ |

It is clear that there is no need to go through the entire simplex process here. The index row of the tableau with the changed objective function coefficient can be generated simply by altering the $c_b$ value in the final tableau, adjusting the appropriate entry in the objective function coefficient row and recalculating the index row values.

For the final tableau to remain optimal, all the values in the index row must be zero or negative. Therefore the value of $\Delta$ must be such that it keeps all of these entries less than or equal to zero. For the above tableau:

Maximum increase ($\Delta$ positive)
$s_2$ column $\quad -3.1 + 0.1\Delta \le 0 \quad \Rightarrow \quad \Delta \le 31$

Maximum decrease ($\Delta$ negative)
$s_3$ column $\quad -1.8 - 0.2\Delta \le 0 \quad \Rightarrow \quad \Delta \ge -9$

Therefore the profit margin on tables may vary from €8 (got by subtracting 9 from 17) to €48 (got by adding 31 to 17) without changing the optimality of the present solution.

If we wanted to examine the effect of a change in the profit margin for benches we can simply alter the final tableau. The new objective function is $17 x_1 + (12 + \Delta) x_2$. The original final optimal tableau is below.

| $c_j$ | | | 17 | 12 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
| 0 | $s_1$ | 21.67 | 0 | 0 | 1 | -0.35 | 0.03 |
| 12 | $x_2$ | 13.33 | 0 | 1 | 0 | 0.40 | -0.13 |
| 17 | $x_1$ | 10 | 1 | 0 | 0 | -0.1 | 0.2 |
| $Z_j$ | | 330 | 17 | 12 | 0 | 3.1 | 1.8 |
| $C_j - Z_j$ | | | 0 | 0 | 0 | -3.1 | -1.8 |

In the new tableau the $c_b$ value for $x_2$ will become $12 + \Delta$ and the objective function coefficient row is also altered accordingly. We can then calculate the new index row and generate the altered final tableau.

| $c_j$ | | | 17 | $12 + \Delta$ | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
| 0 | $s_1$ | 21.67 | 0 | 0 | 1 | -0.35 | 0.03 |
| $12 + \Delta$ | $x_2$ | 13.33 | 0 | 0 | 0 | 0.40 | -0.13 |
| 17 | $x_1$ | 10 | 1 | 1 | 0 | -0.1 | 0.2 |
| $Z_j$ | | $330 + 13.33\Delta$ | 17 | $12 + \Delta$ | 0 | $3.1 + 0.4\Delta$ | $1.8 - 0.13\Delta$ |
| $C_j - Z_j$ | | | 0 | 0 | 0 | $-3.1 - 0.4\Delta$ | $-1.8 + 0.13\Delta$ |

As an exercise for yourself, set up the initial tableau for this problem with the above objective function and go through the simplex procedure to satisfy yourself that the tableau is in fact the optimal one.

Notice that it is not necessary to actually do all the computations. The new $Z_j$ value is the old $Z_j$ value plus the entry in row i of the tableau times the change $\Delta_i$. So to find the new $Z_j$ value for the $s_3$ column it is only necessary to move across row 2 of the tableau to find the entry in the column corresponding to $s_3$, this entry is -0.13. Then the new $Z_j$ value is the old value 1.8 plus ($-0.13\Delta$)

Since all values in the index row must be zero or positive we can calculate the range for which the current solution remains optimal.
Maximum increase ($\Delta$ positive)

| | | | |
|---|---|---|---|
| $s_3$ column | $-1.8 + 0.13\Delta \le 0$ | $\Rightarrow$ | $\Delta \le 13.8$ |

Maximum decrease ($\Delta$ negative)

| | | | |
|---|---|---|---|
| $s_2$ column | $-3.1 - 0.4\Delta \le 0$ | $\Rightarrow$ | $\Delta \ge -7.75$ |

Therefore the present solution will remain optimal if the profit margin on benches is within the range €4.25 (got from subtracting 7.75 from 12) and €25.80 (got by adding 13.80 to 12).

How can we interpret this information? If the profit margin on benches falls below €4.25 then it will no longer be profitable to make benches at all and the entire resources should be used to make tables. This can be seen from the graphical analysis of the problem. In the graph below the present solution can be clearly seen to be optimal while the slope of the objective function lies between the lines of the two constraints. Therefore we would expect the solution to cease to be optimal when the change in the objective function coefficient is sufficient to bring the slope of the line outside the slope of the constraints.

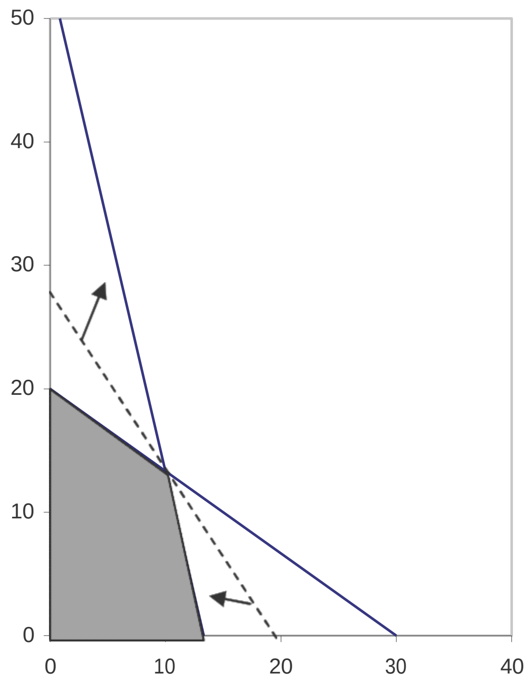



**Figure 3.3: Graphical Solution for Mac's furniture problem**

In fact the simplex analysis has given us exactly the same information. In the case of a change in the profit margin for $x_1$ we have already calculated the minimum profit level for the solution to remain optimal is €8. In this case the objective function becomes $8x_1 + 12\ x_2$ which has the same slope as the assembly time constraint. At the maximum point of the profit level range of the present solution the objective function would be $48x_1 + 12x_2$ giving a line of the same slope as the finishing time constraint.

If the profit level was actually at either one of these extreme points of the range, the slope of the profit function would be identical to one of the constrains giving a multiple optimal solution. This can be seen either from the graph or by substituting an extreme value for $\Delta$ into the tableau. This will cause one of the non-basic slack variables to become equal to zero.

We can repeat the comparison between the graphical and simplex approach for changes in the profit level of benches. If this is done it will be seen that the analysis will give the same results in either case. However, in the calculations of this type, the fact that the entries in the tableau are only accurate to two decimal places may lead to a slight inaccuracy. In this case, the upper limit of the profit margin range for benches is predicted to be €25.80 from the simplex analysis. However, the graphical method indicates a value of €25.50, which is the more accurate solution.

▶ See video **V14_Simplex_7**, which shows how to use the final simplex tableau to find the range for an objective function coefficient. https://goo.gl/Lq9M7q

## *SAQ 11*

*Consider the Linear Program introduced in SAQ 9 above.*

$x_1$ = *quantity of Model 1 PC*
$x_2$ = *quantity of Model 2 PC*

*MAX* $Z = 60\, x_1 + 50\, x_2$
*Subject to*
$4x_1 + 10\, x_2 \leq 100$ *(Assembly Time Constraint)*
$2x_1 + x_2 \leq 22$ *(Inspection Time Constraint)*
$3x_1 + 3x_2 \leq 39$ *(Storage Space Constraint)*
$x_1 \geq 0$ *and* $x_2 \geqslant 0$

*For what range of objective function coefficients does the solution obtained remain optimal?*

*Verify your answer using the graphical approach.*

## *SAQ 12*

*Given the following Linear Program Model:*

*MAX* $Z = 10\, x_1 + 6\, x_2 + 5\, x_3$

*Subject to*
$2x_1 + 3\, x_2 + 4x_3 \leq 25$
$x_1 + 3\, x_2 + 2x_3 \leq 22$
$6x_1 + 3\, x_2 + 4x_3 \leq 32$

$x_1 \geq 0,\ x_2 \geqslant 0$ *and* $x_3 \geqslant 0$

*Find the optimal solution using the Simplex Algorithm.*

*Determine the shadow price for each constraint.*

*Determine the range of optimality for the coefficients of the decision variables that are basic in the optimal solution.*

*By how much would the objective coefficient of* $x_3$ *have to increase before it would enter the optimal solution?*

## *Exercise*

*Now that you have completed all the SAQs, you should verify the solution of all the SAQs and examples using appropriate software. You can use the Solver Add-in in Excel, or linear programming software such as LINDO (a fully functional trial version is available at www.lindo.com and a student solver suite is also available).*

If using Excel's Solver add-in, you may need to install this using 'Add-ins' on the Tools drop-down menu.

▶ See video **V15_Solver_1**, which shows how to enable the Solver add-in in Excel 2003. https://goo.gl/eVUcX7

You should also use software to solve the SAQs in Unit 2 using the simplex algorithm covered in this unit.

The methods used for sensitivity analysis in this unit enable useful information to be obtained from a simplex tableau. However, for large problems with many variables, sensitivity calculations by hand could get quite laborious. As a result, linear programming packages are generally used to perform these calculations automatically and make the results available to the user.

You are advised to check the computer output from your selected software package and, from the system documentation, understand how your software performs sensitivity analysis.

▶ See video **V16_Solver_2**, which uses the Solver add-in in Excel 2003 to solve the loudspeaker linear programming problem (presented in the video V01_LPForm_1). https://goo.gl/GHj591

▶ See video **V17_Solver_3**, which uses the Solver add-in in Excel 2003 to solve the animal feed mix linear programming problem (presented in the video V03_LPForm_3). https://goo.gl/3OaxXR

## ADDITIONAL PROBLEMS

Formulate each of the following problems as Linear Programs. Solve the resulting model using a computer linear programming package.

### *PROBLEM 1*

Four products are produced on two machines. The manufacturing times, in hours per unit, of each product are tabulated for the two machines:

| | Production Time (Hours) per Unit | | | |
|---|---|---|---|---|
| Machine | Product 1 | Product 2 | Product 3 | Product 4 |
| 1 | 2 | 3 | 4 | 2 |
| 2 | 3 | 2 | 1 | 2 |

The total cost of producing 1 unit of each product is based directly on machine time. Assume that the cost per hour of machines 1 and 2 is £10 and £15, respectively. Over the planning horizon, the total hours available on machine 1 is 500 hours, while, 380 hours are available for machine 2. The sales price per unit for products 1, 2, 3 and 4 are £73, £70, £70 and £56. Management wishes to determine the production schedule that will maximize total net profit over the planning horizon.

### *PROBLEM 2*

A market research company has been contracted to conduct a particular survey. For the duration of the survey, which is estimated to last for three weeks, the market research company will employ temporary interviewers. The survey interviews will be conducted by the temporary interviewers during the day shift or night shift, either by telephone or in person.

Interviewers, appointed for the day shift, will work from 8am to 4pm, and interviewers appointed to the night shift will work from 4pm to 11pm. Because of the different skills required an interviewer will be assigned during the duration of the project to either telephone or personal interviews.
Experience suggests that the number of interviews that can be performed by one person during the entire survey depends upon the shift that person works, and on whether the person is assigned to telephone or personal interviews. The following table details the estimated number of interviews that one person can conduct during the survey:

| **Shift/Medium** | **Number of Interviews** |
|---|---|
| Day/Telephone | 100 |
| Day/Person | 120 |
| Night/Telephone | 150 |
| Night/Person | 75 |

The following criterion has been established to ensure a representative survey:

There must be at least 3,000 total interviews.
At least 1,000 interviews must be at night.
At least 1,500 interviews must be personal.

The market research company wants to know the minimum number of interviewers of each type to be assigned to the survey.

### *PROBLEM 3*

The personnel department of a bank is working to develop an efficient work schedule for full-time and part-time tellers. The schedule must provide for efficient operation of the bank at all times. The bank is considering extending its opening hours on Fridays with new opening hours from 9am to 7pm. Research has suggested that the minimum number of tellers necessary to provide adequate customer service during each hour of operation is summarised as follows:

| Time | 9-10 | 10-11 | 11-12 | 12-1 | 1-2 | 2-3 | 3-4 | 4-5 | 5-6 | 6-7 |
|---|---|---|---|---|---|---|---|---|---|---|
| Minimum No. of Tellers | 6 | 4 | 8 | 10 | 9 | 6 | 4 | 7 | 6 | 6 |

Work practices suggest that each full-time employee starts on the hour and works a 4-hour shift, followed by a 1-hour break and then a 3-hour shift. Part-time employees work one 4-hour shift beginning on the hour and lasting for 4 consecutive hours. Considering salary and fringe benefits, full-time employees cost the bank £7.50 per hour and part- time employees cost the bank £4 per hour.

You have been asked to develop a schedule that will satisfy customer service needs at a minimum employee cost.

After reviewing the above formulation, management has realised that some additional constraints must be specified. Specifically, management wants to ensure that one-full time employee is on duty at all times, and that there is a staff of at least five full-time employees. Revise your model to incorporate these additional requirements.

## ANSWERS TO SAQS

### *SAQ 1*

| Cb | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
|---|---|---|---|---|---|---|---|---|
| z | | | 3 | 2 | 0 | 0 | 0 | 0 |
| 0 | $s_1$ | 6 | 1 | 2 | 1 | 0 | 0 | 0 |
| 0 | $s_2$ | 8 | 2 | 1 | 0 | 1 | 0 | 0 |
| 0 | $s_3$ | 1 | -1 | 1 | 0 | 0 | 1 | 0 |
| 0 | $s_4$ | 2 | 0 | 1 | 0 | 0 | 0 | 1 |
| $Z_j$ | | 0 | 0 | 0 | 0 | 0 | 0 | |
| $C_j - Z_j$ | | | 3 | 2 | 0 | 0 | 0 | 0 |

*Initial Simplex Tableau*

### SAQ 2

*Initial Tableau*

| Cb | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
|---|---|---|---|---|---|---|---|
| $C_j$ | | | 50 | 40 | 0 | 0 | 0 |
| 0 | $s_1$ | 150 | 3 | 5 | 1 | 0 | 0 |
| 0 | $s_2$ | 20 | 0 | 1 | 0 | 1 | 0 |
| 0 | $s_3$ | 300 | 8 | 5 | 0 | 0 | 1 |
| $Z_j$ | | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_j - Z_j$ | | | 50 | 40 | 0 | 0 | 0 |

*$x_1$ enters, $s_3$ leaves*

| Cb | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
|---|---|---|---|---|---|---|---|
| $C_j$ | | | 50 | 40 | 0 | 0 | 0 |
| 0 | $s_1$ | 37.5 | 0 | 25/8 | 1 | 0 | -3/8 |
| 0 | $s_2$ | 20 | 0 | 1 | 0 | 1 | 0 |
| 50 | $x_1$ | 37.5 | 1 | 5/8 | 0 | 0 | 1/8 |
| $Z_j$ | | 1875 | 50 | 250/8 | 0 | 0 | 50/8 |
| $C_j - Z_j$ | | | 0 | +70/8 | 0 | 0 | -50/8 |

*$x_2$ enters, $s_1$ leaves*

| Cb | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
|---|---|---|---|---|---|---|---|
| $C_j$ | | | 50 | 40 | 0 | 0 | 0 |
| 40 | $x_2$ | 12 | 0 | 1 | 8/25 | 0 | -3/25 |
| 0 | $s_2$ | 8 | 0 | 0 | -8/25 | 1 | 3/25 |
| 50 | $x_1$ | 30 | 1 | 0 | -5/25 | 0 | 5/25 |
| $Z_j$ | | 1980 | 50 | 40 | 14/5 | 0 | 26/5 |
| $C_j - Z_j$ | | | 0 | 0 | -14/5 | 0 | -26/5 |

*Optimal Answer:*
*$x_1 = 30$*

*$x_2 = 12$*
*Max Z = 1980*

## SAQ 3

*Initial Tableau*

| $c_j$ | | | 100 | 80 | 0 | 0 |
|---|---|---|---|---|---|---|
| $c_b$ | **Basis** | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ |
| 0 | $s_1$ | 12 | 2 | 1 | 1 | 0 |
| 0 | $s_2$ | 8 | 0 | 1 | 0 | 1 |
| | $Z_j$ | 0 | 0 | 0 | 0 | |
| $C_j$ - $Z_j$ | | | 100 | 80 | 0 | 0 |

$x_1$ enters, $s_1$ leaves

| $c_j$ | | | 100 | 80 | 0 | 0 |
|---|---|---|---|---|---|---|
| $c_b$ | **Basis** | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ |
| 100 | $x_1$ | 6 | 1 | 1/2 | 1/2 | 0 |
| 0 | $s_2$ | 8 | 0 | 1 | 0 | 1 |
| $Z_j$ | | 600 | 100 | 50 | 50 | 0 |
| $C_j$ - $Z_j$ | | | 0 | 30 | -50 | 0 |

*$x_2$ enters, $s_2$ leaves*

| $c_j$ | | | 100 | 80 | 0 | 0 |
|---|---|---|---|---|---|---|
| $c_b$ | **Basis** | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ |
| 100 | $x_1$ | 2 | 1 | 0 | 1/2 | -1/2 |
| 80 | $x_2$ | 8 | 0 | 1 | 0 | 1 |
| $Z_j$ | | 840 | 100 | 80 | 50 | 30 |
| $C_j$ - $Z_j$ | | | 0 | 0 | -50 | -30 |

*Optimal:*
*$x_1 = 2$*

*$x_2 = 8$*
*Max $Z = 840$*

## SAQ 4

*Max* $Z = 60\,x_1 + 50\,x_2 + 45\,x_3 + 50\,x_4$

*Subject to:*

$x_2 \quad \le 20$
$x_4 \quad \le 15$
$10\,x_1 + 5\,x_2 \quad \le 120$
$8\,x_3 + 6\,x_4 \quad \le 135$
$x_1, x_2, x_3, x_4 \ge 0$

***Simplex Format***

*Max* $Z = 60\,x_1 + 50\,x_2 + 45\,x_3 + 50\,x_4$

*Subject to:*

$x_2 + s_1 \quad = 20$
$x_4 + s_2 \quad = 15$
$10\,x_1 + 5\,x_2 + s_3 \quad = 120$
$8\,x_3 + 6\,x_4 + s_4 \quad = 135$

***Initial Simplex Tableau***

| $C_j$ / $C_b$ | | | **60** | **50** | **45** | **50** | **0** | **0** | **0** | **0** |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Basis** | $b_i$* | $\mathbf{x_1}$ | $\mathbf{x_2}$ | $\mathbf{x_3}$ | $\mathbf{x_4}$ | $\mathbf{s_1}$ | $\mathbf{s_2}$ | $\mathbf{s_3}$ | $\mathbf{s_4}$ |
| **0** | $\mathbf{s_1}$ | **20** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **0** |
| **0** | $\mathbf{s_2}$ | **15** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **0** |
| **0** | $\mathbf{s_3}$ | **120** | **10** | **5** | **0** | **0** | **0** | **0** | **1** | **0** |
| **0** | $\mathbf{s_4}$ | **135** | **0** | **0** | **8** | **6** | **0** | **0** | **0** | **1** |
| $\mathbf{Z_j}$ | | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| $\mathbf{C_j - Z_j}$ | | | **60** | **50** | **45** | **50** | **0** | **0** | **0** | **0** |

Pivot Row

Pivot Column

$x_1$ *enters,* $s_3$ *leaves.*

***2nd Simplex Tableau***

| $C_j$ / $C_b$ | | | **60** | **50** | **45** | **50** | **0** | **0** | **0** | **0** |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Basis** | $b_i$* | $\mathbf{x_1}$ | $\mathbf{x_2}$ | $\mathbf{x_3}$ | $\mathbf{x_4}$ | $\mathbf{s_1}$ | $\mathbf{s_2}$ | $\mathbf{s_3}$ | $\mathbf{s_4}$ |
| **0** | $\mathbf{s_1}$ | **20** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **0** |
| **0** | $\mathbf{s_2}$ | **15** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **0** |
| **60** | $\mathbf{x_1}$ | **12** | **1** | **5/10** | **0** | **0** | **0** | **0** | **1/10** | **0** |
| **0** | $\mathbf{s_4}$ | **135** | **0** | **0** | **8** | **6** | **0** | **0** | **0** | **1** |
| $\mathbf{Z_j}$ | | **720** | **60** | **30** | **0** | **0** | **0** | **0** | **6** | **0** |
| $\mathbf{C_j - Z_j}$ | | | **0** | **20** | **45** | **50** | **0** | **0** | **-6** | **0** |

Pivot Row

*Pivot Column*

$x_4$ *enters,* $s_2$ *leaves.*

***3rd Simplex Tableau***

| $C_j$ | | | 60 | 50 | 45 | 50 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $C_b$ | **Basis** | $b_i$* | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
| 0 | $s_1$ | 20 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 50 | $x_4$ | 15 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 60 | $x_1$ | 12 | 1 | 5/10 | 0 | 0 | 0 | 0 | 1/10 | 0 |
| 0 | $s_4$ | 45 | 0 | 0 | 8 | 0 | 0 | -6 | 0 | 1 |
| $Z_j$ | | 1470 | 60 | 30 | 0 | 50 | 0 | 50 | 6 | 0 |
| $C_j$ - $Z_j$ | | | 0 | 20 | 45 | 0 | 0 | -50 | -6 | 0 |

Pivot Row

*Pivot Column*
*$x_3$ enters, $s_4$ leaves.*

*4th Simplex Tableau*

| $C_j$ | | | 60 | 50 | 45 | 50 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $C_b$ | **Basis** | $b_i$* | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
| 0 | $s_1$ | 20 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 50 | $x_4$ | 15 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 60 | $x_1$ | 12 | 1 | 5/10 | 0 | 0 | 0 | 0 | 1/10 | 0 |
| 45 | $x_3$ | 45/8 | 0 | 0 | 1 | 0 | 0 | -6/8 | 0 | 1/8 |
| $Z_j$ | | 1723.13 | 60 | 30 | 45 | 50 | 0 | 130/8 | 6 | 45/8 |
| $C_j$ - $Z_j$ | | | 0 | 20 | 0 | 0 | 0 | -130/8 | -6 | -45/8 |

Pivot Row

*Pivot Column*
*$x_2$ enters, $s_1$ leaves.*

*5th Simplex Tableau (Optimal)*

| $C_j$ | | | 60 | 50 | 45 | 50 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $C_b$ | **Basis** | $b_i$* | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
| 50 | $x_2$ | 20 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 50 | $x_4$ | 15 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 60 | $x_1$ | 2 | 1 | 0 | 0 | 0 | -5/10 | 0 | 1/10 | 0 |
| 45 | $x_3$ | 45/8 | 0 | 0 | 1 | 0 | 0 | -6/8 | 0 | 1/8 |
| $Z_j$ | | 2123.13 | 60 | 50 | 45 | 50 | 20 | 130/8 | 6 | 45/8 |
| $C_j$ - $Z_j$ | | | 0 | 0 | 0 | 0 | -20 | -130/8 | -6 | -45/8 |

*$Z^* = 2123.13$*
*$x_1^* = 2$*
*$x_2^* = 20$*
*$x_3^* = 45/8$*
*$x_4^* = 15$*

## *SAQ 5*

*Let:*
$x_1$ *= quantity of Type 1 Watsit*
$x_2$ *= quantity of Type 2 Watsit*

*MAX P =* $60X_1 + 50X_2$

*Subject to:*

$4x_1 + 10x_2 \leq 100$ *(Assembly Time constraint)*
$2x_1 + x_2 \leq 22$ *(Inspection Time constraint)*
$3x_1 + 3x_2 \leq 39$ *(Storage Space constraint)*
$x1, x2 \geq 0$

*The optimal tableau is given below*

| $C_b$ \ $C_j$ | | | 60 | 50 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| | **Basis** | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | s3 |
| **0** | $s_1$ | **24** | **0** | **0** | **1** | **6** | **-16/3** |
| **60** | $x_1$ | **9** | **1** | **0** | **0** | **1** | **- 1/3** |
| **50** | $x_2$ | **4** | **0** | **1** | **0** | **-1** | **2/3** |
| $Z_j$ | | **740** | **60** | **50** | **0** | **10** | **40/3** |
| $C_j - Z_j$ | | | **0** | **0** | **0** | **-10** | **-40/3** |

*From the tableau we can read the optimal solution* $x_1 = 9$ *and* $x_2 = 4$ *and maximum weekly profit is €740*

## *SAQ 6*

*Let*
$y_1$ *= the number of eggs served*
$y_2$ *= the number of bacon strips served*
$y_3$ *= the number of cup of cereal served*

*Thus the model is:*

*Minimise Cost:* $Z = 40 y_1 + 30 y_2 + 20 y_3$

*Subject to:*

$2 y_1 + 4 y_2 + y_3 \geq 16$
$3 y_1 + 2 y_2 + y_3 \geq 12$
$y_1, y_2, y_3 \geq 0$

*Simplex Format*
*Min Z* $= 40y_1 + 30y_2 + 20y_3 + Os_1 + Os_2 + MA_1 + MA_2$
*Subject to:*
$2 y_1 + 4 y_2 + y_3 - s_1 + A_1 = 16$
$3 y_1 + 2 y_2 + y_3 - s_2 + A_2 = 12$

*1st Simplex Tableau*

*Note: In the Case M is very large*

| $c_j$ | | | 40 | 30 | 20 | 0 | 0 | M | M |
|---|---|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $y_1$ | $y_2$ | $y_3$ | $s_1$ | $s_2$ | $A_1$ | $A_2$ |
| M | $A_1$ | 16 | 2 | 4 | 1 | -1 | 0 | 1 | 0 |
| M | $A_2$ | 12 | 3 | 2 | 1 | 0 | -1 | 0 | 1 |
| $Z_j$ | | 28M | 5M | 6M | 2M | -M | -M | M | M |
| $C_j - Z_j$ | | | 40-5M | 30-6M | 20-2M | M | M | 0 | 0 |

Pivot Row

*Pivot Column*

*$y_2$ enters, $A_1$ leaves.*

*2nd Simplex Tableau*

| $c_j$ | | | 40 | 30 | 20 | 0 | 0 | M | M |
|---|---|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $y_1$ | $y_2$ | $y_3$ | $s_1$ | $s_2$ | $A_1$ | $A_2$ |
| 30 | $y_2$ | 4 | ½ | 1 | ¼ | - ¼ | 0 | ¼ | 0 |
| M | $A_2$ | 4 | 2 | 0 | ½ | ½ | -1 | - ½ | 1 |
| $Z_j$ | | 120+4M | 2M+15 | 30 | ½M +7½ | ½M -7½ | -M | -½M+7½ | M |
| $C_j - Z_j$ | | | 25 – 2M | 0 | 12½ – ½M | 7½ – ½M | M | –7½ +1½M | 0 |

*Pivot Row*

*Pivot Column*

*$y_1$ enters, $A_2$ leaves.*

*3rd Simplex Tableau (Optimal)*

| $c_j$ | | | 40 | 30 | 20 | 0 | 0 | M | M |
|---|---|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $y_1$ | $y_2$ | $y_3$ | $s_1$ | $s_2$ | $A_1$ | $A_2$ |
| 30 | $y_2$ | 3 | 0 | 1 | 1/8 | -3/8 | 1/4 | 3/8 | -1/4 |
| 40 | $y_1$ | 2 | 1 | 0 | 1/4 | 1/4 | -1/2 | -1/4 | 1/2 |
| $Z_j$ | | 170 | 40 | 30 | 55/4 | -5/4 | -25/2 | +5/4 | 25/2 |
| $C_j - Z_j$ | | | 0 | 0 | 25/4 | 5/4 | 25/2 | M-5/4 | M-25/2 |

*The optimal solution is:*

*$y_1^* = 2$; $y_2^* = 3$ and Min Z = 170*

## SAQ 7

*Max Z = 200 $x_1$ + 100 $x_2$*

*Subject to:*
$x_1 + x_2 = 1500$
$x_1 \leq 400$
$x_2 \geq 200$
$x_1, x_2 \geq 0$

*Simplex Format*

*Max Z = 200 $X_1$ + 100 $X_2$ + 0 $s_1$ + 0 $s_2$ -M$A_1$ -M$A_2$*

*Subject to:*
$x_1 + x_2 + A_1 = 1500$
$x_1 + s_1 = 400$
$x_2 - s_2 + A_2 = 200$

***1st Simplex Tableau***

| $c_j$ | | | 200 | 100 | 0 | 0 | -M | -M |
|---|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $A_1$ | $A_2$ |
| -M | $A_1$ | 1500 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | $s_1$ | 400 | 1 | 0 | 1 | 0 | 0 | 0 |
| -M | $A_2$ | 200 | 0 | 1 | 0 | -1 | 0 | 1 |
| $Z_j$ | | -1700M | -M | -2M | 0 | M | -M | -M |
| $C_j - Z_j$ | | | M+200 | 2M+100 | 0 | -M | 0 | 0 |

Pivot Row

*Pivot Column*

*$x_2$ enters, $A_2$ leaves.*

***2nd Simplex Tableau***

| $c_j$ | | | 200 | 100 | 0 | 0 | -M | -M |
|---|---|---|---|---|---|---|---|---|
| $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $A_1$ | $A_2$ |
| -M | $A_1$ | 1300 | 1 | 0 | 0 | 1 | 1 | -1 |
| 0 | $s_1$ | 400 | 1 | 0 | 1 | 0 | 0 | 0 |
| 100 | $x_2$ | 200 | 0 | 1 | 0 | -1 | 0 | 1 |
| $Z_j$ | | 20000-1300M | -M | 100 | 0 | -M-100 | -M | M+100 |
| $C_j - Z_j$ | | | M+200 | 0 | 0 | M+100 | 0 | -2M-100 |

*Pivot Column*

*$x_1$ enters, $s_1$ leaves.*

***3rd Simplex Tableau***

| $c_j$ / $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $A_1$ | $A_2$ |
|---|---|---|---|---|---|---|---|---|
| | | | 200 | 100 | 0 | 0 | -M | -M |
| -M | $A_1$ | 900 | 0 | 0 | -1 | 1 | 1 | -1 |
| 200 | $x_1$ | 400 | 1 | 0 | 1 | 0 | 0 | 0 |
| 100 | $x_2$ | 200 | 0 | 1 | 0 | -1 | 0 | 1 |
| $Z_j$ | | 100000-900M | 200 | 100 | M+200 | -M-100 | -M | M+100 |
| $C_j$ - $Z_j$ | | | 0 | 0 | -200 - M | M+100 | 0 | -2M-100 |

*Pivot Column*

*$s_2$ enters, $A_1$ leaves.*

***4th Simplex Tableau (Optimal)***

| $c_j$ / $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $A_1$ | $A_2$ |
|---|---|---|---|---|---|---|---|---|
| | | | 200 | 100 | 0 | 0 | -M | -M |
| 0 | $s_2$ | 900 | 0 | 0 | -1 | 1 | 1 | -1 |
| 200 | $x_1$ | 400 | 1 | 0 | 1 | 0 | 0 | 0 |
| 100 | $x_2$ | 1100 | 0 | 1 | -1 | 0 | 1 | 0 |
| $Z_j$ | | 190000 | 200 | 100 | 100 | 0 | 100 | 0 |
| $C_j$ - $Z_j$ | | | 0 | 0 | -100 | 0 | -M-100 | - M |

$x_1^* = 400$
$x_2^* = 1100$
$Z^* = 190{,}000$

## *SAQ 8*

*Simplex Tableau, Iteration 0: The variable with the most negative index row value is $x_2$ therefore we calculate the ratio values based on this column. Selecting the smallest ratio, we find that $s_2$ is the variable leaving the basis.*

| $c_j$ / $c_b$ | Basis | $b_i$* | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
|---|---|---|---|---|---|---|---|---|
| | | | 15 | 20 | 0 | 0 | 0 | 0 |
| 0 | $s_1$ | 250 | 1 | 2 | 1 | 0 | 0 | 0 |
| 0 | $s_2$ | 150 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | $s_3$ | 125 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | $s_4$ | 140 | 0 | 1 | 0 | 0 | 0 | 1 |
| $Z_j$ | | 0 | 0 | 0 | 0 | 0 | 0 | |
| $C_j$ - $Z_j$ | | | 15 | 20 | 0 | 0 | 0 | 0 |

***Simplex Tableau: Iteration 1:***

| $c_b$ \ $c_j$ | Basis | $b_i$* | 15 | 20 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ | $S_4$ |
| **20** | $x_2$ | **125** | **0.5** | **1** | **0.5** | **0** | **0** | **0** |
| **0** | $s_2$ | **25** | **0.5** | **0** | **-0.5** | **1** | **0** | **0** |
| **0** | $s_3$ | **125** | **1** | **0** | **0** | **0** | **1** | **0** |
| **0** | $s_4$ | **15** | **-0.5** | **0** | **-0.5** | **0** | **0** | **1** |
| $Z_j$ | | **2500** | **10** | **20** | **10** | **0** | **0** | **0** |
| $C_j - Z_j$ | | | **5** | **0** | **-10** | **0** | **0** | **0** |

***Simplex Tableau: Iteration 2:***

| $c_b$ \ $c_j$ | Basis | $b_i$* | 15 | 20 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | $x_1$ | $x_2$ | $S_1$ | $s_2$ | $s_3$ | $S_4$ |
| **20** | $x_2$ | **100** | **0** | **1** | **1** | **-1** | **0** | **0** |
| **15** | $x_1$ | **50** | **1** | **0** | **-1** | **2** | **0** | **0** |
| **0** | $s_3$ | **75** | **0** | **0** | **1** | **-2** | **1** | **0** |
| **0** | $s_4$ | **40** | **0** | **0** | **--1** | **1** | **0** | **1** |
| $Z_j$ | | **2750** | **15** | **2o** | **5** | **10** | **0** | **0** |
| $C_j - Z_j$ | | | **0** | **0** | **-5** | **-10** | **0** | **0** |

*Optimal Answer:* *$x_1 = 50$, $x_2 = 100$ and $Z = 2750$.*

## *SAQ 9*

The optimal tableau is given below:

| $c_b$ \ $c_j$ | Basis | $b_i$* | 60 | 50 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| | | | $x_1$ | $x_2$ | $s_1$ | $s_2$ | $s_3$ |
| **0** | $s_1$ | **24** | **0** | **0** | **1** | **6** | **-16/3** |
| **60** | $x_1$ | **9** | **1** | **0** | **0** | **1** | **-1/3** |
| **50** | $x_2$ | **4** | **0** | **1** | **0** | **-1** | **2/3** |
| $Z_j$ | | **740** | **60** | **50** | **0** | **10** | **40/3** |
| $C_j - Z_j$ | | | **0** | **0** | **0** | **-10** | **-40/3** |

*From the tableau we can read the optimal solution $x_1 = 9$ and $x_2 = 4$.*

*Constraint 1: Slack Capacity = 24 Shadow Price = 0*
*Constraint 2: Slack Capacity = 0 Shadow Price = 10*
*Constraint 3: Slack Capacity = 0 Shadow Price = 13.33*

## *SAQ 10*

*The optimal tableau is given below:*
*MAX* $Z = 4x_1 + 5x_2$

*Subject to*
$4x_1 + 1x_2 \leq 56$
$5x_1 + 3x_2 \leq 105$
$x_1 + 2x_2 \leq 56$
$x_1 \geq 0$ *and* $x_2 \geq 0$

| $c_j$ / $c_b$ | Basis | $b_i$* | 4 | 5 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| | | | **x1** | **x2** | **s1** | **s2** | **s3** |
| **0** | **s1** | **7** | **0** | **0** | **1** | **-1** | **1** |
| **4** | **x1** | **6** | **1** | **0** | **0** | **0.29** | **-0.43** |
| **5** | **x2** | **25** | **0** | **1** | **0** | **-0.14** | **0.71** |
| $Z_j$ | | **140** | **4** | **5** | **0** | **0.43** | **1.86** |
| $C_j - Z_j$ | | | **0** | **0** | **0** | **-0.43** | **-1.86** |

(i) *The shadow price of resource 2 is €0.43*
*The shadow price of resource 3 is €1.86*

(ii) Right Hand Side Ranges
*Constraint 1:*
*Maximum increase ($\Delta$ positive)*
Infinite increase possible

*Maximum decrease ($\Delta$ negative)*
$7 + 1\Delta \geq 0 \quad \Rightarrow \quad \Delta \geq -7$

*Constraint 2:*
*Maximum increase ($\Delta$ positive)*
$7 - 1\Delta \geq 0 \quad \Rightarrow \quad \Delta \leq 7$
$25 - 0.14\Delta \geq 0 \quad \Rightarrow \quad \Delta \leq 178.57$
*Maximum decrease ($\Delta$ negative)*
$6 + 0.29\Delta \geq 0 \quad \Rightarrow \quad \Delta \geq -20.689$

*So we have found that* $-20.689 \leq \Delta \leq 7.0$
*This gives us:*
(105 - 20.689) ≤ Right Hand Side ≤ (105+ 7.0)
84.311 ≤ Right Hand Side ≤ 112

*Constraint 3:*
*Maximum increase ($\Delta$ positive)*
$6 - 0.43\Delta \geq 0 \quad \Rightarrow \quad \Delta \leq 13.953$
*Maximum decrease ($\Delta$ negative)*
$7 + \Delta \geq 0 \quad \Rightarrow \quad \Delta \geq -7$
$25 + 0.71\Delta \geq 0 \Rightarrow \quad \Delta \geq -35.21$
*So we have found that* $-7 \leq \Delta \leq 13.953$
*This gives us:*
*(56 - 7) ≤ Right Hand Side ≤ (56 + 13.953)*
*≤ Right Hand Side ≤ 69.953*

## SAQ 11

*Changing the $x_1$ coefficient from 60 to 60 +$\Delta$ gives a new optimal tableau:*

| $c_b$ \ $c_j$ | Basis | $b_i$* | $60+\Delta$<br>$x_1$ | 50<br>$x_2$ | 0<br>$s_1$ | 0<br>$s_2$ | 0<br>$s_3$ |
|---|---|---|---|---|---|---|---|
| **0** | $s_1$ | **24** | **0** | **0** | **1** | **6** | **-16/3** |
| **60+ Δ** | $x_1$ | **9** | **1** | **0** | **0** | **1** | **-1/3** |
| **50** | $x_2$ | **4** | **0** | **1** | **0** | **-1** | **2/3** |
| $Z_j$ | | **740** | **60+Δ** | **50** | **0** | **10+Δ** | **40/3-1/3 Δ** |
| $C_j - Z_j$ | | | **0** | **0** | **0** | **-10-Δ** | **-40/3+1/3 Δ** |

*The solution is optimal if $-10-\Delta \leq 0$ and $-40/3+1/3\Delta \leq 0$*
*Thus $\Delta \geq -10$ and $\Delta \leq 40$*
*The range is therefore 60-10 to 60 +40 i.e. 50 to 100.*
*Similarly for $x_2$ the range can be shown to be 30 to 60*

## SAQ 12

*The optimal solution is $x_1 = 2$, $x_2 = 6.67$ and $x_3 = 0$, which gives Z = 60.*

*The range for the $x_2$ coefficient is 5 to 30, the range for the $x_1$ coefficient is 4.5 to 12.*

*The coefficient of $x_3$ must exceed 7.2 if $x_3$ is to enter the optimal solution.*

# UNIT 4

## TRANSPORTATION, TRANSHIPMENT AND ASSIGNMENT PROBLEMS

### AIM

This aim of this unit is to introduce transportation, assignment and transhipment problems. These belong to a special class of linear programming problems and may be solved using customised variants of the simplex algorithm.

### OBJECTIVES

- To demonstrate how problems in the distribution of goods or services can be rapidly solved by a special variant of the simplex algorithm

- To demonstrate how problems of allocation can be solved with yet a further variant of the simplex.

### REQUIRED READING

Students should read the following chapters of the mandatory text book (David R. Anderson, Dennis J. Sweeney, Thomas A. Williams and Kipp Martin, *An Introduction to Management Science: Quantitative Approaches to Decision Making*, 13th International Edition, Cengage Learning, ISBN-13: 9780538475655):

Chapter 6.1, 6.2 and 6.3 and Chapter 19 (On CD)

### 4.1 NETWORKS AND SOME AREAS OF APPLICATION

A *network* is the name given to a graphical representation of a situation (or problem) consisting of numbered circles (or *nodes*) joined by a series of lines (or *arcs*). Arrowheads on the arcs indicate the direction of *flow* of whatever is being studied or analysed. Such flows can embrace a wide variety of activities, for instance:

- The transportation of goods and services.

- Cash flows.

- The assignment of skills or persons.

- The flow of utilities such as electricity, water, oil, gas etc.

Graphically, a network may be illustrated as shown in Figure 4.1

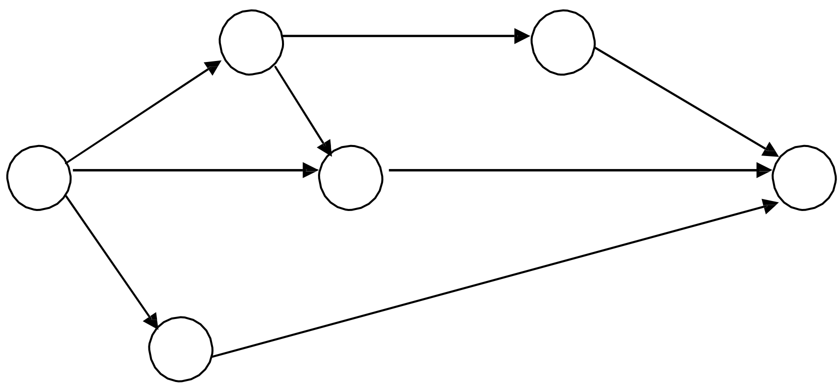


**Figure 4.1: A Typical Network**

- Node (1) is defined as the *start node*
- Node (6) is defined as the *end node*
- Nodes represent *events* whilst the Arrows or Arcs represent *activities.*

It is important to realize that networks don't necessarily commence and end with a single node.

The objective is (usually) to get from the start node to the end node as fast as possible, as cheaply as possible, in the shortest distance, taking the shortest time and so on.

In this unit we will examine a number of special linear programming problems that can be modelled as networks. These can be solved by the simplex algorithm but they are more routinely analysed by specialist algorithms, due to their network structure. In the following three sections we introduce three special linear programming problems: The transportation problem, the assignment problem and the transhipment problem. We then explore some specialist algorithms for solving these problems in section 4.5

## 4.2 THE TRANSPORTATION PROBLEM

In its most general form the transportation problem is concerned with the delivery of a product from a number of sources, with finite supplies, to a number of destinations, with finite demands, at minimum total transportation cost. A typical example is shown in Figure 4.2.

| SOURCE | DESTINATION | | | SUPPLY (TONS) |
|---|---|---|---|---|
| | (j=1) WATERFORD | (j=2) SLIGO | (j=3) ATHLONE | |
| (i=1) DUBLIN | 9 | 11 | 8 | 200 |
| (i=2) CORK | 6 | 15 | 10 | 150 |
| (i=3) GALWAY | 7 | 6 | 6 | 100 |
| DEMAND (TONS) | 180 | 140 | 130 | 450 |

**Figure 4.2: A Typical Transportation Problem**

Transport costs are expressed in € / Ton, and, for simplicity, supply and demand are assumed to be equal. The requirement is to satisfy the needs of Waterford, Sligo and Athlone from Dublin, Cork and Galway, at minimum distribution cost without violating the demand and supply constraints.

This problem is re-expressed as a network in Figure 4.3.

Since the costs are assumed to be linear the example can easily be formulated as a linear program.

Let: $x_{ij}$ = the number of tons of goods to be transported from source i (=1,2,3) to destination j (j=1,2,3).

Then we have:

Minimize
$Z = 9x_{11} + 11x_{12} + 8x_{13} + 6x_{21} + 15x_{22} + 10x_{23} + 7x_{31} + 6x_{32} + 6x_{33}$

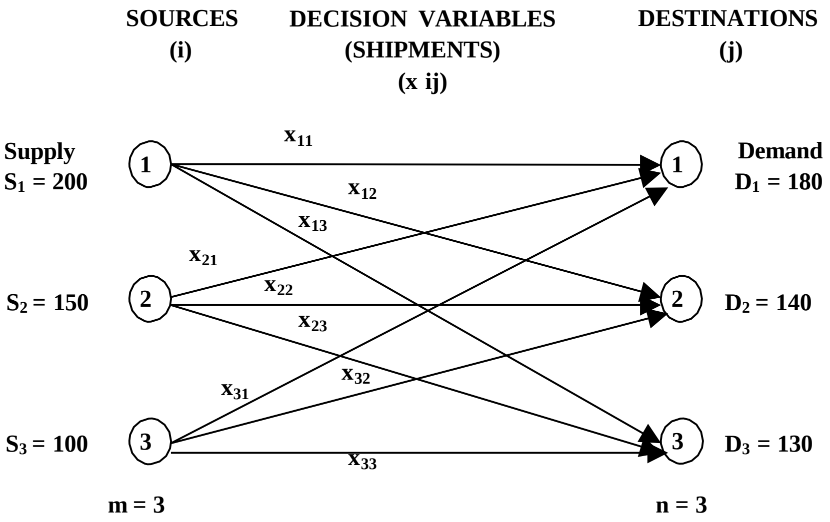



**Figure 4.3: The Network Model for the Transportation Problem**

Subject to:

$$
\begin{aligned}
x_{11} + x_{12} + x_{13} &= 200 \\
x_{21} + x_{22} + x_{23} &= 150 \\
x_{31} + x_{32} + x_{33} &= 100 \\
x_{11} + x_{21} + x_{31} &= 180 \\
x_{12} + x_{22} + x_{32} &= 140 \\
x_{13} + x_{23} + x_{33} &= 130 \\
x_{ij} &\geq 0
\end{aligned}
$$

The first three equations describe the supply conditions, whilst the second three equations describe the demand requirements. Note that these are "equal to" constraints as supply equals demand (and so all of the materials available for supply will be fully utilised and the demand from the three centres will be met). The general form of the model is

Minimize: $$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

Subject to: $\sum_{j=1}^{n} x_{ij} = s_i \qquad i = 1, 2, \ldots\ldots m$

$\sum_{i=1}^{m} x_{ij} = d_j \quad j = 1, 2, \ldots\ldots n$

$x_{ij} \geq 0$

In this general form of the problem there are m sources and n destinations.

The problem can be solved using the simplex algorithm, but even though we have a small 3 × 3 structure, an inspection will show that there are nine decision variables, six constraints and six artificial variables. More realistic problems can become very large and can consume lots of computer resources in their solution. A further examination of the model will reveal that the coefficients of the $x_{ij}$ in the constraint equations are all ones. This unusual feature led Dantzig to develop a special solution procedure, which is computationally more efficient than the simplex method. The procedure he developed is called the MODI method, which is shorthand for Modified (simplex) Distribution method. We shall discuss this solution method in some detail in section 4.5.

▶ See video **V18_Transp_1**, which provides the linear programming formulation of a transportation problem. https://goo.gl/tm6Rlf

## *SAQ 1*

*For the following transportation problem:*

*Supply from source 1=1000*
*Supply from source 2=1500*
*Supply from source 3=1200*

*Demand from destination 1= 2300*
*Demand from destination 2= 1400*

*Cost from source 1 to destination 1 = 80*
*Cost from source 1 to destination 2 = 215*
*Cost from source 2 to destination 1 = 100*
*Cost from source 2 to destination 2 = 108*
*Cost from source 3 to destination 1 = 102*
*Cost from source 3 to destination 2 = 68*

*(a) Formulate the problem as a linear program (LP).*

*(b) Solve the LP model by using a Linear Programming computer package.*

*(You might consider solving the LP model using the Simplex Method. This attempt will show you that while the Simplex Algorithm is a possible methodology, the effort required even for a small problem is too large to be practical.)*

*(c) Present the problem information in the form of a transportation tableau in the standard format.*

## 4.3 The Assignment Problem

The assignment problem is a variation of the transportation problem where there are an equal number of origins and destinations and each origin supply and destination demand is unity. As a consequence, the quantity allocated must either be zero or one.

A typical assignment problem is shown in Figure 4.4.

| **SALESPERSON (i)** | **TERRITORY (j)** | | | **SUPPLY** |
|---|---|---|---|---|
| | **1** | **2** | **3** | |
| **1** | **700** | **300** | **500** | **1** |
| **2** | **800** | **900** | **200** | **1** |
| **3** | **900** | **600** | **800** | **1** |
| **DEMAND** | **1** | **1** | **1** | **3** |

**Figure 4.4: An Assignment Problem**

The above Figure shows the costs in € per week of three sales persons of a particular corporation operating in three different sales territories. The corporation wishes to assign the salespersons so that its total weekly costs are minimized. Although the MODI method is capable of solving this problem, a much simpler procedure is available. This methodology is based on a theorem first proved by the Hungarian Mathematician Dr. A. Konig and is therefore usually referred to as the *Hungarian Method*, which will be illustrated in section 4.5.

The general form of the assignment problem is:

Let
$x_{ij}$ = 1 or 0 according to whether salesperson (i) is assigned to territory (j) or not
and
$c_{ij}$ = the cost of assigning salesperson (i) to territory (j)

Then: $$\text{Min} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\text{-------}n \text{ salespersons}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1,2,\text{------}n \text{ territories}$$

$x_{ij}$ binary (1 or 0)    for all i and j

## 4.4 The Transhipment Problem

An important variation or extension of the transportation problem is the *transhipment problem*. In this model each source and destination can be an intermediate point of shipment, or *transhipment point*, to or from other sources or destinations. The objective in the transhipment problem is to determine how many units should be shipped over each arc in the network so that all destinations demands are met with the minimum possible transportation cost.

Let us take the transportation problem presented in Figure 4.2 and re-cast it into a transhipment problem as presented in Figure 4.5

| To (j) / From (i) | i = 1 | i = 2 | i = 3 | j = 1 | j = 2 | j = 3 | SUPPLY |
|---|---|---|---|---|---|---|---|
| i = 1 | | | | 9 | 11 | 8 | 200 |
| i = 2 | | | | 6 | 15 | 10 | 150 |
| i = 3 | | | | 7 | 6 | 6 | 100 |
| j = 1 | | | | | | | |
| j = 2 | | | | | | | |
| j = 3 | | | | | | | |
| DEMAND | | | | 180 | 140 | 130 | |

**Figure 4.5: A Transhipment Problem**

Our original problem (Figure 4.2) is located in the upper right hand quadrant. The additional new cells represent the possibility of intermediate transhipments. For instance, cell i = 1 to cell i = 2 represents the fact that units can be shipped from source 1 to source 2 before being shipped to a final destination j.

However, Figure 4.5 is incomplete in its present form in that three supply cells, three demand cells and 27 cost cells have, as yet, no entries.

Since this problem is *balanced* in that supply equals demand (both are 450 units), it should be noted that the supply and demand requirements for the new rows and columns must address the reality that each new source and destination can now equal *all* the units supplied and demanded. Therefore, the supply and demand levels must be increased by an amount that is as large as the total demand/supply, 450 units. In addition, the $c_{ij}$ values for the new cells are the transhipment costs as determined by the transport division of the corporation. In this regard is should be noted:

(i) $c_{ij}$ is not necessarily the same as $c_{ji}$ due to possible route conditions and differing models of transportation.
(ii) $c_{ij}$ values on the main left to right diagonal are set at $\infty$ since there is no transportation from a source to itself (i = i) or from a destination to itself (j = j). The completed table is depicted in Figure 4.6.

| To / From | i = 1 | i = 2 | i = 3 | j = 1 | j = 2 | j = 3 | SUPPLY |
|---|---|---|---|---|---|---|---|
| i = 1 | $\infty$ | 5 | 10 | 9 | 11 | 8 | 650 |
| i = 2 | 15 | $\infty$ | 18 | 6 | 15 | 10 | 600 |
| i = 3 | 11 | 19 | $\infty$ | 7 | 6 | 6 | 550 |
| j = 1 | 10 | 7 | 8 | $\infty$ | 20 | 11 | 450 |
| j = 2 | 12 | 16 | 8 | 21 | $\infty$ | 12 | 450 |
| j = 3 | 9 | 12 | 7 | 11 | 13 | $\infty$ | 450 |
| DEMAND | 450 | 450 | 450 | 630 | 590 | 580 | 3150 |

**Figure 4.6: The completed transhipment problem**

This problem can be solved with a slight variation to the MODI method discussed in section 4.5. For completeness, the network representation of the transhipment problem is shown in Figure 4.7

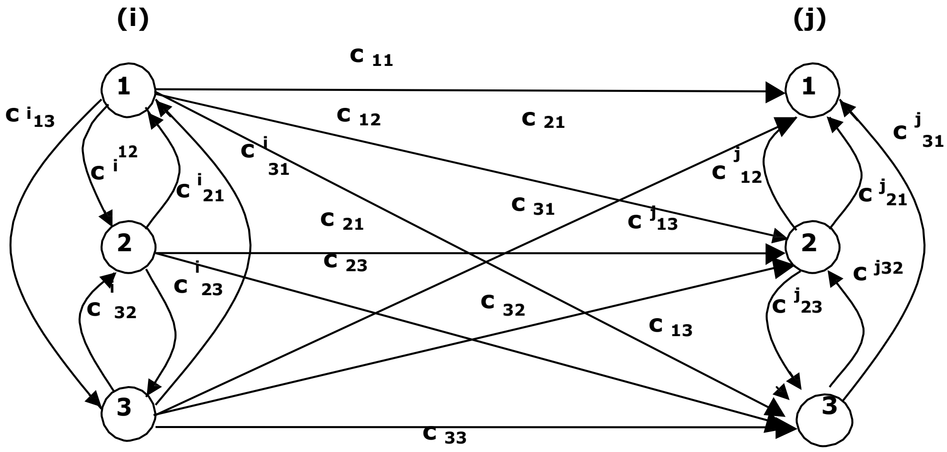



**Figure 4.7: Network representation of the transhipment problem**

In Figure 4.7:
$c_{11}$, $c_{12}$ etc refer to costs between i and j
$c^{i}_{12}$, $c^{i}_{13}$ etc refer to costs between i and i
$c^{j}_{12}$, $c^{j}_{13}$ etc refer to costs between j and j

The general form of the transhipment model is:

$$\text{Min} \sum_{\text{All Arcs}} c_{ij} x_{ij}$$

Subject To:

$$\sum_{\text{Arcs Out}} x_{ij} - \sum_{\text{Arcs In}} x_{ij} = s_i \qquad \text{Origin Nodes (i)}$$

$$\sum_{\text{Arcs Out}} x_{ij} - \sum_{\text{Arcs In}} x_{ij} = 0 \qquad \text{Transhipment Nodes}$$

$$\sum_{\text{Arcs In}} x_{ij} - \sum_{\text{Arcs Out}} x_{ij} = d_j \qquad \text{Destination Nodes (j)}$$

$x_{ij} \geq 0$ all i and j

Finally, is should be noted that *unbalanced problems*, that is, where supply and demand are not equal can be easily accommodated as will be shown in section 4.5.

## 4.5 THE SOLUTION PROCEDURES FOR THE TRANSPORTATION, TRANSHIPMENT AND ASSIGNMENT PROBLEMS

### *4.5.1 INITIAL SOLUTION METHODS*

The first consideration in solving any linear programming problem is the construction of a first feasible solution. That solution is then tested to see if it is optimal – if it is we stop, if not, we generate further feasible solutions until the optimal solution (if it exists) is identified. This is what the simplex algorithm accomplishes in an efficient manner.

For transportation, assignment and transhipment problems, the same logic is followed, although the method of generating the first feasible solution can be selected from:

(i) Any rule-of-thumb procedure or *Heuristic*.

or

(ii) The ***North West Corner*** (NWC) approach.

or

(iii) The ***Low Cost Cell*** approach.

or

(iv) ***Vogel's*** approximation method (VAM).

Extensive research has shown that Vogel's methodology is the most efficient, in that, if it doesn't identify the optimal solution at the first attempt, then very few iterations of the solution method (e.g. MODI) are required to identify it.

With that initial preamble let us now use three different methods to find an initial solution to the transportation problem from section 4.2:

The **North West Corner (NWC)** solution is so called because we start at the top left (i.e. the North-West) corner and we generate a feasible solution.

1. Starting at the top left corner allocate the lesser of the row supply or the column demand to the cell.
2. Subtract the amount allocated to the cell from the row and column totals.
3. If the changed column total is now zero, move to the next cell on the right. If it is the row total that has become zero, then move down to the cell below.
4. Allocate the amount to the new cell as in step 1 unless all demand and supply has been allocated.

| | WATERFORD | | SLIGO | | ATHLONE | | |
|---|---|---|---|---|---|---|---|
| **DUBLIN** | 180 * | 9 | 20 | 11 | | 8 | 200 |
| **CORK** | | 6 | 120 | 15 | 30 | 10 | 150 |
| **GALWAY** | | 7 | | 6 | 100 | 6 | 100 |
| | 180 | | 140 | | 130 | | |

The cell marked * is the North West corner cell. The row total is 200 and the column total is 180, hence we allocate 180 units to this cell. The free demand for column 1 (Waterford) is now zero and the free supply for row 1 (Dublin) is 20 units. As the column total is now zero, we move across one cell to the right.

The free demand for the second column (Sligo) is 140 and the free supply for row 1 (Dublin) is 20. We therefore allocate 20 units to this cell. The free supply for row 1 (Dublin) is now reduced to zero, while the column 2 (Sligo) free demand is now 120. As the row total is now zero, we move down one cell.

The free demand for the second column (Sligo) is 120 and the free demand for the second row (Cork) is 150. We therefore allocate 120 units to this cell. The free supply for the second row then becomes 30 and the free demand for the second column becomes zero. As the column total is now zero, we move one cell to the right.

We are now in the cell supplying Athlone from Cork. The free demand for this column (column 3) is 130, while the free supply for this row (row 2) is 30. We therefore allocate 30 units to this cell. The free supply for this row now becomes zero, and the free demand 100. As the free demand for the row is zero, we move down one cell.

Following the same method as before, we place 100 units in this cell. The initial table is now completed. It is easy to show that this solution costs €4540.

▶ See **video V19_Transp_2**, shows how to obtain an initial solution using the North West Corner method. https://goo.gl/6gKKq0

The North West Corner Method is not the most efficient as it does not take into account the cost of each supply route when deciding on the allocations to each cell. One method which does this is the **Least Cell Cost Method**. The basic steps of this method are as follows:

1. Allocate as much as possible to the cell with the lowest cost. Adjust its Row and Column Free Supply figure.
2. From the remaining cells whose column has some free demand and whose row has some free supply select the one with the lowest costs.
3. Stop when all supply and demand has been allocated.

We use this approach to establish a second starting solution for the transportation problem from section 4.2.

| | WATERFORD | | SLIGO | | ATHLONE | | |
|---|---|---|---|---|---|---|---|
| DUBLIN | 30 | 9 | 40 | 11 | 130 | 8 | 200 |
| CORK | 150 | 6 | | 15 | | 10 | 150 |
| GALWAY | | 7 | 100 | 6 | | 6 | 100 |
| | 180 | | 140 | | 130 | | |

The lowest cell cost value is 6 and there are three cells with this cost. Of these we pick the cell that we can allocate the most units to. Hence we begin by allocating 150 units to the (Cork, Waterford) cell. The free supply for row 2 (Cork) becomes zero, while the free demand for column 1 (Waterford) is now 30 units. We can no longer use Cork as a supply centre as all of the resources it has available are being sent to Waterford.

We look for the next cell with least cost. There are two cells available to us with associated cost 6. We can allocate 100 units to the (Galway, Sligo) cell or 100 units to the (Galway, Athlone) cell. We pick one of these at random and allocate 100 units to the (Galway, Sligo) cell. The free supply for row 3 (Galway) is now zero and the free demand for column 2 is now 40 units. This means that we can no longer use the Galway supply centre as it is sending all its available resources to Sligo.

Of the remaining available routes, the cell with the lowest cost is the (Dublin, Athlone) cell which has cost 8 units. There are 200 units available for supply from Dublin and Athlone has a demand of 130 units. Hence we can supply 130 units from Dublin to Athlone. Athlone now has a demand of zero, while 70 units remain available for supply from Dublin.

There are only two cells remaining available for supply (Dublin, Waterford) and (Dublin, Sligo). We pick the cell with the lowest cost which is the (Dublin, Waterford) route. There are 70 units of free supply left in Dublin, while Waterford has a free demand of 30 units. Hence we supply 30 units to this route.

The remaining 40 units in Dublin are supplied to the (Dublin, Sligo) route to satisfy the demand there. We have now found a second feasible initial solution using the Least Cost Method. The cost of this solution is easily calculated to be:

| | | | |
|---|---|---|---|
| DUBLIN | $(30 \times 9) + (40 \times 11) + (130 \times 8)$ | = | €1750 |
| CORK | $(150 \times 6)$ | = | €900 |
| GALWAY | $(100 \times 6)$ | = | €600 |
| Total Cost | | = | €3250 |

Note that this is €1290 cheaper than the initial solution found using the North-West Corner Method.

▶ See video **V20_Transp_3**, shows how to obtain an initial solution using the Least Cost method.
https://goo.gl/UY38Rm

The least cost approach utilises cells with lowest absolute cost, it does not take into account the fact that one row or column may have several cells which have lower costs than others in the tableau. Vogel's Approximation Method (VAM) attempts to identify the greatest marginal advantage of a cell over the next best cell in its row or column. This approach generally yields a close to optimum starting solution and is superior to the least cost method and very much better than the North-West corner approach. The basic steps of Vogel's method are the following:

1. Calculate a penalty for each row and each column by subtracting the smallest cost element in the row or column from the next smallest cost for the same row or column.
2. Identify the row or column with the greatest penalty. Allocate as much as possible to the cell with the least cost in this row or column. Adjust the free demand and supply totals and cross out the row or column whose demand or supply is entirely used up. If both the row and the column have no supply or demand left only cross out one.
3. If all columns and rows have been satisfied then we have an initial solution. Otherwise, recalculate the penalties for uncrossed out rows and columns without counting rows and columns with zero free demand or supply.

| | WATERFORD (j = 1) | SLIGO (j = 2) | ATHLONE (j = 3) | |
|---|---|---|---|---|
| DUBLIN (i = 1) | 9 | 11 | 8 | 200 |
| CORK (i = 2) | 6 | 15 | 10 | 150 |
| GALWAY (i = 3) | 7 | 6 | 6 | 100 |
| | 180 | 140 | 130 | |

Step I

| | | |
|---|---|---|
| Row 1: min cost = 8 | second best = 9 | penalty = 9-8 =1 |
| Row 2: min cost = 6 | second best = 10 | penalty = 10-6 =4 |
| Row 3: min cost = 6 | second best = 6 | penalty = 6-6 =0 |

| | | |
|---|---|---|
| Column 1: min cost = 6 | second best = 7 | penalty = 7-6 =1 |
| Column 2: min cost = 6 | second best = 11 | penalty = 11-6 =5 |
| Column 3: min cost = 6 | second best = 8 | penalty = 8-6 =2 |

Therefore the maximum penalty is associated with column 2. We allocate as many units as possible to the lowest cost cell in this column. This is the Galway Sligo route and we allocate it 100 units. All the units available for supply from Galway are now being sent to Sligo, so we cross out the Galway row.

Step II

We must recalculate the penalties ignoring row 3 (Galway).

| | | |
|---|---|---|
| Row 1: min cost = 8 | second best = 9 | penalty = 9-8 =1 |
| Row 2: min cost = 6 | second best = 10 | penalty = 10-6 =4 |

| | | |
|---|---|---|
| Column 1: min cost = 6 | second best = 9 | penalty = 9-6 =3 |
| Column 2: min cost = 11 | second best = 15 | penalty = 15-11 =4 |
| Column 3: min cost = 8 | second best = 10 | penalty = 10-8 =2 |

We now have a situation where both row 2 and column 2 have the same penalty of 4. In this case we arbitrarily choose one, say row 2. The least cost cell in this row is the (Cork, Waterford) cell. Cork can supply 150 units and Waterford has a demand of 180 units. Therefore we can allocate 150 units to this cell. Cork has no more units left for supply so we cross out row 2. Waterford has a remaining demand of 30 units.

Step III

We are now left with only one row and we allocate units to each of the cells according to their remaining demands. Waterford has a remaining demand of 30 units, so this is supplied from Dublin. Similarly, Sligo and Athlone have remaining demands of 40 and 130 units respectively.

This results in the following initial feasible solution.

| | WATERFORD | SLIGO | ATHLONE | |
|---|---|---|---|---|
| DUBLIN | 30 \| 9 | 40 \| 11 | 130 \| 8 | 200 |
| CORK | 150 \| 6 | 15 | 10 | 150 |
| GALWAY | 7 | 100 \| 6 | 6 | 100 |
| | 180 | 140 | 130 | |

This is the same solution as obtained using the Least Cost Method, and therefore has the same associated cost of €3250.

The NWC method solution costs €4540, that is, €1290 or 39.7% greater than the other two methods. The NWC method is a very inefficient starting procedure and should be abandoned in favour of VAM.

▶ See **video V21_Transp_4**, shows how to obtain an initial solution using Vogel's Approximation method.
https://goo.gl/unnRqo

## *SAQ 2*

*Irish Widgets Plc has two manufacturing plants, one in Cork and one in Belfast. The company has important customers in Derry, Ballina, Dublin and Wexford. The quantities supplied and demanded by each centre and the costs of sending a widget between them is given as follows:*

*Production Quantities*
*Cork 200*
*Belfast 160*

*Demand Quantities*
*Derry 60*
*Ballina 75*
*Dublin170*
*Wexford 55*

| | *Derry* | *Ballina* | *Dublin* | *Wexford* |
|---|---|---|---|---|
| *Cork* | *€28* | *19* | *16* | *12* |
| *Belfast* | *7* | *16* | *11* | *19* |

*Set up the transportation tableau and determine the initial feasible solution and determine the cost of the solution*

### 4.5.2 OPTIMALITY AND OBTAINING A FINAL SOLUTION

Test for Optimality
Is €3250 the minimum possible cost of transportation for the problem from section 4.2? We will now explore two methods that can be used to obtain the final, optimal solution, the Stepping Stone methods and the MODI method.

Before we begin we must check that the initial feasible solution obtained is not degenerate. (We will discuss this concept in more detail later on). This is done by checking that the number of cells used is equal to the number of rows (m) + the number of columns (n) – 1. This concept of checking that the current solution contains exactly m + n – 1 routes is fundamental to the successful implementation of the transportation algorithm.

The **Stepping Stone Method** evaluates the change in total cost of using a cell which not currently being used. If we decide to use such a cell, then this will affect other cells which are not currently being used. The total sent via all routes from any source must continue to equal the supply available at that source. Also, the amount sent to any destination must continue to equal the demand at that destination. Therefore, if we decide to send a unit via a previously unused route, the amount sent via another route from that source must decrease by one. In the tableau, if we use a currently unused cell then the value of one other cell in the row and column of the new cell must decrease by one.

If we look at the initial solution obtained using the VAM method for the problem from section 4.2, we might be interested in the effect of using the route from Cork to Sligo. If we send one unit from Cork to Sligo, then the amount sent from Dublin to Sligo must decrease by 1. This will in turn require that the quantity sent from Dublin to Waterford must increase. This change will mean that less will be sent from Cork to Waterford. This makes sense since we require less to be shipped from Cork to a destination other than Sligo so that we some can be sent to Sligo. Waterford is such a destination. Therefore, the changes form a unique "loop" in the tableau, thus ensuring that the tableau remains valid.

| | **WATERFORD** | **SLIGO** | **ATHLONE** | |
|---|---|---|---|---|
| **DUBLIN** | **9**<br>**30 + 1** | **11**<br>**40 – 1** | **8**<br>**130** | **200** |
| **CORK** | **6**<br>**150 – 1** | **15**<br>**+ 1** | **10** | **150** |
| **GALWAY** | **7** | **6**<br>**100** | **6** | **100** |
| | **180** | **140** | **130** | |

Exercise: Confirm that the above loop is unique and that no other way of compensating for the inclusion of a one unit on the Cork-Sligo route can be found. In your attempt at finding an alternative, ensure that you remember the importance of the m + n – 1 rule.

So what effect will this change have on the cost? Some cells will have an extra unit, while others will have one unit less. If the costs of the cells with one unit less are greater in total than the cells with one unit more then the total cost will fall. In this case:

| Increased amounts: | Cost |
|---|---|
| Cork-Sligo: | 15 |
| Dublin-Waterford: | 9 |
| Total: | 24 |

| Decreased amounts: | Cost |
|---|---|
| Dublin-Sligo: | 11 |
| Cork-Waterford: | 6 |
| Total: | 17 |

Overall Change in Cost: 24 – 17 = 7. Therefore there will be an increase in costs of 7 units if we sent one unit via the Cork Sligo route, hence it will not reduce our costs to include the route in our solution.

We examine the other possible routes for inclusion

Cork-Athlone: +10 – 8 + 9 – 6 = +5
Galway – Waterford: +7 – 9 + 11 – 6 = +3
Galway – Athlone: +6 – 8 + 11 – 6 = +3

None of these routes offer a potential decrease in the total cost. Therefore the current solution is optimal. If this was not the case then we would have to bring the route which offered the highest cost savings into the solution. We will see how to do this later.

This method is called the stepping stone method as you step between the "stones" (i.e. filled cells) to form the "loop" for each unfilled cell.

▶ See video **V22_Transp_5**, which demonstrates the use of the Stepping Stone method (using the initial tableau found in the video V21_Transp_4). https://goo.gl/Yz8iQ7

Next we consider an alternate method for obtaining the optimal solution, the **MODI** method. It is intuitively obvious that if a lower cost solution exists, then some of the unallocated cells CORK / SLIGO, CORK / ATHLONE, GALWAY / WATERFORD and GALWAY / ATHLONE must be included in the new solution. This implies that there is an "opportunity cost" that is not being availed of by excluding some, or all, of these cells from the solution. It is therefore necessary to compute the opportunity costs of these cells and determine their effect on the current feasible solution. Now, we know that the equation of a straight line is:

y = a + b x
Where:
a is the intercept on the y axis and b is the slope of the line

Let us assume (for simplicity) that the slope, b = 1, and the resulting equation is redefined as:

| **y** | **= a** | **+ x** |
|---|---|---|
| Unit Distribution Cost of the Cell | Distribution Cost Component due to Location of Source | Distribution Cost Component due to Location of Destination |

Or, in more general terms:

$c_{ij} = u_i + v_j$ $\qquad$ $i = 1, 2, 3$ $\qquad$ $j = 1, 2,3$

We start the optimality testing routine by concentrating on the *occupied* cells and writing down their cost equations as follows:

| | $v_1$ | $v_2$ | $v_3$ |
|---|---|---|---|
| | 9 | 11 | 8 |
| $u_2$ | 6 | | |
| | | 6 | |

$\mathbf{c}_{ij}$
$u_1 + v_1 = 9$
$u_1 + v_2 = 11$
$u_1 + v_3 = 8$
$u_2 + v_1 = 6$
$u_3 + v_2 = 6$

In order to be able to solve these equations (for u and v) we must firstly test for the following condition:

No of Rows (**R**) + No of Columns (**C**) – 1 = No of Allocations (**A**)

That is,

3 + 3 – 1 = 6 – 1 = 5 = No of Allocations

As you can see, that condition is satisfied. If it is not then the problem is *degenerate,* but that is another story. Again, as you can see, we have five (5) equations and six (6) unknowns, thus the solution is *indeterminate* or:

$u_1 = f(u_2, u_3, v_1, v_2, v_3)$

$u_2 = f(u_1, u_3, v_1, v_2, v_3)$ etc.

However, if we let, for instance, $u_1 = 0$ (or any number you like) then the indeterminacy is broken, and we can evaluate the remaining variables as follows:

| | | |
|---|---|---|
| $u_1 + v_1 = 9$ | Let $u_1 = 0$ | $v_1 = 9$ |
| $u_1 + v_2$ | $= 11$ | $v_2 = 11$ |
| $u_1 + v_3$ | $= 8$ | $v_3 = 8$ |
| $u_2 + v_1$ | $= 6$ | $u_2 = -3$, since $v_1 = 9$ |
| $u_3 + v_2$ | $= 6$ | $u_3 = -5$, since $v_2 = 11$ |

We are now in a position to evaluate the opportunity costs of the unallocated cells as follows:

| | $v_1 = 9$ | $v_2 = 11$ | $v_3 = 8$ |
|---|---|---|---|
| $u_1 = 0$ | | | |
| $u_2 = -3$ | | 8 15 | 5 10 |
| $u_3 = -5$ | 4 7 | | 3 6 |

Now:

$8 = -3 + 11$: $5 = -3 + 8$

$4 = -5 + 9$: $3 = -5 + 8$

For each empty cell the opportunity cost is defined by:

$OC = c_{ij} - (u_i + v_j)$, or:

| | |
|---|---|
| Cell ($u_2$, $v_2$) | $OC = 15 - 8 = 7$ |
| Cell ($u_2$, $v_3$) | $OC = 10 - 5 = 5$ |
| Cell ($u_3$, $v_1$) | $OC = 7 - 4 = 3$ |
| Cell ($u_3$, $v_3$) | $OC = 6 - 3 = 3$ |

Recall that in a minimization simplex tableau the optimal solution to the problem is arrived at when there are zeros and positives in the Index Row. Since the occupied cells, are "in the solution" the differences for them will be zero and therefore we have, in total, zeros and positives for all of the cells. Hence €3250 is the optimal solution.

▶ See video **V23_Transp_6**, which demonstrates the use of the MODI method (using the initial tableau found in the video V21_Transp_4). https://goo.gl/aaTUdD

## *SAQ 3*

*Perform the optimality test for the solution to SAQ2*

## *SAQ 4*

*Consider the following transportation problem:*

*Sources*
*Plant A 400 units available*
*Plant B 200 units available*
*Plant C 350 units available*

*Demand*
*Warehouse 1 300 units required*
*Warehouse 2 250 units required*
*Warehouse 3 400 units required*

*Unit Transport Cost, from Plant to Warehouse*

| | *Warehouse1* | *Warehouse2* | *Warehouse3* |
|---|---|---|---|
| ***Plant A*** | *50* | *32* | *40* |
| ***Plant B*** | *16* | *30* | *20* |
| ***Plant C*** | *35* | *28* | *42* |

*Develop a distribution plan that minimises total transport cost.*

If, however, one or more of the unoccupied cells were to show a negative opportunity cost then the optimal solution would not have been found. In this situation, select that cell which has the larger negative value and make an allocation $\theta$ to that cell. Use the stepping stone method to adjust all other cells which this affects by adding or subtracting $\theta$. Determine the maximum value which $\theta$ may have, adjust the allocations accordingly and check again for optimality. Continue the process until the optimal solution is found.

## *SAQ 5*

*Consider the following transportation problem.*

*Supply Data*
*Source A = 70*
*Source B = 40*
*Source C = 100*

*Demand Data*
*Destination 1 = 30*
*Destination 2 = 60*
*Destination 3 = 90*
*Destination 4 = 30*

*Cost Data*

| | **Dest.1** | **Dest.2** | **Dest.3** | **Dest.4** |
|---|---|---|---|---|
| **Source A** | 7 | 6 | 2 | 12 |
| **Source B** | 3 | 9 | 8 | 7 |
| **Source C** | 10 | 4 | 11 | 5 |

*Find the optimal solution.*

***UNBALANCED PROBLEMS***

In reality many transportation problems will not have equal supply and demand. In most situations supply will exceed demand but the opposite can also occur. However, this unbalance between supply and demand is not a problem when using the transportation technique. We simply make the problem a balanced one by adding a dummy source or destination. A dummy source is introduced if the demand is greater than the supply and a dummy destination, if the supply is greater than the demand. We allocate a cost of 0 to the cells in the dummy source or destination, as no transportation cost is incurred if goods are not shipped.

***PROHIBITED ROUTES***

In many realistic situations there will be routes between pairs of sources and destinations which cannot be used. In this case we allocate a very large cost to this prohibited route. Since we are minimising cost this will ensure that the route will not be used. In a similar way to the use of artificial variables in the simplex tableau we usually refer to this very large cost as M. We then proceed with the solution procedure in the normal way. Even if the initial solution uses the prohibited route, the algorithm will alter the solution to use a different route.

## *SAQ 6*

*A brick manufacturing company transports bricks from three plants to three constructions sites. The supply capacities of the three plants, the demand requirements at the three sites and the transportation costs (€/ton) are as follows:*

| | ***To Construction Site*** | | | ***Supply*** |
|---|---|---|---|---|
| ***From Plant*** | ***(1)*** | ***(2)*** | ***(3)*** | ***(Tons)*** |
| ***A*** | *10* | *7* | *8* | *240* |
| ***B*** | *17* | *12* | *14* | *160* |
| ***C*** | *5* | *11* | *12* | *200* |
| ***Demand(Tons)*** | *300* | *140* | *120* | |

*You are required to:*

1. *Portray the problem as a network model*
2. *Develop a linear programming formulation of the problem*
3. *Determine the optimal solution to the problem using the MODI algorithm*
4. *Ascertain the optimal transportation cost/ton*

## *SAQ 7*

*A company wishes to supply its 3 customers from 3 depots at minimum cost. Given the data below, identify the optimal routes to use.*

*Cost data*

| | **Dest.1** | **Dest.2** | **Dest.3** |
|---|---|---|---|
| **Source 1** | 9 | 10 | 6 |
| **Source 2** | M* | 5 | 13 |
| **Source 3** | 5 | 4 | 10 |

** the route from Source 2 to Destination 1 may not be used.*

*Supply Data*
*Source 1 = 100*
*Source 2 = 80*
*Source 3 = 90*

*Demand Data*
*Destination 1 = 105*
*Destination 2 = 80*
*Destination 3 = 85*

### *4.5.3 THE TRANSHIPMENT PROBLEM*

The solution to the transhipment problem depicted in Fig 4.6 is the same as that described for the transportation problem except that in the initial VAM, those cells with costs of ∞ are ignored. Thus no allocations to these cells can take place.

### *4.5.4 THE ASSIGNMENT PROBLEM*

The assignment problem can be solved by the transportation method. However, a simpler procedure known as The Hungarian Method is usually used because it is more efficient.

Figure 4.4 depicted the following situation:

| ***SALESPERSON (i)*** | ***TERRITORY (j)*** | | | ***SUPPLY*** |
|---|---|---|---|---|
| | ***1*** | ***2*** | ***3*** | |
| ***1*** | ***700*** | ***300*** | ***500*** | ***1*** |
| ***2*** | ***800*** | ***900*** | ***200*** | ***1*** |
| ***3*** | ***900*** | ***600*** | ***800*** | ***1*** |
| ***DEMAND*** | ***1*** | ***1*** | ***1*** | ***3*** |

The problem is to allocate a salesperson to a territory so that total weekly costs are minimized. The Hungarian Method commences by developing an "opportunity cost" table which is similar to the opportunity cost procedure discussed in the VAM method. It is usual to start with the rows and then continue with the columns. For each row, find the lowest cost in that row and subtract it from all other costs in the row. This results in:

| SALES PERSON \ TERRITORY | ***1*** | ***2*** | ***3*** | |
|---|---|---|---|---|
| ***1*** | ***400*** | ***0*** | ***200*** | ***(Subtracting 300)*** |
| ***2*** | ***600*** | ***700*** | ***0*** | ***(Subtracting 200)*** |
| ***3*** | ***300*** | ***0*** | ***200*** | ***(Subtracting 600)*** |

Since Supply and Demand always have a value of one, the Supply and Demand requirements are not required in the table. Next, subtracting the lowest cost in each column from all other costs in the column gives:

| SALES PERSON \ TERRITORY | ***1*** | ***2*** | ***3*** |
|---|---|---|---|
| ***1*** | ***100*** | ***0*** | ***200*** |
| ***2*** | ***300*** | ***700*** | ***0*** |
| ***3*** | ***0*** | ***0*** | ***200*** |

The zero elements represent the lowest opportunity costs of assignment. For an optimal assignment to exist the table must result in three unique assignments. A methodology for determining this condition is to draw the minimum number of lines, vertically and /or horizontally, to cover all the zeros. If the minimum number of lines is equal to the number of rows or columns, then an optimal assignment exists.

For the problem under consideration we have an optimum solution, because three lines are required to cover all the zeros

| SALES PERSON \ TERRITORY | ***1*** | ***2*** | ***3*** |
|---|---|---|---|
| ***1*** | ***100*** | ***0*** | ***200*** |
| ***2*** | ***300*** | ***700*** | ***0*** |
| ***3*** | ***0*** | ***0*** | ***200*** |

| ***Sales Person*** | ***Territory*** | ***Cost*** |
|---|---|---|
| 1 | 2 | €300 |
| 2 | 3 | €200 |
| *3* | *1* | *€900* |
| | | |
| Total cost | | €1400 |

To illustrate the procedure if the minimum number of lines is not equal to the number or rows or columns, consider the following enlarged problem:

| SALES PERSON \ TERRITORY | ***1*** | ***2*** | ***3*** | ***4*** |
|---|---|---|---|---|
| ***1*** | ***2000*** | ***3000*** | ***3200*** | ***3600*** |
| ***2*** | ***2600*** | ***3400*** | ***3200*** | ***2000*** |
| ***3*** | ***2200*** | ***1800*** | ***1600*** | ***3600*** |
| ***4*** | ***2600*** | ***2600*** | ***3200*** | ***1800*** |

Row reduction gives:

| SALES PERSON \ TERRITORY | ***1*** | ***2*** | ***3*** | ***4*** |
|---|---|---|---|---|
| ***1*** | ***0*** | ***1000*** | ***1200*** | ***1600*** |
| ***2*** | ***600*** | ***1400*** | ***1200*** | ***0*** |
| ***3*** | ***600*** | ***200*** | ***0*** | ***2000*** |
| ***4*** | ***800*** | ***800*** | ***1400*** | ***0*** |

Column reduction gives:

| SALES PERSON \ TERRITORY | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **1** | ***0*** | ***800*** | ***1200*** | ***1600*** |
| **2** | ***600*** | ***1200*** | ***1200*** | ***0*** |
| **3** | ***600*** | ***0*** | ***0*** | ***2000*** |
| **4** | ***800*** | ***600*** | ***1400*** | ***0*** |

Thus the number of lines is three when it should be four, and therefore we have not reached an optimal assignment. The procedure that is now adopted is to subtract the smallest uncrossed cell from all other uncrossed cells and to add the same number to those cells at the intersection of lines leaving the single crossed cells values the same. As 600 is the smallest uncrossed number we now obtain:

| SALES PERSON \ TERRITORY | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **1** | ***0*** | ***800*** | ***1200*** | ***2200*** |
| **2** | ***0*** | ***600*** | ***600*** | ***0*** |
| **3** | ***600*** | ***0*** | ***0*** | ***2600*** |
| **4** | ***200*** | ***0*** | ***800*** | ***0*** |

We now have four required lines, in this case horizontally (but note that four vertical lines or other combinations of four lines are also possible) and the optimal assignment is (marked by the zeroes in the table), therefore:

| Sales Person | Territory | Cost |
|---|---|---|
| 1 | 1 | 2000 |
| 2 | 4 | 2000 |
| 3 | 3 | 1600 |
| 4 | 2 | 2600 |
| Total cost | | 8200 |

▶ See video **V24_Assign_1**, which demonstrates the initial stages of the Hungarian Method. https://goo.gl/IfLzUE

▶ See video **V25_Assign_2**, which completes the application of the Hungarian Method. (This follows from the previous video, V24_Assign_1.) https://goo.gl/021fgc

▶ See video **V26_Assign_3**, which provides an interpretation of the final table from the Hungarian Method (from the previous video, V25_Assign_2). https://goo.gl/EqYZIc

## *SAQ 8*

*A courier company has divided Dublin into four regions and wishes to assign four of its drivers, one to each region, on the basis of minimising its total cost. After carrying out a series of studies the company produced the following set of cost figures:*

*Daily driver cost (€) per region*

| ***Region/ Driver*** | ***North*** | ***South*** | ***East*** | ***West*** |
|---|---|---|---|---|
| ***Andy*** | *50* | *75* | *80* | *90* |
| ***Billy*** | *70* | *65* | *80* | *50* |
| ***Con*** | *55* | *45* | *40* | *90* |
| ***David*** | *65* | *65* | *55* | *45* |

*You are required to:*

1. *Assign drivers to areas in optimal way*
2. *Compute the minimum daily cost incurred by the company*
3. *Check if the solution is unique.*
4. *Generate other solutions if they exist.*

### SUMMARY

In this unit three types of linear programming problems were examined and solved using tailored and efficient methods of tableau solution taking into account their particular structure. The three areas considered were the transportation problem, the more general transhipment problem and the assignment problem.

Each can be formulated as linear programming problems and in solving them it was assumed that the objective was one of minimisation (typically a cost function). In the case of the transportation problem we are presented with a set of supply nodes and a set of demand nodes and unit transportation costs for each possible route. As such the information can be presented as a network type diagram or for solution purposes a transportation tableau. Minimising the total transportation cost whilst satisfying the supply and demand constraints as well as any prohibited routes is the objective. Naturally we may also wish to maximise a profit or revenue function and we need to amend the approach slightly as detailed below in the summary of steps. The solution procedure consisted of a few well defined steps. The first step consisted of constructing the initial tableau incorporating all the information and constraints and costs. An initial solution using one of three approaches is then determined and checked for optimality. If suboptimal several iterations of the Modi method can be applied until the optimal solution is achieved.

The transhipment problem is a generalisation of the transportation problem and as such can be represented in network diagram form and solved by a more general and larger transportation tableau. Solving these problems by hand rapidly becomes laborious and consequently software solutions are the norm.

The assignment problem is a simplified and specialised version of a transportation problem and as such also consists of a set of supply and demand constraints and unit assignment costs. The defining feature of the problem is that of unit demand and supply levels so that the decision variables are binary (assign or not assign). Given the structure a specialised algorithm termed the Hungarian Method is available for an efficient solution. Once again the problem may be represented in network diagram form but in terms of determining a solution a tableau is utilised. The simple steps consisted of row and column reductions followed by a test for optimality. If suboptimal alterations are made to the table entries according to a simple rule until the optimal solution is found.

As always in all the above problem types the solution may change if some of the features of the problem change. Conducting sensitivity analyses allows one to assess the impact on the solution if for example unit costs changed.

When the optimal solution is found, as before it is good practice to generate a simple report on the solution detailing which routes/allocations are made and the cost implications discussed.

## SUMMARY OF STEPS

The transportation problem:

1. Identify the structure as that of a transportation problem and identify all the decision variables and unit costs. If a formulation as a linear programming problem is required, all supply and demand constraints and prohibited routes need to be identified**.** If required a network diagram can be used as a representation of the problem.

2. Construct the **initial transportation tableau** with unit costs and supply and demand totals detailed.

3. Check whether the issue is one of **maximisation or minimisation**. If the problem is one of maximisation (unit profits instead of unit costs available for each route) convert to an equivalent minimisation problem by subtracting all unit profits in the table from the largest (most profitable route) value and proceed as normal with the new table treating as a minimisation problem.

4. Check whether the problem is balanced or not. **If unbalanced** add a dummy row (if supply is less than demand) or a dummy column (if supply is more than demand). The unit costs for each fictitious route will be zero and the total of the dummy row or column will be such as to make up the shortfall between supply and demand.

5. Check for **prohibited routes** and where they exist assign a large cost (usually denoted by 'M') relative to the other routes. This has the effect of ultimately ensuring that this costly route is avoided in the final optimal solution.

6. The initial tableau is now ready for inserting an **initial feasible solution**.

7. Use one of the three methods (North West Corner, Least Cost Method, Vogel's method) to generate an initial feasible but typically suboptimal solution.

8. Check whether **degeneracy** exists by seeing if m+n-1 routes are being used when the table consists of m rows and n columns. If less than this number of routes is being used degeneracy exists and an extra route needs to be added to the current solution set (occupied routes) by treating a currently empty route as occupied but with zero units being routed along it. The hope is that the degeneracy will be removed in the next iteration and m+n-1 cells will be occupied. If not, then try a different candidate route for allocating the zero value to.

9. Apply the Modi method to **check for optimality**. This consists of calculating so called u and v values using $u_1 = 0$ and the condition:

   $c_{ij} = u_i + v_j$ for occupied routes.

   For each empty route calculate the **opportunity cost** defined by:

   $$OC = c_{ij} - (u_i + v_j)$$

   If all the opportunity costs are positive the current solution is optimal. If any are negative then choose the route with the most negative value and assign as much as possible (use the stepping stone) to this cost reducing route. This new route enters the solution set and an existing route leaves.

10. Repeat step 9 until all opportunity costs are positive. The optimal solution can be identified and the total transportation cost determined. If there are other optimal solutions (multiple optimal solutions) they can be identified if an opportunity cost is zero for an unused cell in an otherwise optimal solution.

11. **Generate** a solution report and implement and review the findings as needed. If necessary, sensitivity analyses can be conducted.

The Assignment problem:

1. Identify the structure as that of an Assignment problem and identify all the decision variables (possible assignments) and unit costs. If a formulation as a linear programming problem is required, all supply and demand constraints and prohibited assignments need to be identified**.** If required a network diagram can be used as a representation of the problem.

2. Check whether the issue is one of **maximisation or minimisation**. If the problem is one of maximisation (unit profits/revenues instead of unit costs available for each assignment) convert to an equivalent minimisation problem by subtracting all unit profits in the table from the largest (most profitable assignment) value and proceed as normal with the new table treating as a minimisation problem.

3. Check whether the problem is balanced or not. **If unbalanced** add a dummy row (if supply is less than demand) or a dummy column (if supply is more than demand). The unit costs for each fictitious cell will be zero.

4. Check for **prohibited assignments** and where they exist assign a large cost (usually denoted by ‘M’) relative to the other routes. This has the effect of ultimately ensuring that this costly assignment is avoided in the final optimal solution.

5. The initial tableau is now ready for **row and column reduction**. Perform row reduction (For each row, find the lowest cost in that row and subtract it from all other costs in the row) followed by column reduction.

6. **Check whether** the resulting tableau is **optimal** by crossing out all the zeros with as few horizontal and/or vertical lines as possible. If the minimum number of lines is equal to the number of rows or columns, then an optimal assignment exists. Assignments are flagged by the cells where the zeros are and the optimal solution can be identified.

Multiple optimal solutions (if sufficient zeros are present) may be present at the same overall cost.

7. If the current solution is not optimal (less than m or n lines are needed to cross out all zeros) the procedure that is now adopted is to subtract the smallest uncrossed cell from all other uncrossed cells and to add the same number to those cells at the intersection of lines leaving the single crossed cells values the same. Repeat step 6 until optimality is achieved.

8. Generate a solution report.

## FINAL COMMENTS

The problem types encountered above represent an important first step in the analysis of many logistic type problems. Many important problem types fit into the template of a transportation or transhipment problem. Each is a form of linear programming and as usual a first step is the correct formulation of the problem incorporating the various constraints.

Solving realistic problems however will often require including many variables and constraints and hence software solutions are the norm. The student should augment the examples and exercises provided by studying some further problems to be found in most management science/Operational Research textbooks and web resources. It would be useful to solve some of the exercises using an Excel spreadsheet and try varying some of the cells and conditions and resolving to see the effects of parameter changes on the optimal solution as determined by the Solver add-in.

# ANSWERS TO SAQS

## *SAQ 1*

*(a) Linear programming formulation*

*Let $x_{ij}$ represent the amount transported from source I to destination j.*

*MIN* $80x_{11}+215x_{12}+100x_{21}+108x_{22}+102x_{31}+68x_{32}$

*Subject to:*

$x_{11} + x_{12} = 1000$ *(total flows from source 1 = supply)*
$x_{21} + x_{22} = 1500$ *(total flows from source 2 = supply)*
$x_{31} + x_{32} = 1200$ *(total flows from source 3 = supply)*

$x_{11} + x_{21} + x_{31} = 2300$ *(total flows to destination 1 = demand)*
$x_{12} + x_{22} + x_{32} = 1400$ *(total flows to destination 2 = demand)*

*(b) Optimal solution is:*
*x11 = 1000*
*x21 = 1300*
*x22 = 200*
*x32 = 1200*

*(c) Standard transportation tableau*

| | Dest 1 | Dest 2 | Supply |
|---|---|---|---|
| **Source 1** | **80** | **215** | **1000** |
| **Source 2** | **100** | **108** | **1500** |
| **Source 3** | **102** | **68** | **1200** |
| **Demand** | **2300** | **1400** | **3700** |

## *SAQ 2*

*The initial transportation tableau is:*

| | Derry | Ballina | Dublin | Wexford | Supply |
|---|---|---|---|---|---|
| **Cork** | 23 | 19 | 16 | 12 | **200** |
| **Belfast** | 7 | 16 | 11 | 19 | **160** |
| **Demand** | **60** | **75** | **170** | **55** | **360** |

*Using Vogel's Approximation method, we get the following penalties:*

*Row 1: 16-12 =4*
*Row 2: 11-7 = 4*

*Column 1: 28-7 = 21*
*Column 2: 19-16 = 3*
*Column 3:16-11 = 5*
*Column 4: 19-12 = 7*
*Therefore the maximum penalty is associated with column 1, and we allocate as many units as possible to the Belfast- Derry route.*

| | Derry | Ballina | Dublin | Wexford | Supply |
|---|---|---|---|---|---|
| **Cork** | 28 | 19 | 16 | 12 | **200** |
| **Belfast** | 7<br>60 | 16 | 11 | 19 | **100** |
| **Demand** | **0** | **75** | **170** | **55** | |

*We recalculate the penalties:*
*Row 1: 16-12 = 4*
*Row 2: 16-11 = 5*

*Column 2: 19-16 = 3*
*Column 3: 16-11 = 5*
*Column 4: 19-12 = 7.*
*Now the maximum penalty is associated with column 4, so we allocate as much as possible to the cell in this column with the lowest costs, this is Cork- Wexford:*

| | Derry | Ballina | Dublin | Wexford | Supply |
|---|---|---|---|---|---|
| **Cork** | 28 | 19 | 16 | 12 | **130** |
| **Belfast** | 7 | 16 | 11 | 19 | **100** |
| **Demand** | **0** | **75** | **170** | **0** | |

*We recalculate the penalties, ignoring the rows and columns with zero supply and demand:*
*Row 1: 19-16 = 3*
*Row 2: 16-11=5*
*Column 2: 19-16=3*
*Column 3: 16-11=5*

*We now have a situation where row2 and column3 have the same penalty, 5. We arbitrarily choose one, say row 2. We allocate as much as possible to the lowest cost cell left in the row, this is Belfast-Dublin. If we had selected the column we would also have used this cell.*

*At this stage only one row remains unsatisfied and so we just allocate the remaining amount. Finally, we get the Initial Tableau*

| | Derry | Ballina | Dublin | Wexford | Supply |
|---|---|---|---|---|---|
| **Cork** | 28 | 19<br>75 | 16<br>70 | 12<br>55 | **200** |
| **Belfast** | 7<br>60 | 16 | 11<br>100 | 19 | **160** |
| **Demand** | **60** | **75** | **170** | **55** | |

*The cost of this solution is:*

| | | |
|---|---|---|
| *Cork – Ballina 75 × 19* | | *1425* |
| *Cork – Dublin 70 × 16* | | *1120* |
| *Cork – Wexford* | *55 × 12* | *660* |
| *Belfast – Derry* | *60 × 7* | *420* |
| *Belfast – Dublin* | *100 × 11* | *1100* |
| *Total Cost* | | *4725* |

*Note that the table has 5 routes (3+3-1) and so is not degenerate.*

## *SAQ 3*

*The initial feasible solution is:*

| | Derry | Ballina | Dublin | Wexford | Supply |
|---|---|---|---|---|---|
| **Cork** | 28 | 19<br>75 | 16<br>70 | 12<br>55 | **200** |
| **Belfast** | 7<br>60 | 16 | 11<br>100 | 19 | **160** |
| **Demand** | **60** | **75** | **170** | **55** | |

*Taking the allocated cells:*

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $u_1$ | | 19 | 16 | 12 |
| $u_2$ | 7 | | 11 | |

| | $c_{ij}$ |
|---|---|
| $u_1 + v_2$ | $= 19$ |
| $u_1 + v_3$ | $= 16$ |
| $u_1 + v_4$ | $= 12$ |
| $u_2 + v_1$ | $= 7$ |
| $u_2 + v_3$ | $= 11$ |

*No of Rows (**R**) + No of Columns (**C**) - 1 = No of Allocations (**A**)*

*That is,*

*4 + 2 – 1 = 6 – 1 = 5 = No of Allocations*

*The condition is satisfied.*

*Let* $u_1 = 0$

| | | |
|---|---|---|
| $u_1 + v_2$ | $= 19$ | $v_2 = 19$ |
| $u_1 + v_3$ | $= 16$ | $v_3 = 16$ |
| $u_1 + v_4$ | $= 12$ | $v_4 = 12$ |
| $u_2 + v_3$ | $= 11$ | $u_2 = -5$ |
| $u_2 + v_1$ | $= 7$ | $v_1 = 12$ |

*We are now in a position to evaluate the opportunity costs of the unallocated cells as follows:*

$u_1 + v_1 = 12$
$u_2 + v_2 = 14$
$u_2 + v_4 = 7$

| | $v_1 = 12$ | $v_2 = 19$ | $v_3 = 16$ | $v_4 = 12$ |
|---|---|---|---|---|
| $u_1 = 0$ | 12 / 28 | | | |
| $u_2 = -5$ | | 14 / 16 | | 7 / 19 |

*The opportunity costs are now:*
*Cell ($u_1$, $v_1$) = 28 – 12 = 16*
*Cell ($u_2$, $v_2$) = 16 – 14 = 2*
*Cell ($u_2$, $v_4$) = 19 – 7 = 12*

*All these costs are positive and hence the solution is optimal.*

## *SAQ 4*

*Initial Feasible Solution*

| | Warehouse1 | Warehouse2 | Warehouse3 | Supply |
|---|---|---|---|---|
| Plant A | 50 | 32 | 400 / 40 | 400 |
| Plant B | 200 / 16 | 30 | 20 | 200 |
| Plant C | 100 / 35 | 250 / 28 | 42 | 350 |
| Demand | 300 | 250 | 400 | 950 |

*Note that, as there are less than five allocations, this tableau is degenerate. To solve the opportunity cost equations, we need an extra allocation. This is achieved by entering a zero in an unoccupied cell. A low-cost cell should be chosen, but any unoccupied cell will work if occupying it allows us to compute all the u's and v's. Entering a zero in, for instance, the cell for Plant B to Warehouse 3 will allow their calculation. When this is done, the opportunity costs for the four other unoccupied cells turn out to be positive, and so the initial solution is the optimum solution.*

*Cost:*

| | |
|---|---|
| $16 \times 200$ | 3200 |
| $100 \times 35$ | 3500 |
| $250 \times 28$ | 7000 |
| $400 \times 40$ | 16000 |
| | |
| Total cost | 29700 |

## *SAQ 5*

*VAM For First Allocation*

| | $D_1$ | $D_2$ | $D_3$ | $D_4$ |
|---|---|---|---|---|
| $S_A$ | [7] | [6] | [2] 70 | [12] |
| $S_B$ | [3] 30 | [9] | [8] 10 | [7] |
| $S_C$ | [10] | [4] 60 | [11] 10 | [5] 30 |

*Occupied Cells Cost Equations*

| | | |
|---|---|---|
| $u_1 + v_3 = 2$ | *Let* $u_1 = 0$ | $v_3 = 2$ |
| $u_2 + v_1 = 3$ | | $v_1 = -3$ |
| $u_2 + v_3 = 8$ | | $u_2 = 6$ |
| $u_3 + v_3 = 11$ | | $u_3 = 9$ |
| $u_3 + v_2 = 4$ | | $v_2 = -5$ |
| $u_3 + v_4 = 5$ | | $v_4 = -4$ |

*Now calculate the u and v values to get:*

*$u_1 = 0$ so $v_3 = 2$ hence $u_2 = 6$ and $u_3 = 9$ hence $v_3 = 2$ and $v_4 = -4$ and $v_2 = -5$ and $v_1 = -3$.*

*Test for optimality means calculating for unoccupied cells the quantity $OC = c_{ij} - (u_i + v_j)$ which gives the encircled entries in the currently unused routes in the table below. Note that all are positive so using these routes would increase overall cost. Hence the current table is optimal.*

*The corresponding cost is 810:*

*This solution (given by the initial VAM table) is optimal. This solution (given by the initial VAM table) is optimal and unique (since no OC =0 in optimal table).*

| | $v_1 = -3$ | $v_2 = -5$ | $v_3 = 2$ | $v_4 = -4$ |
|---|---|---|---|---|
| $u_1 = 0$ | 10 [7] | 11 [6] | [2] | 16 [12] |
| $u_2 = 6$ | [3] | 8 [9] | [8] | 5 [7] |
| $u_3 = 9$ | 4 [10] | [4] | [11] | [5] |

*Minimum cost is 810.*

## *SAQ 6*

*(i) The Network Model*

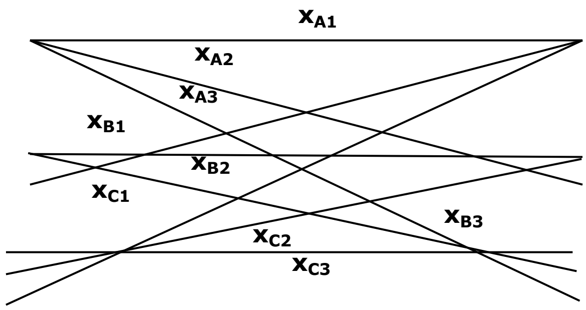
| *Plants (Sources)* | *Decision Variables (Shipments)* | *Construction Sites (Sinks)* |
|---|---|---|

*Supply* | *Demand*

240 (A) | (1) 300

160 (B) | (2) 140

200 (C) | (3) 120




*(ii) The Linear Programming Formulation*

*Let $x_{A1}$ represent the number of tons transported from plat A to Site 1, etc.*

*Minimise* $Z = 10x_{A1} + 7x_{A2} + 8x_{A3} + 17x_{B1} + 12x_{B2} + 14x_{B3} + 5x_{C1} + 11x_{C2} + 12x_{C3}$

*Subject to:* $x_{A1} + x_{A2} + x_{A3} \leq 240$

$x_{B1} + x_{B2} + x_{B3} \leq 160$

$x_{C1} + x_{C2} + x_{C3} \leq 200$

$x_{A1} + x_{B1} + x_{C1} = 300$

$x_{A2} + x_{B2} + x_{C2} = 140$

$x_{A3} + x_{B3} + x_{C3} = 120$

$x_{ij} \geq 0$

*(iii) The Optimal Solution Using the MODI*

*The Transportation Tableau*

*In this table, the problem has been balanced by adding a dummy destination, D whose fictitious demand of 40 balances supply and demand and zero costs assigned for each dummy route.*

*VAM for First Allocation*

| | $S_1$ | $S_2$ | $S_3$ | D | |
|---|---|---|---|---|---|
| $P_A$ | 100 (10) | 20 (7) | 120 (8) | (0) | 240 |
| $P_B$ | (17) | 120 (12) | (14) | 40 (0) | 160 |
| $P_C$ | 200 (5) | (11) | (12) | (0) | 200 |
| | 300 | 140 | 120 | 40 | 600 |

*No of Allocations*   *= R + C −1*
*6*   *= 3 + 4- 1 = 6: Not Degenerate*

*Occupied Cells Cost Equations*

| | | |
|---|---|---|
| $u_1 + v_1 = 10$ | Let $u_1 = 0$ | $v_1 = 10$ |
| $u_1 + v_2 = 7$ | | $v_2 = 7$ |
| $u_1 + v_3 = 8$ | | $v_3 = 8$ |
| $u_2 + v_2 = 12$ | | $u_2 = 5$ |
| $u_2 + v_4 = 0$ | | $v_4 = -5$ |
| $u_3 + v_1 = 5$ | | $u_3 = -5$ |

*Unoccupied Cell Cost Comparisons*

| | $v_1 = 10$ | $v_2 = 7$ | $v_3 = 8$ | $v = -5$ |
|---|---|---|---|---|
| $u_1 = 0$ | 10 | 7 | 8 | -5   0 |
| $u_2 = 5$ | 15   17 | 12 | 13   14 | 0 |
| $u_3 = -5$ | 5 | 2   11 | 3   12 | -10   0 |

- *Since all of the opportunity costs are positive, the solution is optimal*
- *Optimal cost is: 1000 + 140 + 960 +1440 + 0 + 1000 = €4540*
- *The excess supply of 40 tons remain at Plant B*

*Optimal transportation cost/ton = 4540/560 = €8.11*

## *SAQ 7*

| | Destination1 | Destination2 | Destination3 | Supply |
|---|---|---|---|---|
| Source1 | 9 | 10 | 6 | 100 |
| Source2 | M | 5 | 13 | 80 |
| Source3 | 5 | 4 | 10 | 90 |
| Demand | 105 | 80 | 85 | 270 |

| | Destination1 | Destination2 | Destination3 | Supply |
|---|---|---|---|---|
| Source1 | 15 (9) | (10) | 85 (6) | 100 |
| Source2 | (M) | 80 (5) | (13) | 80 |
| Source3 | 90 (5) | (4) | (10) | 90 |
| Demand | 105 | 80 | 85 | 270 |

*As R+C-1 = 5 and only four routes are used, the tableau is degenerate. Entering a zero in the cell for Source 3 to Destination 2 allows the calculation of all the u's and v's. When this is done, all the opportunity costs turn out to be positive, and hence the solution is optimal.*

*Total Cost: 1495*

## *SAQ 8*

Assignment Tableau with Row Reductions

| | ***Region*** | | | |
|---|---|---|---|---|
| ***Driver*** | ***North*** | ***South*** | ***East*** | ***West*** |
| ***Andy*** | *0* | *25* | *30* | *40* |
| ***Billy*** | *20* | *15* | *30* | *0* |
| ***Con*** | *15* | *5* | *0* | *50* |
| ***David*** | *20* | *20* | *10* | *0* |

Assignment Tableau with Row and Column Reductions

| | ***Region*** | | | |
|---|---|---|---|---|
| ***Driver*** | ***North*** | ***South*** | ***East*** | ***West*** |
| ***Andy*** | *0* | *20* | *30* | *40* |
| ***Billy*** | *20* | *10* | *30* | *0* |
| ***Con*** | *15* | *0* | *0* | *50* |
| ***David*** | *20* | *15* | *10* | *0* |

For an optimal solution the minimum number of lines = 4. Since this is not so we proceed. Thus we subtract 10 from all uncrossed cells and add 10 to those cells at the intersection of lines.

| | ***Region*** | | | |
|---|---|---|---|---|
| ***Driver*** | ***North*** | ***South*** | ***East*** | ***West*** |
| ***Andy*** | *0* | *20* | *30* | *50* |
| ***Billy*** | *10* | *0* | *20* | *0* |
| ***Con*** | *15* | *0* | *0* | *60* |
| ***David*** | *10* | *5* | *0* | *0* |

Since the minimum number of lines is 4 we have an optimal solution.

(i) The optimal assignment is therefore
Andy → North
Billy → West
Con → South
David → East

(ii) The minimum daily cost is therefore: 50 + 50 + 45 + 55 = €200

(iii) The solution is not unique

(iv) Andy→ North = 50
Billy→ South = 65
Con → East = 40
David → West = 45

Total Cost €200

# UNIT 5

## INTEGER LINEAR PROGRAMMING

### AIM

The aim of this unit is to give an understanding of the basic ideas behind Integer Linear Programming.

### OBJECTIVES

To formulate and solve special types of Linear Programming problems called Integer Linear Programming problems. At the end of this unit you will:

- Be able to recognise the types of situations where integer linear programming problem formulations are desirable.
- Know the difference between all-integer, mixed integer and {0,1} linear programming problems.
- Understand how to solve integer linear programmes by graphical method
- Be able to construct a graphical representation of an integer linear programme
- Be able to construct the solution space and to appreciate its significance
- Recognise the limitations of the graphical procedure
- Identify the types of problems that can be solved using integer programming techniques.

### REQUIRED READING

Students should read the following chapters of the mandatory text book David R. Anderson , Dennis J. Sweeney , Thomas A. Williams and Kipp  Martin An Introduction To Management Science: Quantitative Approaches To Decision Making, 13th International Edition, Cengage Learning, ISBN-13: 9780538475655

Chapter 7.1, 7.2 and 7.3

### INTRODUCTION

In this unit we will look at two types of problem in Integer Programming. The first one deals with the situation where some or all of the variables must take on integer values. The second one deals with the situation where the variables must be in the set {0,1}. Examples of integer programming would be if one required that only a complete article was manufactured during a time period. An example of {0.1} programming would be in an investment situation if one was to invest or not in a particular stock or share.

## SOLUTION OF INTEGER LINEAR PROGRAMMES

The programme is formulated in the usual way for example looking at the example in 8.2 of the text.

Max Z = 10x + 15y

Subject to:

$282x + 400y \leq 2000$
$4x + 40y \leq 140$
$x \leq 5$

$x,y \geq 0$ and Integer

This problem is first solved ignoring the restriction of integers to get the absolute maximum value of the objective function or the "Upper Bound". See figure 5.1 below.

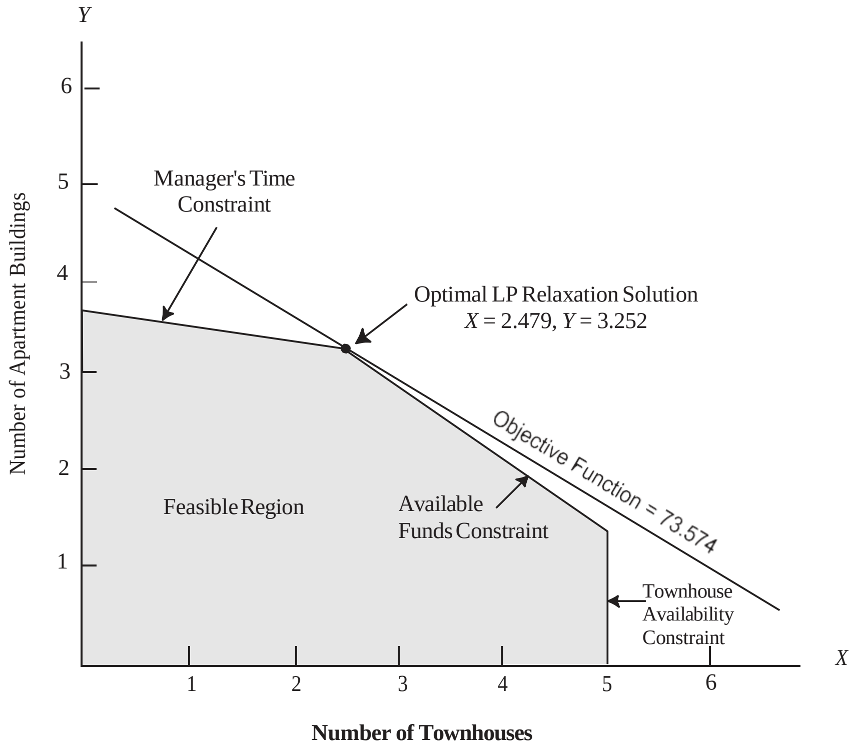



**Figure 5.1**

This gives a value of 73.574 for the Objective Function and x = 2.479 and y = 3.252. This violates the condition that both the variables should be integer. We now need to amrk all the integer values inside the feasible solution space and select the one that maximises the function. See figure 5.2 below.

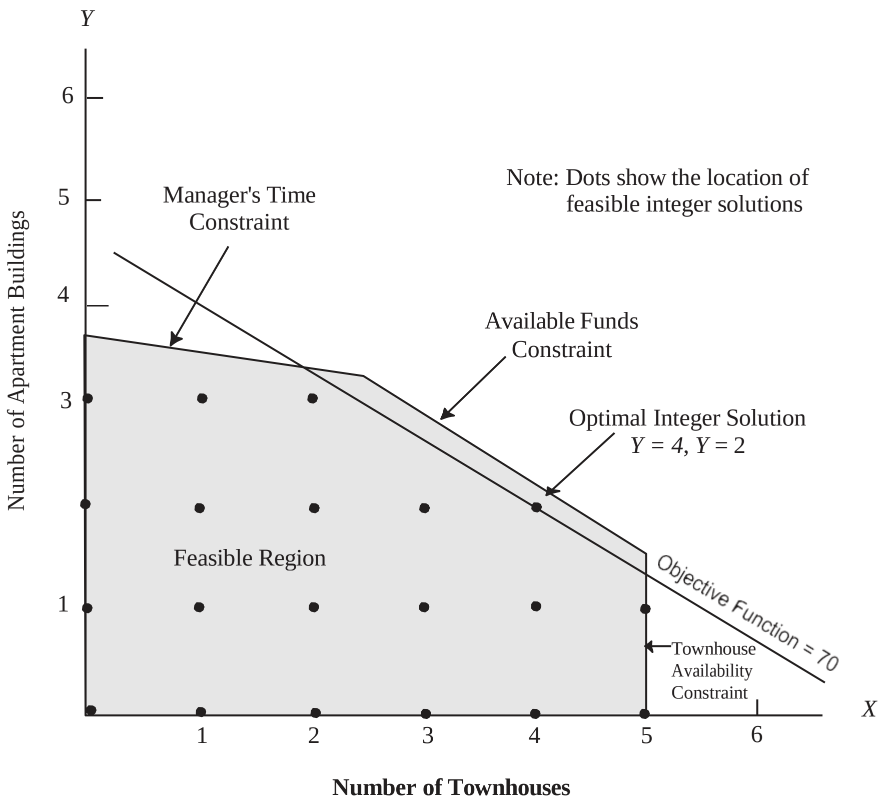



**Figure 5.2**

The gives a value of 70 for the Objective Function and  x = 4 and y -= 2. The integer restriction is now satisfied.

## *SAQ 1*

*Consider the following all-integer linear problem.*

$$Max\ Z = 5x_1 + 8x_2$$

*Subject to:*

$$6x_1 + 5x_2 \leq 30$$
$$9x_1 + 4x_2 \leq 36$$
$$x_1 + 2x_2 \leq 10$$

$x_1, x_2 \geq 0$ *and Integer*

(a) *Graph the constraints for this problem. Use dots to indictte all feasible inteer soslutions.*

(b) *Fid the optimal solution to the L.P. Relaxation. Round down to find a feasible integer solution.*

(c) *Find the optimal integer solution. Is it the same as the solution obtained in part (b) by rounding down?*

## ANSWERS TO SAQS

### *SAQ 1*

*$x_3 = 1$, $x_4 = 1$, $x_6 = 1$; Value = 17,500*

# UNIT 6

## GRAPH THEORY AND APPLICATIONS

### AIM
The aim of this unit is to show how certain classes of problem can be modelled using graph theory. These models are then solved using graph theory techniques.

### OBJECTIVES
To formulate and solve, using graph theory techniques a range of problems that include:

- The Chinese Postman
- The shortest route problem
- The minimal spanning-tree problem
- The maximal flow problem
- The travelling salesman problem
- The vehicle scheduling problem

### REQUIRED READING
Students should read the following chapters from the mandatory text book (David R. Anderson, Dennis J. Sweeney, Thomas A. Williams and Kipp Martin, *An Introduction to Management Science: Quantitative Approaches to Decision Making*, 13th International Edition, Cengage Learning, ISBN-13: 9780538475655):

Chapter 6.1, 6.2 and 6.3 and Chapter 19 (On CD)

### 6.1 TERMINOLOGY AND NOTATION
A graph model has two main components namely:

**A NODE:**
Usually denoted by a numbered circle

**BRANCHES:**
These connect pairs of nodes. The branches, represented by *arrows (→)* are assumed to represent a *flow* of some entity e.g. goods, oil, finances etc. Branches are also called *arcs* or *links*. The total entity of nodes and branches connected together is known as a *Network*.

An example of such a network is represented in Figure 6.1

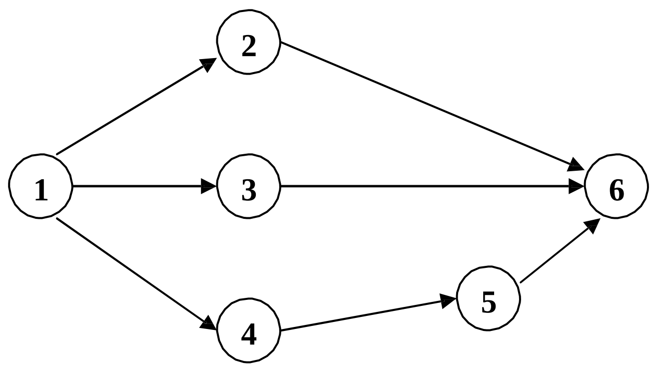
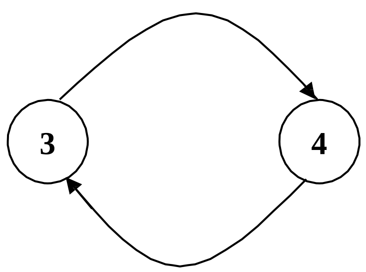



**Figure 6.1 A typical Network Model**

The lack of an arrow between nodes 2 and 3, between nodes 3 and 4 and between nodes 3 and 5 indicates that no flows exist between these nodes. Whereas *arrows* represent *flows* between nodes*, nodes* usually represent *locations* e.g. distribution centres, cities, towns, villages, airports etc.

The branch between nodes i and j is represented using (i,j). Thus in Fig 6.1 we have branches (1,2), (1,3), (1,4), (3,6), (4,5), (2,6) and (5,6).

It is important to note that branch (i, j) may not be the same as branch (j, i) for example:




could represent a one-way street system. For example, Dawson Street is a one-way street that joins St. Stephen’s Green and Nassau Street. We may represent this by the graph above by identifying the nodes in the following way.

Dawson Street, St. Stephen’s Green Junction (node 3)

Dawson Street, Nassau Street Junction (node 4)

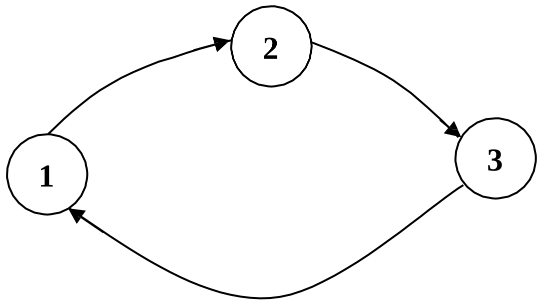
If the dimension of (3, 4) is 5 minutes and (4, 3) is 15 minutes then, in a shortest route (time) problem, traffic flow will always proceed in the correct direction, that is, from St. Stephen’s Green to Nassau Street down Dawson Street. Such a specified direction is known as a *directed branch*. All the branches shown in Figure 6.1 are directed branches.

A sequence of connecting branches between any two nodes i and j is known as a *chain* from i to j. Thus for instance (1,2) and (2,6) is a *chain* from 1 to 6 and (1,4) and (4,5) and (5,6) is another *chain* from 1 to 6.

Since both of these chains are *directed* they are also known as *paths* and thus for Figure 6.1 we have:

(1→2→6) is a path from 1 to 6
(1→3→6) is a path from 1 to 6
(1→4→5→6) is a path from 1 to 6

A *cycle* is a particular type of path where a node is connected to itself:




This structure will be examined later in ***Travelling Salesman*** and ***Vehicle Scheduling*** problems.

Finally, the origin of a network is referred to as the *source node*, whilst the destination is called the *sink node*.

We will now examine some frequently occurring network models of practical situations and indicate how graph theory is used in the approach to their solution.

## 6.2 The Shortest Route Problem

The shortest route problem is concerned with finding the shortest path in a connected network from origin to destination where the distance associated with each branch of the network is known. Such problems are most commonly encountered in transportation and scheduling. However similar situations can arise in communication and production applications.

There are three basic problems namely:

1. The shortest path between two particular nodes.
2. The shortest path from one particular node to all other nodes.
3. The shortest path between all pairs of nodes.

In terms of everyday activities, the three problems could be re-worded as:

1. Find the shortest route between Dublin and Cork.
2. Find the shortest route between Dublin and all major towns in Ireland.
3. Find the shortest route between all major towns in the country.

Computer programs now exist which routinely solve these problems. Indeed, in some countries when you hire a car you may specify the places you wish to visit. Many satellite navigation systems use a shortest-path algorithm to determine the shortest route to these destinations. These routes are determined using digitised road maps. Programs to determine the shortest routes from say, Paris to Bordeaux, Berlin to Munich, London to Skye and so on may be easily found on the Internet. Again a shortest-path algorithm is used to determine such routes given that a digitised road map for the countries concerned exists.

The Shortest-Path algorithm that is most widely used is known as ***Dijkstra's Algorithm*** and is a label setting algorithm where each node is given a permanent label. It is best explained by means of a simple example as depicted in Figure 6.2.

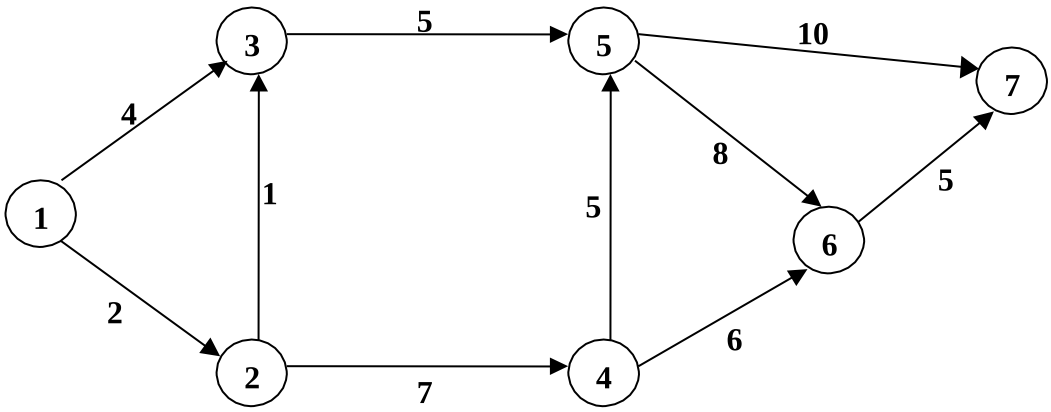



**Figure 6.2 A Shortest Route Problem**

This is a directed graph with distances in kilometres, and the problem is to find the shortest path from the source 1 to the sink 7.

Dijkstra's algorithm uses the concept of a *Permanent Set* of nodes and an *Adjacent Set* of nodes. At first only node 1, the origin, is placed in the permanent set. The nodes in the adjacent set are the nodes directly connected to the permanent set by connecting branches. Thus we have:

| *Permanent Set* | *Adjacent Set* |
|---|---|
| 1 | 2,3 |

The algorithm progresses through the network in an orderly fashion creating labels for each node. Each label is the shortest distance found so far from one node to another node. If the label is permanent, then it is the shortest distance. If the label is temporary, we are not yet sure. At each iteration of the algorithm one label is taken from the adjacent set (temporary label) and added to the permanent set. When no further labels can be added to the permanent set the shortest distance has been found.

The algorithm can be stated formally as follows.

Suppose that $\theta$ is the source node and S the sink node. Let L (i) represent the label for node *i*, let d(i,j) represent the length of the branch (i,j), and let p represent the last node made permanent.

*Step 1*

Set $L(\theta) = 0$, because the distance from the origin is zero, and let $L(i) = \infty$ (infinity) for all other nodes. Make the source node $\theta$ permanent, so that $p = \theta$.

*Step 2*

For each node in the adjacent set (and thus with a temporary label) that is directly connected to p:

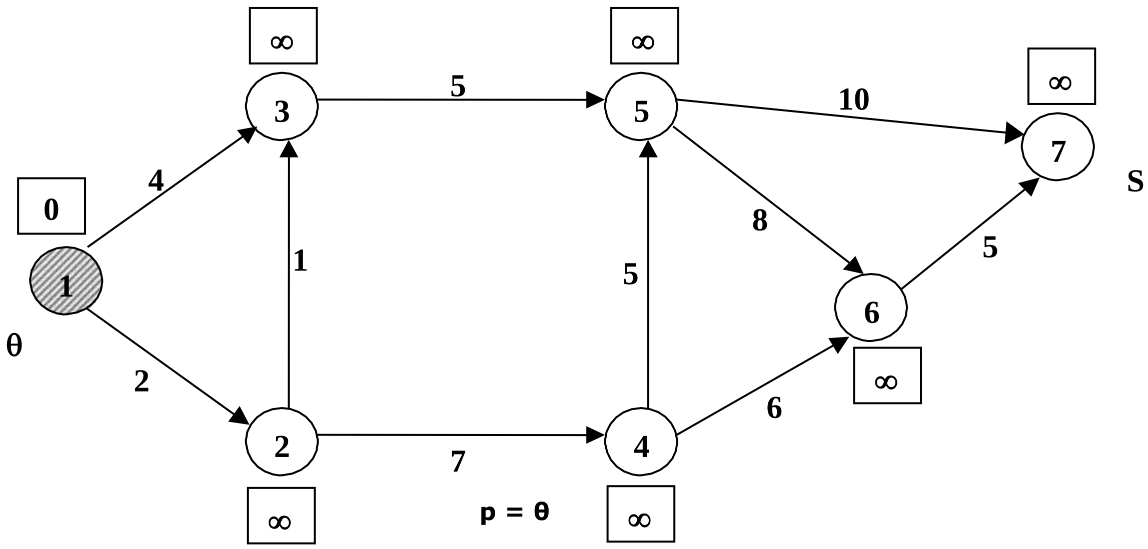
If $L(i) > L(p) + d(p,i)$, then set $L(i) = L(p) + d(p,i)$.

Then, from the adjacent set, select the node i with the smallest L(i). Make this node permanent, so that p=i.

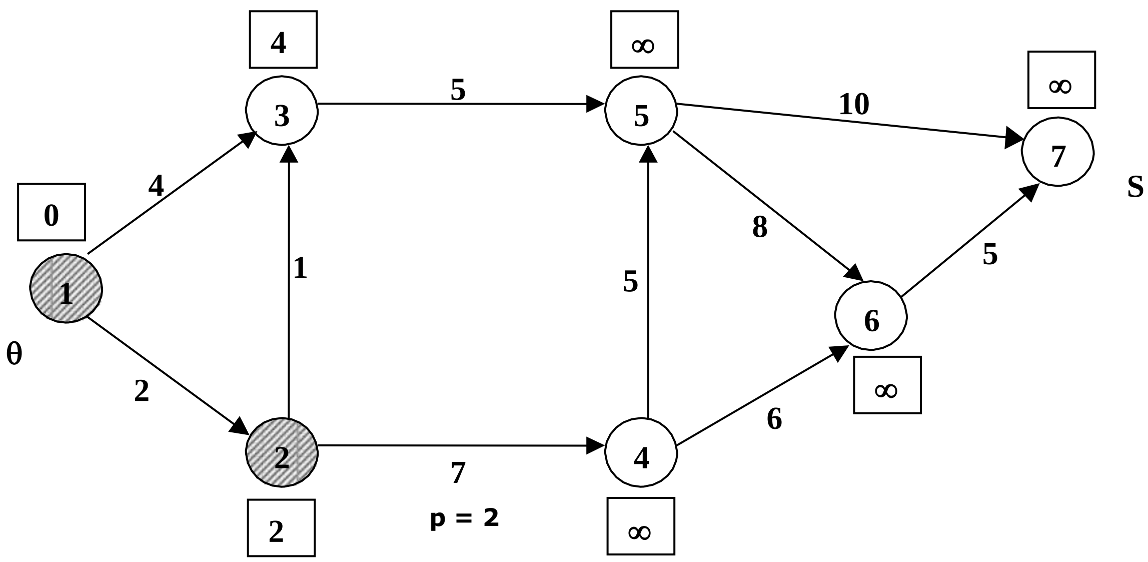
Step 3

If the sink S has been made permanent, then the problem is solved and L(S) is the shortest distance from $\theta$ to S. Otherwise, repeat Step 2.

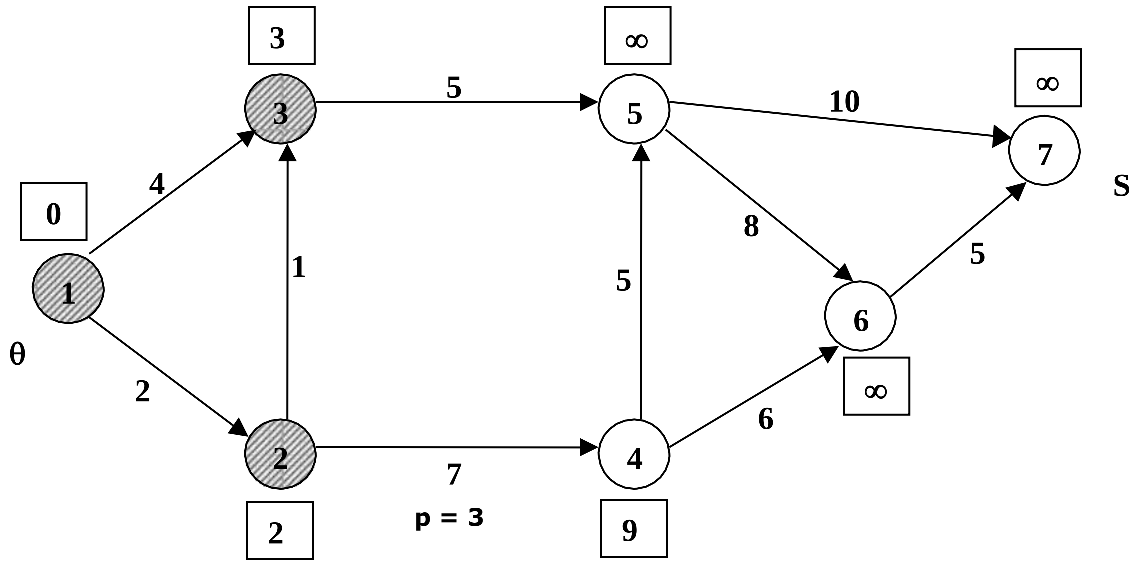
Applying Step 1 to our example (and using shading to represent the permanent node), we have:



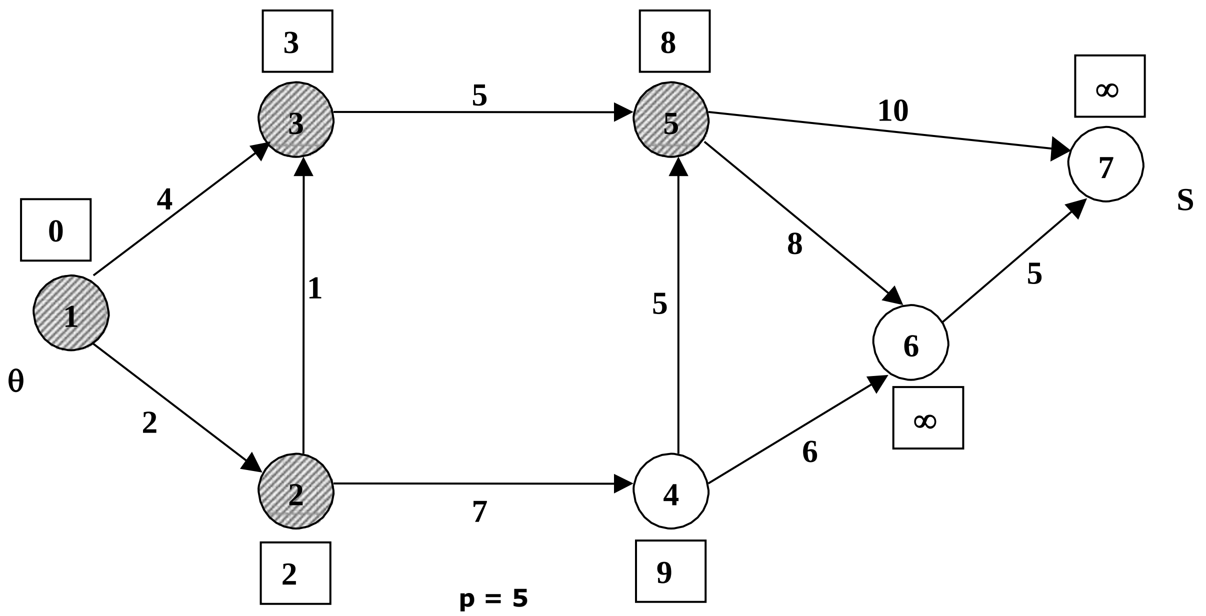

For Step 2, the nodes directly connected to node θ are nodes 2 and 3. Change the temporary labels for these nodes to L(3)=4 and L(2)=2. Since the node in the adjacent set with the smallest label is node 2, we make node 2 permanent:



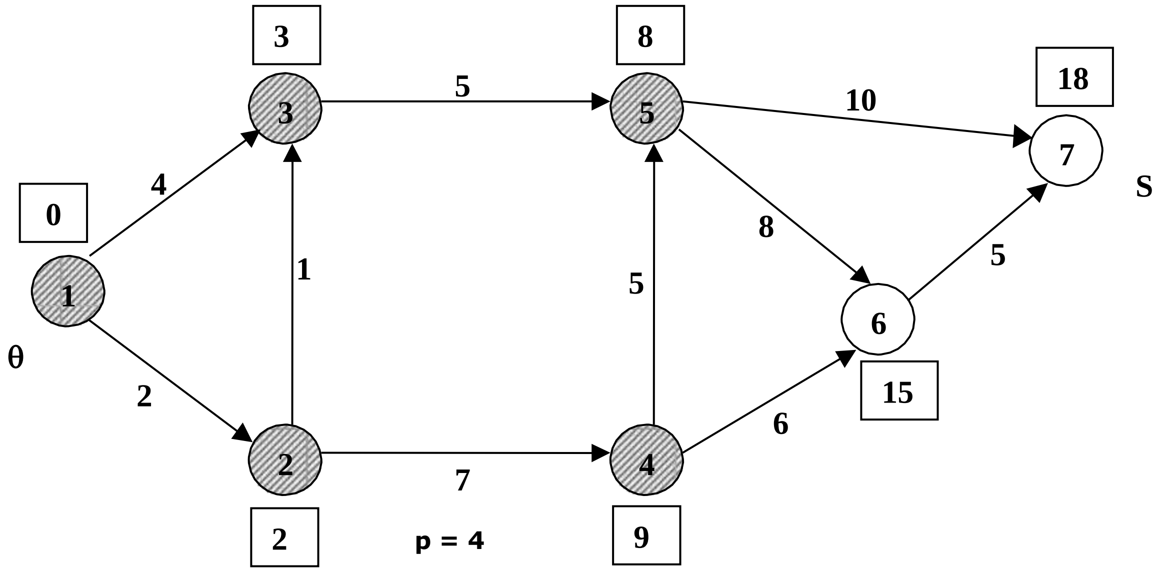
We now examine the nodes connected to node 2, the node we just made permanent. The label for node 3 is reduced from 4 to 3, and the label for node 4 becomes 9. Node 3 is made permanent.



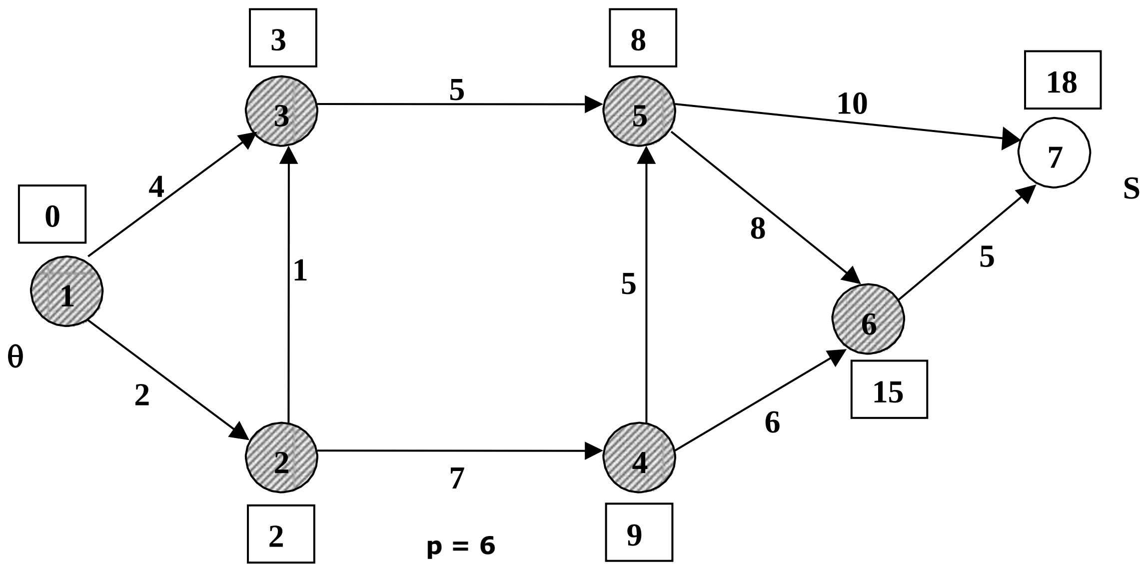

The label for node 5 becomes 8, and node 5 is made permanent.



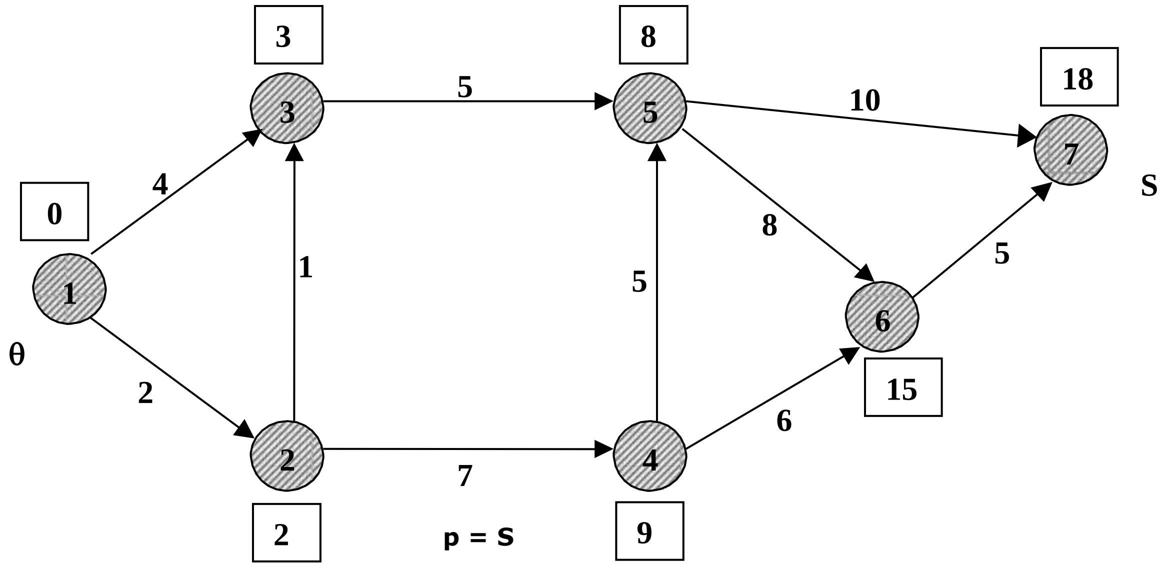
The label for node 7 becomes 18. Of the three nodes in the adjacent set, node 4 is made permanent.




No changes are required to the labels. Node 6 is made permanent.




The final node, node 7, is now made permanent:




Since node S has been made permanent the algorithm stops. The following should be noted:

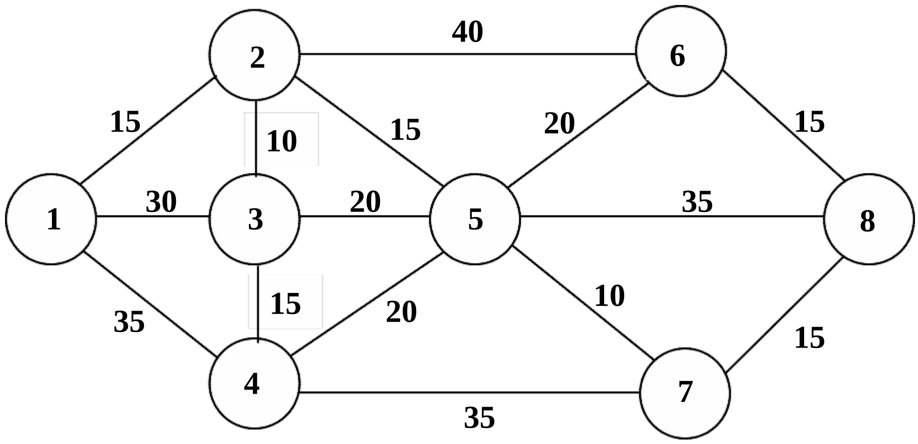
(i) The Optimal Path is 1→2→3→5→7 and the shortest distance is 18 kilometres

(ii) The shortest distances from the Origin (Node 1) to all other nodes are given in the labels. For instance, the shortest distance to node 6 is 15 kilometres, the shortest path being 1→2→4→6

▶ See the video **V27_Dijkstra_1**, which demonstrates the use of Dijkstra's Algorithm. https://goo.gl/QGbNbj

## *SAQ 1*

*Suppose you are the Transport Manager of a Distribution Company which operates in a region with 8 major towns. The road network joining these towns, and the distances between them in kilometres, is:*




*Use the shortest path algorithm to ascertain the shortest route from 1 to 8 and the shortest distances between 1 and all other towns.*

## 6.3 THE MINIMAL SPANNING TREE PROBLEM

Whereas in the shortest route problem the objective is to determine the minimum distance route (or cost) from origin to sink, the objective of the minimal spanning-tree problem is to *connect* all the nodes of the network so that the total length of all arcs is minimized for distance, cost or time. The shape of the solution resembles a tree whose branches connect (span) all the nodes which, for example, can represent towns, cities, computer terminals, shops etc.

There are many practical examples of the application of this algorithm such as: Cable TV connections to households; bus routes to scheduled stops; connecting computer terminals to servers or mainframe computers etc. The algorithm is known as a *Greedy Algorithm* since at each stage we can be greedy and take the action that looks most promising at that stage. Greedy algorithms are often excellent heuristics. (*Heuristics* involves the use of commonsense procedures to attempt to find a solution to a problem.)

There are two steps in the minimal spanning-tree algorithm namely:

**STEP 1:**

Begin at *any* node and connect it to the nearest node (in terms of distance, cost, time etc). Such nodes are termed *connected*, whereas the remainder are *unconnected*.

**STEP 2:**

Choose the nearest unconnected node to the connected nodes. Add this node to the connected set and repeat until nodes have been connected. If ties exist, then it is unclear which node to add to the connected set; this is resolved by use of any random procedure (e.g. tossing a coin) to break the tie. Figure 6.3 depicts a typical problem, which concerns a Cable TV network.

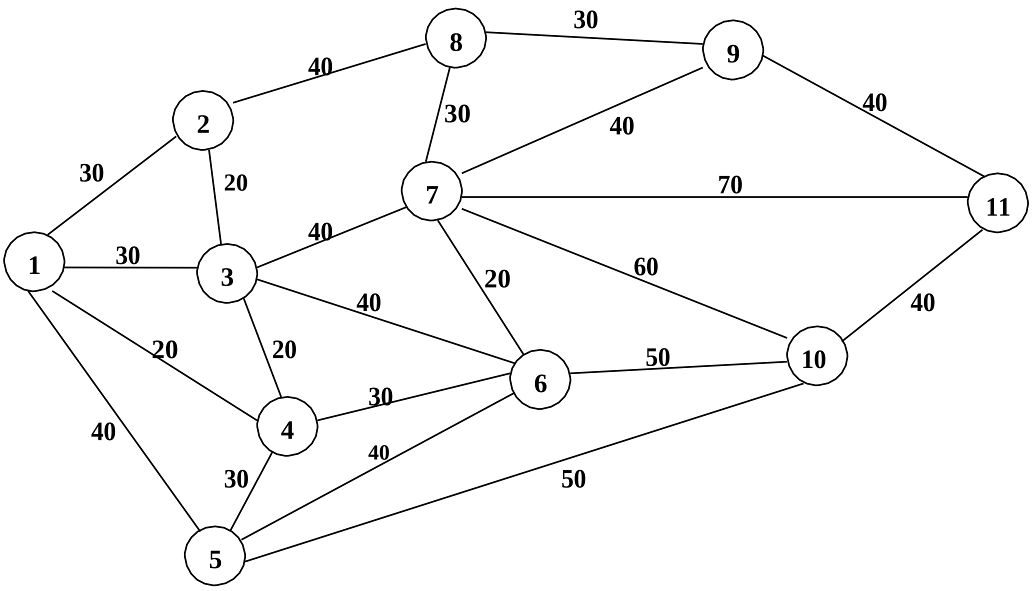



**Figure 6.3: A Cable Television Network on a Small Housing Estate**

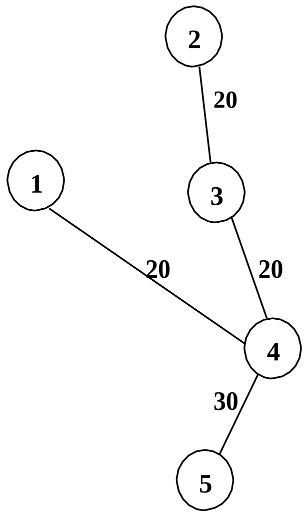
The nodes of the network show the houses that must be reached by the cable television company's transmission lines. The arcs of the network show the number of metres between the houses to be connected. The company is interested in knowing how to connect all the houses using the minimum length of cable.

The initial steps of the calculation are as follows:

Step 1
- ❑ Connect nodes 1 and 4: Distance 20 m

Step 2
- ❑ Connect nodes 4 and 3: Distance 20 m
- ❑ Connect nodes 3 and 2: Distance 20 m
- ❑ Connect nodes 4 and 5: Distance 30 m

*Position After Four Connections*




Complete the connections and confirm that the minimal spanning-tree is as is shown in Figure 6.4.

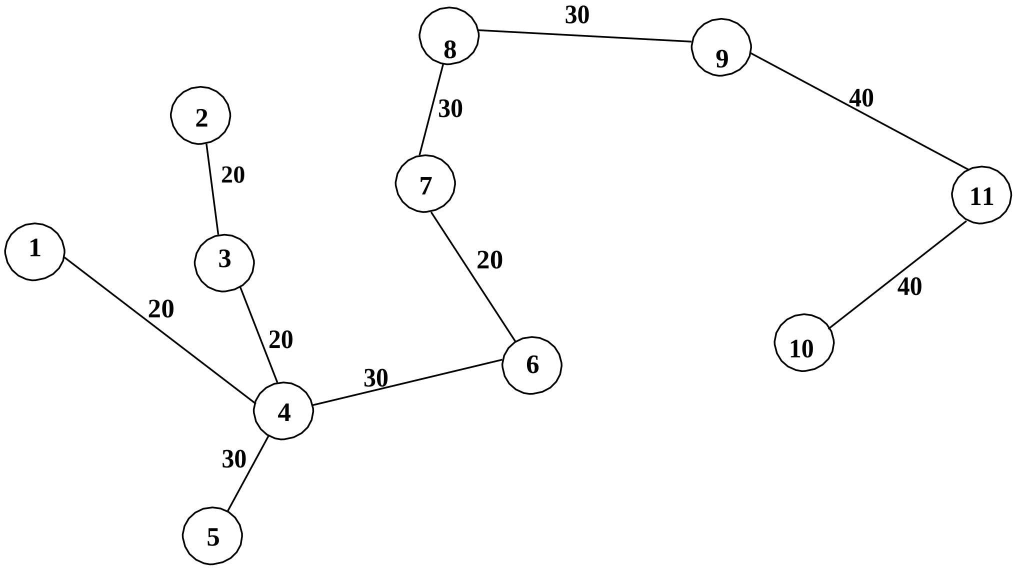



**Figure 6.4: The Minimal Spanning Tree for the Cable Television Company**

Total length of cable required is 280 metres.

▶ See the video **V28_Spanning_1**, which demonstrates the use of the Minimal Spanning Tree algorithm. [https://goo.gl/040y0F](https://goo.gl/040y0F)

## *SAQ 2*

*Suppose the network in SAQ 1 describes a small housing development. The distances given are measured in metres. An underground cable is to be laid to connect the houses (nodes) for cable television. Determine the minimum length of cable required for the project.*

## 6.4 The Maximal Flow Problem

The Maximal flow problem is concerned with routing flows through a network in order to maximize the total flow (vehicles, data, oil, etc) from source to sink. There are many examples of this situation in everyday life; these include:

- The flow of water from a reservoir through a network of pipelines to users, both corporate and private.
- The flow of data through a communications network.

As in our already described problems, a network is used where the nodes might represent pumping stations, road intersections or switches depending on the application. And in similar manner the arcs might represent pipelines, roads or cables. In each case the objective would be to determine the maximum flow that can be achieved from source to sink where each arc, or branch, has a specified capacity restriction in one direction (or perhaps both directions) depending on the application.

An essential assumption of the maximal flow problem is the concept of *Conservation of Flow,* which states that the total flow into any node must be equal to the total flow out of the node. This concept is illustrated in Figure 6.5.

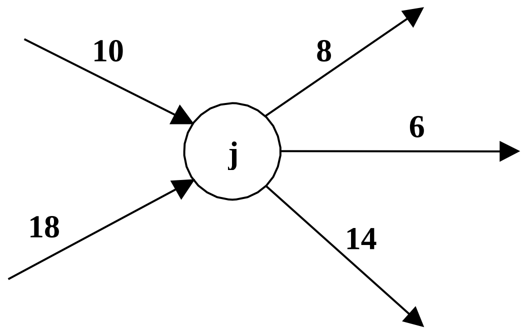



**Figure 6.5: Conservation of Flow**

Total Flow into Node j = 10 + 18 = 28
Total Flow Out Of Node j = 8 + 6 + 14 = 28

Let us now consider an oil company that has a pipeline across a country from its loading port (source) to its refinery (sink). There are three pumping stations in the network and the arcs (pipes) have differing capacities. Figure 6.6 illustrates the situation.

Note:

(i) The numbers in brackets on each arc denote the lower and upper capacities of the pipelines and are expressed in thousands of barrels per hour
(ii) Flows between nodes 3 and 4 can go in either direction with different capacities.
(iii) All other flows are uni-directional.

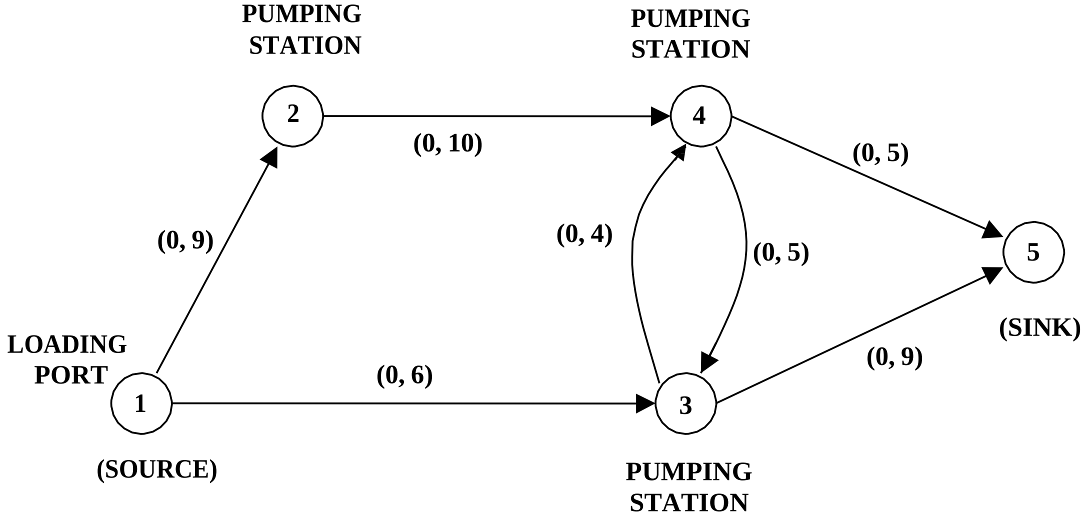



**Figure 6.6: An Oil Pipeline**

In order to compute the maximum flow in this network we use a linear programming model. Define:

$x_{ij}$ = the flow through the arc (i →j)

$d_{ij}$ = the capacity limit of each arc

This leads to the constraints needed to enforce the capacity restrictions (here it is assumed that minimum capacities are zero):

$0 \leq x_{ij} \leq d_{ij}$

In addition, we have a Conservation of Flow consideration at each intermediate node: the sum of the flows into the node must equal the sum of the flows out of the node. In our oil pipeline example, this leads to the three constraints:

| ***Node*** | ***Equation*** |
|---|---|
| 2 | $x_{12} = x_{24}$ (or $x_{12} - x_{24} = 0$) |
| 3 | $x_{13} + x_{43} = x_{34} + x_{35}$ (or $x_{13} + x_{43} - x_{34} - x_{35} = 0$) |
| 4 | $x_{24} + x_{34} = x_{43} + x_{45}$ (or $x_{24} + x_{34} - x_{43} - x_{45} = 0$) |

The *objective* is to maximize the flow into the refinery or sink, thus:

MAX $F = x_{45} + x_{35}$

Alternatively, we could maximise $x_{12} + x_{13}$, the flow from the loading port or source. This would lead to exactly the same solution. Thus the complete model can be stated as follows:

MAX $F = x_{45} + x_{35}$

Subject To:
$x_{12} - x_{24} = 0$
$x_{13} + x_{43} - x_{34} - x_{35} = 0$
$x_{24} + x_{34} - x_{43} - x_{45} = 0$
$x_{12} \leq 9$
$x_{13} \leq 6$
$x_{24} \leq 10$
$x_{34} \leq 4$
$x_{43} \leq 5$
$x_{45} \leq 5$
$x_{35} \leq 9$
$x_{ij} \geq 0$ all i, j

This model can be solved by the simplex algorithm or by using appropriate software or a spreadsheet.

▶ See the video **V29_Maxflow_1**, which derives the linear programming formulation for a maximal flow problem. https://goo.gl/i1eLYe

## 6.5 THE CHINESE POSTMAN PROBLEM

This problem is so-called because most of the original analysis of the problem was done by the Chinese mathematician Mei-ko-Kwan.

This problem is concerned with minimising the total distance that must be travelled so that every arc in an undirected graph must be crossed at least once. This problem has relevance for garbage collection, postal deliveries, road sweeping, and door-to-door milk deliveries.

In the following examples, where the arcs of the graphs correspond to the network of roads, a postman wishes to travel along each road in such a way as to minimise his total travel distance. In solving this problem, the postman must obviously travel each of the roads in this route at least once but he clearly wishes to avoid covering too many roads more than once.

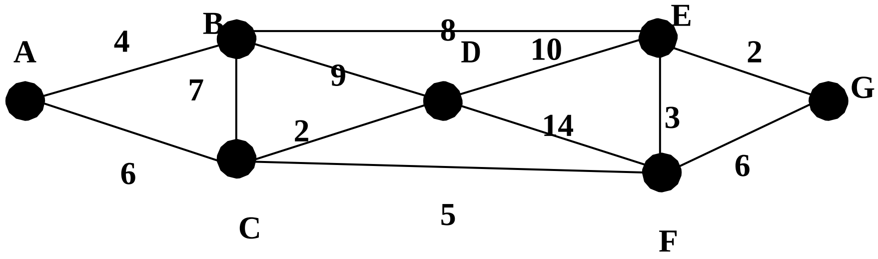



**Figure 6.7 A Chinese Postman Problem**

In the Figure 6.7 above, the postman wishes to start and finish his route at node A on the graph. Since all arcs must be travelled and all roads contain deliveries, he must travel at least the sum of the arc lengths. In the figure above, the arc lengths sum to 76. Therefore, the minimum distance he must travel is 76. The question to be answered is, can he start at A, travel every arc and return to A without re-crossing an arc? For example, in Figure 6.7 above the answer is yes.

A possible solution is
A → B → E → D → C → F → G → E → F → D → B → C → A
A distance of 76.

Now consider the network of roads as represented in the figure below

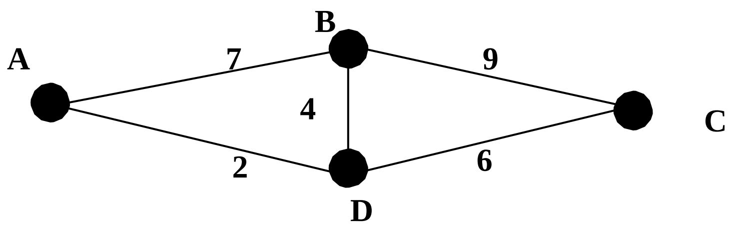



**Figure 6.8 A Road Network**

A postman, now, cannot start and finish at A and cover all arcs exactly once. Thus, the minimum total distance he must travel is the sum of the arc lengths, 28, plus the additional distance arising from re-crossing some arcs.

In this example, it can be shown that arc BD must be crossed twice and so the minimum total distance is 28 + 4= 32.

In Figure 6.7, a tour can be found that crossed each arc exactly once, such a tour could not be found in Figure 6.8. Why?

The answer to this question, believed to be the first question answered in graph theory, was supplied by Euler in 1736. Euler showed that a tour, through all arcs, and covering all arcs exactly once, will exist in a graph if the degree of all nodes in the graph is even. The **degree of a node** is defined as the number of arcs meeting at the node.

In the Figure 6.7, the nodes of the graph have the following degrees:

| Nodes | Degree |
|---|---|
| A | 2 |
| B | 4 |
| C | 4 |
| D | 4 |
| E | 4 |
| F | 4 |
| G | 2 |

Since all the above degrees are even, the required tour must exist in the graph.

In contrast, the degrees of nodes B and D in Figure 6.8 are odd and therefore no tour will exist in the graph, and some arcs must be re-crossed.

The proof of Euler's theorem is beyond the scope of this course.

**Example**
Find the minimum distance required to travel all arcs in the graph shown in Figures 6.4 at least once.

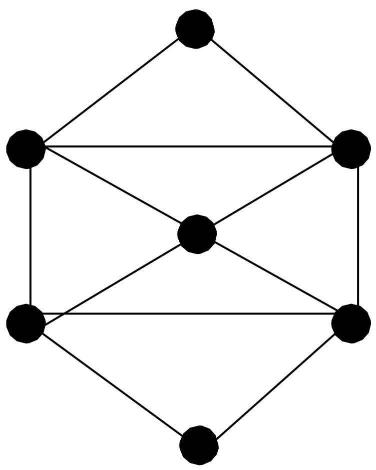


**Figure 6.9 An Example Network**

By inspection, it can be seen that the degree of every node in the graph in Figure 6.9 is even, and thus a tour can be found which will travel all arcs exactly once.

Thus the Minimum Total Tour Length = $\sum$ arc length = 89.

In the event of the graph containing nodes of **odd degree**, then the following **algorithm** is used to find the tour minimum length.

Step 1: Identify all nodes of odd degree in the graph.

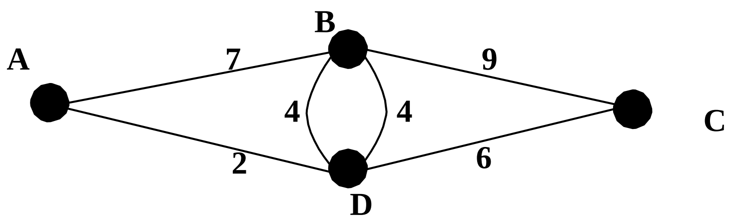
Step 2: Find the minimum cost way of connecting into pairs all nodes of odd degree.

Step 3: Add the arcs identified in Step 2 to the graph.

Step 4: Calculate the Minimum Tour Length as the sum of the original arcs in the graph plus the arcs added in Step 3.

If this algorithm is applied to Figure 6.8, then the workings are as follows:

Step 1: Nodes of odd degree: B and D
Step 2: The minimum cost connection of B and D is through arc BD, a distance of 4 units.
Step 3: Add arc BD to the graph. The revised graph is now as shown below.



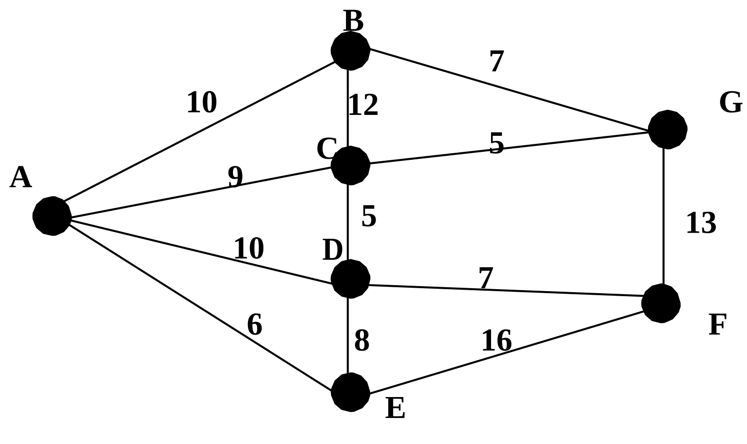

Since all nodes in the revised graph are even, then the required tour exists.

Step 4: Calculate the Minimum Tour Distance as:
$\Sigma$ arc lengths + arc added = 28 + 4 = 32.

▶ See the video **V30_Chinese_1**, which demonstrates the use of the Chinese Postman method.
https://goo.gl/ZpeOEK

## *SAQ 3*

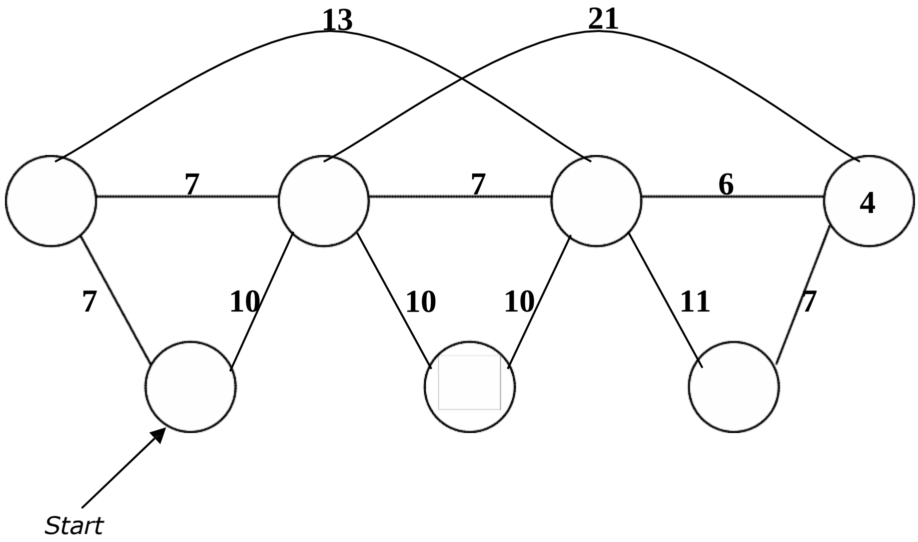
*The roads in a housing estate can be represented in a graph as shown below.*




*You have been requested by Dublin City Council to design a tour of minimum total length for a garbage trunk that must service every road in the estate.*

## SAQ 4

*Find an optimal postman tour in the network shown below*




## 6.6 THE TRAVELLING SALESMAN PROBLEM

This problem is so-called because it simulates the actions of a salesperson who, every day, wishes to visit a number of customers and then return to the point from which they started whilst covering the least possible distance. The Travelling Salesman Problem (TSP) is not concerned with arcs but seeks to find a *tour* of minimum length that passes through each node on the graph exactly once. Figure 6.10 depicts the situation.

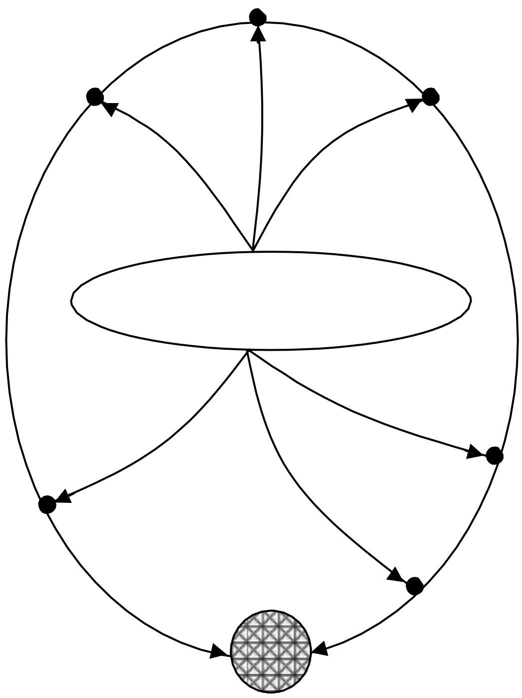



**Figure 6.10: The Travelling Salesman Problem**

The TSP is fundamental to all routing problems in that every time a delivery truck leaves a depot to call on a given list of customers, then the order in which the truck calls to the customers (in order to minimize distance travelled) is a solution to the travelling salesman problem. The number of ways in which these tours can be constructed is large. If there are n calls to be made, counting the starting node as one of the nodes on the tour, the number of tours is:

$(n - 1)!$

For instance, if n = 10 (not a large number) the possible number of tours is:

$(10 - 1)! = (9)! = 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 362{,}880.$

It is therefore obvious that the TSP is a "hard" problem to solve. To date, solution procedures are divided into two camps namely:

1. *Exact*
   - Complete enumeration
   - {0 ,1} Programming
   - Branch and Bound

2. *Heuristics*
   - Nearest Neighbour
   - Nearest Insertion
   - 2 - OPT

### ***Exact Methods***

Exact methods can only solve relatively small problems, whilst the *heuristics* yield approximate solutions which may not be near-optimal.

The {0,1} Programming method is based on linear programming, wherein each $x_{ij}$ can take the value 1 (if the branch is used) or 0 (if it is not). Apart from the fact that this method is unsuitable for large networks, another problem is that *sub-tours* can arise in the linear programming solution. A sub-tour exists among k nodes if the number of arcs connecting those nodes is equal to k.

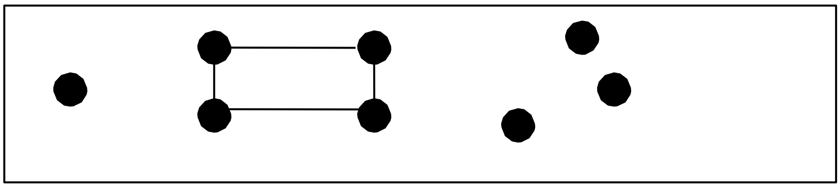


**Figure 6.11:**
**Sub-Tours**

Mathematically, the sub-tour can be eliminated for the k nodes involved by introducing the following constraint and solving the formulation again.

$$\sum_{ij \in k} x_{ij} \leq k - 1$$

The complete procedure for the exact solution of the TSP is as follows:

*Step 1*

Solve

Min: $\sum_{i \neq j} c_{ij} x_{ij}$

Subject to:

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \text{all } i$$

$$\sum_{i=1}^{n} x_{ji} = 1 \qquad \text{all } j$$

$x_{ij} = 0$ or 1

*Step 2*

If there are no sub-tours in the solution found in Step 1, the optimal solution has been found. If there are sub-tours, introduce elimination constraints as required.

*Step 3*

Solve new problem.
Return to step 2

Let us look at a four city TSP Problem as shown in Figure 6.12

**Distance Matrix (Kilometres)**

| | ***A*** | ***B*** | ***C*** | ***D*** |
|---|---|---|---|---|
| ***A*** | — | **4** | **6** | **9** |
| ***B*** | **4** | — | **9** | **3** |
| ***C*** | **6** | **9** | — | **7** |
| ***D*** | **9** | **3** | **7** | — |

**Figure 6.12:**
**A Four City TSP Problem**

The problem is to find the minimum length tour that starts and ends at city A and passes through every other city exactly once. The complete formulation of this problem is:

*Step 1*

Let $x_{AB} = 1$ if the branch (A, B) is used and zero otherwise, and let T represent the total length of the route.

Min: $T = 4\,x_{AB} + 6x_{AC} + 9\,x_{AD} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots + 7\,x_{DC}$

Subject to:

$$
\begin{array}{l}
x_{AB} + x_{AC} + x_{AD} = 1 \\
\quad\vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
x_{DA} + x_{DB} + x_{DC} = 1
\end{array}
$$

$$
\begin{array}{l}
x_{BA} + x_{CA} + x_{DA} = 1 \\
\quad\vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
x_{AD} + x_{BD} + x_{CD} = 1
\end{array}
$$

$x_{ij} = \{0, 1\}$

The optimal solution to this integer linear programming problem is:

$x_{AC} = 1$; $x_{BD} = 1$ ; $x_{CA} = 1$ ; $x_{DB} = 1$

and

Min T = 18 kilometres.

However, this does not solve the TSP problem because the solution uses two sub-tours:

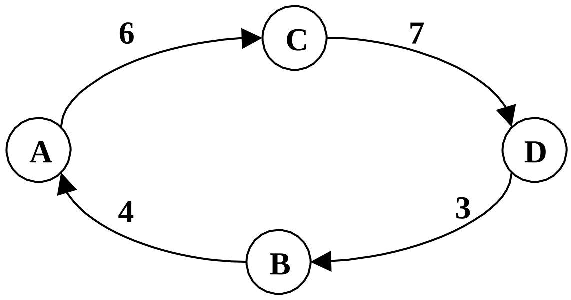
*Step 2*
Since sub-tours exist we add the following constraints:

$x_{AC} + x_{CA} \leq 1$
$x_{BD} + x_{DB} \leq 1$

and then we solve the linear programming problem again.

*Step 3*
On re-solving we get:
$x_{AC} = 1$; $x_{BA} = 1$ ; $x_{CD} = 1$ ; $x_{DB} = 1$

We now have a valid tour. The solution is:




The tour length is T = 6 + 7 + 3 + 4 = 20 Kilometres.

## *SAQ 5*

*Find the minimum length tour that starts and ends at city 1 and passes through every other city exactly once. The distance matrix is:*

| | **Cities** | | | | | |
|---|---|---|---|---|---|---|
| **Cities** | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | **~** | **13** | **12** | **18** | **7** | **14** |
| **2** | **13** | **~** | **21** | **26** | **15** | **25** |
| **3** | **12** | **21** | **~** | **11** | **6** | **4** |
| **4** | **18** | **26** | **11** | **~** | **12** | **14** |
| **5** | **7** | **15** | **6** | **12** | **~** | **9** |
| **6** | **14** | **25** | **4** | **14** | **9** | **~** |

### *HEURISTIC ALGORITHMS FOR THE TSP*

The Travelling Salesman Problem is very difficult to solve optimally, and, in any event, only fairly small problems can be analysed by exact methods. The alternative approach is to use heuristic methods, which often give answers reasonably close to optimality. Heuristics for the TSP can be classified into two genres:

(i) Tour construction heuristics which generate a feasible tour.

and

(ii) Tour improvement heuristics which begin with a feasible tour and attempts to make improvements to it.

There are quite a number of these heuristics, and we will consider two: the Nearest Neighbour algorithm and the Nearest Insertion algorithm.

The **Nearest Neighbour algorithm** is very simple and consists of the following four steps:

1. Select a starting node.
2. Find the unvisited node that is nearest to the last node selected and add it to the tour.
3. Repeat Step 2 until all nodes have been visited.
4. Complete the tour by adding the starting node to the end of the tour and calculate the total distance.

The Nearest Insertion heuristic is only slightly more complicated. To demonstrate it, let us return to the distance matrix exhibited in Figure 6.12, namely:

**Distance Matrix (Miles)**

| | ***A*** | ***B*** | ***C*** | ***D*** |
|---|---|---|---|---|
| ***A*** | — | **4** | **6** | **9** |
| ***B*** | **4** | — | **9** | **3** |
| ***C*** | **6** | **9** | — | **7** |
| ***D*** | **9** | **3** | **7** | — |

The rules of the **Nearest Insertion heuristic** are:

- Select any node i to start
- Choose the nearest node j and form the sub-tour {i-j-i}
- At each insertion find a node s not in the sub-tour that is closest to any node in the sub-tour.
- Find the arc (x,y) in the sub-tour which minimises d(x, s) + d(s, y) - d(x, y)
- Insert node s between x and y.
- Repeat this process until a valid tour is constructed.

Applying these rules, we have:

- Select node A
- Closest node is B, sub-tour A – B – A
- The nearest node is D where d (B, D) = 3
- Add D to the sub tour in one of two ways namely A-B-D-A or A-D-B-A; both distances = 16 kilometres.
- Arbitrarily choose A-D-B-A
- C is the only remaining node so add C as efficiently as possible

Namely:
A – C – D – B – A →20 kilometres
A – D – C – B – A→ 29 kilometres
A – D – B – C – A→27 kilometres

- Thus the Nearest Insertion solution is 20 kilometres, which is the same as the optimal one we found earlier using an exact method. In larger, more realistic, problems this equality of solutions is unlikely to exist.

## *SAQ 6*

*Use the nearest neighbour heuristic to generate a travelling salesman solution through 7 towns whose distance matrix is:*

**Towns (Distance in Miles)**

| **Towns** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|---|
| **1** | ~ | 13 | 14 | 17 | 9 | 10 | 12 |
| **2** | 13 | ~ | 12 | 8 | 10 | 7 | 17 |
| **3** | 14 | 12 | ~ | 14 | 17 | 12 | 11 |
| **4** | 17 | 8 | 14 | ~ | 13 | 8 | 12 |
| **5** | 9 | 10 | 17 | 13 | ~ | 10 | 14 |
| **6** | 10 | 7 | 12 | 8 | 10 | ~ | 17 |
| **7** | 12 | 17 | 11 | 12 | 14 | 17 | ~ |

## *SAQ 7*

In a fudge-making factory, several flavours of fudge are mixed in sequence on one machine. After each flavour, the machine must be cleaned in readiness for the next flavour. The table gives the time (in minutes) required to clean the machine between each pair of flavours. Find the sequence that minimises the total time spent cleaning the machine.

**Next Product**

| | **Vanilla** | **Strawberry** | **Coffee** | **Chocolate** | **Devon Cream** | **Mint** |
|---|---|---|---|---|---|---|
| **Vanilla** | ~ | 100 | 110 | 110 | 60 | 100 |
| **Strawberry** | 150 | ~ | 170 | 150 | 150 | 200 |
| **Coffee** | 140 | 170 | ~ | 100 | 150 | 200 |
| **Chocolate** | 140 | 160 | 90 | ~ | 100 | 110 |
| **Devon Cream** | 50 | 90 | 80 | 90 | ~ | 110 |
| **Mint** | 170 | 130 | 150 | 150 | 150 | ~ |

## 6.7 THE VEHICLE SCHEDULING PROBLEM

The *Vehicle Scheduling Problem* (VSP) can be explained by referring to figure 6.13.

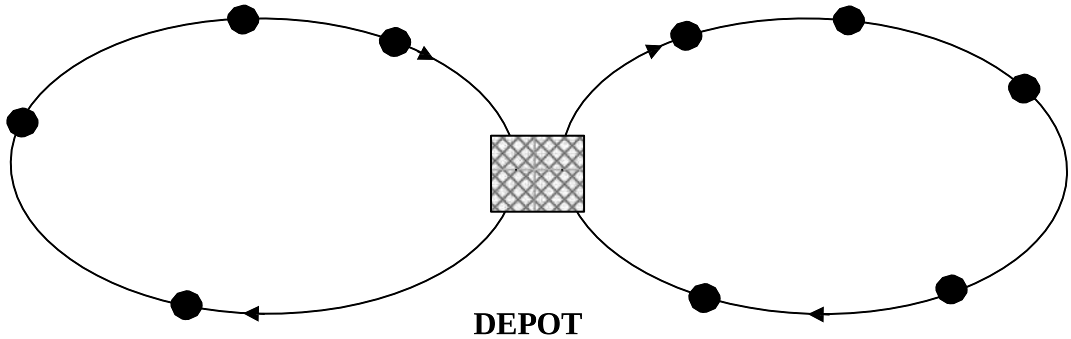



**Figure 6.13:**
**The Vehicle Scheduling Problem**

There are nine customers. The task is to service the delivery or collection requirements (or both) of these customers in such a way that the distances the vehicles have to travel is minimized. Applications include:

- Collecting milk from farmers and delivery to creameries
- Collecting letters and parcels from post offices for delivery to sorting centres
- Delivering petrol/diesel to filling stations
- Delivering goods to supermarkets etc, etc.

There are also capacity restrictions on the vehicles, in terms of gallons of milk, weight of goods, and so on. In the absence of capacity restrictions, the VSP reduces to the TSP.

The basic ingredients of the VSP are:

- A set of customers with requirements for collection and/or delivery.
- A set of vehicles, which do not necessarily have the same capacities
- A depot out of which all vehicles begin their journey and to which they return.
- A distance matrix containing the distances between all customers and the distances from the depot to each customer.
- Minimizing the total distance travelled by the vehicle fleet

The VSP is not an easy problem to solve. As with the TSP, exact methods such as {0, 1} programming exist, but they are computationally inefficient and can be used only for relatively small networks. Heuristic methods, therefore, must be used to get acceptable solutions (usually non-optimal) in a reasonable computing time.

Heuristics generally fall into two categories namely *sophisticated* or *unsophisticated*. The former are based on abstract mathematical techniques such as *Lagrangian Relaxation* and are beyond the scope of this module. The latter include:

- The Savings Heuristic
- The Sweep Heuristic
- Simulated Annealing

and other approaches. We will look at two of these - *The Savings Heuristic* and *the Sweep Heuristic*

### THE SAVINGS HEURISTIC

The *Savings Heuristic* was developed by Clarke and Wright in 1963 and subsequently computerized by IBM and launched under the software title *Vehicle Scheduling Problem X* or VSPX. This software product was discontinued by IBM some years ago, but it is instructive to note the simplicity of the Clarke /Wright heuristic. Figure 6.14 displays the essential idea.

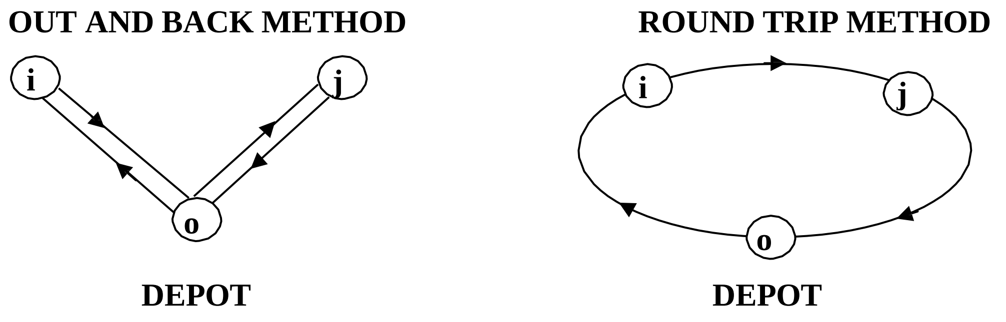



**Figure 6.14:**
**The Savings Heuristic**

In the case of just two customers, the simple algebra is as follows. As usual we let d(i,j) represents the distance from node i to node j. The depot is located at node o.

*Out and Back:*
Total Distance Travelled $D_1 = 2\,d(o, i) + 2d(o, j)$

*Round Trip:*
Total Distance Travelled: $D_2 = d(o, i) + d(o, j) + d(i, j)$

The Savings Realized:

$S(i, j) = D_1 - D_2$

Or

$S(i, j) = d(o, i) + d(o, j) - d(i, j)$

If S(i, j) is positive, it is worthwhile to join (i) and (j) on a route. With the use of slightly more difficult algebra it is possible to extend the idea to 3,4,5,6 etc customers.

The steps in the *Savings Heuristic* are therefore as follows:

**STEP 1** Compute the savings S(i ,j) for all pairs of customers

**STEP 2** Rank the savings in descending order of magnitude.

**STEP 3** Select the highest ranked savings and determine if it is feasible (within, for instance, capacity limitations) to join then together. If so, join them together. If not, go to the next highest saving.

**STEP 4** Continue with step 3 until all positive savings are exhausted.

The following example should explain the workings of the heuristic.

Acme Brewery distributor has received orders for the next day's delivery. The distance matrix and the number of cases required by each customer is shown in Figure 6.15.

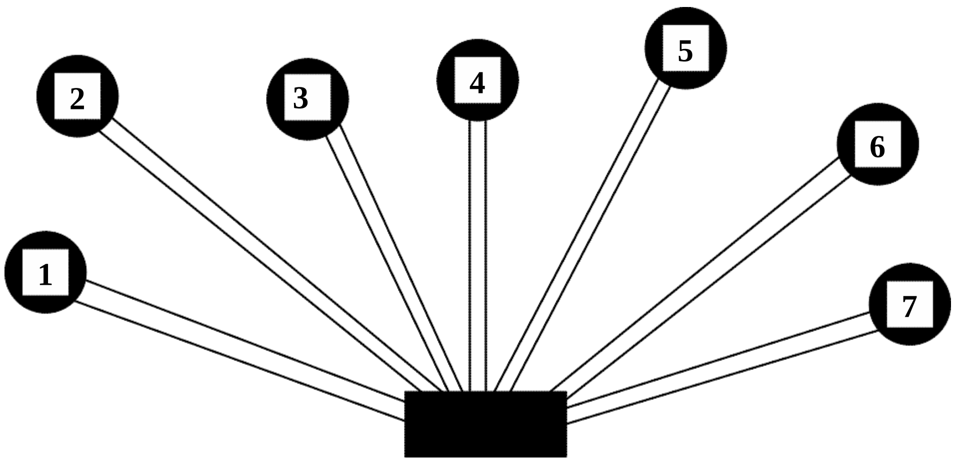
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| **0** | ~ | | | | | | | |
| **1** | 20 | ~ | | | | | | |
| **2** | 57 | 51 | ~ | | | | | |
| **3** | 51 | 10 | 50 | ~ | | | | |
| **4** | 50 | 55 | 20 | 50 | ~ | | | |
| **5** | 10 | 25 | 30 | 11 | 50 | ~ | | |
| **6** | 15 | 30 | 10 | 60 | 60 | 20 | ~ | |
| **7** | 90 | 53 | 47 | 38 | 10 | 90 | 12 | ~ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Customer** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **Cases** | 46 | 55 | 33 | 30 | 24 | 75 | 30 |

**Delivery Trucks have a capacity for 80 cases**

**Figure 6.15:**
**Acme Brewery Distribution**

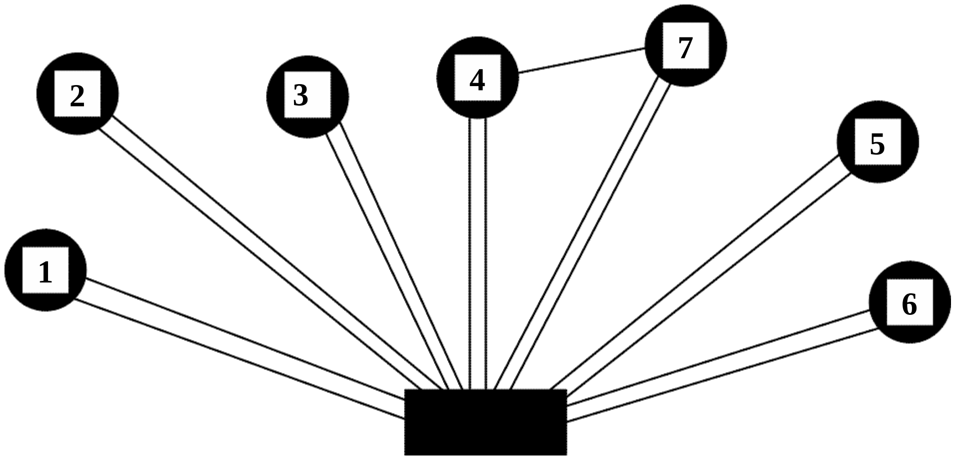
*Initial Solution*




*Savings Matrix*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| **1** | ~ | | | | | | |
| **2** | 26 | ~ | | | | | |
| **3** | 61 | 58 | ~ | | | | |
| **4** | 15 | 87 | 51 | ~ | | | |
| **5** | 5 | 37 | 50 | 10 | ~ | | |
| **6** | 5 | 62 | 6 | 5 | 5 | ~ | |
| **7** | 57 | 100 | 103 | 130 | 10 | 93 | ~ |

Select largest savings – 130 (customers 4 and 7)
Can we join? Yes, as demand (30+30 = 60) < 80
Join 4 and 7



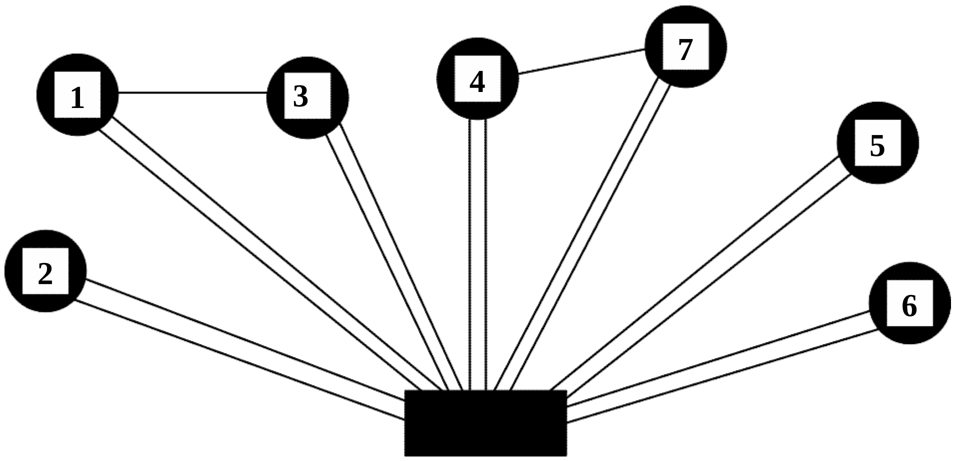

Select largest savings - 103 (customers 3 and 7)
Can we join? No, as 60 (already on truck) + 33 > 80

Next largest savings - 100 (customers 2 and 7)
Can we join? No, as 60 (already on truck) + 55 > 80

Next largest savings - 93 (customers 6 and 7)
Can we join? No, as 60 (already on truck) + 75 > 80
a
Next largest savings - 87 (customers 2 and 4)
Can we join? No, as 60 (already on truck) + 55 > 80

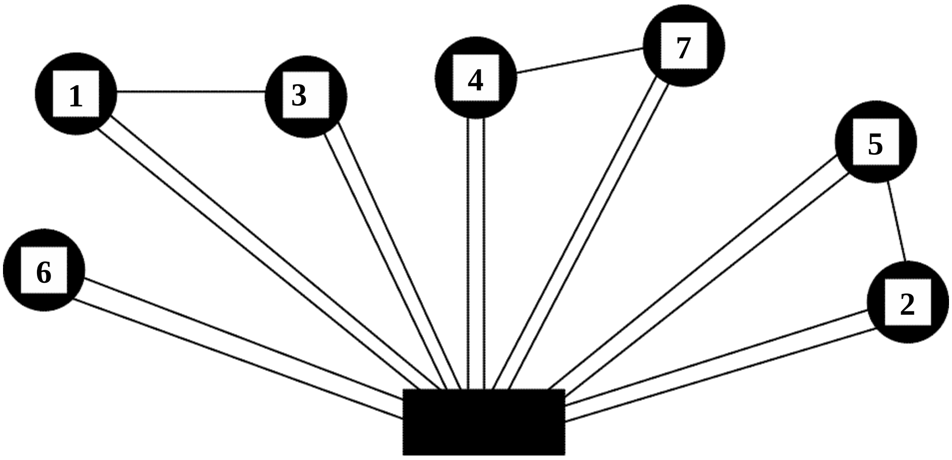
Next largest savings - 62 (customers 2 and 6)
Can we join? No, as 55 + 75 > 80

Select largest savings – 61 (customers 1 and 3)
Can we join? Yes, as 46 + 33 < 80
Join 1 and 3.



We continue with the rest, with the following results:

3 and 2 – No
7 and 1 – No
4 and 3 – No
5 and 3 – No
5 and 2 – Yes – join 5 and 2




2 and 1 – No
4 and 1 – No

All remaining positive savings are infeasible. The final set is as shown above.

▶ See the video **V31_Savings_1**, which uses the Savings Heuristic to find a solution to a vehicle scheduling problem. https://goo.gl/1ytdJd

## *SAQ 8*

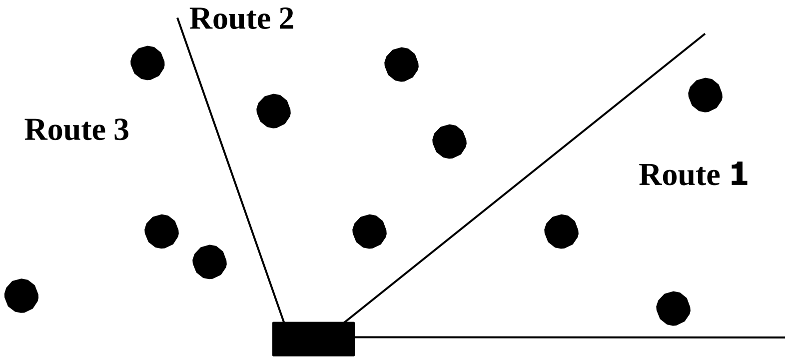
*6 customers are to be serviced from a single depot with a fleet of trucks of capacity 15 tons. The table below shows the distance matrix and the quantity to be delivered to each customer. Use the Savings Heuristic to determine the best routes*

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| **0** | ~ | | | | | | |
| **1** | 24 | ~ | | | | | |
| **2** | 19 | 17 | ~ | | | | |
| **3** | 20 | 31 | 16 | ~ | | | |
| **4** | 27 | 44 | 29 | 15 | ~ | | |
| **5** | 16 | 36 | 35 | 34 | 40 | ~ | |
| **6** | 12 | 23 | 25 | 28 | 37 | 11 | ~ |
| **Customer** | 1 | 2 | 3 | 4 | 5 | 6 | |
| **Tons** | 4 | 10 | 4 | 3 | 3.5 | 5 | |

## THE SWEEP HEURISTIC

This falls into a class of heuristics called *Cluster First, Route Second*. The idea behind the heuristic is to first cluster the customers into routes, having regard for vehicle capacity and customer demand. The next stage is to use a TSP heuristic to find a *good* vehicle tour through the customers in a cluster.

In applying the sweep heuristic, it is assumed that the location of each customer is known in terms of an (x, y) co-ordinate.




Consider the problem:

| Customer | Quantity (tons) | Location (x,y) |
|---|---|---|
| 1 | 10 | 41,49 |
| 2 | 19 | 55,20 |
| 3 | 5 | 20,50 |
| 4 | 16 | 30,60 |
| 5 | 23 | 30,25 |
| 6 | 19 | 10,20 |
| 7 | 17 | 15,60 |
| 8 | 18 | 45,10 |
| 9 | 6 | 65,20 |

The trucks are all based at the same depot which is located at co-ordinate (35,35) and each truck has a capacity of 45 tons.

The steps in the Sweep Heuristic are as follows:

*Step 1*

Locate the depot and all customers on a graph as shown in Figure 6.16.

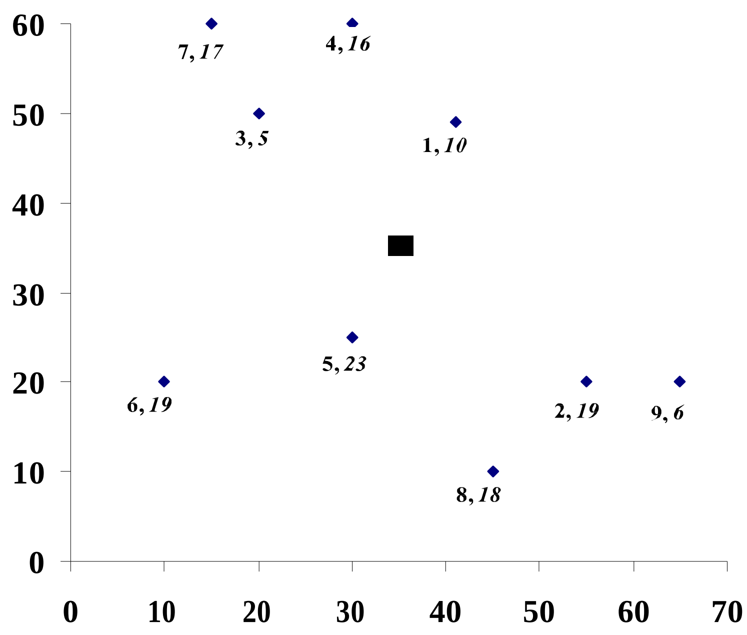
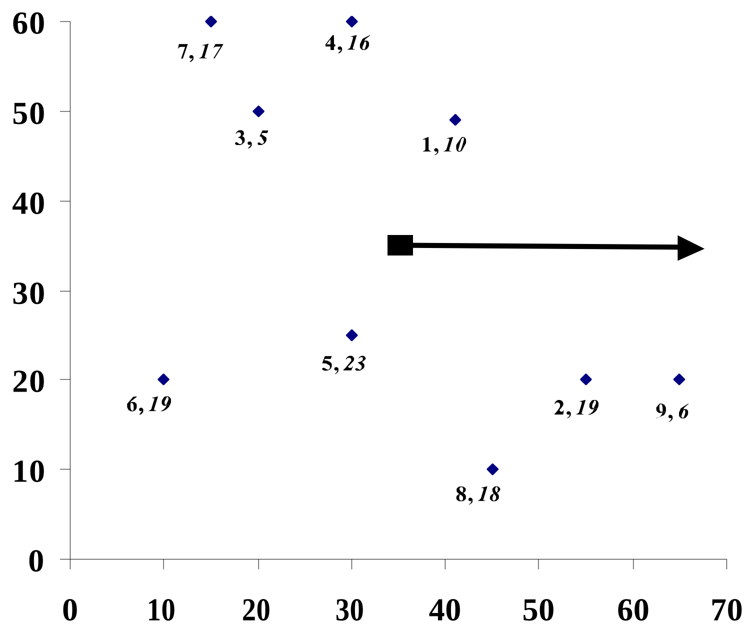



**Figure 6.16:**
**Customers and Depot Locations**

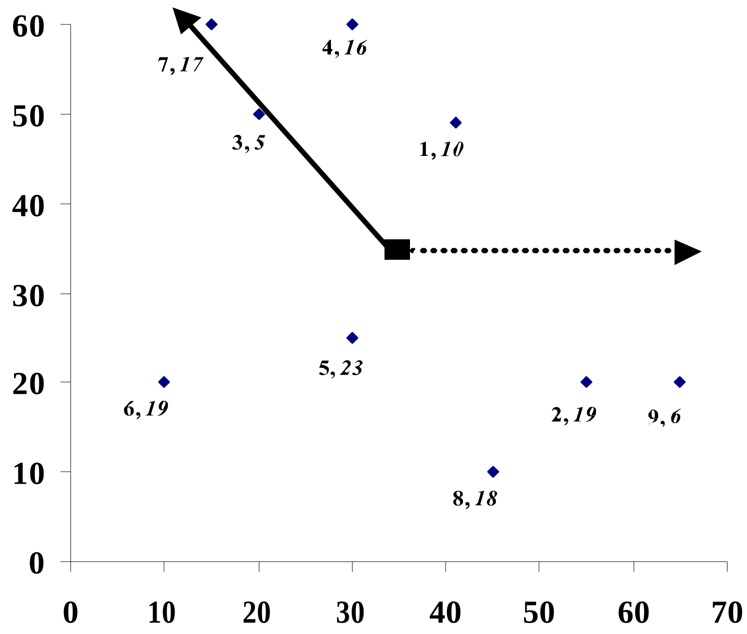
In Figure 6.16 the numbers below the points are the customer number and the requirement of that customer.

*Step 2:*
Centre a sweep-hand at the depot and pointed horizontally to the right.

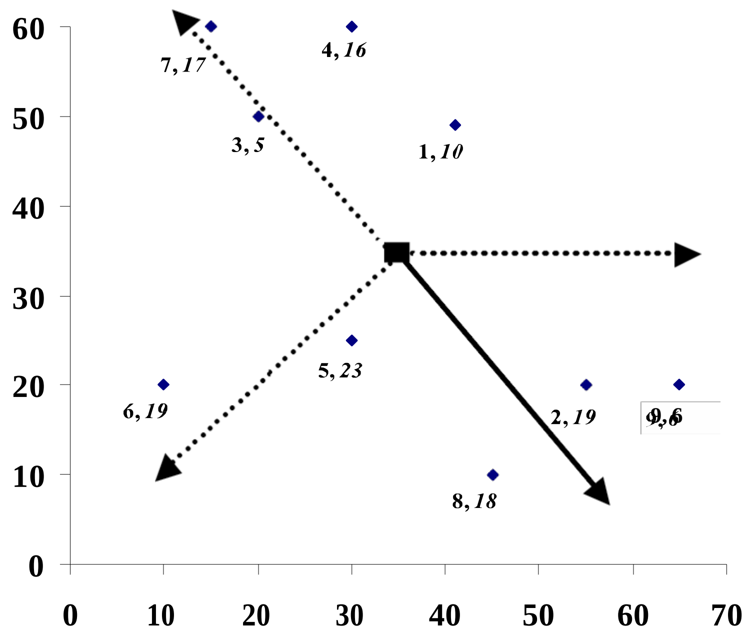


*Step 3:*
Rotate the sweep-hand anti-clockwise. All customers included in the sweep will be allocated to cluster 1.

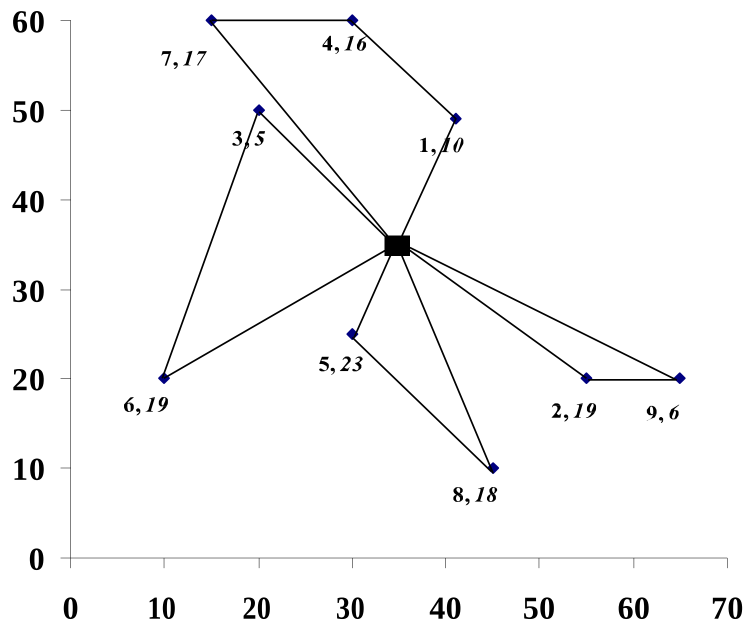
*Step 4:*
Stop the sweep-hand when the inclusion of the next customer will cause vehicle capacity to be exceeded.




*Step 5:*
Restart the sweep-hand to form remaining clusters.



*Step 6:*
Use one of the TSP heuristics to find a good tour through each cluster and depot.




## *SAQ 9*

*Use the sweep heuristic to find an allocation of customers to vehicles for the following problem:*

| Customer | Quantity Demanded Tons | Location (x , y) |
|---|---|---|
| 1 | 15 | (10 , 30) |
| 2 | 18 | (50 , 55) |
| 3 | 20 | (30 , 50) |
| 4 | 12 | (20 , 40) |
| 5 | 10 | (30 , 60) |
| 6 | 5 | (20 , 20) |
| 7 | 17 | (70 , 30) |
| 8 | 12 | (70 , 50) |
| 9 | 17 | (40 , 70) |
| 10 | 18 | (30 , 10) |
| 11 | 21 | (60 , 25) |
| 12 | 10 | (40 , 25) |

*The depot is located at (40, 40). Truck capacity is 50 tons, and up to 5 trucks are available.*

## SUMMARY

In this unit various types of network problems were briefly introduced and solution techniques and heuristics were applied. As before we see that in some cases an optimal solution can be achieved following well defined algorithm steps and in other cases such as the TSP the problems rapidly assume a complexity non immediately evident from seemingly benign network representations of the problem. The problems at hand whilst being amenable to formulation mathematically as linear programming problems, are not so easily solved especially when many combinations and possibilities exist and requiring exhaustive enumeration is computationally intensive. The discipline of network analysis is a well-developed field theoretically, with network and graph theory being a formalised branch of mathematics replete with many theorems and theoretical results having a wide range of applications ranging from genetics to computer science and logistics. Our interest has been briefly one of exposing the student to the many problem types that exist and have relevance especially in applied areas such as scheduling and logistics. We therefore briefly considered a range of problems and using graph theory techniques that include:

- The Chinese Postman
- The shortest route problem
- The minimal spanning-tree problem
- The maximal flow problem
- The travelling salesman problem
- The vehicle scheduling problem

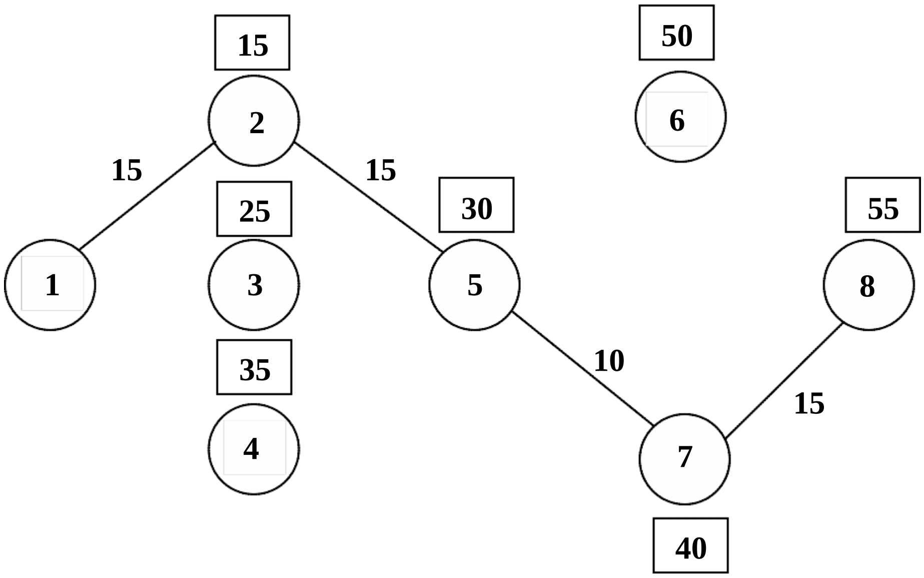
The last two were tackled using heuristics to yield ‘good’ solutions whilst the earlier problems are more amenable to solution by hand or using a spreadsheet especially for modest sized problems.

## SOME USEFUL REFERENCES AND RESOURCES

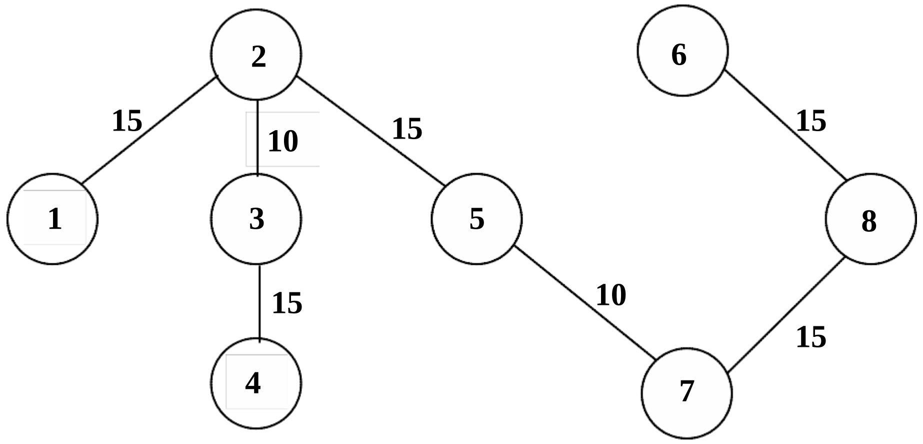
http://www.graph-magics.com

## ANSWERS TO SAQS

### *SAQ 1*




### *SAQ 2*



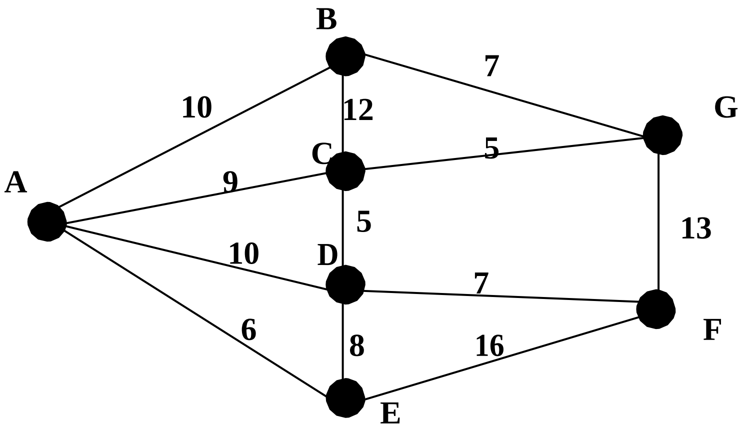

*Total length of cable required is 95 metres.*

## *SAQ 3*

*The degree of the nodes of the graph are as follows:*

| *Nodes* | *Degree* |
|---|---|
| *A* | *4* |
| *B* | *3* |
| *C* | *4* |
| *D* | *4* |
| *E* | *3* |
| *F* | *3* |
| *G* | *3* |

*The odd numbers suggest that in designing the tour some arcs will be travelled more than once. To decide which arcs to re-cross the algorithm is applied as follows:*




*Step 1: Nodes of odd degree: B, E, F and G.*

*Step 2: Pairing of nodes of odd degree. Three possible pairings must be considered:*
*1.B – E and F – G*
*2.B – F and E – G*
*3.B – G and E – F*

*The costings of these pairings are as follows:*
*B – E and F – G: 16* and 13 = 29*
*B – F and E – G: 20 and 18 = 38*
*B – G and E – F: 7 and 15 = 22*

**Note: 16 is the cost of pairing B – E, because it is the shortest distance on the graph from B to E.*

*The cheapest of the above pairings is B – G and E – F at a cost of 22.*

*Step 3: The arcs of the pairing B – G and E – F are added to the graph.*

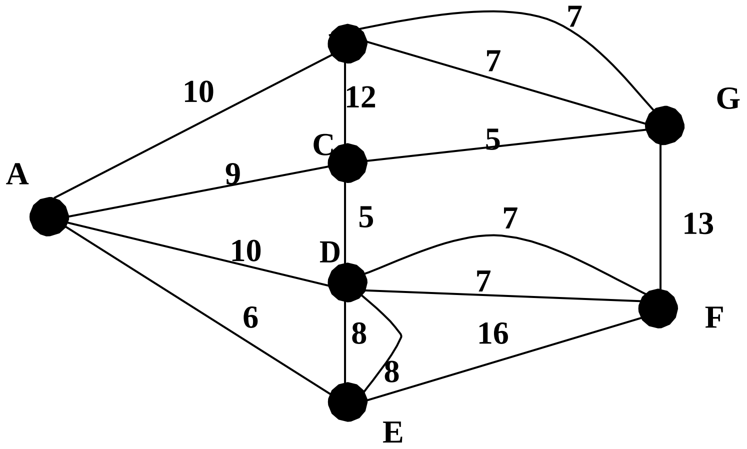
*For pairing B – G arc BG is added.*
*For pairing E – F arcs ED and DF are added. These arcs are added in preference to arc EF because their total length of 8 and 17 i.e. 15 is less than the length of EF at 16.*

*Step 4: Minimum tour length = $\Sigma$ all arcs + added arcs = 108 +22 = 130*

*Thus the tour of minimum length for the garbage truck is one that covers all arcs exactly once in the revised graph.*

*The figure below is the original graph with arcs BG, FD and DE added.*

*All nodes in the revised graph are even, and thus the required tour, covering all arcs exactly once, exists.*




*Given that the tour exists, a simple search procedure will find such a tour. For the above example such a tour is*

*A → B → C → G → B → G → F → D → C → A → D → F → E → D → E → A*

## *SAQ 4*

| *Node* | *Degree* |
|---|---|
| *1* | *3* |
| *2* | *5* |
| *3* | *5* |
| *4* | *3* |
| *5* | *2* |
| *6* | *2* |
| *7* | *2* |

*4 Nodes of Odd Degree; hence 3 possible matchings. These include:*

*1 – 2: 3 – 4: Distance = 7 + 6 = 13*
*1 – 3 : 2 – 4: Distance = 13+13 = 26*
*1 – 4: 2 – 3: Distance = 19 +7 = 26*

*The shortest of these matchings is, by inspection,*

*1 – 2: 3 – 4: Distance = 13*

*The arcs 1 – 2, 3– 4 are added to the original graph.*

*The minimal distance covered is, thus,*
*13 + sum of original arcs = 13 + 109 = 122.*

## *SAQ 5*

*Solving the LP formulation gives:*

$x_{12} = 1$
$x_{21} = 1$
$x_{36} = 1$
$x_{43} = 1$
$x_{54} = 1$
$x_{65} = 1$
*with Minimum tour length of 62.*

*This solution contains two sub-tours:*

*1 - 2 - 1, and*
*3 - 6 - 5 - 4 - 3.*

*To prevent both of these subtours from future solutions add the constraints:*

$x_{12} + x_{21} \leq 1$, *and*
$x_{36} + x_{65} + x_{54} + x_{43} \leq 3$

*Re-solving the problem with these additional constraints gives:*

$x_{15} = 1$
$x_{21} = 1$
$x_{36} = 1$
$x_{43} = 1$
$x_{52} = 1$
$x_{64} = 1$

*with Minimum tour length of 64.*

*The subtours now are:*

*1 - 5 - 2 - 1, and*
*3 - 6 - 4 - 3.*

*These subtours are prevented by adding the constraints:*

$x_{15} + x_{52} + x_{21} \leq 2$
$x_{36} + x_{64} + x_{43} \leq 2$

*After several further iterations the optimal solution is:*

$x_{12} = 1$
$x_{25} = 1$
$x_{54} = 1$
$x_{43} = 1$
$x_{36} = 1$
$x_{61} = 1$

*with Minimum tour length of 69. This SAQ shows how difficult it is to solve a TSP with only six nodes.*

## *SAQ 6*

*If we start with Town 1, the heuristic leads to the following route:*

*$1 \rightarrow 5 \rightarrow 2 \rightarrow 6 \rightarrow 4 \rightarrow 7 \rightarrow 3 \rightarrow 1$, which has a length of*
*9 + 10 + 7 + 8 + 12 + 11 + 14 = 71 miles.*

*It must be noted that this is not necessarily the optimal answer, but a 'good' answer provided by the Nearest Neighbour Heuristic.*

## *SAQ 7*

*The problem can be viewed as a TSP problem, with the cleaning times between flavours corresponding to the distances between towns.*

*Choosing the Nearest Neighbour Heuristic, and starting with Vanilla, we obtain the "route"*
*Vanilla*
*→Devon Cream*
*→Coffee*
*→ Chocolate*
*→Mint*
*→ Strawberry*
*→Vanilla,*

*with a total time of*
*60 + 80 + 100 + 110 + 130 +150 = 630 minutes.*

*As with SAQ 2, this solution can be considered good, but there is no guarantee that it is optimal.*

## *SAQ 8*

*The savings Matrix is:*

| | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|
| **1** | **~** | | | | | |
| **2** | **26** | **~** | | | | |
| **3** | **13** | **23** | **~** | | | |
| **4** | **7** | **17** | **32** | **~** | | |
| **5** | **4** | **0** | **2** | **3** | **~** | |
| **6** | **13** | **6** | **4** | **2** | **17** | **~** |

*The routes are formed as follows:*

*Suggestions*
*Join 4 and 3 – yes*
*Join 2 and 1 – yes*
*Join 3 and 2 – no, combined routes exceed vehicle capacity.*

*and so on….*

*The final solution is:*
*Route 1: Depot – 4 – 3 – Depot*
*Route 2: Depot – 2 – 1 – Depot*
*Route 3: Depot – 5 – 6 – Depot*

## *SAQ 9*

*The order in which customers would be allocated by the Sweep hand is as follows:*

| ***Customer*** | ***Quantity*** | ***Cumulative*** | |
|---|---|---|---|
| *8* | *12* | *12* | |
| *2* | *18* | *30* | ***Cluster 1*** |
| *9* | *17* | *47* | |
| *5* | *10* | *10* | |
| *3* | *20* | *30* | ***Cluster 2*** |
| *4* | *12* | *42* | |
| *1* | *15* | *15* | |
| *6* | *5* | *20* | ***Cluster 3*** |
| *10* | *18* | *38* | |
| *12* | *10* | *48* | |
| *11* | *21* | *21* | ***Cluster 4*** |
| *7* | *17* | *38* | |

*The nearest neighbour heuristic is now used to generate tours through the depot and customers. The results are:*

| ***Truck*** | ***Tour*** |
|---|---|
| *1* | *Depot* $\rightarrow$ *2* $\rightarrow$ *9* $\rightarrow$ *8* $\rightarrow$ *Depot* |
| *2* | *Depot* $\rightarrow$ *3* $\rightarrow$ *5* $\rightarrow$ *4* $\rightarrow$ *Depot* |
| *3* | *Depot* $\rightarrow$ *12* $\rightarrow$ *10* $\rightarrow$ *6* $\rightarrow$ *1* $\rightarrow$ *Depot* |
| *4* | *Depot* $\rightarrow$ *11* $\rightarrow$ *7* $\rightarrow$ *Depot.* |

# UNIT 7

## INVENTORY AND STOCK CONTROL METHODS

### AIM

The aim of this unit is to give an understanding of the basic ideas behind inventory management.

### OBJECTIVES

- To introduce the concept of an optimal order quantity.
- To show how a consideration of various inventory costs leads to the computation of the economic order quantity (EOQ).
- To introduce factors such as the re-order point, lead time and safety stock and to consider their role in the minimisation of stock-outs.
- To develop a procedure to effectively deal with uncertain demand.
- To present a stock classification system.

### REQUIRED READING

Students should read the following chapters of the mandatory text book (David R. Anderson, Dennis J. Sweeney, Thomas A. Williams and Kipp Martin, *An Introduction to Management Science: Quantitative Approaches to Decision Making*, 13th International Edition, Cengage Learning, ISBN-13: 9780538475655) in conjunction with this unit:

Chapter 10.1, 10.2, 10.4 and 10.6.

### 7.1 FUNDAMENTAL INVENTORY DECISIONS

In this course, management science techniques such as linear programming have been introduced. These techniques have a wide variety of applications but it is clear from some of the examples used that these approaches are of particular use in production applications. If we consider a typical linear programming formulation, it might be concerned with maximising profit in the production of several products. The main constraints on this production will be limited resources such as labour, time, and raw materials. Therefore, the organisation will generally have a *stock* or *inventory* of these raw materials on hand to enable production to take place.

Inventory serves a number of purposes in an organisation. In the case of raw materials such stocks are used because it is more convenient to have large quantities of a product delivered at each delivery. This product will then remain in stock until it is all used up. In the case of a finished product the use of inventories can allow demand to be satisfied if there is an unexpected increase in demand or delay in production. Additional reasons why organisations hold inventories include:

- To be able to buy or produce in economic batch sizes.
- To guard against shortages.
- To maintain flexibility in scheduling.
- To display items in order to accommodate customer selection.

- To allow for goods in transit.

The most basic of inventory decisions is *what quantity* to order and *when* to order it. These decisions are primarily based on *cost* and the different types of cost to be considered include:

*Ordering Costs:*
These include any costs that arise each time an order is made. Most of these costs will be internal to the company, mainly the extra labour required if a large number of orders are made. There may also be overheads such as telephone or postage associated with ordering. There may also be transport charges associated with each order; these may be in the form of a delivery charge by suppliers.

*Holding Costs:*
These include the cost of capital tied up in the inventory. The costs of actual storage and the wages of warehouse staff must also be included. In many cases there will also be costs resulting from the deterioration of goods in stock. In the real world there is often considerable "shrinkage" when large inventories are held.

*Purchase costs:*
This is the amount that a company must pay for the inventory items. In some instances, purchasing in large quantities can result in quantity discounts.

*Shortage costs:*
These are also known as *stock-out* costs and can either be planned or accidental. This category of costs is not as easy to quantify as the previous categories. The appropriate cost might be the cost of an emergency order to replenish stocks, if this is possible. If a company is unable to supply its customers, there will be a loss of customer goodwill. If raw materials run short, it may be possible to calculate the costs of lost production.

In general, the objective of inventory management is to determine a stock strategy that will minimise the total cost of carrying, ordering and purchasing inventories. More often than not, inventory costs are computed for one year. Thus, the goal is to minimise total annual cost, where:

| | | | | | |
|---|---|---|---|---|---|
| Total | | Annual | Annual | Annual | Annual |
| Annual | = | holding + | order + | purchase + | shortage |
| Cost | | cost | cost | cost | cost |

As we will see later, not all of these costs apply to every situation. For example, if quantity discounts are not a factor, purchase price is independent of order size, and order size decisions need not involve unit price. Likewise, shortages may be prohibited in some cases, making it unnecessary to include that component in the decision.

## 7.2 THE ECONOMIC ORDER QUANTITY MODEL (EOQ)

The objective of this, the classic inventory model, is to minimize the totality of costs incurred in holding inventory. This is generally achieved by determining the optimal order quantity Q. The assumptions underpinning this model are:

(i) Annual demand is known with certainty.
(ii) Items will be drawn from inventory at a constant rate.
(iii) A constant order size Q will be used.
(iv) Unit purchase costs are independent of order size. Hence, quantity discounts are not a consideration.

(v)  Ordering lead times are known and constant.
(vi) No shortages are permitted.

The purpose of EOQ calculations is to determine the amount that should be ordered. As this new order will arrive when stocks reach zero then the maximum amount in stock will be the amount ordered. The average amount of stock will be half this amount. Therefore, if we order small quantities frequently our average stock will be less than if we order large amounts infrequently.

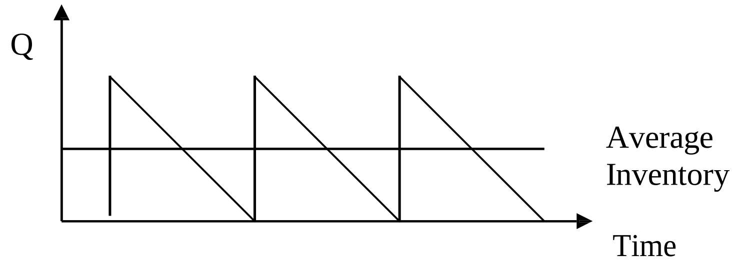



**Figure 7.1**
**Inventory level with infrequent ordering**

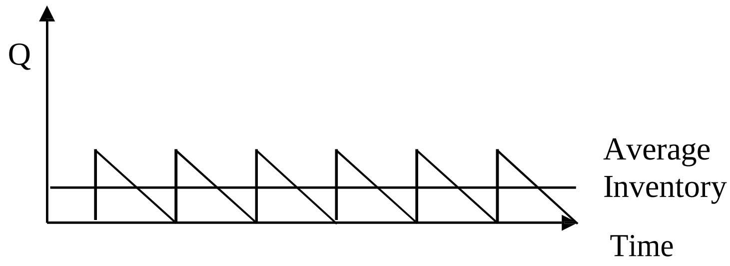



**Figure 7.2**
**Inventory level with frequent ordering**

In order to ensure that the new order arrives when the old stock is used up it is normally necessary to order some time in advance of the stock actually running out. The interval between an order being made and an order being delivered is known as the *lead time*. Because of the existence of lead time we do not order when stock is zero but when it falls to some other level known as the *re-order point*. In order to calculate the re-order point in the simple EOQ model we can take the demand over the lead time.

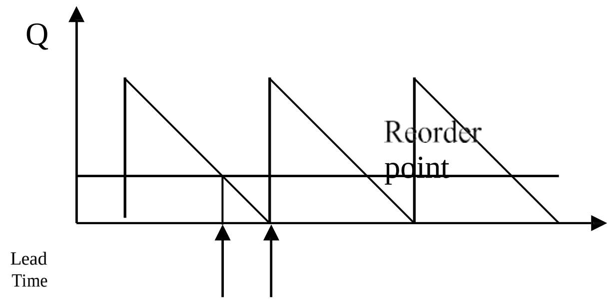



**Figure 7.3**
**Re-order point and lead time**

As stated previously, the objective of an EOQ analysis is to identify the order size that will minimise the sum of holding cost and ordering cost. Purchase costs are not included in the analysis because they are the same regardless of order size, and shortage costs need not be considered because it is assumed that shortages can be avoided.

*Deterministic Demand*
The order quantity is received instantly. The on-hand inventory is replenished instantly when its level becomes zero. Figure 7.4 illustrates this model, whilst Figure 7.5 shows the costs associated with the EOQ model.

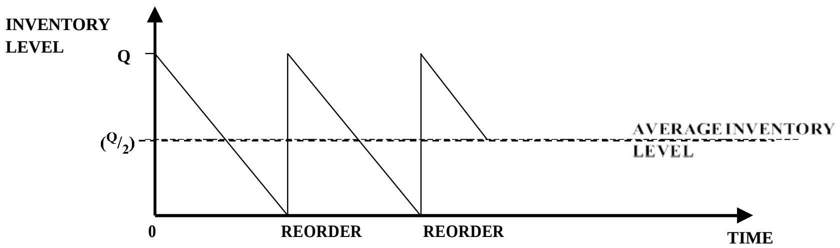



**Figure 7.4**
**The Classic Inventory Model**

This saw-tooth shape indicates that demand is known and constant and the reorder decision is triggered when on-hand inventory becomes zero.

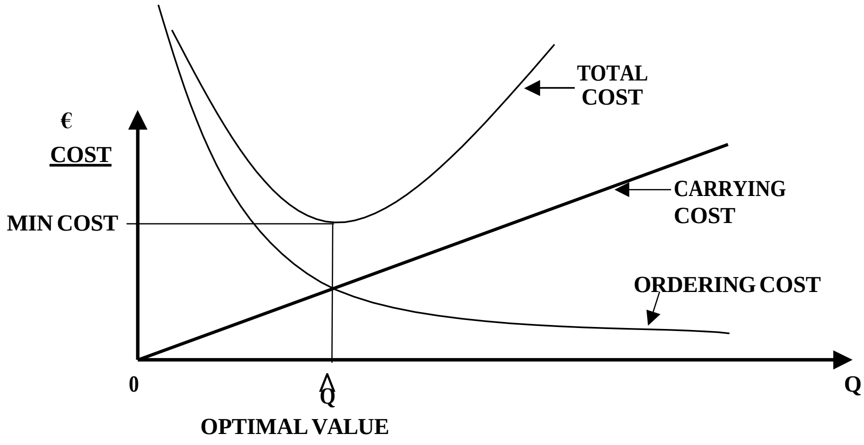



**Figure 7.5**
**Costs Associated with the EOQ Model**

In order to develop the EOQ model and to determine the optimal order quantity, $\hat{Q}$ , the following variables are defined:

Q = the order quantity

D = the annual demand, or usage, for the inventory goods.

$C_h$ = the annual holding cost of one inventory unit.

$C_o$ = the ordering cost per order

TC = the total annual inventory cost

Using these symbols, we can develop the following mathematical model:

Number of Orders per Year = $\frac{D}{Q}$

Total Ordering Cost = $C_o . \frac{D}{Q}$

Total Carrying Cost = $\frac{1}{2} Q . C_h$

(This is obtained by multiplying the mean inventory level by the holding cost per unit.)

Total Annual Inventory Cost = Total Ordering Cost + Total Carrying Cost

$$TC = C_o \frac{D}{Q} + \frac{1}{2} Q . C_h$$

For *minimum* total cost we differentiate with respect to Q and equate to zero. This is so because the function we want to minimise is a function of one variable Q and a necessary condition for a minimum is that the first derivative is zero at some point.

$$\frac{d}{dQ}\text{(TC)} = 0 = -C_o \frac{D}{Q^2} + \frac{1}{2} C_h$$

hence $Q^2 = \frac{2C_oD}{C_h}$

or

$$\hat{Q} = \sqrt{\frac{2C_oD}{C_h}}$$

**Note**:

1. The optimal value is determined by the three parameters/inputs $C_o$, D and $C_h$. notice the implications for example as the holding cost reduce the order quantity increases. The sensitivity of the optimal order to changes in these three inputs can be easily seen and investigated with a spreadsheet/graphically.
2. At the optimal level we have the result that Total Ordering Cost = Total Carrying Cost for this simple model.

***Example***

To demonstrate the use of this simple model let us assume:

$C_o$ = €100 per order

$C_h$ = €200 per unit held in inventory per year

D = 8100 units annually

Thus $\hat{Q} = \sqrt{\frac{2.(100)(8100)}{200}}$

$= \sqrt{8100}$

or $\hat{Q}$ = 90 units

Using this result, we can also determine the following:

$\hat{N}$ = Optimal number of orders per year = $\frac{D}{\hat{Q}} = \frac{8100}{90}$ = 90 orders

$\hat{T}_b$ = Optimal time between orders = $\frac{T}{\hat{N}}$

(where T = Total time i.e 1 year or 365 days)

Thus

$\hat{T}_b = \frac{365}{90} = 4\frac{5}{90}$ i.e. 4.056days

and

$$\text{TC}_{\text{Opt}} = \text{C}_o \cdot \frac{D}{\hat{Q}} + \tfrac{1}{2}\,\hat{Q} \cdot \text{C}_h$$

$$= \frac{C_o.D}{\sqrt{\dfrac{2C_oD}{C_h}}} + \tfrac{1}{2}\,\text{C}_h \cdot \sqrt{\frac{2C_oD}{C_h}}$$

or

$$\text{TC}_{\text{Opt}} = \sqrt{2C_oC_hD} = \sqrt{2(100)(200)(8100)}$$

Thus $\text{TC}_{\text{Opt}}$ = €18,000

Thus although some of the basic assumptions may not hold in practice (e.g. constant demand) the model can give useful management information. However, some simple “add-ons” and adjustments can make the model more realistic.

## *SAQ 1*

*Gadget Assemblers have a large factory requiring 1500 employees. 1% of the staff leaves every week. New employees must undergo a one-week training programme. The training is undertaken by outside experts and each one-week training programme costs €5000 to run regardless of the numbers who attend it. In order to maintain production, the company must have 1500 employees available. If extra staff are hired they cost €200 a week each while they are not producing. Formulate this as an inventory problem and identify the optimal number which should attend each training course.*

## *SAQ 2*

*A firm is faced with the attractive situation in which it can obtain immediate delivery of an item it stocks for retail sale. The firm has therefore not bothered to order the item in any systematic way. It has recently hired a management consultant to study their inventory control. The consultant has determined that the various costs associated with making an order are approximately €30 per order. In addition, she has determined that the cost of carrying one unit of the item in inventory for one year amounts to approximately €20 (primarily direct storage cost and forgone profit on investment in inventory). The demand forecast for the item is 19,200 units per year and is reasonably constant over time. When an order is placed for the item, the entire order is immediately delivered to the firm from the supplier. The firm operates six days a week plus a few Sundays, or approximately 320 days per year. Determine the following:*

*(a) Optimal order quantity per order.*

*(b) Optimal number of orders to place per year.*

*(c) Number of operating days between orders, based on the optimal ordering policy.*

*(d) Total holding costs and carrying costs associated with the optimal order size policy.*

## 7.3 The When-To-Order And How-Much-To-Order Decisions

The assumption that goods are received as soon as the order is placed would not work in practice. There is usually a delay, i.e. a *lead-time,* between placing an order and its arrival. Furthermore, it is equally unrealistic to order when the stock level becomes zero, rather it is usual to re-order when there is a finite quantity of inventory on hand. This is known as the *reorder point*. These changes to the basic model are depicted in Figure 7.6.

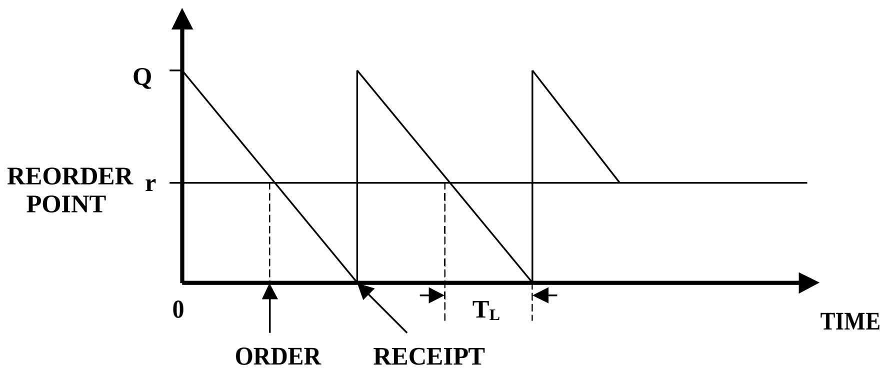



**Figure 7.6**
**Lead Time and Reorder Point Additions to the Basic Model**

In this model r is the *reorder point* and $T_L$ is the *lead time*, which is usually expressed in days. It is a simple matter to compute the reorder point as follows:

D = Annual Demand

Demand / day = $\frac{D}{365}$

Therefore, the reorder point, which is equal to the demand during the lead time, is

$$r = T_L \cdot \frac{D}{365}$$

Suppose daily demand is 25 units and the lead time is 10 days, then

r = (10) (25) = 250 units

It is important to note that $\hat{Q}$ is not affected by lead time.

## *SAQ 3*

Referring to SAQ 2, assume that rather than receiving immediate delivery of an order the firm must wait two days for the order to arrive.

(a) At what inventory level (reorder point) should the firm reorder?

(b) Since the firm has to wait two days for the order should it recalculate the optimal order size quantity?

(c) Suppose the lead time required to receive an order was twelve days; how many orders would the firm have outstanding at any given point in time?

(d) Shouldn't the firm simply lump these orders together and receive them all at once? Why or why not?

(e) Complete a table showing the annual ordering costs, annual carrying costs, and total inventory related costs for the following two cases:

(i) order every twelve days,
(ii) order at the interval determined in the solution to SAQ 2.

The assumption of deterministic demand is somewhat out of step with what happens in reality. Demand is nearly always *probabilistic* rather than *deterministic* and it is usually not possible to determine what probability distribution best describes the demand for a product. Rather, the approach used is to maintain buffer or *safety stocks* of inventory in order to avoid shortages or *stock-outs*. Figure 7.7 illustrates an inventory situation with uncertain demand and safety stocks.

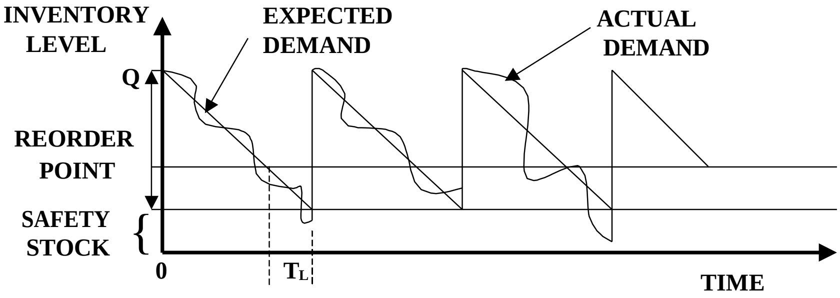



**Figure 7.7**
**Uncertain Demand and Safety Stocks**

This graph illustrates that the *safety stock* stops *stock-outs* occurring when the actual demand (probabilistic) is greater than the expected demand (deterministic).

There are several approaches to calculating safety stock levels. The most usual is for an organisation to define its preferred ***service level,*** that is, its objective is to satisfy some specified percentage of its customers throughout the year. These *service levels* are usually 99%, 95% or 90%. To specify 100% would result in very large inventories with resulting large inventory holding costs. In reality a 95% service level is the one most commonly chosen.

Typically the demand is assumed to be Normally distributed with a particular standard deviation. Naturally demand can be monitored and modelled using various distributions or even simulated. For simplicity we adopt the normal distribution assumption. Using the tables for the Normal distribution you can calculate the value below which the demand lies for the chosen service level.

To achieve the chosen service level a *safety stock* is added to the average maximum inventory level (which is defined to be Q). Thus in this approach the reorder point is:

**r =** $\overline{D}_L$ **+ Z** $\sigma_L$

Where:

$\overline{D}_L$ = the average demand during lead time.
$_L$ = the standard deviation of demand during lead time
Z = the Normal deviate at the specified service level.

The formula implies that:
Safety stock = Z$\sigma_L$

**Note**: we are essentially getting a confidence interval for mean demand during lead time and using the upper bound/tail as our safety stock.

***Example***
By way of illustration let us assume that an organisation wishes to have a service level of 95% i.e 95% of all their customers requiring stock during its lead time will receive it. This implies that Z = 1.645 (obtained from the Normal probability tables). If lead time is constant and is equal to 10 days and the daily demand is a normal distribution with mean 100 units and a standard deviation of 12 units, there arises:

$D_L$ = 10 × 100 = 1000 Units

$_L$ = $\sqrt{10 \times (12)^2}$ ≅ 38 units

This is so since the variance for 10 days can be taken as ten times the variance for one day if they are independent and identically distributed. The standard deviation is of course the square root of the variance

$\sigma_L$
as usual.

And

r = 1000 + 1.645 (38)
= 1000 + 63
=1063 units
and

Safety stock = 63 units

Now if C $_o$ = €50 and C $_h$ = € 0.50 per unit then:

$$Q = \sqrt{\frac{2(50)(100)(365)}{0.50}} = \sqrt{200 \times 100 \times 365}$$

Therefore $\hat{Q}$ = 2,702 Units

Thus the organisation's inventory policy is an Order Quantity of 2,702 units when inventory levels fall to 1063 units which includes a safety stock of 63 units.

▶ See the video **V32_Inventory_1**, which shows how to determine the reorder level when the demand is variable. https://goo.gl/UUXFqG

## SAQ 4

The daily demand for a commodity is 60 units on average with a standard deviation of 5 units. Lead time is constant at 9 days. The Order Cost is €100 and the Inventory carrying cost is €0.25 per unit on an annual basis.

The firm who deals with this commodity desires to establish an inventory policy that will maintain a 95% service level. Assuming that the commodities demand distribution is normally distributed, establish such a policy and portray it graphically.

## SAQ 5

Average demand for a product is 10 kg per day. Calculate the stock level at which an order should be placed so that 95% of the time demand during the lead-time is met. The lead time is 6 days, and daily demand can be assumed to be normally distributed with a standard deviation of 2 kg.

## 7.4 SENSITIVITY ANALYSIS AND THE EOQ MODEL

Order costs and holding costs can be difficult to estimate and are therefore prone to error. However, they play a central part in the estimation of $\hat{Q}$.

Consider the example from Section 7.2. Table 7.1 shows the effects on the optimum order quantity $Q$ and the total inventory costs $TC_{Opt}$ if we increase the order cost, or the holding cost, or both, by 5%, 10% and 15%.

| **% INCREASES** | | $\hat{Q}$ | | | **TC Opt** | | |
|---|---|---|---|---|---|---|---|
| **$C_o$** | **$C_h$** | **$C_o$ AND $C_h$** | **$C_o$** | **$C_h$** | **$C_o$ AND $C_h$** | **$C_o$** | **$C_h$ €** |
| **0** | **0** | **90** | **90** | **90** | **€ 18,000** | **€ 18,000** | **18,000** |
| **5** | **5** | **90** | **92** | **88** | **18,900** | **18,445** | **18,445** |
| **10** | **10** | **90** | **94** | **86** | **19,800** | **18,879** | **18,879** |
| **15** | **15** | **90** | **96** | **84** | **20,700** | **19,303** | **19,303** |
| **MAXIMUM CHANGE (%)** | | **0** | **6.7** | **-6.7** | **15.0** | **7.2** | **7.2** |

**Table 7.1**
**Sensitivity Calculations On The EOQ Model**

To interpret this table, consider for example the 10% row. The entries in the $\hat{Q}$ columns show that the EOQ increases from 90 to 94 units if the order cost increases by 10%, and it falls to 86 if the holding cost increases by 10%; the EOQ does not change if both costs increase by the same percentage. The entries in the $TC_{Opt}$ columns show that the total inventory cost increases from €18,000 to €18,879 if either the order cost or the holding cost increases by 10%, while it increases to €19,800 if both costs go up by 10%. The bottom line of the table shows that, if either the order cost or the holding cost increases by 15%, the total inventory cost increases by only 7.2%.

It is unlikely that both $C_o$ and $C_h$ will increase together by the percentages indicated and thus it can be concluded that this EOQ model is relatively insensitive to small changes or errors in the cost estimates. This makes it an attractive /robust tool in assessing optimal order quantities even if the costs are not totally deterministic.

## 7.5 THE PRODUCTION LOT SIZE MODEL

Let us assume that an organisation manufactures its own products for inventory instead of ordering from an outside supplier. Hence we can consider the production run (or lot size) as the order quantity. In this situation the products are received at a constant rate over time rather than instantaneously. Figure 7.8 depicts the situation:

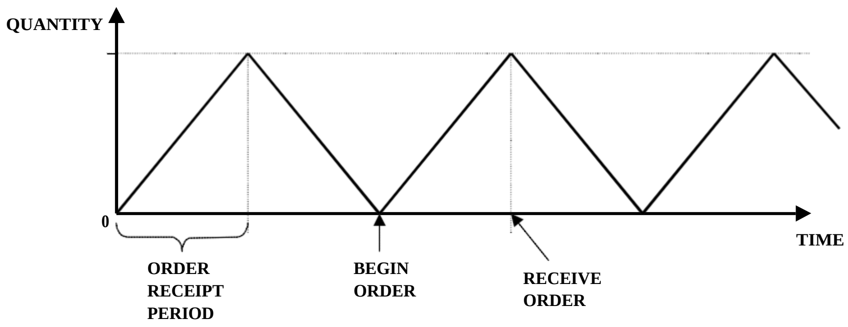



**Figure 7.8**
**The Production – Lot Size Model with Continuous Receipt**

The development of the model is as follows:

Let P represent the annual production rate. Then

$\frac{Q}{P}$ = the period necessary, as a proportion of a year, required to receive one entire order.

(For example, if goods can be produced at the rate of 1,000 per annum and if the order quantity, or lot size, is 100 units, then it will take 1/10 of a year to receive the order.)

$\frac{Q}{P}$. D = the usage rate during the order receipt period.

Q – $\left(\frac{Q}{P}.D\right)$ = Maximum inventory level for any order

Hence

½ $\left[Q - \left(\frac{Q}{P}.D\right)\right]$ = the average inventory level

$$\equiv \frac{Q}{2}\left[1 - \frac{D}{P}\right]$$

Therefore

Ordering Cost = $C_o . \frac{D}{Q}$

Carrying Cost = $C_h . \frac{Q}{2}\left[1 - \frac{D}{P}\right]$

And Total Cost = TC = $C_o . \frac{D}{Q} + C_h . \frac{Q}{2}\left[1 - \frac{D}{P}\right]$

And thus for a minimum

$$\frac{d}{dQ}\text{(TC)} = \frac{d}{dQ}\left\{C_o.\frac{D}{Q} + C_h.\frac{Q}{2}\left[1-\frac{D}{P}\right]\right\}$$

= 0

Which on solution yields

$$\hat{Q} = \sqrt{\frac{2C_oD}{C_h\,1-D/P)}}$$

***Example***
As an example let us assume that an organisation has an Annual Production Rate (P) = 12,000 units and an Annual Demand (D) = 8000 units. The set up cost ($C_o$) = €200 per order and the Carrying Cost ($C_h$) = €20.0 per unit. Substitution of these values into $\hat{Q}$ yields

$$\hat{Q} = \sqrt{\frac{2(200).8000}{(20.0)(1-\frac{8000}{12000})}}$$

Therefore the optimum production run is $\hat{Q} \cong 693$ units.

We may further calculate:

$$\hat{N} = \frac{D}{\hat{Q}} = \frac{8000}{693} \cong 12 \text{ production runs per year}$$

If we assume that the number of production days in a year is T=250, then the length between the start of each production run is:

$$\frac{T}{\hat{N}} = \frac{250}{12}$$

$\cong$ 21 working days

The Daily Production Rate = $\frac{12000}{250} = 48$ units / day

The Run Completion Rate = $\frac{693}{48} = 14.4$ days

The Maximum Inventory Level = $\hat{Q}\left(1-\frac{D}{P}\right)$ = 693 $\left(1-\frac{8000}{12000}\right)$

$\cong$ 231 Units

It should be noted that *reorder point* and *safety stock* considerations can also be incorporated in this model.

## SAQ 6

(a) A company uses a particular part for which there is an annual demand of 4,000 units. This part has a carrying cost of 25 cents per annum. The company can buy these parts at 20 cents each. Orders cost €30 per order and the part can only be supplied on a particular date in each month.

(i) Identify a close to optimal quantity which can be ordered at a regular interval of a whole number of months (e.g. 1, 2, 3, 4 .... 12 month interval).

(ii) What is the total cost of this inventory policy, including the purchase price?

(b) The production department in the company have offered to produce this part using batch production on an existing machine. Production at an equivalent rate of 10,000 units per annum is possible. Each unit produced will cost €0.15. In order to set the machine up for each batch a cost of €125 is incurred.

(i) What batch size should be used for lowest cost production?

(ii) Should the company buy in the part or produce it internally?

## SAQ 7

A company sells 60 litres of a particular type of soft drink each day. The company can produce this particular product at the rate of 210 litres per day. The drink sells for €5 per Litre. The company operates 7 days per week, and the weekly carrying cost is estimated at €0.05 per litre per week. Production set up cost is estimated at €7.50.

Determine the following:

1. The optimum run size and the number of runs per week.
2. The length of a production run in days.
3. The maximum amount of litres in stock.
4. The weekly total of holding and set-up costs.

## 7.6 QUANTITY DISCOUNTS AND THE EOQ MODEL

Quantity discounts are part of everyday business. For instance, in many off-licences various discounts can be obtained depending on the number of cases of wine purchased. In this situation discounts usually start at 5% for 1 case purchased and can go up to 10% + for 12 or more cases of your favourite tipple.

Suppose that the discount strategy is as portrayed in Table 7.2.

| DISCOUNT TYPE | DISCOUNT % | ORDER SIZE | UNIT COST |
|---|---|---|---|
| **A** | **0** | **0-899** | **€ 10** |
| **B** | **4** | **900-1799** | **9.60** |
| **C** | **6** | **1800-2699** | **9.40** |

**Table 8.2**
**Trader's Discount Strategy**

We will further assume that the trader's other terms of trade are:
Cost / Order: €50 ($C_o$)
Annual Holding Cost Rate: 15% (I)
Annual Demand: 9000 Units (D)

The calculation to ascertain the most advantageous order policy proceeds as follows:

***Step One***

For each discount type, compute $\hat{Q}$, the economic order quantity. In each case, $\hat{Q}$ depends on the holding cost $C_h$ which in turn depends on the unit cost. We have $C_h$ = IC, where I is the annual holding cost rate (15% in this example) and C is the unit cost (€10.00, €9.60, or €9.40 depending on the order size).

$$\hat{Q}_A = \sqrt{\frac{2(9000)(50)}{(0.15)(10)}} = 775$$

$$\hat{Q}_B = \sqrt{\frac{2(9000)(50)}{(0.15)(9.6)}} = 791$$

$$\hat{Q}_C = \sqrt{\frac{2(9000)(50)}{(0.15)(9.4)}} = 799$$

***Step Two***

For those $\hat{Q}$ that are too small to qualify for their category discount adjust the order quantity to that level which will allow them to qualify:

Thus: $\hat{Q}_B \rightarrow 900$
$\hat{Q}_C \rightarrow 1800$

***Step Three***

Now compute the total annual cost (TC) for all discount types according to:

$$TC = \frac{D}{Q} C_o + \frac{Q}{2} C_h + DC$$

ORDER COST HOLDING COST PURCHASE COST

This calculation is shown in Table 7.3

| DISCOUNT TYPE | UNIT COST | ORDER QUANTITY | ANNUAL COST | | | |
|---|---|---|---|---|---|---|
| | | | ORDER | HOLD | PURCHASE | TOTAL |
| | | | $^{D}/_{Q}.C_o$ | $^{Q}/_{2}.C_h$ | DC | TC |
| **A** | **€ 10** | **775** | **€ 581** | **€ 581** | **€ 90,000** | **€ 91,162** |
| **B** | **9.60** | **900** | **500** | **648** | **86,400** | **87,548** |
| **C** | **9.40** | **1800** | **250** | **1269** | **84,600** | **86,119** |

**Table 7.3**
**Total Cost Calculations for the Discount Strategy**

Thus, although having a significantly higher holding cost, Type C has the lowest total cost. This implies that order quantities of 1800 should be placed resulting in a discount of 6%. It should be noted the second cheapest - B - is only €1429 dearer than C. If inventory storage space is an issue, then B might be a more manageable option. A further calculation will show if the trader is tempted (for whatever reason) to lower type C discount to 5%, there is little to choose between B and C.

▶ See the video **V33_Inventory_2**, which involves an inventory problem with a discount schedule. https://goo.gl/BjJJda

## *SAQ 8*

*Given the price list below, determine the order size that will minimise the total of annual holding, ordering, and purchasing costs. The parameters of the problem are as follows:*

*Annual demand is 500 units.*
*Ordering Cost is €25 per order.*
*Holding Cost is €10 per unit per annum.*

| *Order Quantity* | *Unit Price €* |
|---|---|
| *1 to 30* | *130* |
| *31 to 40* | *125* |
| *41 or more* | *120* |

## *SAQ 9*

Given the price list below, determine the order size that will minimise the total of annual holding, ordering, and purchasing costs. The parameters of the problem are as follows:

Annual demand is 1,200 units.
Ordering Cost is €41 per order.
Holding Cost is €2 per unit per annum.

| *Order Quantity* | *Unit Price* € |
|---|---|
| 1 to 999 | 27 |
| 200 to 299 | 26 |
| 300 to 399 | 25 |
| 400 or more | 24 |

## 7.7 ABC ANALYSIS OF STOCKS

Stocks in warehouses have differing monetary values, as well as differing rates of demand. Rather than designing a global inventory policy which takes in all of the products in an inventory, it is more useful to classify the products by way of a *Pareto* or *ABC* analysis. This makes use of the Pareto Distribution. If expressed in terms of value this analysis would typically show that:

| | | |
|---|---|---|
| A | 80% of VALUE | 20% BY NUMBER |
| B | 15% of VALUE | 30% BY NUMBER |
| C | 5% of VALUE | 50% BY NUMBER |

i.e. 20% of the goods in the inventory represent 80% of the monetary value of the inventory etc. These statistics translate into the well-known *Pareto curve*, which is shown in Figure 7.9.

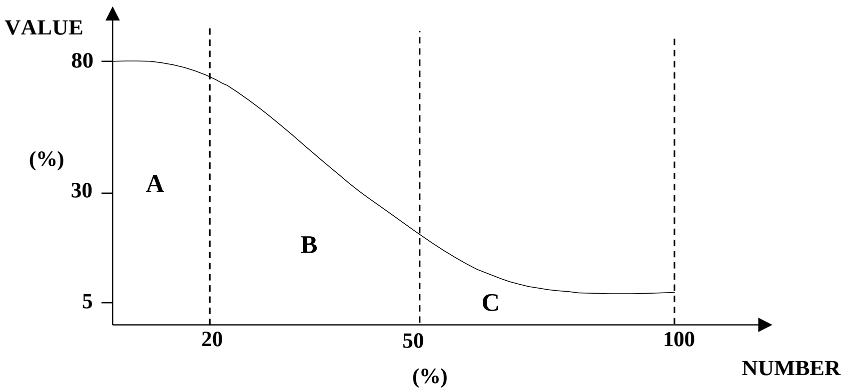



**Figure 7.9**
**The Pareto Curve**

Thus the tightest controls should be imposed on the 'A' classification, including all calculations regarding $\hat{Q}$, *Reorder Point* (r), *Safety Stock* (SS) and *Total Cost* (TC). Constant tracking on *Demand* (D) is essential to ensure that the *Inventory Control* situation is well in hand.

'B' classification items should be checked periodically whilst the C items need only be checked in a random manner.

## *SAQ 10*

(a) A furniture store stocks a very popular type of dining table which is in constant demand. Each time the store orders these tables from the manufacturer it costs €50. Because these tables are popular, and large, they absorb a large amount of storage space. Thus the store manager has computed that storage cost per table per year is €100. Current demand is expected to be 49,000 units a year, based on past sales records. Compute the various parameters of this inventory model.

(b) Suppose however that the manager after formulating his inventory policy on (a) discovers, after some three months into the current trading year, that actual demand is 50% ahead of that estimated, what errors in inventory policy is the store managing facing? Discuss the impact of these errors.

### SUMMARY

In this unit a brief examination of some simple but insightful inventory models was conducted. The importance of a properly managed and understood inventory policy cannot be understated and indeed can make the difference between success and failure for a company. It must be realised that the notion of inventory is wider than simple physical stock units on shelves and generally can extend to the notion of unused resources including finished stock, in process production, cash and individuals even etc.

Inventory can in some cases extend to up to 40% of total company assets. Managing the inventory policy will involve many aspects ranging from identifying what and when to stock, maintaining accurate forecasts of demand, interfacing with suppliers, customers and other intermediaries. Inventory may be in place for reasons such as matching supply and demand efficiently and avoiding stock outs and possibly also to facilitate lower production costs.

Our approach to the subject like many others is predicated on some adopted simplifying assumptions. Thus we start by adopting the classic assumptions of the Economic Order Quantity (EOQ) model. These adopted assumptions included:

- Annual demand being known with certainty.
- Demand/consumption from inventory is at a constant rate.
- A constant order size Q is used.
- Quantity discounts are not present.
- Ordering lead times are known and constant.
- No shortages are permitted etc.

The approach then focused on total (annual usually) costs being minimised. The total cost itself was decomposed into its constituent components namely, those of ordering, holding and purchasing, with stock out/shortage costs being ignored.

A simple expression for total costs was derived:

$$TC_{Opt} = C_o \cdot \frac{D}{\hat{Q}} + \tfrac{1}{2}\,\hat{Q}\,.\,C_h$$

and its minimisation resulted in a result whereby:

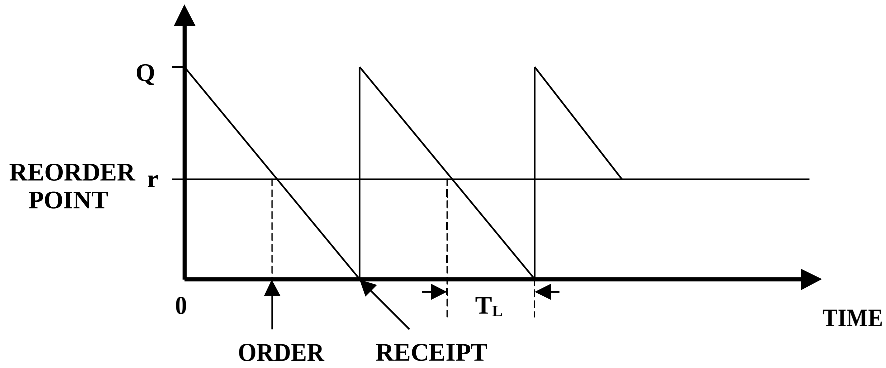
Total Ordering Cost = $C_o . \frac{D}{Q}$ =Total Carrying Cost = $\frac{1}{2}$ Q. $C_h$

or $\hat{Q} = \sqrt{\frac{2C_oD}{C_h}}$

with

TC $_{Opt}$ = $\sqrt{2C_oC_hD}$

Lead time is easily accommodated and reorders are placed at the reorder point allowing enough stock for lead time demand.

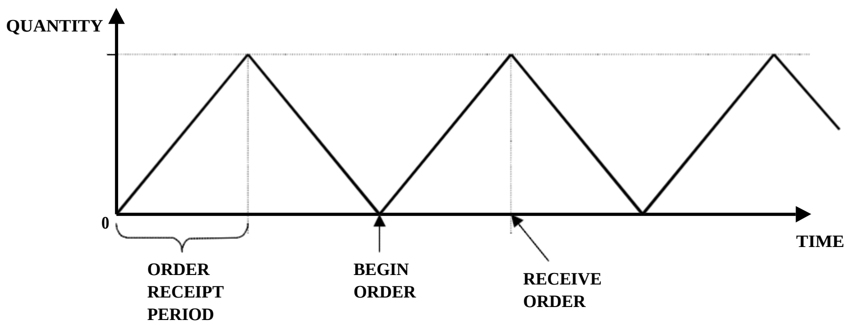



The concept of a **service level** was then introduced as a pragmatic recognition of uncertain demand during lead time. This relaxation of deterministic demand is a slight generalisation of earlier assumptions but once again to achieve tractable results it was assumed that demand followed the normal distribution during lead time. A simple result for service levels then followed:

$$\mathbf{r} = \overline{D}_L + \mathbf{Z}\ \sigma_L .$$

With(Z.$\sigma_L$) being the **safety/buffer stock.**

A second simple model/adaptation took the form of the so called **production lot model** the products being received at a constant rate P over time rather than instantaneously, as depicted below:



Following a similar treatment with cost minimisation being pursued, results were obtained which are seen to be simple modifications of the earlier model, namely;

$$\hat{Q} = \sqrt{\frac{2C_o D}{C_h \left(1 - D/P\right)}} \text{ and TC}_{\text{Opt}} = \sqrt{2C_o C_h D(1 - \frac{D}{P})}$$

So that a simple factor of $(1-\frac{D}{P})$ distinguishes the two model results.

In reality many different pricing/purchasing structures are possible and one simple quantity discount structure was examined and its impact on the optimum order quantity was explored. Many variants and deviations from the above simple models and assumptions are possible. In all cases one is faced with a set of inputs typically cost and demand structures and forecasts and a set of outputs typically total cost exposure over a period and across possibly different inventory items. The objective of cost minimisation (or indeed multiple objectives) requires that these inputs be reliable and any solution/prescription be investigated and assessed for reliability and reviewed regularly depending on the context. Hence sensitivity analysis is integral to evaluating the impact of variations or errors in these inputs and forecasts. The simple models were seen to be relatively robust and attractive in this regard. Tools such as simulation (see next unit) are invaluable in sampling from different scenarios and assessing the impact on decisions and outcomes. Given the importance of inventory policies many decision support platforms and inventory models software have been developed for various sectors such as the manufacturing and production sector. Micro managing approaches are possible if one chooses to classify the products by way of a *Pareto* or *ABC* analysis. This allows for greater care and attention being directed where it is needed most. For example, group A stock might benefit from quantitative analysis and forecasting techniques given their high value etc. modern material requirement planning (MRP) systems are the focus of much attention with Just in Time (JIT) approaches being advocated and adopted by some sectors. Inventory theory is undergoing continuous theoretical and practical development in the search for efficiencies and in tandem with ever changing business models and customer supply models and channels, not least driven by technological advances and customer expectations.

## SUMMARY OF STEPS

- Identify the context as one of inventory management.

- Examine the various components and adopt assumptions/descriptions of the various components. Particular attention is needed for modelling the three key inputs, namely; the ordering costs the demand and the holding costs.

- See if the situation is described by a simple standard model such as those considered above and familiarise yourself with the key features/assumptions.

- Determine the best values of the inputs $C_0$, $C_h$ ,D etc. using if necessary forecasting techniques.

- Calculate the EOQ and any other metrics/outputs of interest, these are typically total cost, reorder point, safety stock (if any), time between and number of orders etc.

- Use these outputs to facilitate appraisal of the system performance. If other economic implications are present then explore various options and inputs under your control such as the service level, different cost structure options and conduct sensitivity analyses.

- Implement and review the decisions preferably with clear report generation and interactive spreadsheet or simulation packages supporting the analysis and decision prescriptions.

## FINAL COMMENTS

The student should seriously consider developing a spreadsheet for the interactive calculations of the order quantity, total cost and associated outputs of interest. Thus by varying the inputs the outputs can be interactively reviewed, charted and form the basis of efficient decisions based on sound analysis. One can vary the inputs directly of course or simulate them drawing randomly from appropriate distributions used in modelling the input values. For example, the demand might be sampled from a normal distribution with some specified mean and variance and the EOQ determined in each case and possibly assessed for efficiency.

## SOME USEFUL REFERENCES AND RESOURCES

http://scm.ncsu.edu/public/inventory/
http://www.blandon.co.uk/finance/topic5/t5other.htm
http://www.solver.com/purframe.htm

# ANSWERS TO SAQS

## *SAQ 1*

$C_h$ - holding or carrying cost per unit per week = €200
$C_o$ - order cost per order = €5000
D - requirement for the product = 15 (per week)

Using the formula for $\hat{Q}$

$$\hat{Q} = \sqrt{\frac{2 \times 15 \times 5000}{200}} = 27.39 \cong 30$$

The optimal number for each course is 30, and a course will be needed every 2 weeks.

## *SAQ 2*

(a) Using the formula for $\hat{Q}$

$$\hat{Q} = \sqrt{\frac{2 \times 30 \times 19200}{20}} = 240$$

The optimal quantity is 240

(b) To get the number of orders in a year, divide the total demand by the order quantity: 19200/240 = 80 orders per year.

(c) To get the number of operating days between orders, divide the number of operating days in the year by the number of orders: 320/80 = 4 days between orders.

(d) The holding costs are

$\frac{1}{2}\hat{Q}\, C_h = 120 \times 20 =$ €2,400.

The order costs are

$C_o \dfrac{D}{\hat{Q}} = 30 \times (19200/240) = 30 \times 80 =$ €2,400.

The total annual investment cost is therefore €4,800.

Observe that, at the optimum, the order cost and the holding cost are equal.

## SAQ 3

(a) Since the demand per operating day is 19200/320 = 60, the reorder point (the demand over the lead time of two days) is 120.

(b) No, the EOQ remains the same.

(c) Since the time between orders is four days, there will be three orders outstanding (one is ordered as another is delivered).

(d) No. If orders are joined, the order quantity will move away from optimal value and inventory costs will increase.

(e) If we order every 12 days, the order quantity will increase from 240 to 720. In that case, the holding costs are

$$\tfrac{1}{2}\hat{Q}\,C_h = 360 \times 20 = €7{,}200,$$

and the order costs are

$$C_o \frac{D}{\hat{Q}} = 30 \times (19200/720) = 30 \times 26.67 = €800.$$

This leads to the following comparative table:

| *Order period* | *Holding cost* | *Order cost* | *Total cost* |
|---|---|---|---|
| 4 days | €2,400 | €2,400 | €4,800 |
| 12 days | €7,200 | €800 | €8,000 |

## *SAQ 4*

Such an inventory policy will require the establishment of an:

- Order Quantity
- Re-order Point
- Safety Stock

*Order Quantity*

$$\hat{Q} = \sqrt{\frac{2\times100\times60\times365}{0.25}} = 4186 \text{ units}$$

*Re-Order Point*

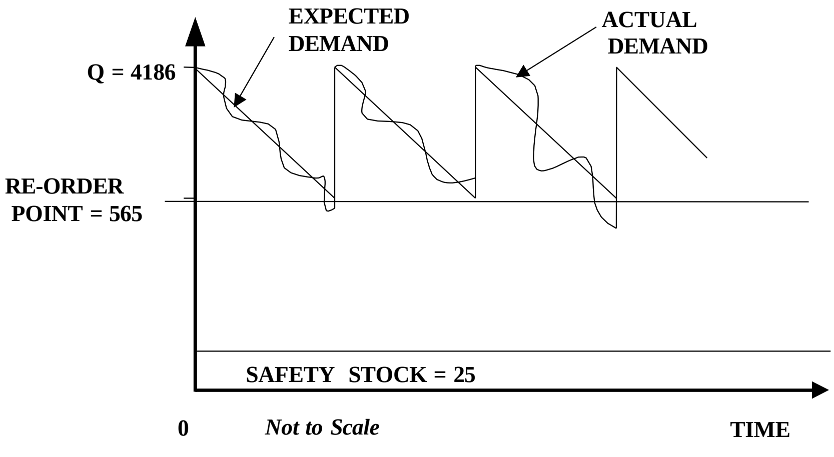
r = $\overline{D}_L$ + $Z_{0.95}\,\sigma_L$

$\overline{D}_L$ = 60 $\times$ 9 = 540
and
$_L$ = $\sqrt{9\times(5)^2}$ = 15

Thus:
r = 540 + 1.645 (15)
= 540 + 24.675
= 565 Units

*Safety Stock*

SS = $Z_{0.95}\,\sigma_L$
= 1.645 $\times$ 15
= 25 Units



## *SAQ 5*

Since the lead time is 6 days, the mean demand over the lead time is 60kg and the standard deviation of this demand is 4.90kg.

Therefore 95% of the values are under

60 + 1.645(4.90) = 68.06 ≅ 68kg.

A safety stock of 8kg is needed, and an order is placed when stock levels fall to 68kg.

## *SAQ 6*

(a)

| | | | | |
|---|---|---|---|---|
| $C_h$ | - | holding or carrying cost per unit | = | €0.25 |
| $C_o$ | - | order cost per order | = | €30 |
| D | - | requirement for the product | = | 4000 |

Using the formula for $\hat{Q}$

$$\hat{Q} = \sqrt{\frac{2 \times 30 \times 4000}{0.25}} = 979.79 \cong 980$$

The optimal order quantity is 980.

This implies just over 4 orders per annum so ordering 1000 units every 3 months is very close to optimal and meets our requirements.

The inventory cost of this policy is the holding cost plus the order cost.

| | | | |
|---|---|---|---|
| Order cost | €30 $\times$ 4 | = | €120 |
| Holding cost | €0.25 $\times$ 500 | = | €125 |
| Inventory cost | | | €245 |
| Purchase cost | €0.20 $\times$ 4000 | = | €800 |
| Total Cost | | | €1045 |

(b)

| | | |
|---|---|---|
| $C_h$- holding or carrying cost per unit | = | €0.25 |
| $C_o$- batch set-up cost per order | = | €125 |
| D - requirement for the product | = | 4,000 |
| P - production rate for the product | = | 10,000 |

Using the production run equation

$$\hat{Q} = \sqrt{\frac{2C_oD}{C_h\,1 - D/P)}}$$

$$\hat{Q} = \sqrt{\frac{2 \times 125 \times 4000}{0.25 \times \left(1 - \frac{4000}{10000}\right)}} = 2581.99 \cong 2582$$

A batch size of 2582 units is optimal.

Cost of batch production

set-up cost €125 $\times$ 1.55 = €193.65

holding cost €0.25 $\times$ 2582 $\times$ (1 - 4000/10000) $\times$ 0.5 = €193.65

Note that, at the optimum, the order or set-up cost is equal to the holding cost.

Production cost €0.15 $\times$ 4000 = €600.00

Total Cost = €987.30

The total cost of production is €987 which is less than the €1045 cost involved in buying the part.

Therefore the part should be produced rather than purchased.

## *SAQ 7*

| | | | | |
|---|---|---|---|---|
| $C_h$ | - | holding or carrying cost per unit | = | €0.05 per litre per week |
| $C_o$ | - | batch set-up cost per order | = | €7.50 |
| D | - | requirement for the product | = | 60 litres /day |
| | | | = | 420 litres /week. |
| P | - | production rate for the product | = | 210 litres /day |
| | | | = | 1470 litres/week |

Using the production run equation gives a batch size of 420 litres. Hence, there should be one run per week.

Length of a production run = 420 /210 = 2 days.

Maximum Stock Level = 2 days $\times$ (210 - 60) per day = 300 litres.

Weekly Costs:

Set-up Cost = €7.50 $\times$ 1 = €7.50

Holding Cost = 300/2x�.05 = €7.50

Total Weekly cost = €15.00

Note again the equality of holding and set-up costs.

## *SAQ 8*

Annual demand is 500 units.
Ordering Cost is €25 per order.
Holding Cost is €10 per unit per annum.

Using the formula for $\hat{Q}$

$$\hat{Q} = \sqrt{\frac{2 \times 500 \times 25}{10}} = 50$$

Because this EOQ quantity falls in the range associated with the lowest unit price, it will yield a lower total cost than any other possible quantity. Therefore, it is unnecessary to compare the total cost of the EOQ with the total cost of other quantities.

## *SAQ 9*

Annual demand is 1,200 units.
Ordering Cost is €41 per order.
Holding Cost is €2 per unit per annum.

. Using the formula for $\hat{Q}$

$$\hat{Q} = \sqrt{\frac{2 \times 1200 \times 41}{2}} = 222 \text{ units}$$

The options that now must be costed are:

222 units at €26 each
300 units at €25 each
400 units at €24 each

Total cost is computed as follows:

TC = $Q \times C_h/2 + D \times C_o/Q + C \times D$

TC (at Q = 222) = 222 + 222 + 31200 = €31,644
TC (at Q = 300) = 300 + 164 + 30000 = €30,464
TC (at Q = 400) = 400 + 123 + 28800 = €29,323

Hence, the best order size is 400 units because it will yield the lowest total annual cost.

## *SAQ 10*

(a) Order Cost — $C_o$ = €50
Storage (holding) Cost — $C_h$ = €100;
Demand — D = 49,000 units

Thus:

$$\hat{Q} = \sqrt{\frac{2 \times 50 \times 49000}{100}}$$ = 221 tables per order

and

$\hat{N}$ = 490,000/221 = 222 order/year

$T_b = T/\hat{N}$ = 365/222 = 1.64 or 2 days between orders

$TC_{opt} = \sqrt{2CoChD} = \sqrt{2(50)(100)49{,}000}$

= €22,136 per year

(b) Order Cost — $C_o$ = €50
Storage (holding) Cost — $C_h$ = €100;
Demand — D = 49,000 units + 50% = 73,500

*Then*

$$\hat{Q} = \sqrt{\frac{2 \times 50 \times 73{,}500}{100}}$$ = 271 tables per order

$\hat{N}$ = 73,500/271 = 271 orders/year

$T_b = T/\hat{N}$ = 365/271 = 1.35 days between orders

$Tc_{opt} = \sqrt{2CoChD} = \sqrt{2(50)(100)(73500)}$

= €27,111 per year

(c) Therefore, with the higher demand, the minimum inventory cost is €27,111. This occurs when the order size is 271 tables. But as the manager is ordering 221 tables per order, the actual inventory costs are

$TC = C_o \frac{D}{Q} + \frac{1}{2} QC_h$ = 16629 + 11050 = €27,629,

which is €518 or 2% higher than the minimum cost.

This analysis shows that a 50% error in demand forecasting results in a 23% increase in the EOQ and only a 2% increase in total inventory costs. This lack of sensitivity to errors in parameter estimates accounts to a certain extent for the popularity of the EOQ model.

# UNIT 8

## QUEUING & WAITING LINE MODELS

### AIM

This unit aims to introduce the concepts of queuing theory. The emphasis of the unit is on both the derivation and the application of the models involved.

### OBJECTIVES

- To show how finite-difference mathematics can be used to model and investigate queue behaviour.
- To determine the steady state condition of queues and compute critical queue parameters.
- To economically analyse various queue configurations.
- To recognise and be able to apply a variety of queuing models.
- To be familiar with queue classification systems.

### REQUIRED READING

Students should read the following chapters of the mandatory text book (David R. Anderson, Dennis J. Sweeney, Thomas A. Williams and Kipp Martin, *An Introduction to Management Science: Quantitative Approaches to Decision Making*, 13th International Edition, Cengage Learning, ISBN-13: 9780538475655) in conjunction with this unit:

Chapter 11.1, 11.2, 11.3, 11.4 and 11.5

### 8.1 THE BASIC COMPONENTS OF QUEUES

*Waiting Lines* or *Queues* are part of the fabric of modern living. We queue for services at Supermarkets, Petrol Stations, Banks …. We queue at Traffic Lights, Air Terminals, Train/Bus Stations … - the list is endless. Aircraft, waiting to land, are stacked (queued) over large International Airports. Telephone calls can be queued waiting for an operator to become available, internet traffic can be queued at routers, and somewhat more unusually, water particles are queued at the sluice gates of the large hydro-electric dams. The study of waiting lines, known as *Queuing Theory* is nearly a century old and can be traced back to the work of A.K Erlang, a Danish Mathematician who studied the fluctuating load on the Danish Telephone Exchange in the early part of the 20th Century.

Queuing theory is concerned with building mathematical models to help us describe and understand different queuing situations and is unlike most other areas of Operations Research/Management Science that seek to build a model for the purpose of optimising some objective function. Queuing theory is concerned with describing what will happen in a particular queuing situation. In our everyday lives we tend to think of queues as relating to people only. However, in dealing with queuing theory we must broaden our viewpoint considerably. Queues may consist of people, cars, components awaiting processing, telephone calls, airplanes, indeed any discrete item.

All queuing situations can be divided into four basic components. These are:

1. The arrivals distribution.

2. The queuing discipline.

3. The number of servers.

4. The service time distribution.

When queuing theory is applied to a particular situation, assumptions must be made, and justified, for each of the four components. The basic components of a waiting line process, or queue, are depicted in figure 8.1.

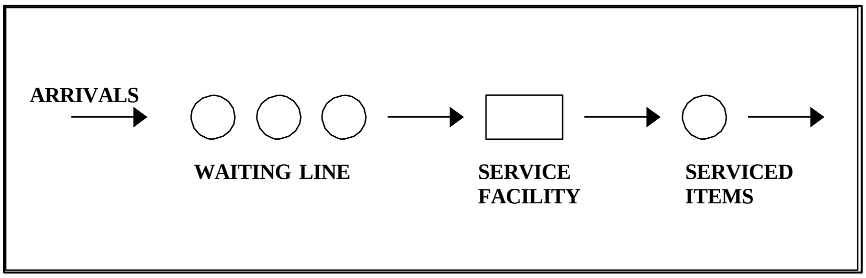



**Figure 8.1:**
**The Basic Components of a Queue or Waiting Line**

## 8.2 FUNDAMENTAL ASSUMPTIONS OF QUEUING THEORY

With regard to Figure 8.1, the assumptions essential to the development of the mathematical theory of queues are:

a. *Distribution of Arrivals:*
This component is concerned with how items (people, telephone calls, cars
etc.) arrive into the queuing situation. Possible variations include:

- Groups of items arriving together. For example, a bus-load of tourists arriving at a shop.

- Items arrive singly and with a fixed time between arrivals. For example, finished cars arriving off a production line.

- Items arrive singly and the time between arrivals is variable and best described by a random variable.

The third situation above is the one of interest here. Considerable empirical analysis has been done on the best random variable to model the time between arrivals in queuing situations. For the models we consider, the number of arrivals per unit time is assumed to follow a Poisson distribution. This assumption is underpinned by empirical research which indicates that, for many queuing processes, the arrival pattern is, indeed, Poisson distributed. The Poisson distribution is mathematically expressed as:

$$P(x) = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

Where:

x = the number of arrivals
P(x) = the probability of x arrivals
$\lambda$ = the mean arrival rate per time unit
e = the base of natural logarithms = 2.71828…….
x! = x (x − 1) (x − 2) …..3.2.1 (read "x factorial")

This distribution depends only on one parameter $\lambda$ which is both the mean and variance. Thus if we know the value of $\lambda$ we have a complete description of the arrivals process. The distribution is an example of a discrete distribution being used to model a discrete event namely unit arrivals. It is skewed (tail is to the right) to the right and becomes more symmetric (like the normal distribution) as $\lambda$ increases. We can calculate the probability of x arrivals in a given period by simply inserting the value of x into the formula together with the value of $\lambda$.

*b.* *Distribution of Service Time*:
The time taken to service an item may be constant, but, more likely, the time will vary. In the latter case a random variable will be used to describe the service time distribution. The statistical distribution of service time is assumed to be the *Negative Exponential* distribution and there is some empirical evidence to underpin this assumption. However, the correlation is not as strong as that for arrivals. The negative exponential distribution is mathematically expressed as:

$f(t) = \mu e^{-\mu t}$

Where:
t = the service time (which is a continuous random variable taking values from 0 to possibly infinity theoretically)

f (t) = the probability associated with t.

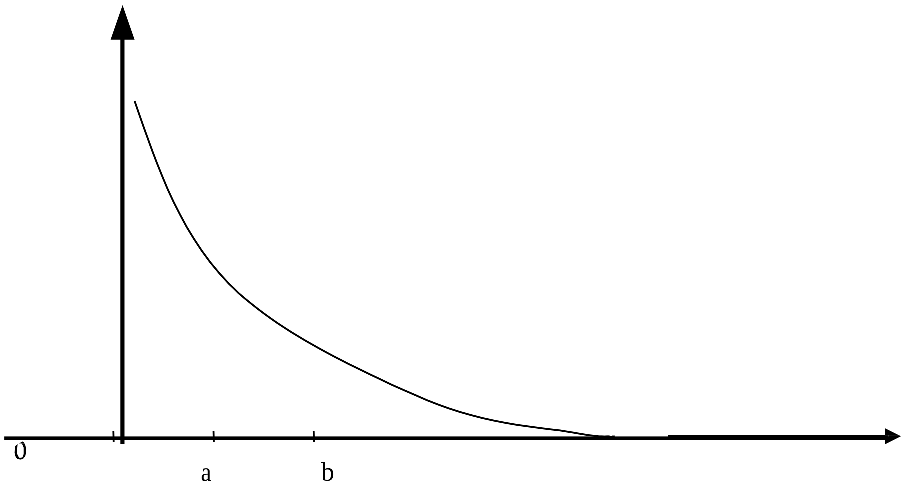
μ = the mean service rate (therefore 1/ μ = the mean service time)

e = the base of the natural logarithm

Using the wrong value for μ is a common source of error in applying queuing theory formulae.

This distribution depends only on one parameter μ which is both the mean and standard deviation of service times. Thus if we know the value of μ we have a complete description of the service process. The distribution is an example of a continuous distribution being used to model a continuous random variable namely service time. Its tail is to the right approaching the horizontal axis as service times get larger. The total area under the curve is of course 1.0 We can calculate the probability of service times being between reference values by simply inserting the value of t into the formula together with the value of μ. The following results are easily developed:

Prob(service time ≥b) = $e^{-\mu b}$ this is area above b
Prob(service time ≤a) =1- $e^{-\mu a}$ this is area below a
Prob(a≤ service time ≤b) = $e^{-\mu a}$ - $e^{-\mu b}$ this is area between a, b




As μ increases the curve above becomes steeper with the tail tapering off more quickly.

Another interesting result is the connection between the Poisson and negative exponential distributions, namely; if the arrivals are Poisson then the inter-arrival time is negative exponential as a consequence. Hence these two distributions are intrinsically linked.

c. *The Queue Discipline*:
This component is concerned with what happens between the arrival of an item requiring service and the commencement of the service. The queue discipline is a decision rule as to how the customers will be serviced. The usual rule is First In, First Out (FIFO). Other disciplines are possible and these include Last In, First Out (LIFO) and priority methods based on urgency or importance.

d. *The Calling Population*:
The *calling population* is the source of arrivals. Generally, the basic queuing models assume an *infinite* calling population - people coming to shop at a supermarket, for example. However *finite* populations can exist - machines in a machine shop, for example. Queuing models for the finite population have been developed, but this unit will concentrate on infinite populations.

e. *Queue Length*:
In queuing models, the *service rate* must be greater than the *arrival rate* that is:

$$\lambda / \mu < 1$$

If this were not the case, then items would arrive for service more quickly than they can be dealt with and the queue would grow longer and longer over time i.e. become infinite in length. Hence, as depicted in Figure 8.2, as the ratio (λ/μ) approaches 1 the expected queue length will approach infinity.

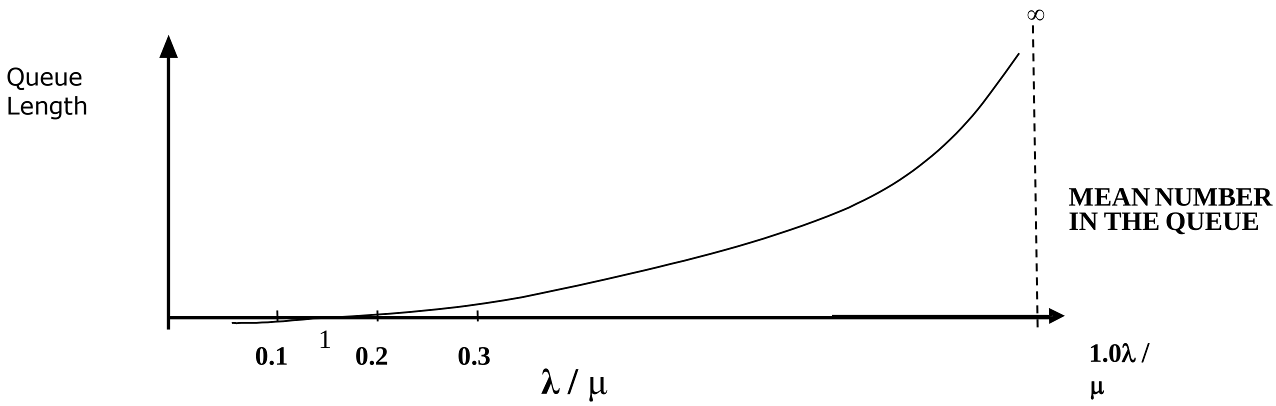



**Figure 8.2**
**Queue Length and λ / μ**

In practical situations the queue length is finite as customers will by-pass service stations, ATM machines etc if the queue is deemed to be too long. Thus queue length has economic implications.

f. *Number of Servers*:
In queuing theory a distinction is made between situations that have a single service area (e.g. a server or counter) and situations that have more than one service area. With the single server model, all arrivals must be served by the one server. In contrast, with the multi-server models, only a proportion of arrivals are served by a particular server.

The picture of the single server situation is as follows:

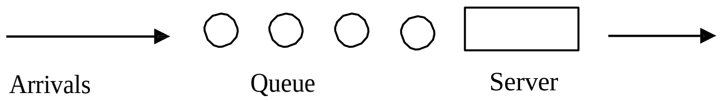



**Figure 8.3**
**A Single Server Queuing System**

With the multi-server model two situations are possible. These are

(i) Multi-server/Multi-queue

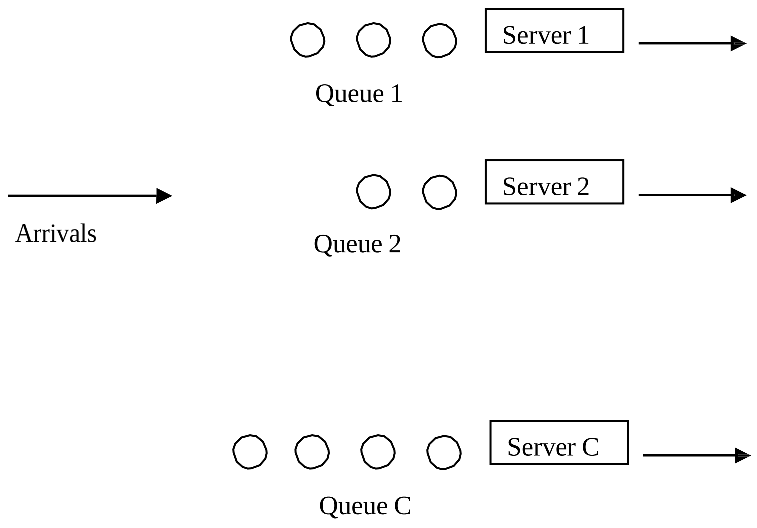



**Figure 8.4**
**A Multi- Server/ Multi Queue System**

The above situation occurs at supermarket checkout desks where a queue can form at each desk. An arrival must decide which queue to join. (This leads to another branch of queuing theory where different policies are used to determine which queue to join in a multi-queue system e.g. the "join the shorter queue" policy).

(ii) Multi-server/Single queue

This situation is used at some banks and department stores. A single queue is formed, and the person at the top of the queue is served by the first cashier to become available. This system is also used by some supermarkets for customers who wish to purchase a small number of items.

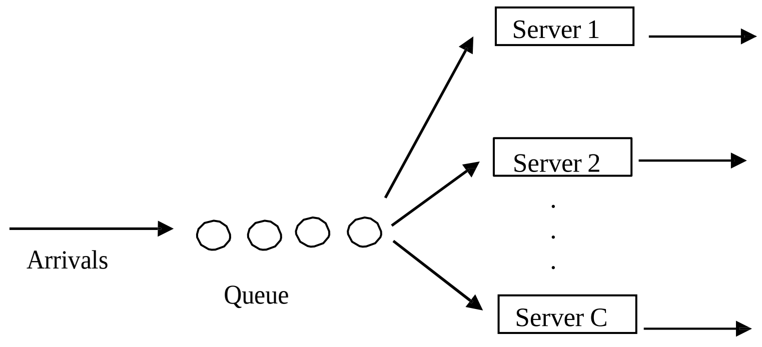



**Figure 8.5**
**A Multi- Server/ Single Queue System**

g. *Steady State Operation*:
Almost all basic queuing models assume a steady-state condition: after the system has been in operation for a period of time its

operating characteristics, such as queue length and average waiting time, will migrate towards constant average values. These are known as the steady state values

## 8.3 THE MATHEMATICS OF WAITING LINE MODELS

We shall start the analysis by considering the simplest case of the *Single-Channel Single-Phase* queuing model. The number of channels in a queuing process is the number of parallel servers available to service arrivals. The number of phases is the number of sequential service steps each individual arrival must go through. For the single-channel single-phase model, we shall assume that:

1. The arrival pattern follows a Poisson Probability Distribution with $\lambda$ = the mean number of arrivals per time period.

2. The service pattern follows an exponential probability distribution with $\mu$ = the mean number of services per time period and $\lambda / \mu < 1$.

3. The queue discipline is first in first out or FIFO.

4. The system achieves a steady-state condition over finite time.

Figure 8.1 illustrates such a system.

It can be proved mathematically that:

$$P_0 = \left(1 - \lambda/\mu\right)$$

This then yields the general result:

$$P_n = \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^n$$

Let us define the following:

L = the expected (mean) number of units in the system (i.e. in the queue or being serviced).

$L_q$ = the expected number in the queue

W = the expected time in the system

$W_q$ = the expected time in the queue

Recall that the *expectation function* of the random variable x is defined to be:

$$E(x) = \bar{x} = \text{Mean} = \sum_{n=0}^{\infty} x\,P(x)$$

Therefore the expected number of units in the system is

$$L = \sum_{n=0}^{\infty} n \,.\, P_n$$

$$= \sum_{n=0}^{\infty} n \left(1-\lambda/\mu\right)\left(\lambda/\mu\right)^n$$

Algebraic manipulation of this relationship yields:

$$L = \frac{\lambda}{\mu-\lambda}$$

Since $1 / \mu$ is the expected time in service:

$$L_q = L - (\lambda/\mu) = \frac{\lambda^2}{\mu(\mu-\lambda)}.$$

There are λ arrivals per unit time, and the time each of these spends in the system on average is W. Therefore the total time in the system for the λ arrivals is λW. Suppose that λ = 3 customers arrive, and that each spends an average of W = 2 hours in the system. Then the total number of customer hours is 6. It follows that there are 6 customers on average in the system: customers 1, 2 and 3 in their second hour in the system and customers 4, 5 and 6 in their first hour. This reasoning gives L = λW.

Using the same reasoning gives the relationship between the average number in the queue and the waiting time in the queue:
$L_q = \lambda W_q$.

These two relationships, in conjunction with the formulae derived earlier for L and $L_q$, give:

$$W = \frac{1}{\mu-\lambda}$$

$$W_q = \frac{\lambda}{\mu(\mu-\lambda)}$$

Finally, if P is the service facility utilisation factor (the proportion of time that the server is in use) and if I is proportion of the time the server is idle, then we have

$P = \lambda / \mu$

$I = 1 - \lambda / \mu = P_0$

All of these measures define the *Single-Line Single-Phase* system (also known as the *Single Server Queuing Model*) and it should be noted that they are all expressions involving $\lambda$ and $\mu$ (the arrival and departure rates for the system).

L, $L_q$, W and $W_q$ are known as the ***Operating Characteristics*** of the system and it should be noted that $\lambda$ and $\mu$ are needed for their computation.

A final point to note is that all of the above analysis only applies if $\mu > \lambda$ (i.e. ($\lambda/\mu < 1$).

***Example 1:***
Let us assume that we are examining a drive-in bank teller window. Measurements have been carried out which establish Poisson arrivals and exponential service with the following values:

*Arrivals:* $\lambda$ = 20 / hr or 1 every 3 mins

*Services:* $\mu$ = 30 / hr or 1 every 2 mins.
$P = \lambda / \mu = 20/30 < 1$

Thus we have the following Operating Characteristics:

$P_0 = 1 - \lambda / \mu$

$= 1 - 20/30 = 0.33$ 33% idle time.

$L = \dfrac{\lambda}{\mu - \lambda} = \dfrac{20}{30-20}$ = 2 people / cars on average in the system

$L_q = \dfrac{\lambda^2}{\mu(\mu-\lambda)} = \dfrac{20^2}{30(30-20)}$ = 1.33 people / cars in the waiting line

$W = \dfrac{1}{\mu-\lambda} = \dfrac{1}{30-20} = \dfrac{1}{10}$ hrs = 6 mins, on average, in the system

$W_q = \dfrac{\lambda}{\mu(\mu-\lambda)} = \dfrac{20}{30(30-20)}$ = 4 mins, average waiting time per customer

The above information could be used in a number of ways, including the financial analysis of the systems as follows:

***Example 2***

Suppose that, in Example 1, the service time can be reduced by engaging a more experienced teller who can service 40 customers per hour. However, this teller will cost €15 per hour as against €8 per hour for the current teller. The drive-in bank is in operation for 8 hours a day. The bank has estimated that customer waiting time in the queue is costing €2 per minute in terms of customer dissatisfaction. Should the existing teller be replaced by the more experienced teller?

To answer this question, we need to make our decision on the basis of two costs. The first is the cost of customer waiting time in the queue, which we can derive by calculating $W_q$. The second is the cost of the server being idle, where we take the value of lost productivity to be equal to the wage of the teller during the time that he or she is idle.

With the existing teller, from Example 1 we know that the average waiting time is $W_q$ = 4 minutes, and the server (teller) is idle for $1-\lambda/\mu$ = 1-20/30 = 0.33 or 33% of the time.

If we employ the more experienced teller, the waiting time is

$W = \dfrac{\lambda}{\mu(\mu-\lambda)} = \dfrac{20}{40(40-20)} = \dfrac{1}{40}$ hours,

or 1.5 minutes average waiting time per customer, and the server is idle for $1-\lambda/\mu$ = 1-20/40 = 0.5, or 50% of the time.

Over the 8-hour day, the expected number of customers is 160. So the total queuing time with the current teller is 160 × 4 = 640 minutes. With the new teller, this reduces to 160 × 1.5 = 240 minutes. Moreover, in the first case, the service is idle for 8/3 = 2.64 hours, while in the second it is idle for 8/2 = 4 hours.

The table summarises the costs.

| | ***Example 1*** | ***Example 2*** |
|---|---|---|
| | Current teller | Experienced teller |
| Customer Waiting Time | 640 × 2 = €1,280 | 240 × 2 = €480 |
| Server Idle Time | 2.64 × 8 = €21.20 | 4 × 15 = €60 |

Since the reduction in customer waiting cost (€1280-€480 = €800) greatly exceeds the increased cost of the server idle time (€60.00-€21.20 = €38.88), the experienced teller should be engaged.

In answering this question, we assumed that the value of lost productivity while the server is idle is equal to the wages of the teller during that time. If this is not so, then we can simply compare the total cost of the two server levels. In this case, the increased labour cost is 8 × 15 - 8 × 8 = €56, so we reach the same decision as before: take on the new teller.

- ▶ See video **V34_Queueing_1**, which involves the calculation of the operating characteristics for a single-server queue. https://goo.gl/4Zp49Y
- ▶ See the video **V35_Queueing_2**, which solves a probability problem concerning a single-server queue. https://goo.gl/7wm4nr

## *SAQ 1*

*The ticket booth on a college campus is operated by one person, who sells tickets for a football match. The ticket seller can serve 12 customers per hour (exponentially distributed). An average of 10 customers arrive to purchase tickets every hour.*

*Determine the average time a ticket buyer must queue to buy a ticket and the proportion of time the ticket seller is busy.*

## *SAQ 2*

*A team of 15 men is employed to unload lorries at a terminal. The team works a 6-hour day during which, on average, 36 lorries arrive (i.e. an hourly average of 6 lorries). It is estimated that it takes on average 7.5 minutes to unload one lorry with the team acting as a single unit.*

*It has been estimated that the cost of keeping lorries waiting is €20 per hour.*

*Members of the team are each paid €10 per hour.*

*Management is considering a proposal to increase the size of the team to 20 men. With 20 men in a team it is estimated that the average time to unload a lorry will fall to 5 minutes. Advise management.*

## *SAQ 3*

*A Cinema has customers arriving, on average, 60 per hour, and in accordance with a Poisson distribution. The clerk at the ticket window can complete 90 services every hour and in accordance with an exponential distribution. Any queues are dealt with on a first time, first served basis, customers coming from an infinite population and there is no limit to queue length. Compute the queuing system's operating characteristics for a steady state condition.*

## *SAQ 4*

*Suppose the management of the cinema is able, by a re-arrangement of the Clerk's workplace, to increase the service to 120 per hour. In doing so the management expects the clerk to look for a productivity increase of 50% on the hourly rate which is currently €8 per hour. Management has further estimated the cost of customer waiting time to be €0.30 per minute in terms of customer dissatisfaction. Advise management as to whether the productivity bonus of 50% is justified. Assume the Clerk is effectively working 6 hrs per day.*

## 8.4 Single Server, Finite Queue Model

In many queuing systems the length of the queue may be limited by the physical area in which the queue forms, so that only a limited number of customers can enter the queue. When the maximum queue length is reached, potential arrivals cannot enter the system and they go elsewhere for service.

Practical applications of this model include:

- A telephone switchboard with a limited number of incoming lines. This means that only a certain number of calls can be placed on hold.

- A petrol station fronting onto a busy main road. When the available queuing space in the garage forecourt is full, potential customers cannot gain access to the system and must go elsewhere.

The model described in this section is identical to the single server model with the additional restriction that the system has a maximum size of M, (the maximum queue length is M-1). Thus, with this model the possible number of items in the system are 0, 1, 2, 3…M and the only non-zero probabilities are $P_o$, $P_1$, $P_3$,…, .$P_M$

The assumptions behind the single serve, finite queue model are:

- The number of arrivals per unit time is Poisson with mean $\lambda$
- Single server.
- Service time is exponential with rate $\mu$
- Maximum system size of M.

Note that state M is the highest state that this system can be in. In state M the system is full and no further arrivals are allowed.

The following formulae can be proved for this case:

$$P_0 = \frac{1-\lambda/\mu}{1-(\lambda/\mu)^{M+1}} \text{ if } \lambda \neq \mu$$

$$P_0 = \frac{1}{M+1} \text{ if } \lambda = \mu$$

$$P_i = \left(\frac{\lambda}{\mu}\right)^i P_0 \text{ with } i \leq M$$

$P_M$, the probability of M items in the system, is the proportion of potential arrivals that is lost to the system. When the system contains M items the system is full, and arrivals are refused entry. This situation occurs with probability $P_M$.

Consider the following example.

***Example 3***
A petrol station is situated on a busy road and analysis would suggest that on average 12 cars per hour could stop at the station. The average profit per petrol sale is €3. The petrol station has a single petrol pump, and it takes, on average, 4 minutes to fuel a car. The physical layout of the garage forecourt prevents more than 3 cars being in the garage at the same time.

Calculate the potential profit that is lost to this garage every hour.
The single server, finite queue model is appropriate in this example. The parameters of the model are:

$\lambda$ = 12 per hour
$\mu$ = 15 per hour
M = 3

If all 12 potential customers per hour could enter the system, then the potential profit per hour is:

12 x €3 = €36 per hour

Because of the finite system, all of the 12 potential customers may not be able to enter the garage. The potential customers who arrive when the system is full go elsewhere and their potential profit is lost.

$P_M$ = Probability that the system is full. Thus

$\lambda P_M$ = number of lost customers per hour.

For this example

$$P_0 = \frac{1-12/15}{1-(12/15)^4} = 0.339$$

$$P_1 = \left(\frac{\lambda}{\mu}\right) P_0 = 0.271$$

$$P_2 = \left(\frac{\lambda}{\mu}\right)^2 P_0 = \left(\frac{\lambda}{\mu}\right) P_1 = \frac{12}{15}(0.271) = 0.217$$

$$P_3 = \left(\frac{\lambda}{\mu}\right)^3 P_0 = \frac{12}{15}(0.217) = 0.173$$

Note that the four probabilities above sum to 1. These are the only 4 states that the petrol station can be in – no customers, 1, 2 or 3 customers.

$P_3 = 0.173$. Thus, 17.3% of the time the system is full and arrivals are turned away.

$\lambda P_3 = 12 \times 0.173 = 2.1$ customers are lost every hour.

The expected lost profit per hour is therefore €3 × 2.1 = €6.3.

### *OPERATING CHARACTERISTICS FOR THE SINGLE SERVER, FINITE QUEUE MODEL*

The operating characteristics for the model with parameters $\lambda$, $\mu$ and M are calculated as follows:

L = Average number in the system

$$= (0 \times P_0) + (1 \times P_1) + (2 \times P_2) + \ldots + (M \times P_M)$$

$$= \sum_{i=1}^{M} i \times P_i$$

$W_q = L / \mu$

$W = W_q + 1 / \mu$

i.e. the average time in the system is comprised of the average time waiting to be served and average service time.

Applying these formulae to the example above, gives

$$L = 0 \times P_0 + 1 \times P_1 + 2 \times P_2 + 3 \times P_3$$

$$= 0 \times 0.339 + 1 \times 0.271 + 2 \times 0.217 + 3 \times 0.173$$

$$= 1.224 \text{ customers}$$

$W_q$, the average wait in the queue = $L / \mu$ = 1.224/15 = 0.0816 hours = 4.9 minutes

W, the total wait of a customer = $W_q$ + service time = 4.9 + 4 = 8.9 minutes

## *SAQ 5*

*The student services officer is in charge of advising undergraduates. To eliminate some of the congestion in her suite of offices, the associate dean has placed 4 chairs in the outer office, and established a rule that if a chair is not vacant, a student cannot wait to see her. The students arrive at a rate of 6 per hour (Poisson distributed), and the average time she spends advising a student is 15 minutes (exponentially distributed). She desires to put up a sign on the notice board, indicating how long students can expect to be in the office waiting and getting advice so that they can plan not to miss a class.*

*What length of time should be on the sign? What fraction of students arriving at the student services officer's outer office will have to leave and come back another time? In light of your answers to the above two questions, what is your evaluation of the adequacy of this system?*

## *SAQ 6*

*A restaurant can seat five customers and can serve fifteen customers in one hour. The potential arrival rate for customers is also fifteen per hour. What is the average wait for service? If the average cost of a meal is €30.00 calculate the lost revenue due to potential customers going elsewhere.*

### 8.5 MULTIPLE – CHANNEL QUEUES

The two models described above are appropriate when the service is provided by a single server. In many practical situations, more than one server is available in the service area. This section analyses those situations which utilise more than one server. It is important to note that the theory of this section is only appropriate if a single queue is used to order the arrivals waiting area. A *Multiple Channel Single Phase* queuing system is depicted in Figure 8.6.

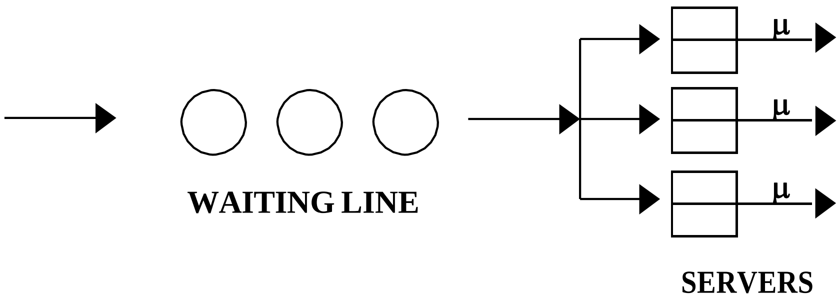



**Figure 8.6:**
**A Multiple Channel Single Phase Queuing System**

Again assuming Poisson arrival and exponential service time distributions*, Operating Characteristics* can be computed using mathematics. The operating characteristics for a multiple channel single phase queuing system are:

$$P_0 = \frac{1}{\left[\sum_{n=0}^{s-1} \frac{\left(\lambda/\mu\right)^n}{n!}\right] + \left[\frac{\left(\lambda/\mu\right)^s}{s!\left(1 - \lambda/s\mu\right)}\right]}$$

Where s is the number of servers, and s.μ > λ

$$P_n = \frac{\left(\lambda/\mu\right)^n}{n!} P_0 \text{ where } n \leq s$$

$$P_n = \frac{\left(\lambda/\mu\right)^n}{s!s^{n-s}}$$ $P_0$ where n > s

$$P = \frac{\lambda}{s.\mu}$$ is the service utilisation factor.

$$L_q = \frac{P_o\left(\lambda/\mu\right)^s P}{s!\left(1-P\right)^2}$$

$L = L_q + \lambda / \mu$

$W_q = L_q / \lambda$
$W = W_q + 1 / \mu$

As can be seen, the calculations for $P_0$ are quite complex. These can be done with the assistance of a computer/spreadsheet or using tables that give values of $P_0$ for selected values of s and the ratio $\lambda/\mu$. These tables are contained in many texts on Management Science, thus the computation of the Operating Characteristics is considerably simplified.

When the level of complexity is such that mathematical analysis is extremely difficult *simulation* is usually the preferred method of analysis. *Simulation* will be discussed in Unit 10.

***Example 3***
A three server queuing situation exists with, on average, 5 trucks arriving for service per hour. The average service time is 30 minutes.

The model parameters are therefore:

| | | | |
|---|---|---|---|
| $\lambda$ | = | 5 per hour | |
| $\mu$ | = | 2 per hour | |
| S | = | 3 servers | [Check that $\lambda < S\mu$] |

Applying the formula gives:

$$P_0 = \frac{1}{\left[\sum_{n=0}^{2}\frac{\left(\frac{5}{2}\right)^n}{n!}\right]+\left[\frac{\left(\frac{5}{2}\right)^3}{3!\left(1-\frac{5}{6}\right)}\right]} = \frac{1}{6.625+15.625} = 0.0449$$

L = 6 trucks

Wq = 0.7 hours

W = 1.2 hours

Thus for this queuing system, an arriving truck will spend 0.7 hours (42 minutes) waiting for service to start. Once service has started, it will take on average 30 minutes, and, thus, the total time spent in the system by the truck is 42+30 = 72 minutes.

▶ See the video **V36_Queueing_3**, which compares the costs of single-server and multi-server queues.
https://goo.gl/cnUUz5

## *SAQ 7*

*A surgery has 2 dentists who see patients daily. Patients arrive at the surgery according to a Poisson distribution with a mean rate of 6 per hour. The time a dentist spends with a patient is exponentially distributed with a mean of 15 minutes. Patients wait in a waiting area until one of the 2 dentists is able to examine them. Since patients typically are nervous when they come to the surgery, the dentists do not believe it is good practice to have the patients wait longer than an average of 15 minutes. Should this surgery hire a third dentist, and, if so, will this reduce the average waiting time to less than 15 minutes?*

## *SAQ 8*

*An "As You Wait" shoe repair shop employs three operatives. On average, it takes 12 minutes to repair a shoe. It is estimated that an average 12 customers per hour require service at the counter.*

*Calculate the average wait to get a shoe repaired.*

## *SAQ 9*

*Articulated lorries arrive at a supermarket to be unloaded. The supermarket has three unloading bays and the fully-loaded lorries dock at one of the three spaces when they become available. The arrival rate follows a Poisson distribution with an average rate of 6 per hour. The service distribution is exponential and the average rate of unloading is 3 per hour. There is no significant limit on the number of lorries that can be waiting for unloading at any particular time. What are the operating characteristics of this system under steady state conditions?*

## 8.6 The Implied Value of Customer Waiting Time

Occasionally, management decides to spend money to improve the speed at which customers can be served in a service facility. This implies that the waiting time of customers has an implied value which is used to offset the additional costs incurred.

In the following example, management decides to spend additional money so as to reduce average customer waiting time. By comparing the reduction in waiting time with the additional costs spent, a value is obtained for customer waiting time.

***Example 4***

Management currently use a two channel system to service customers. The average arrival of customers is 12 per hour and it takes, on average, 6 minutes to service a customer. The total cost of the facility is €30 per hour.

A decision is made to decrease the average service time to 4 minutes by increasing the hourly cost from €30 to €45.

The implied value of customer waiting time is calculated as follows:

| Service time 6 minutes | Service time 4 minutes |
|---|---|
| S = 2 | S = 2 |
| $\lambda$ = 12 | $\lambda$ = 12 |
| $\mu$ = 10 | $\mu$ = 15 |
| $P_0$ = 0.25 | $P_0$ = 0.428 |
| W = 0.156 hours | W = 0.079 hours |

Cost difference = €45 - €30 = €15 per hour
= €(15/12) per customer
= €1.25 per customer

The difference in waiting timer per customer is (0.156 – 0.079) = 0.077 hours.

Thus 0.077 hours is equivalent to €1.25, which implies that 1 hour is equivalent to €1.25/0.077 = €16.23.

## *SAQ 10*

*You have been asked to consider three systems for providing service when customers arrive with an average arrival rate of 24 per hour.*

***Option 1 A single channel with an average service time of 2 minutes.***
*This service costs €5 per customer with a fixed cost of €50 per hour.*

***Option 2: 2 channels in parallel.***
*The average service time of a customer is 4 minutes. This system costs €4 per customer with a fixed cost of €30 per channel per hour.*

***Option 3: 3 channels in parallel.***
*The average service time of a customer is 6 minutes. This system costs €3 per customer with a fixed cost of €25 per hour per channel.*

*You are required to calculate:*

*(a) The average time a customer spends in the system for each of the 3 options.*

*(b) The implied value to the firm of customer waiting time if the firm is indifferent between options 2 and 3.*

### SUMMARY

In this unit several types of waiting line situations were examined in detail. Waiting lines are pervasive in business and dynamic systems generally and as such a dedicated branch of applied probability has developed. Applications and instances vary from system/machine failure to accident services response times to customers arriving in queues, to incoming call traffic in customer call centres etc. Analysing a waiting line requires certain assumptions being made particularly regarding the key components. These components being:

- The arrivals distribution.
- The queuing discipline.
- The number of servers.
- The service time distribution.

The starting point for most analysis is the adoption of reasonable distributions to describe the arrival and service processes. The adopted distributions need to balance empirical support and theoretical justification and tractability. The classic assumption of Poisson arrivals was adopted and the simplifying assumption of the negative exponential was also assumed to apply. Other assumptions revolve around the nature of the calling population, the number of servers/channels and the queue discipline.

Whatever the specific waiting line being examined the analysis is typically concerned with developing values for the operating characteristics of the queue. These values are themselves decision support tools allowing the decision maker to monitor and if necessary adjust the performance of the waiting line system if it is judged economical or justified. These characteristics are average performance values across a range of simple metrics. Typically, one will wish to develop estimates for the following:

$L$ = the mean number in the system
$L_q$ = mean number in the queue
$W$ = the mean time in the system
$W_q$ = the mean waiting time
$P_0$ = the probability of zero units in the system
$P_n$ = the probability of n units in the system
$P$ = the service facility utilisation factor

*It was seen that in each case that calculation of these quantities was relatively straightforward particularly when the quantities were seen to be interdependent for example:*

$$L = L_q + \lambda / \mu$$

$$W_q = L_q / \lambda$$

$$W = W_q + 1 / \mu$$

*Key to calculating all the quantities above was knowledge of the values of the key inputs, namely the arrival rate* $\lambda$ and the service rate $\mu$. *Various waiting line configurations are possible and to facilitate a classification system the* Kendall's notation can be adopted:
For example the simplest model is M / M / 1 → Poisson Arrivals, Exponential Service and 1 Channel. More generally retaining the arrival process as Poisson and relaxing the service process to one of an undefined distribution with known mean and variance constitute an M / G / 1 model→ Poisson Arrivals, Arbitrary Service and 1 channel. Other systems might include M / D / 1 → Poisson Arrivals, Deterministic Service and 1 channel and M / M / S → Poisson Arrivals, Exponential Service and S channels. It is relatively easy to appreciate the variety and complexity of waiting line configurations that are possible by varying the various components of the system from the arrival process to the service process to the queue discipline and so on. This complexity presents analytical challenges and consequently simulation based approaches are typically invoked. There is an abundance of waiting line theory with supporting software packages and in the case of simulation various programming languages can be efficiently utilised.

## SUMMARY OF STEPS

- ❑ Identify the context as one of a waiting line.
- ❑ Examine the various components and adopt assumptions/descriptions of the various components. Particular attention is needed for modelling the two key inputs, namely; the arrival process and the service process distribution.
- ❑ Determine the values of $\lambda$ and $\mu$ taking care to express them in the same units (rates per unit time) and if useful classify your system using the Kendall notation and deploy the corresponding formulae.
- ❑ Calculate the operating characteristics of the waiting line and any other metrics of interest, these are typically average values and are performance indicators of the system.
- ❑ Use these outputs to facilitate appraisal of the system performance. If economic implications are present, then explore various options and inputs under your control such as the service rate and conduct sensitivity analyses. Typically, as service rates increase the service cost does whilst the waiting costs if quantifiable tend to be reduced. The total expected costs can be compared under different service scenarios and optimal service rates identified.
- ❑ Implement and review the decisions preferably with clear report generation and interactive spreadsheet or simulation packages supporting the analysis and decision prescriptions.

## FINAL COMMENTS

The student should seriously consider developing a spreadsheet for the interactive calculations of the operating characteristics and economic analyses. Thus by varying the inputs the outputs can be interactively reviewed, charted and form the basis of efficient decisions based on sound analysis. Spreadsheet add-ins allow for simulation of waiting lines and these are also widely available and worth applying for more complex waiting line analyses.

## SOME USEFUL REFERENCES AND RESOURCES

http://www.egr.msu.edu/~ziad/qt/qt.html
http://people.brunel.ac.uk/~mastjjb/jeb/or/queue.html
http://new-destiny.co.uk/andrew/past_work/queueing_theory/
http://www2.uwindsor.ca/~hlynka/qsoft.html

## ANSWERS TO SAQS

### *SAQ 1*

- *Arrival rate is 10 per hour* $\lambda$ *= 10/hour*
- *Average service rate is 12 per hour* $\mu$ *= 12/hour.*

$$W = \frac{1}{\mu - \lambda} = 1/(12 - 10) = 1/2 \text{ hour}$$

$$= 30 \text{ minutes.}$$

$$W_Q = \frac{\lambda}{\mu(\mu - \lambda)} = \frac{10}{12(12-10)} = 5/12 \text{ hours}$$

$$= 25 \text{ minutes.}$$

*Thus, an arrival will spend, on average, 30 minutes purchasing a ticket. This 30 minute comprises an average wait of 25 minutes in the queue, and 5 minutes being served.*

*The proportion of time the ticket seller is idle is calculated through* $P_0$*.*

$$P_0 = (1 - \lambda/\mu) = 1 - 10/12$$

$$= 0.167$$

*Thus, the ticket seller is idle 16.7% of the time, and is busy the remaining 83.3% of the time.*

## *SAQ 2*

| ***15 man team*** | ***20 man team*** |
| --- | --- |
| *$\lambda$ = 6 per hour*<br>*$\mu$ = 8 per hour* | *$\lambda$ = 6 per hour*<br>*$\mu$ = 12 per hour* |
| $W = \dfrac{1}{\mu - \lambda}$ *= 1/2 hour* | $W = \dfrac{1}{\mu - \lambda}$ *= 1/6 hour* |
| ***Hourly Cost*** | ***Hourly Cost*** |
| *Lorry waiting time*<br>$\lambda \times W \times$ *€20* | *Lorry waiting time*<br>$\lambda \times W \times$ *€20* |
| *6* $\times$ $^1/_2$ $\times$ *€20 = €60* | *6* $\times$*.167* $\times$ *€20 = €20* |
| *Team Cost*<br>*15* $\times$ *€10 = €150* | *Team Cost*<br>*20* $\times$ *€10 = €200* |
| *Total Hourly Cost €210* | *Total Hourly Cost €220* |

*The 15 man team is more economical by €10 per hour.*

## *SAQ 3*

- *The mean arrival rate is: Poisson.* $\lambda = 60/60 = 1/Min$
- *The mean Service rate is= Exponential* $\mu = 90/60 = 1.5/Min$
- *Thus* $\lambda / \mu = 1/1.5 = 2/3 < 1$
- *The mean expected number in the system is:*

$$L = \frac{\lambda}{(\mu - \lambda)} = 1/(1.5 - 1) = 2 \text{ persons}$$

- *The mean number in the queue is:*

$$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)} = \frac{1^2}{1.5(1.5 - 1)} = 4/3 \text{ persons}$$

- *The mean time in the system is:*

$$W = \frac{1}{(\mu - \lambda)} = 1/(1.5 - 1) = 2 \text{ mins}$$

- *The mean waiting time is:*

$$W_q = \frac{\lambda}{\mu(\mu - \lambda)} = \frac{1}{1.5(1.5 - 1)} = 4/3 \text{ mins}$$

- *The probability of zero customers in the system is:*

$$P_0 = (1 - \lambda/\mu) = 1 - 1/(3/2) = 1 - 2/3 = 0.33$$

## *SAQ 4*

*Look on the situation as a Case I and Case II situation and compute $W_q$ and I for each case.*

***Case I***

*Current service rate ($1/\mu$) = 1/(3/2) = 2/3 customers per min.*

$$W_{q\ I} = \frac{\lambda}{\mu(\mu-\lambda)} = \frac{1}{1.5(1.5-1)} = 4/3 \text{ mins}$$

$I_I$ = $(1 - \lambda/\mu)$ = 1 – 1/(3/2) = 1- 2/3 = 33% idle time

*As can be seen, the calculations for $P_0$ are quite complex. However, to help simplify the calculations of these multi-channel expressions, tables are available that give values of $P_0$ for selected values of s and the ratio $\lambda/\mu$.*

***Case II***

*Current service rate ($1/\mu$) = 120/60 = 2 customers per min.*

$$W_{q\ II} = \frac{\lambda}{\mu(\mu-\lambda)} = \frac{1}{2(2-1)} = 1/2 \text{ mins}$$

$I_{II}$ = $(1 - \lambda/\mu)$ = 1 – 1/2 = 1/2 = 50% idle time

*Using these values of $W_Q$ and I we can summarise the costs for one day as follows:*

*Working day is 6 hrs = 360 mins*

*$\lambda$ = 1/min, thus expect 360 customers/day*

| | ***Case I*** | ***Case II*** |
|---|---|---|
| ***Customer Waiting Time cost*** | *360 × 4/3 × 0.3 = €144* | *360 × 1/2 × 0.3 = €54* |
| ***Server Idle Time cost*** | *6 × 0.33 × 8 = €16* | *6 × 0.5 × 12 = €36* |

*The reduction in the waiting time cost is 144-54 = €90.*
*The increase in the server idle time cost is 36-16 = €20.*
*As the reduction in customer waiting time cost exceeds the increase in server idle time cost, the productivity increase is justified.*

## *SAQ 5*

*The single server, finite queue model is appropriate with parameters*

| | | |
|---|---|---|
| $\lambda$ | = | *6 per hour* |
| $\mu$ | = | *4 per hour* |
| *M* | = | *5* |

*The possible probabilities for this situation are* $P_0$, $P_1$, $P_2$, $P_3$, $P_4$, *and* $P_5$. *Their values are:*

$$P_0 = \frac{1-6/4}{1-(6/4)^6} = 0.048$$

| | | |
|---|---|---|
| *P1* | = | *.072* |
| *P2* | = | *.108* |
| *P3* | = | *.162* |
| *P4* | = | *.244* |
| *P5* | = | *.365* |

*For this situation* $P_5$ *= .365. Thus 36.5% of the time all 4 chairs are occupied and a student will have to come back at another time.*

*The average time spent by a student waiting and getting advice is W. W is calculated as follows:*

$$W = W_Q + \text{service time}$$

$$= L / \mu + \text{service time}$$

$$L = \sum_{i=1}^{5} i \times P_i = 3.576 \text{ students.}$$

*Thus W* = $(1/4 \times 3.576 + 1/4)$ *hours*

= *1.14 hours.*

*A wait of 1.14 hours seems totally unacceptable from the students' point of view.*

## *SAQ 6*

*The single server finite queue model is appropriate in this application with*

$\lambda$ = 15 per hour
$\mu$ = 15 per hour
$M$ = 5

*Applying the formulae from the text suggest that*

$P_i = 1/6$ for $i = 0, 1, 2, \ldots, 5$

*L is the average number in the restaurant*

$$L = \sum_{i=0}^{5} i \times P_i = 2.5 \text{ customers}$$

$W_q$ = Average wait for service = $L / \mu$ = 2.5 / 15 = 0.167 hours = 10 minutes

*Lost Revenue:*
*The space limitations in the question suggest that when the restaurant contains 5 customers, all further customers go elsewhere. The restaurant will contain 5 customers with probability* $P_5 = {}^1/_6$.

*The average arrivals rate per hour is* $\lambda = 15$, *of which* ${}^1/_6$ *go elsewhere because the system is full. This implies that* ${}^1/_6$ *of 15, (i.e. 2.5) customers do not gain access to the restaurant and go elsewhere. The lost revenue from these is calculated as follows.*

*The average revenue per customer is €30, thus, on average, € 75 is lost each hour due to the fact that customers go elsewhere.*

## *SAQ 7*

*The current situation in the dentist's surgery can be modelled as a multi-server queuing model with parameters as follows:*

| | | |
|---|---|---|
| $\lambda$ | = | *6 per hour* |
| *s* | = | *2 dentists* |
| $\mu$ | = | *4 per hour (average time of 15 minutes)* |

*$W_q$ is of concern in this problem and its value is calculated as:*

*$P_0 = 1/(1 + 1.5 + 4.5) = 0.143$*

$$P = \frac{\lambda}{s.\mu} = 0.75$$

$$L_q = \frac{0.143(6/4)^2 0.75}{2(1-0.75)^2} = 1.9304$$

*$W_q = L_q/\lambda = 1.9304/6 = 0.32$ hours, or 19 minutes.*

*Thus, with the current arrangement, patients will spend, on average 19 minutes waiting for a dentist.*

*This wait of 19 minutes is deemed unacceptable and the practice is considering hiring a third dentist. If a third dentist is hired then the parameter of the model becomes*

| | | |
|---|---|---|
| $\lambda$ | = | *6 per hour* |
| *S* | = | *3 doctors* |
| $\mu$ | = | *4 per hour* |

*Using the above parameters, the calculations leading to Wq are as follows:*
*$P_0 = 0.211$*
*$P = 0.5$*
*$L_q = 0.2374$*
*$W_q = L_q/\lambda = 0.2374/6 = 0.039$ hours, or 2.4 minutes.*

*Employing the third dentist reduces the average waiting time for a patient from 19 minutes to 2.4 minutes.*

## *SAQ 8*

*"As You Wait" shoe repair.*

*Multi-server*

| | | |
|---|---|---|
| $S$ | = | 3 |
| $\lambda$ | = | 12 per hour |
| $\mu$ | = | 5 per hour |

*Using the formula in the text we get*

$P_0 = 0.056$

*P = 12/15 = 0.8*

$$L_q = \frac{0.0562\left(12/5\right)^3 0.8}{6\left(1-0.8\right)^2} = 2.59$$

$W_q = L_q/\lambda = 2.59/12 = 0.216$ *hours.*

*Therefore, the average wait in the queue is 13 minutes, and the average total wait of a customer in the system is 13 + 12 = 15 minutes.*

## SAQ 9

$$P_0 = \frac{1}{\left[\sum_{n=0}^{s-1} \frac{\left(\lambda/\mu\right)^n}{n!}\right] + \left[\frac{\left(\lambda/\mu\right)^s}{s!\left(1-\frac{\lambda}{s\mu}\right)}\right]} = 1/(Lterm + Rterm)$$

$$Lterm = \frac{(\lambda/\mu)^0}{0!} + \frac{(\lambda/\mu)^1}{1!} + \frac{(\lambda/\mu)^2}{2!}$$

$$= \frac{1}{0!} + \frac{(6/3)^1}{1!} + \frac{(6/3)^2}{2!}$$

*= 1 + 2 + 2 = 5*

$$Rterm = \frac{(6/3)^3}{3!(1-6/9)} = \frac{8}{6\times1/3} = 4$$

*Thus*

*$P_0$ = 1/(5 + 4) = 1/9 = 0.111.*

*That is the system is idle 11.11% of the time.*

*Note that, if you wanted to avoid the calculations for $P_0$, the value of 0.111 can be read directly from the $P_0$ table in the Anderson, Sweeney and Williams textbook.*

$$L_q = \frac{P_o\left(\lambda/\mu\right)^s P}{s!\left(1-P\right)^2} = \frac{0.11\left(6/3\right)^3 \times 2/3}{3!\left(1-2/3\right)^2} = 0.88 \text{ trucks queuing}$$

*$L = L_q + \lambda/\mu = 0.88 + 6/3 = 2.88$ trucks in the system*

*$W_q = L_q/\lambda = 0.88/6 = 0.147$ hours or 8.80 minutes waiting time*

*$W = W_q + 1/\mu = 0.147 + 1/3 = 0.48$ hours or 28.8 minutes in the system.*

## *SAQ 10*

***Option 1:*** *Single-server model*

$\lambda$ = 24 per hour
$\mu$ = 30 per hour

*W = 1/(μ−λ) = 1/(30−24) = 1/6 hours or 10 minutes.*

*Hourly cost = €50 + 24 × €5 = €170*

***Option 2:*** *Multi-server model*

$S$ = 2
$\lambda$ = 24 per hour
$\mu$ = 15 per hour

$$P_0 = \frac{1}{\left[1 + \frac{24}{15}\right] + \left[\frac{(24/15)^2}{2\times(1/5))}\right]} = 0.111$$

$$P = \frac{\lambda}{s.\mu} = 24/30 = 0.8$$

$$L_q = \frac{0.1111(24/15)^2\,0.8}{2(1-0.8)^2} = 2.844$$

*$W_q = L_q/\lambda$ = 2.844/24 = 0.1185 hours, or 7.11 minutes.*

*W = 4 minutes + 7.11 minutes = 11.11 minutes*

*Hourly Cost = €30 × 2 + 24 × €4 = €156*

***Option 3:*** *Multiserver Option*

$S$ = 3
$\lambda$ = 24 per hour
$\mu$ = 10 per hour

*Using the formulae for $P_0$, we get $P_0$ = 0.0562. (This value can also be taken directly from the $P_0$ table in the Anderson, Sweeney and Williams textbook.) Continuing with the calculations gives:*

*P = 24/30 = 0.8*

*$L_q$ = 2.590*

*$W_q$ = 0.108 hours, or 6.5 minutes.*

*The expected total time in the system is therefore 6.5 + 6 = 12.5 minutes.*

*Hourly cost = 3 × €25 + 24 × €3 = €147.*

…/

*Contd...*

*Implied value of customer waiting time:*

*Cost Difference per hour between options 2 and 3*

*= €156 - €147 = €9 per hour.*

*On average 24 customers per hour use the system, thus the €9 per hour is equivalent to €9/24 = €0.375 per customer.*

*The difference in waiting times between options 2 and 3 is:*

*12.5 minutes - 11.1 minutes = 1.4 minutes.*

*The above implies that the company is willing to spend €0.375 on each customer in order to reduce his or her waiting time by 1.4 minutes.*

*Thus 1.4 minutes is worth €0.375*

*60 minutes is worth €0.375 $\times$ 60 /1.4*

*= €16.07*

*Thus the implied value of 1 hour of customer waiting time is €16.07.*

# UNIT 9

## SIMULATION

### AIM

Simulation is a mathematical technique which is widely used to analyse and describe problem situations in both business and industry. This unit aims to introduce the theory and application of simulation.

### OBJECTIVES

- To present the basic elements of simulation and demonstrate why it is used to analyse different situations.
- To illustrate the use and scope of Monte Carlo simulation.
- To create an understanding of the economics of computer simulation.

### REQUIRED READING

Students should read the following chapters of the mandatory text book David R. Anderson , Dennis J. Sweeney , Thomas A. Williams and Kipp Martin An Introduction To Management Science: Quantitative Approaches To Decision Making, 13th International Edition, Cengage Learning, ISBN-13: 9780538475655 in conjunction with this unit.

Chapter 12

### 9.1 THE SIMULATION PROCESS

Simulation is a technique widely used in Management Science for the analysis and solution of problems in business and industry. Through your study of previous Management Science techniques you are familiar with the emphasis that these techniques place on optimisation, and the need to find the best answer.

Not all real-world problems can be conveniently analyzed by mathematical models, due, in many cases, to the complexity of the mathematics involved. While many mathematical models usually have the objective of *optimality*, not all real-world scenarios can be cast in this mode. A further consideration is *complexity* and *stochastic relationships* which make formal mathematical analysis extremely difficult, if not impossible.

Simulation is in contrast with these techniques in that it does not attempt to find the optimum solution. Simulation is concerned with building a model of a problem, and then observing that model repeatedly so that the behaviour of the problem situation can be analysed and useful information obtained. The simulation technique is experimental in nature in the sense that a simulation run can be regarded as a sample in a statistical experiment. The success of the simulation process depends mainly on the ability of the model to reflect the reality of the problem situation.

A simulation experiment differs from a regular laboratory experiment in that it can be conducted totally on the computer. By expressing the interactions among the components of the system as mathematical relationships, we are able to gather the necessary information in very much the same way as if we were observing the real system. The nature of simulation thus allows greater flexibility in representing complex systems that are normally difficult to analyse using standard mathematical models.

Variability and unpredictability are a fact of life in real problems. Problems contain variables whose value cannot be predicted with certainty, and, at best, probability distributions can be used to describe their performance. Simulation models must, therefore, accommodate a probability dimension. In general, this is achieved through random number generators.

Hence, when using simulation to solve a problem the method of attack is to create, usually on a computer, a system which closely models the characteristics of the real-world situation. The majority of such simulation models are stochastic models - involving the use of one or more probability distributions. The complete simulation process can be depicted as a flow chart and is exhibited as Figure 9.1.

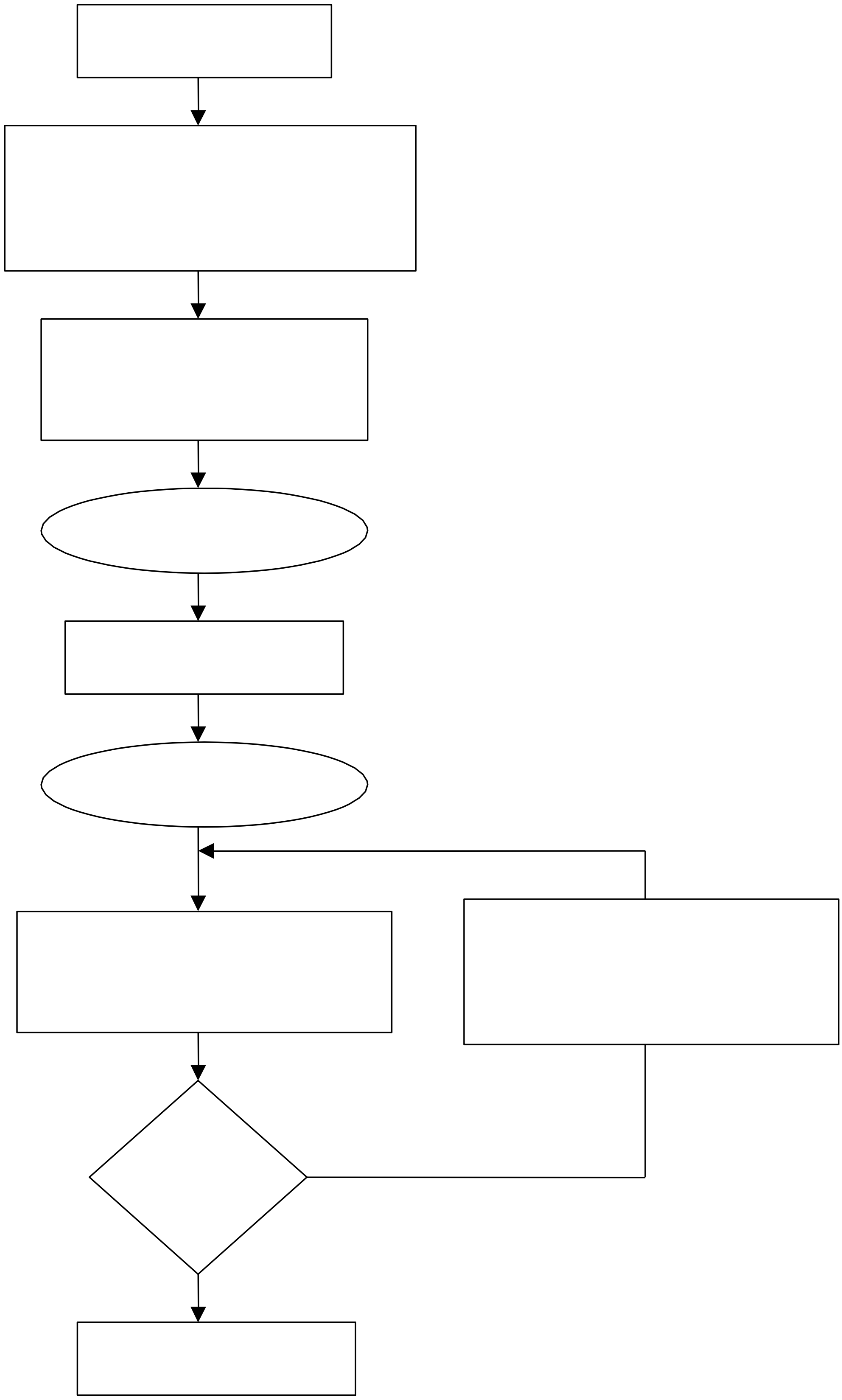


**Figure 9.1**
**The Simulation Process**

There are two main reasons why simulation is the preferred solution procedure in certain situations. These are:

- The problem situation is such that the assumptions required by the standard Management Science techniques do not apply. For example, in linear programming, it is assumed that all the equations of the model are linear. If the linearity assumption is not valid in a particular application then the linear programming technique is not appropriate.

- In some applications, even though a realistic model can be built, it may not be possible to derive a solution using the available techniques. This is especially true in the areas of queuing theory and inventory control.

In the above cases simulation may prove to be a useful approach.

The two main types of simulation are:

- Analogue
- Monte Carlo

With analogue simulation a physical device is built to model a situation. Typical analogue simulation includes flight simulators for trainee pilots and wind tunnels for car design. Analogue simulation is not of concern in this unit.

This unit is concerned only with Monte Carlo Simulation. In these cases a mathematical model is built to describe the problem.

## 9.2 MONTE CARLO SIMULATION

The term Monte Carlo is a mathematical technique used within simulation and is concerned with generating independent values of the random variable described by the probability distribution. The term was coined by Professor John Von Neumann of Princeton University who used the technique in the development of the atomic bomb in the 1940s. The basic principles behind the technique are similar to the games of chance one can play in the casinos of Monte Carlo or, for that matter, Las Vegas. Consider the roulette wheel with the ball spinning around and finally settling in a numbered hole. If one records the frequency with which the ball settled in the '6' or '10' or '20' holes over a complete day's play; then a probability model of the play of the roulette wheel can be constructed. A simplified set of results from such a simulation is given in Table 9.1.

| NUMBER OF HOLE ENTERED | NO. OF TIMES HOLE ENTERED | PROBABILITY OF ENTERING HOLE P(X) | CUMULATIVE PROBABILITY F(X) |
|---|---|---|---|
| 2 | 20 | 0.20 | 0.20 |
| 18 | 40 | 0.40 | 0.60 |
| 24 | 20 | 0.20 | 0.80 |
| 30 | 10 | 0.10 | 0.90 |
| 36 | 10 | 0.10 | 1.00 |
| TOTAL | 100 | 1.00 | |

**Table 9.1**
**Simplified results of roulette wheel simulation**

This process was actually observed, some years ago, by several Ph.D. students at a leading American University over several days of play. They recorded the frequency of entering the holes and subjected the data to statistical analysis. They discovered that the roulette wheel was biased, in that certain holes were more favoured than others. Using their data they constructed a simulation model of the wheel and computed how much they would have to spend to win a very considerable amount of money. They did!

A Random Process Generator produces random numbers in accordance with a particular probability distribution. This is done using the appropriate cumulative frequency distribution **F(x)** to generate values of the random variable **x**. This is usually done using a computer. The first step in the process involves the generation of random numbers.

## 9.3 RANDOM NUMBERS AND THEIR GENERATION

Random numbers are a sequence of numbers produced by a process that ensures that all digits have an equal probability of appearing at each point in the sequence. In mathematical terms, random numbers are numbers that are statistically independent of one another.

Nowadays we use computers to generate pseudo-random numbers using embedded sub-routines. Numbers generated in this fashion are known as pseudo-random as they are not fully randomized. However, for all practical purposes they may be considered as random numbers. Prior to the advent of computers, statisticians used generated tables of random numbers. A section of such a table is shown in Table 9.2. This contains a selection of random numbers that lie in the range between 0 and 99.

| | | | | |
|---|---|---|---|---|
| **36** | **63** | **70** | **35** | **33** |
| **98** | **16** | **04** | **41** | **67** |
| **56** | **20** | **11** | **32** | **44** |
| **26** | **99** | **76** | **75** | **63** |
| **60** | **82** | **29** | **20** | **25** |
| **79** | **88** | **01** | **97** | **30** |
| **14** | **85** | **11** | **47** | **23** |

**Table 9.2**
**A Random Number Table**

Such random numbers, when allied to a *Cumulative Frequency Distribution*, generate statistically independent values of the random variable (**x**), in this case the number of the hole that the ball falls into on the roulette wheel. Table 10.3 depicts a limited simulation of the situation.

- The first random number is 36 which falls in the range 20 → 59, hence the numbered hole is 18.
- The second random number is 98 which falls in the range 90 → 99 hence the numbered hole is 36.

This process can be continued as long as is desired. As a general rule, the longer the period of simulation the closer the simulated results mirror the real-life situation. Thus after exhausting the numbers in the first column of the random number table, we could continue into the second column, and so on. It should be noted however that nowhere near all of the numbers of a roulette wheel have been included. This could only be achieved by observing the real-life situation for a considerable period of time and then constructing the appropriate distributions for **P(x)** and **F(x)**.

| | | | | | LIMITED SIMULATION | |
|---|---|---|---|---|---|---|
| **NUMBER OF HOLE ENTERED x** | **NUMBER OF TIMES HOLE ENTERED** | **PROBABILITY OF ENTERING HOLE P(x)** | **CUMULATIVE PROBABILITY F(x)** | **RANGE OF RANDOM NUMBERS** | **RANDOM NUMBER FROM TABLE** | **NUMBER OF HOLE ENTERED** |
| **2** | **20** | **0.2** | **0.2** | **0 - 19** | **36** | **18** |
| **18** | **40** | **0.4** | **0.6** | **20 - 59** | **98** | **36** |
| **24** | **20** | **0.2** | **0.8** | **60 - 79** | **56** | **18** |
| **30** | **10** | **0.1** | **0.9** | **80 - 89** | **26** | **18** |
| **36** | **10** | **0.1** | **1** | **90 - 99** | **60** | **24** |
| **Total** | **100** | **1** | | | **79** | **24** |
| | | | | | **14** | **2** |
| | | | | | **1st Column of table of random numbers** | **Number of hole pertaining to random number** |

**Table 9.3**
**A Simulation of the Roulette Wheel**

## *SAQ 1*

A petrol station is being planned and it is required to optimise the forecourt layout. Traffic studies suggest that customers will arrive into the new garage with the following distribution.

| No. of customers in a minute | Probability |
|---|---|
| 0 | .52 |
| 1 | .20 |
| 2 | .11 |
| 3 | .09 |
| 4 | .08 |

Use the following sequence of random numbers to simulate 10 minutes in the life of the garage.

20 18 94 83 27 56 49 83 17 74 26 75

## *SAQ 2*

An analysis of a single server queue yields the following probability distributions for Arrival Intervals and Service times.

| Arrival Interval (Mins) | Probability |
|---|---|
| 1 | .1 |
| 2 | .1 |
| 3 | .4 |
| 4 | .2 |
| 5 | .2 |

| Service time (Mins) | Probability |
|---|---|
| 2 | .2 |
| 3 | .5 |
| 4 | .3 |

For example, the probability that there is a three-minute interval between one arrival and the next is 0.4. Similarly, the probability that a customer will take four minutes to service is 0.3.

Use a simulation of 10 customers to determine the average customer waiting time and the percentage of customers who must wait for service.

Random Number sequence to be used is:

76 23 47 25 79 08 15 71 58 56 31

Use the first digit of each pair to generate random arrival intervals, and the second to generate random service times.

## *SAQ 3*

The daily demand for a product has the following probability distribution:

| Demand Per Day | Associated Probability Distribution |
|---|---|
| 140 Units | 0.20 |
| 150 Units | 0.40 |
| 160 Units | 0.20 |
| 170 Units | 0.10 |
| 180 Units | 0.10 |

Given the following set of random numbers

39 65 76 45 45
73 71 23 70 90
71 20 47 33 84
75 17 25 69 17
37 48 79 88 74

Simulate the demand of the product for twenty periods. Compute the mean demand and compare it with the expected value of demand.

## *SAQ 4*

A company selling electrical light fittings to the building industry wishes to determine the optimal stock ordering policy for those fittings. Demand for the fittings is not certain and there is a lead time for stock replenishment. The following table details information on daily demand.

| *Demand Units per day* | *Probability* |
|---|---|
| 3 | .1 |
| 4 | .25 |
| 5 | .25 |
| 6 | .3 |
| 7 | .1 |

| | | |
|---|---|---|
| Carrying cost (per unit per day, and based on stock at end of day) | = | €0.20 |
| Ordering Cost | = | €5.00 |
| Lead time | | 3 days |

Stock ordered at the end of day i will be received into stock at the start of day i + 3.

| | | |
|---|---|---|
| Stock-out cost (per unit per day) | = | €0.75 |

Stock on hand at the start of the simulation period is 21 units.

Carry out a simulation exercise for 8 days to evaluate the following re-ordering rule:

Order 15 units when end-of-day stock level falls below 15 units.

Use the following sequence of random numbers in the simulation:

09 11 51 86 65 71 29 96.

Discuss how the simulation model could be extended to find the optimal re-ordering rule.

## 9.4 COMPUTER SIMULATION

Before the computer age, simulations had to be carried out by hand using random numbers drawn from random number tables. This was a laborious process. Like in many other branches of Management Science, the advent of the computer revolutionized the application of simulation to real-world problems. To carry out any sort of realistic simulation, the use of a computer becomes a necessity. This is not because of the complexity involved, but merely because of the large number of times that the model must be run. An important advantage of computer simulation was that the simulation model and its computer program represented a "managerial" experimental laboratory. This was accomplished through asking "what if" questions and observing their result.

However caution must be exercised in developing computer simulation models. Much time, effort and cost can be expended on their construction. Hence one should not start on their development without assessing potential gains versus likely development costs. The risk factor can be reduced if one can use some of the simulation applications that have been developed over the past several decades.

To assist in the computerisation of simulation models several computer packages have been developed. These include GPSS (General Purpose Systems Simulator), SIMSCRIPT and GASP (General Activity Simulation Program).

## 9.5 ADVANTAGES AND DISADVANTAGES OF SIMULATION

The main advantage of simulation is that it is applicable to complex cases which defy investigation by analytical procedures. This is often the case where systems have probabilistic components embedded in them.
The greater their number, the more likely is the use of the computer simulation approach. In tandem with other Management Science techniques, simulation provides a forum to investigate "what if" scenarios. Thus it can also be used as a "management laboratory" to test the effects of differing conditions and assumptions.

A disadvantage of simulation is that it does not normally provide an optimal solution to the problem under investigation. In this regard, it is akin to Goal Programming, which attempts to find a *satisfactory* solution. If the user of a simulation model is concerned with finding the optimal set of inputs, then the model must be run repeatedly with different input values. The optimal set of inputs is the set that maximises (or minimises) the performance criteria. The disadvantage of this procedure is that the optimal answer produced is not necessarily the optimal for the problem; it is simply the best of the alternatives tested by the user of the simulation.

As has already been mentioned, Simulation can be expensive if the simulation program has to be specially developed. Thus the costs of model development must be weighed against the anticipated gains before the decision to proceed is taken.

## 9.6 CONTINUOUS SIMULATION

The previous sections of this unit were concerned with discrete simulation in which discrete random variables are used to model the various variable quantities. In some applications, however, it is not appropriate to use discrete random variables and the variable quantities are best modelled using continuous random variables.

Continuous simulation involves using continuous random variables in the models. As is the case with discrete simulation, continuous simulation uses random variables to incorporate the probability dimension into the model. However, with continuous simulation advanced probability theory is required to identify a variable value with a random number range. This theory involves the inverse of the probability density function and is beyond the scope of this unit.

## 9.7 SOME APPLICATIONS OF SIMULATION

Simulation is widely used in many management, economic, financial, military and space-related investigations. Indeed NASA constantly uses Computer Simulation in its various space-exploration programmes. Providing a definitive list of applications would be an impossible task. The following therefore is only a small sample of the range of application of this very useful technique:

- War Games
- Space Exploration
- Environmental Investigations
- Nuclear Reactions (chain reactions)
- Public Service Operations
- Financial Markets
- National and International Economic Systems
- Flight Simulators
- Queuing Problems in Road/Rail/Aircraft Control Systems
- Inventory Systems
- Production Systems
- Electrical /Water / Computer Networks.

## *SAQ 5*

A public quoted company in the chemical industry is concerned about the number of large trucks arriving at one of its warehouses to collect some of its products. The Managing Director has asked the Transport Manager to carry out a survey to ascertain the arrival distribution of customers' vehicles at the warehouse and the service distribution of those vehicles whilst at the warehouse. In due course of time the Transport Manager sends his Managing Director the following information.

***The arrival distribution of vehicles***

| *Time Between Arrivals (x) (Mins)* | *Probability P(x)* |
|---|---|
| 1 | 0.10 |
| 2 | 0.30 |
| 3 | 0.30 |
| 4 | 0.20 |
| 5 | 0.10 |

***The service distribution of vehicles***

| *Service Times (y) (Mins)* | *Probability P(y)* |
|---|---|
| 2 | 0.40 |
| 3 | 0.40 |
| 4 | 0.20 |

You, as Technical Assistant to the Managing Director are handed these distributions with the instruction, "Tell me what is going on and write me (the M.D.) a brief Report".

*Hint*
Simulate for, say, 20 customers and analyse the results.

Use the following random numbers:

*Arrivals*

| | |
|---|---|
| 28 | 69 |
| 24 | 97 |
| 0 | 28 |
| 92 | 39 |
| 7 | 87 |
| 18 | 6 |
| 93 | 26 |
| 99 | 97 |
| 87 | 79 |
| 33 | 79 |

*Service*

| | |
|---|---|
| 77 | 12 |
| 94 | 73 |
| 30 | 73 |
| 5 | 99 |
| 39 | 12 |
| 28 | 49 |
| 10 | 99 |
| 99 | 57 |
| 0 | 94 |
| 27 | 82 |

## ***Exercise***

*Repeat the work in SAQ 5 using a different set of random numbers and see if the results are similar.*

## 9.8 BINOMIAL, POISSON AND NORMAL RANDOM VARIABLES

So far in this unit, the random variables have all been discrete with probabilities allocated to values according to tables. In the following examples the appropriate random variables are named random variables, such as binomial, Poisson or Normal. The following examples demonstrate the principles of using such random variables.

**Example 1: (Poisson Random Variable)**
The number of machines that break down in a factory each day can be modelled as a Poisson random variable with an average of two breakdowns per day. Each breakdown results in a loss of production to the company. The value of lost production depends on the nature of the breakdown. The following table details the value of lost production

associated with each breakdown type, together with the probability of that type of breakdown occurring.

| Nature of Breakdown | Value of Lost Production | Probability |
|---|---|---|
| Severe | €2,000 | 0.08 |
| Medium | €800 | 0.37 |
| Normal | €400 | 0.55 |

Simulation is to be used to model eight days of production and estimate the daily average value of lost production.

Random Numbers:

20 43 74 78 79 89 63 23 47 26 90
03 73 70 59 12 93 90 24 52 34 55
17 73 69 13 44 00 76

The Poisson random variable with average of two can take on the following values:

0, 1, 2, 3, 4, 5, 6, 7, 8, …..

with probabilities given by the formula

$$P[x] = \frac{e^{-\lambda}\lambda^x}{x!}$$

Using the above formula the following table can be generated:

| Number of breakdowns | Probability | Number Range |
|---|---|---|
| 0 | 0.14 | 00 – 13 |
| 1 | 0.27 | 14 – 40 |
| 2 | 0.27 | 41 – 67 |
| 3 | 0.18 | 68 – 85 |
| 4 | 0.09 | 86 – 94 |
| 5 | 0.04 | 95 – 98 |
| 6 | 0.01 | 99 |

The table for the value of lost production is:

| Breakdown Type | Value | Probability | Number Range |
|---|---|---|---|
| Severe | €2,000 | 0.08 | 00 – 07 |
| Medium | €800 | 0.37 | 08 – 44 |
| Normal | €400 | 0.55 | 45 – 49 |

The simulation results are

| Day | Random No. | No. of Breakdowns | Random No. | Value (€) |
|---|---|---|---|---|
| 1 | 20 | 1 | 43 | 800 |
| 2 | 74 | 3 | 78 | 400 |
| | | | 79 | 400 |
| | | | 89 | 400 |
| 3 | 63 | 2 | 23 | 800 |
| | | | 47 | 400 |
| 4 | 26 | 1 | 90 | 400 |
| 5 | 03 | 0 | | |
| 6 | 73 | 3 | 70 | 400 |
| | | | 59 | 400 |
| | | | 12 | 800 |
| 7 | 93 | 4 | 90 | 400 |
| | | | 24 | 800 |
| | | | 52 | 400 |
| | | | 34 | 800 |
| 8 | 55 | 2 | 17 | 800 |
| | | | 73 | 400 |
| Total | | 16 | | €8,800 |

Average per day = (€8,800 / 8) = €1,100

## *SAQ 6*

*A van rental company has 4 vans available for daily hire. From the past records of daily demand, the management estimates that daily demand for vans can be considered as a Poisson random variable with a daily average of 3.*

*Management is concerned about the number of occasions when the daily demand exceeds 4, and potential customers must be told that the company's full fleet of 4 vans has already been allocated. Use a simulation of 10 days to estimate the proportion of days on which demand exceeds supply and calculate the average utilisation of 4 vans.*

*Use the following random numbers:*
*20 43 74 78 79 89 63 23 47 26 90*
*03 73 70 59 12 93 90 24 52 34 55*
*17 73 69 13 44 00 76*

**Example 2: (Normal Random Variable)**
Weekly demand for a product is normally distributed with a mean of 50 units and a standard deviation of 2 units.

Use the following random numbers to simulate the demand over six weeks:

03 72 49 46 83 71 26

The probability distribution can be found by dividing the range of weekly demand volumes into class intervals.

| Class Interval | Probability (from Normal Tables) |
|---|---|
| < 43 | - |
| 43 – 45 | 0.01 |
| 45 – 47 | 0.06 |
| 47 – 49 | 0.24 |
| 49 – 51 | 0.38 |
| 51 – 53 | 0.24 |
| 53 – 55 | 0.06 |
| 55 – 57 | 0.01 |
| > 57 | - |

The above probabilities are calculated using the standard Normal tables.

Random Numbers are now allocated as follows, using the mid-point of the class interval as actual demand.

| Class Interval | Mid-Point | Probability | Random No. |
|---|---|---|---|
| 43 – 45 | 44 | 0.01 | 00 |
| 45 – 47 | 46 | 0.06 | 01 - 06 |
| 47 – 49 | 48 | 0.24 | 07 - 30 |
| 49 – 51 | 50 | 0.38 | 31 - 68 |
| 51 – 53 | 52 | 0.24 | 69 - 92 |
| 53 – 55 | 54 | 0.06 | 93 - 98 |
| 55 – 57 | 56 | 0.01 | 99 |

The simulation of demand over 6 weeks is as follows:

| Week | Random Number | Demand |
|---|---|---|
| 1 | 03 | 46 |
| 2 | 72 | 52 |
| 3 | 49 | 50 |
| 4 | 46 | 50 |
| 5 | 83 | 52 |
| 6 | 71 | 52 |

## *SAQ 7*

*The time taken for a mechanic to repair a machine can be considered to be a Normal random variable with mean of 4 hours and standard deviation of 1 hour.*

*A bonus per machine is paid to the mechanic according to the following table:*

*If repair time is 2 hours or less a bonus of €3 is paid.*
*If repair time is greater than 2 but is 4 hours or less, a bonus of €2 is paid.*
*If repair time is between 4 and 5 hours, a bonus of €1 is paid.*
*Repair times in excess of 5 hours yield no bonus.*

*Use a simulation of 12 machines to calculate the average bonus per machine that the mechanic obtains.*

*Use the following random numbers:*

*20 43 74 78 79 89 63 23 47 26 90*
*03 73 70 59 12 93 90 24 52 34 55*
*17 73 17 73 69 13 44 00 76*

**Example 3: (Binomial Random Variable)**
A part-time salesperson is allocated five houses to visit every evening. Past records suggest that, on average, 60% of all calls result in a sale. Use the following random numbers to simulate the performance of the sales person over five evenings.

57 3 97 48 81 09

The number of sales achieved each evening can be considered as a binomial random variable with 5 trials and a probability of success at each trial of 0.6. Using the binomial formula the following table is produced.

| No. of Sales | Probability | | No. Range |
|---|---|---|---|
| 0 | ${}^5C_0\,(0.6)^0\,(0.4)^5$ | = 0.01 | 00 |
| 1 | ${}^5C_1\,(0.6)^1\,(0.4)^4$ | = 0.08 | 01 – 08 |
| 2 | ${}^5C_2\,(0.6)^2\,(0.4)^3$ | = 0.23 | 09 – 31 |
| 3 | ${}^5C_3\,(0.6)^3\,(0.4)^2$ | = 0.34 | 32 – 65 |
| 4 | ${}^5C_4\,(0.6)^4\,(0.4)^1$ | = 0.26 | 66 – 91 |
| 5 | ${}^5C_5\,(0.6)^5\,(0.4)^0$ | = 0.08 | 92 – 99 |

The simulation of 6 evenings is:

| Evening | Random Number | No. of Sales |
|---|---|---|
| 1 | 57 | 3 |
| 2 | 23 | 2 |
| 3 | 97 | 5 |
| 4 | 48 | 3 |
| 5 | 81 | 4 |
| 6 | 09 | 2 |

## *SAQ 8*

*ABC Ltd. manufactures a product X, which is used as a raw material by five other companies. In addition to ABC Ltd. other companies manufacture a similar product, and at the start of each year ABC Ltd. estimate that there is a probability of 0.7 that any one of the 5 potential customers will purchase its entire annual requirement from ABC Ltd. The decision by each of the 5 companies to buy either from ABC Ltd. or from its competitors is not dependent on the decision of the other four.*

*If a company decides to buy its annual requirement from ABC Ltd. then the amount purchased is Normally (Gaussian) distributed with a mean of 2,000 units and a standard deviation of 100 units.*

*Use simulation to model 3 years for ABC Ltd.*

*Random Numbers*
*27 05 20 30 85 21 04 67 19 13 12 46 98 16 26*

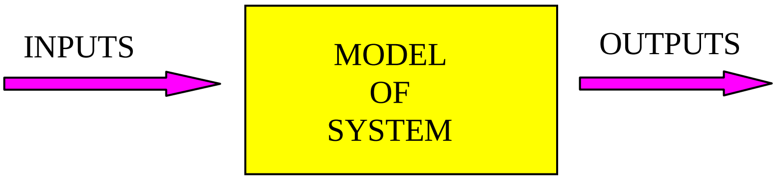
## SUMMARY

In this unit a brief introduction to the expansive and widely used tool of simulation was presented. The student should realise that many real world problems present significant complexity and uncertainty in their structure lending themselves intractable to standard analytical tools and closed form solutions. In particular where uncertainty and risk are present in decision scenarios the decision maker will find it difficult to make the best decision. Consequently one may then wish to assess the likelihood of various outcomes given various inputs and make a more informed decision. Typically a simulation context is characterised by a set of inputs and these are usually subject to uncertainty and consequently the outputs of interest (often some sort of performance metrics) are themselves subject to uncertainty. This then attaches risk to the outcomes and this risk needs to be assessed and quantified. Varying the inputs or equivalently sampling from their distributions assuming they can be adequately modelled allows for a distributional snapshot of possible outcomes. **Summary statistics** can then in turn yield estimates of among other things:

- Mean or expected value of quantities of interest (e.g. expected profit, cost, waiting times etc.)

- Variance (a proxy risk measure) pertaining to quantities of interest and other measures of dispersion (e.g. semi-variance, range, quartiles, skewness, kurtosis etc.)

- Probabilities and confidence intervals of quantities of interest etc.




Inputs might be for instance different interest rate levels and stock price movements for an investment portfolio whilst the outputs might be the value at risk (VAR) and expected profit in the case of a bank simulating portfolio exposure to price fluctuations. In the case of an insurance company inputs could be claim incidence and size from a portfolio of motor and home insurance and outputs might include the probability of ruin, expected loss etc.

For a meaningful simulation the various inputs must be identified, understood and modelled often from past data. They may be a mixture of discrete and continuous random variables and capturing accurately their range of behaviours is not to be underestimated for many situations. The quality and accuracy of the inputs determines the value of the outputs. Various assumptions may be adopted and need to be tested. For instance one might assume Poisson arrivals in a waiting line situation and a Pareto distribution for claim size in an insurance case or lognormal distribution for stock price movement etc.

The model (or models) is itself the link between the inputs and outputs. This may be a simple formula calculating for example a profit figure or a complicated set of relationships linking the inputs to the outputs. The outputs are simply any quantity of central interest to the decision maker such as profit, costs, waiting times, project completion time, stock out incidence etc.

In their final summarised form following the simulation exercise the outputs constitute (when statistically analysed and summarised) a basis for decisions, now supported by an analysis that if conducted properly is a sound basis for risk assessment.

## SUMMARY OF STEPS

- Identify the various inputs to the model. Some will be fixed in value and others will be random in nature. Identify also which are under your control and which are outside.

- Try to develop a description of the behaviour of the inputs, this may mean adopting assumptions or modelling from past data to arrive at some distributional description of the random components.

- Develop and construct the various models and formulae linking the inputs to the outputs. This may require some effort if the system is complex. Knowledge of the context and stakeholder input will be necessary here.

- Identify the various outputs and summary statistics and performance measures required by the stakeholders.

- Perform the simulations by randomly sampling from the input distributions and in each case calculating through the model the outputs of interest. This is typically automatically repeated many times in a spreadsheet environment and the performance measures can then be statistically analysed. This will involve calculating measures of central tendency, spread, likelihoods and probability of extreme outcomes being assessed etc.

- Develop graphical and tabular summaries that are easily understood and that facilitate any decision support.

- Sensitivity analyses are integral to determining the high impact inputs and stress testing in terms of any assumptions concerning inputs and the model.

- Review the simulation exercise and if useful archive the results and process.

## FINAL COMMENTS

The topic of simulation allows one to embrace in a realistic way the complexity found in most real systems. These may be systems examined in the units studied such as project management, inventory, waiting lines etc. or other applications such as finance, capital budgeting, marketing etc. there is a multiplicity of tools available that btring this method within reach of the novice. At the simplest level various add-ins to Excel are available such as Crystal –Ball and @RISK. These are available quite often as CD-ROM attachments to texts and st5udent/trial versions are easily downloaded from the web. These make the process of random number generation easy and a library of distributions one can sample from is available together with easy to use charting and statistical analysis menus. Consequently the student should attempt to incorporate this spreadsheet capability in some of the exercises and possible assignment tasks. A quick survey of applications and software vendors on the web will convince the student of the value of this tool especially when the decision context leaves one vulnerable and exposed to risk.

## USEFUL REFERENCES

http://www.solver.com/monte-carlo.htm
www.palisade.com
http://www.lionhrtpub.com/orms/surveys/Simulation/Simulation.html
http://www.idsia.ch/~andrea/simtools.html
http://www.simulation.ie/cim.htm
http://www.simuledge.ie

## ANSWERS TO SAQS

### *SAQ 1*

*Since accuracy to two places of decimal is used in the probability table, the random number sequence will be read in blocks of 2 numbers. The table below associates ranges for the random numbers with Number of Customers values.*

| *No. of customers* | *Probability* | *Range* |
|---|---|---|
| *0* | *.52* | *00 to 51* |
| *1* | *.20* | *52 to 71* |
| *2* | *.11* | *72 to 82* |
| *3* | *.09* | *83 to 91* |
| *4* | *.08* | *92 to 99* |

*The simulation results are:*

| *Run No.* | *Minutes* | *Random No.* | *Range* | *No. of Customers* |
|---|---|---|---|---|
| *1* | *1* | *20* | *00 to 51* | *0* |
| *2* | *2* | *18* | *00 to 51* | *0* |
| *3* | *3* | *94* | *92 to 99* | *4* |
| *4* | *4* | *83* | *83 to 91* | *3* |
| *5* | *5* | *27* | *00 to 51* | *0* |
| *6* | *6* | *56* | *52 to 71* | *1* |
| *7* | *7* | *49* | *00 to 51* | *0* |
| *8* | *8* | *83* | *83 to 91* | *3* |
| *9* | *9* | *17* | *00 to 51* | *0* |
| *10* | *10* | *74* | *72 to 82* | *2* |

*Total number of customers in the 10 minutes is 13*

## *SAQ 2*

*In this example, both Arrival Interval and Service Time use probability distributions accurate to 1 place of decimal. Thus for both of these, variable random numbers in blocks of length 1 will be used.*

*The following tables associate random number ranges for both Arrival Interval and Service Time.*

| *Arrival Interval (Mins)* | *Probability Range* | *Random Number* |
|---|---|---|
| *1* | *.1* | *0 to 0* |
| *2* | *.1* | *1 to 1* |
| *3* | *.4* | *2 to 5* |
| *4* | *.2* | *6 to 7* |
| *5* | *.2* | *8 to 9* |

| *Service time (Mins)* | *Probability Range* | *Random Number* |
|---|---|---|
| *2* | *.2* | *0 to 1* |
| *3* | *.5* | *2 to 6* |
| *4* | *.3* | *7 to 9* |

*Random Number sequence used is:*

*76 23 47 25 79 08 15 71 58 56 31*

*Simulation results for 10 customer arrivals are as follows:*

| ***Customer*** | ***Random No.*** | ***Arrival Interval*** | ***Arrives*** | ***Start*** | ***Wait*** | ***Random No*** | ***Service Time*** | ***Ends*** | ***Time in System*** |
|---|---|---|---|---|---|---|---|---|---|
| *1* | *7* | *4 mins* | *4* | *4* | *0 min* | *6* | *3 mins* | *7* | *3 mins* |
| *2* | *2* | *3* | *7* | *7* | *0* | *3* | *3* | *10* | *3* |
| *3* | *4* | *3* | *10* | *10* | *0* | *7* | *4* | *14* | *4* |
| *4* | *2* | *3* | *13* | *14* | *1* | *5* | *3* | *17* | *4* |
| *5* | *7* | *4* | *17* | *17* | *0* | *9* | *4* | *21* | *4* |
| *6* | *0* | *1* | *18* | *21* | *3* | *8* | *4* | *25* | *7* |
| *7* | *1* | *2* | *20* | *25* | *5* | *5* | *3* | *28* | *8* |
| *8* | *7* | *4* | *24* | *28* | *4* | *1* | *2* | *30* | *6* |
| *9* | *5* | *3* | *27* | *30* | *3* | *8* | *4* | *34* | *7* |
| *10* | *5* | *3* | *30* | *34* | *4* | *6* | *3* | *37* | *7* |

*The total time waiting for service is 20 minutes for 10 customers, and therefore the average wait time per customer is 2 minutes.*

*Of the 10 customers 6 had to wait before service could start, i.e. 60%.*

## *SAQ 3*

| ***Demand*** | ***Probability Of Demand*** | ***Cumulative probability*** | ***Random Numbers Range of Random Numbers*** |
|---|---|---|---|
| ***x*** | ***P(x)*** | ***f(x)*** | ***(r)*** |
| *140* | *0.20* | *0.2* | *00 - 19* |
| *150* | *0.40* | *0.6* | *20 - 59* |
| *160* | *0.20* | *0.8* | *60 - 79* |
| *170* | *0.10* | *0.9* | *80 - 89* |
| *180* | *0.10* | *0.1* | *90 - 99* |

*Expected value of demand*

*= 140(0.20) + 150(0.40)+160(0.20)+170(0.10)+180(0.10)*

*= 28+60+32+17+18*

*=155*

*Simulation over twenty periods*

| **Demand Period** | **Random Number** | **Demand** |
|---|---|---|
| | (r) | (x) |
| 1 | 39 | 150 |
| 2 | 73 | 160 |
| 3 | 71 | 160 |
| 4 | 75 | 160 |
| 5 | 37 | 150 |
| 6 | 65 | 160 |
| 7 | 71 | 160 |
| 8 | 20 | 150 |
| 9 | 17 | 140 |
| 10 | 48 | 150 |
| 11 | 76 | 160 |
| 12 | 23 | 150 |
| 13 | 47 | 150 |
| 14 | 25 | 150 |
| 15 | 79 | 160 |
| 16 | 45 | 150 |
| 17 | 70 | 160 |
| 18 | 33 | 150 |
| 19 | 69 | 160 |
| 20 | 88 | 170 |

*Average Demand*
*=(140+9×150+9×160+170)/20*
*= (140 + 1350 + 1440 + 170)/20*
*= 3100/20*
*= 155*

*Thus the expected value of demand is equal to the average of the simulated demand.*

## *SAQ 4*

*The allocation of random numbers for daily demand would be as follows:*

| *Demand per day* | *Probability* | *Random Number Range* |
|---|---|---|
| 3 | .1 | 00 to 09 |
| 4 | .25 | 10 to 34 |
| 5 | .25 | 35 to 59 |
| 6 | .3 | 60 to 89 |
| 7 | .1 | 90 to 99 |

*The simulation would be as follows:*

| *Day* | *Opening Stock* | *Random Number* | *Daily Demand* | *Closing Stock* | *Order Cost* | *Hold Cost* | *Short Cost* | *Total Cost* |
|---|---|---|---|---|---|---|---|---|
| 1 | 21 | 09 | 3 | 18 | - | 3.60 | - | 3.60 |
| 2 | 18 | 11 | 4 | 14* | 5 | 2.80 | - | 7.80 |
| 3 | 14 | 51 | 5 | 9 | - | 1.80 | - | 1.80 |
| 4 | 9 | 86 | 6 | 3 | - | .60 | - | .60 |
| 5 | 3+15=18 | 65 | 6 | 12* | 5 | 2.40 | - | 7.40 |
| 6 | 12 | 71 | 6 | 6 | - | 1.20 | - | 1.20 |
| 7 | 6 | 29 | 4 | 2 | - | .40 | - | .40 |
| 8 | 2+15=17 | 96 | 7 | 10* | 5 | 2.00 | - | 7.00 |
| | | | | | | | Total | €29.80 |

** Order for 15 units placed.*

*Total Inventory cost over the eight days is €29.80, an average of €3.73 per day.*

*The above estimate of €3.73 per day is based upon a very short simulation, and as such the accuracy of the estimate must be questioned.*

*€3.73 is an estimate of the average cost per day for the re-order policy: order 15 units when stock level falls below 15 units. The optimal policy can only be found using different re-ordering policies.*

## *SAQ 5*

***Arrival Distribution***

| ***Time Between Arrivals (x) (Mins)*** | ***Probability P(x)*** | ***Cumulative Probability C(x)*** | ***Range of Random Numbers ($r_1$)*** |
|---|---|---|---|
| *1* | *0.10* | *0.10* | *00 - 09* |
| *2* | *0.30* | *0.40* | *10-39* |
| *3* | *0.30* | *0.70* | *40 - 69* |
| *4* | *0.20* | *0.90* | *70 - 89* |
| *5* | *0.10* | *1.00* | *90 - 99* |

***Service Distribution***

| ***Service Times (y) (Mins)*** | ***Probability P(y)*** | ***Cumulative Probability C(y)*** | ***Range of Random Numbers ($r_2$)*** |
|---|---|---|---|
| *2* | *0.40* | *0.40* | *00 - 39* |
| *3* | *0.40* | *0.80* | *40 - 79* |
| *4* | *0.20* | *1.00* | *80 - 99* |

*Carry out 20 simulations using random numbers (RNs).*

| ***RNs For Arrivals*** | | | ***RNs For Service*** | | |
|---|---|---|---|---|---|
| ***Start*** ↓ | *28* | *69* | ***Start*** ↓ | *77* | *12* |
| | *24* | *97* | | *94* | *73* |
| | *0* | *28* | | *30* | *73* |
| | *92* | *39* | | *5* | *99* |
| | *7* | *87* | | *39* | *12* |
| | *18* | *6* | | *28* | *49* |
| | *93* | *26* | | *10* | *99* |
| | *99* | *97* | | *99* | *57* |
| | *87* | *79* | | *0* | *94* |
| | *33* | *79* | | *27* | *82* |

***20 Simulations***

| Customer | Random Numbers ($r_1$) | Time Between Arrivals x(Mins) | Arrival Clock | Service Clock | Customer Waiting Time (mins) | System Idle Time (mins) | Random Number ($r_2$) | Service Time y (mins) | Dept Clock |
|---|---|---|---|---|---|---|---|---|---|
| *1* | *28* | *2* | *2* | *2* | *0* | *2* | *77* | *3* | *5* |
| *2* | *24* | *2* | *4* | *5* | *1* | *0* | *94* | *4* | *9* |
| *3* | *0* | *1* | *5* | *9* | *4* | *0* | *30* | *2* | *11* |
| *4* | *92* | *5* | *10* | *11* | *1* | *0* | *5* | *2* | *13* |
| *5* | *7* | *1* | *11* | *13* | *2* | *0* | *39* | *2* | *15* |
| *6* | *18* | *2* | *13* | *15* | *2* | *0* | *28* | *2* | *17* |
| *7* | *93* | *5* | *18* | *18* | *0* | *1* | *10* | *2* | *20* |
| *8* | *99* | ***5*** | *23* | *23* | *0* | *3* | *99* | *4* | *27* |
| *9* | *87* | *4* | *27* | *27* | *0* | *0* | *0* | *2* | *29* |
| *10* | *33* | *2* | *29* | *29* | *0* | *0* | *27* | *2* | *31* |
| *11* | *69* | *3* | *32* | *32* | *0* | *1* | *12* | *2* | *34* |
| *12* | *97* | *5* | *37* | *37* | *0* | *3* | *73* | *3* | *40* |
| *13* | *28* | *2* | *39* | *40* | *1* | *0* | *73* | *3* | *43* |
| *14* | *39* | *2* | *41* | *43* | *2* | *0* | *99* | *4* | *47* |
| *15* | *87* | *4* | *45* | *47* | *2* | *0* | *12* | *2* | *49* |
| *16* | *6* | *1* | *46* | *49* | *3* | *0* | *49* | *3* | *52* |
| *17* | *26* | *2* | *48* | *52* | *4* | *0* | *99* | *4* | *56* |
| *18* | *97* | *5* | *53* | *56* | *3* | *0* | *57* | *3* | *59* |
| *19* | *79* | *4* | *57* | *59* | *2* | *0* | *94* | *4* | *63* |
| *20* | *79* | *4* | *61* | *63* | *2* | *0* | *82* | *4* | *67* |
| | | | | | *(29)* | *(10)* | | *(57)* | |

*Adding the customer waiting time and service time yields the following:*

| ***Customer*** | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *13* | *14* | *15* | *16* | *17* | *18* | *19* | *20* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***Customer time in system*** | *3* | *5* | *6* | *3* | *4* | *4* | *2* | *4* | *2* | *2* | *2* | *3* | *4* | *6* | *4* | *6* | *8* | *6* | *6* | *6* |

*Total customer time in system = 86 mins*

*The key system parameters are:*

- *Average waiting time = 29/20 = 1.45 mins*
- *7 of the 20 customers (35%) do not wait at all*
- *The 13 customers who have to wait for service, wait on average 29/13 = 2.2mins*
- *Total simulation extends over 67 minutes for which the system is idle for 10 mins (15%), hence it is busy for 85% of the time*
- *Average time a customer spends in system = 86/20 = 4.3 mins*

- *The 7 customers who do not have to wait only spend an average of 18/7 = 2.6 mins in system*
- *The 13 customers who must wait do so for an average of 68/13 or 5.2 mins in the system*
- *Generally 50% of all customers will wait 1 min, or less, for service and 20% of all customers must wait for 3 or more mins.*

## *SAQ 6*

*Simulation is to be used to model 10 days of demand for vans. If demand is 4 or less, then the number of vans is equal to demand. If demand exceeds 4 then only 4 vans can be hired, and potential customers are disappointed.*

*Daily demand is a Poisson random variable with an average of three and can take on the following values;*

*0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, … etc*

*with probabilities given by the formula*

$$P[x] = \frac{e^{-3}3^{x}}{x!}$$

*Using the above formula the following table can be generated*

| *Daily Demand* | *Probability* | *Random Number Range* |
|---|---|---|
| *0* | *0.05* | *00 – 04* |
| *1* | *0.15* | *05 – 19* |
| *2* | *0.22* | *20 – 41* |
| *3* | *0.22* | *42 – 63* |
| *4* | *0.17* | *64 – 80* |
| *5* | *0.10* | *81 – 90* |
| *6* | *0.05* | *91 – 95* |
| *7* | *0.02* | *96 – 97* |
| *8* | *0.01* | *98* |
| *9* | *0.01* | *99* |
| *10* | *0.00* | *-* |

*Random numbers:*
*20 43 74 78 79 89 63 23 47 26 90*
*03 73 70 59 12 93 90 24 52 34 55*
*17 73 69 13 44 00 76*

*The following table can be generated*

| *Day Number* | *Random Number* | *Daily Demand* | *Vans in use* | *Disappointed Customers* |
|---|---|---|---|---|
| *1* | *20* | *2* | *2* | *No* |
| *2* | *43* | *3* | *3* | *No* |
| *3* | *74* | *4* | *4* | *No* |
| *4* | *78* | *4* | *4* | *No* |
| *5* | *79* | *4* | *4* | *No* |
| *6* | *89* | *5* | *4* | *Yes* |
| *7* | *63* | *3* | *3* | *No* |
| *8* | *23* | *2* | *2* | *No* |
| *9* | *47* | *3* | *3* | *No* |
| *10* | *26* | *2* | *2* | *No* |

*On only 1 day in 10 do disappointed customers exist.*

*Over the 10 days, 31 vans were hired out. A utilisation rate of 31/40.*

## *SAQ 7*

*The time taken for a mechanic to repair a machine can be considered to be a Normal random variable with mean of 4 hours and standard deviation of 1 hour.*

*A bonus per machine is paid to the mechanic according to the following table:*

*If repair time is 2 hours or less a bonus of €3 is paid.*
*If repair time is greater than 2 but is 4 hours or less, a bonus of €2 is paid.*
*If repair time is between 4 and 5 hours, a bonus of €1 is paid..*
*Repair times in excess of 5 hours yield no bonus.*

*Using Standard Normal Tables the following probabilities can be determined.*

| *Repair time* | *Probability* | *Random Number Range* |
|---|---|---|
| *< 2* | *0.02* | *00 – 01* |
| *2 – 4* | *0.48* | *02 – 49* |
| *4 – 5* | *0.34* | *50 – 83* |
| *> 5* | *0.16* | *84 – 99* |

*Use a simulation of 12 machines to calculate the average bonus time that the mechanic obtains.*

*Random numbers:*
*20 43 74 78 79 89 63 23 47 26 90*
*03 73 70 59 12 93 90 24 52 34 55*
*17 73 69 13 44 00 76*

| *Machine Number* | *Random Number* | *Repair Time* | *Bonus* |
|---|---|---|---|
| *1* | *20* | *2 – 4* | *€2* |
| *2* | *43* | *2 – 4* | *€2* |
| *3* | *74* | *4 – 5* | *€1* |
| *4* | *78* | *4 – 5* | *€1* |
| *5* | *79* | *4 – 5* | *€1* |
| *6* | *89* | *> 5* | *-* |
| *7* | *63* | *4 – 5* | *€1* |
| *8* | *23* | *2 – 4* | *€2* |
| *9* | *47* | *2 – 4* | *€2* |
| *10* | *26* | *2 – 4* | *€2* |
| *11* | *90* | *> 5* | *-* |
| *12* | *03* | *2 – 4* | *€2* |

*Total Bonus = € 16.*

*This gives an average bonus of €1.33 per repair.*

## *SAQ 8*

*The probability distribution for the number of companies who purchase from ABC Ltd. is:*

| *No. of Sales* | *Probability* | | *No. Range* |
|---|---|---|---|
| 0 | $^5C_0\ (0.7)^0\ (0.3)^5$ | = 0.0 | - |
| 1 | $^5C_1\ (0.7)^1\ (0.3)^4$ | = 0.03 | 00 – 02 |
| 2 | $^5C_2\ (0.7)^2\ (0.3)^3$ | = 0.13 | 03 – 15 |
| 3 | $^5C_3\ (0.7)^3\ (0.3)^2$ | = 0.31 | 16 – 46 |
| 4 | $^5C_4\ (0.7)^4\ (0.3)^1$ | = 0.36 | 47 – 82 |
| 5 | $^5C_5\ (0.7)^5\ (0.3)^0$ | = 0.17 | 83 – 99 |

*The annual demand from each customer is normal with mean of 2,000 and standard deviation of 100.*

| *Interval* | *Mid-point* | *Probability* | *No. Range* |
|---|---|---|---|
| < 1650 | | | |
| 1650 – 1750 | 1700 | 0.01 | 00 |
| 1750 – 1850 | 1800 | 0.06 | 01 – 06 |
| 1850 – 1950 | 1900 | 0.24 | 07 – 30 |
| 1950 – 2050 | 2000 | 0.38 | 31 – 68 |
| 2050 – 2150 | 2100 | 0.24 | 69 – 92 |
| 2150 – 2250 | 2200 | 0.06 | 93 – 98 |
| 2250 – 2350 | 2300 | 0.01 | 99 |
| < 2350 | | | |

*Simulation Results*

| *Year* | *Random Number* | *No. of Customers* | *Random Number* | *Sales* |
|---|---|---|---|---|
| 1 | 27 | 3 | 05 | 1,800 |
| | | | 20 | 1,900 |
| | | | 30 | 1,900 |
| 2 | 85 | 5 | 21 | 1,900 |
| | | | 04 | 1,800 |
| | | | 67 | 2,000 |
| | | | 19 | 1,900 |
| | | | 13 | 1,900 |
| 3 | 12 | 2 | 46 | 2,000 |
| | | | 98 | 2,200 |

*A total of 19,300 units are sold during the three year period.*

# UNIT 10

## DECISION ANALYSIS

### AIM
The aim of this unit is to give an understanding of the basic ideas behind Decision Analysis.

### OBJECTIVES
- ❒ Learn how to describe a problem situation in terms of decisions to be made, chance event and consequences.
- ❒ Be able to analyse a simple decision analysis problem from both a payoff table and decision tree point of view.
- ❒ Be able to develop a risk profile and interpret its meaning.
- ❒ Be able to determine the potential value of additional information.

### REQUIRED READING
Students should read the following chapters of the mandatory text book David R. Anderson , Dennis J. Sweeney , Thomas A. Williams and Kipp Martin An Introduction To Management Science: Quantitative Approaches To Decision Making, 13th International Edition, Cengage Learning, ISBN-13: 9780538475655 in conjunction with this unit.

Chapter 13.1, 13.2 and 13.3

### INTRODUCTION
We first of all need to define the problem. We do this by identifying from the problem what the alternatives are and what each of the consequences associated with each of the alternatives will be.

There are two main situations – Decision making under uncertainty (without probabilities) and Decision making under risk. We will look at each of these separately.

### DECISION MAKING UNDER UNCERTAINTY
We first set up a payoff table which lists all the alternatives against their returns.

Example – Consider a builder trying to decide on the size of apartment complex to build. Suppose he has three alternative options small, medium or large and with each is an associated return depending on the state of the rental market. His return on his investment is as given in the following table which is a "Payoff" table.

| | States of Nature | |
|---|---|---|
| Decision Alternatives | Strong Demand | Weak Demand |
| Small | 8 | 7 |
| Medium | 14 | 5 |
| Large | 20 | -9 |

**Table 10.1**
**Payoff table for apartment complex**

The same information can also be displayed in a Decision Tree as in the following diagram. Figure 10.1

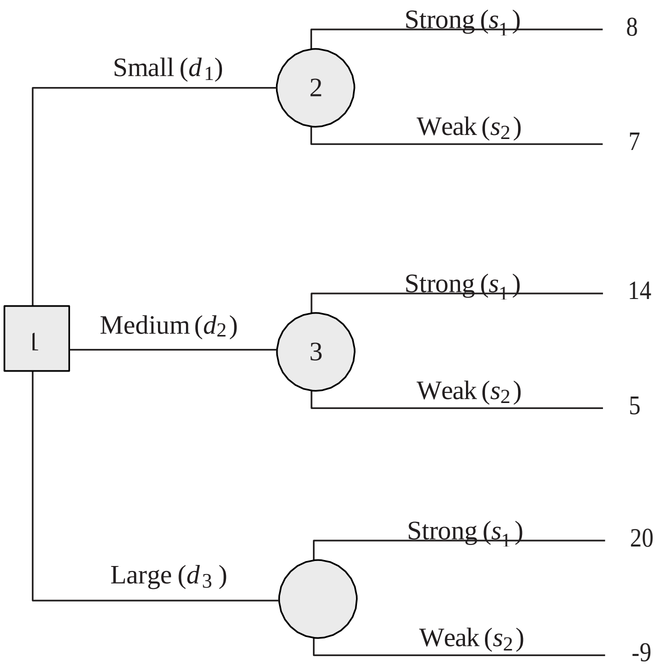



**Figure 10.1**
**Decision Tree for the apartment complex**

We will consider four different methods of Decision making without probabilities. The Optimistic approach, the Conservative approach, the Opportunity Loss and La Place.

### *OPTIMISTIC APPROACH*

This evaluates the alternatives looking at the best possible outcome and is known as the Maximax approach.

We take the maximum payoff for each alternative and then select the maximum from among these.

| Decision Alternative | Maximum Payoff |
|---|---|
| Small | 8 |
| Medium | 14 |
| Large | 20 |

**Table 10.2**
**Maximum Payoff for each Decision Alternative**

From the above table we now select the Large Decision Alternative as it gives the maximum payoff of 20.

### *CONSERVATIVE APPROACH*
This evaluates the alternatives looking at the minimum payoff for each alternative and fro those we select the best. This is known as the Maximin approach.

| Decision Alternative | Maximum Payoff |
|---|---|
| Small | 7 |
| Medium | 5 |
| Large | -9 |

**Table 10.3**
**Minimax Payoff for each Decision Alternative**

From the above table we now select the Small Decision Alternative as it gives the maximum payoff of 7.

### *OPPORTUNITY LOSS APPROACH*
This is also known as the Regret Approach. The regret approach gets its name from the amount you would lose by not making the right decision i.e. regretting your decision if wrong. A regret or Opportunity Loss table is set up showing how much the wrong decision costs. It is calculated by subtracting all the values in the column from the largest in each column. The table is shown below 10.4.

| | States of Nature | |
|---|---|---|
| Decision Alternatives | Strong Demand | Weak Demand |
| Small | 12 | 0 |
| Medium | 6 | 2 |
| Large | 0 | 16 |

**Table 10.4**
**Opportunity Loss table**

Having set up the Opportunity Loss table we now select the Decision Alternative that minimises the maximum opportunity loss. In this case we select Medium at a value of 6 as in the table below 10.5.

| Decision Alternative | Maximum Regret |
|---|---|
| Small | 12 |
| Medium | 6 |
| Large | 16 |

**Table 10.5**
**Maximum Regret or Opportunity Loss Table**

### *LA PLACE APPROACH*

This assigns equal probabilities to each State of Nature and calculates an expected value for each and selects the best return. This method is very unreliable as the probabilities assumed could be very inaccurate.

| | States of Nature | |
|---|---|---|
| Decision Alternatives | Strong Demand | Weak Demand |
| Small | 8 | 7 |
| Medium | 14 | 5 |
| Large | 20 | -9 |
| Probability | 0.5 | 0.5 |

**Table 10.6**
**La Place probabilities assigned**

Using the probabilities the expected value of the return can be calculated as:

Small 0.5 x 8 + 0.5 x 7 = 7.5

Medium 0.5 x 14 + 0.5 x 5 = 9.5

Large 0.5 x 20 + 0.5 x -9 = 5.5

Selecting the best option gives Medium at 9.5.

### DECISION MAKING UNDER RISK (WITH PROBABILITIES)

The calculations for this method are the same as those for La Place except that the probabilities have been calculated for the different states of nature. Let us assign the probabilities of 0.8 for strong demand n 0.2 for weak demand. From this we can calculate the Expected Value.

| | States of Nature | |
|---|---|---|
| Decision Alternatives | Strong Demand | Weak Demand |
| Small | 8 | 7 |
| Medium | 14 | 5 |
| Large | 20 | -9 |
| Probability | 0.8 | 0.2 |

**Table 10.7**
**Table for Expected Value calculations**

Using the probabilities the expected value of the return can be calculated as:

Small 0.8 x 8 + 0.2 x 7 = 7.8

Medium 0.8 x 14 + 0.2 x 5 = 12.2

Large 0.8 x 20 + 0.2 x -9 = 14.2

Selecting the best option gives Large at 14.2

## CONCEPT OF PERFECT INFORMATION

If information were available to ascertain with certainty the outcome of a scenario then we would need to know how much we should pay for such information. We call this the Value of Perfect Information.

| | States of Nature | |
|---|---|---|
| Decision Alternatives | Strong Demand | Weak Demand |
| Small | 8 | 7 |
| Medium | 14 | 5 |
| Large | 20 | -9 |

**Table 10.8**
**Pay off table for Perfect Information**

From the table we see that if there is a strong demand then we should select the Large Alternative and if there is a weak demand then the Small alternative. Given that the Strong demand has a probability of 0.8 and the Weak demand a probability of 0.2 the expected value can be calculated as follows:

Expected Value = 0.8 x 20 + 0.2 x 7 = 17.4

Earlier we calculated that the Expected value was 14.2 we have now calculated that with perfect information the value would be 17.4 therefore we should invest no more than 17.4 – 14.2 = 3.2 to acquire the information.

## *SAQ 1*

*A new air service is to be started which will depend on the market reaction. The decision to be made is as to the size of plane to use on the route either a large or small size. The estimated profit based on each size of plane is as shown in the following table.*

| | *Demand for Service* | |
|---|---|---|
| *Service* | *Strong* | *Weak* |
| *Large Plane* | *€960* | *-€490* |
| *Small Plane* | *€670* | *€320* |

*(a) What is your recommended decision using the Maximax, Maximin, Regret and La Place approaches?*

*(b) If the probability is confirmed as 0.7 for Strong Demand and 0.3 for Weak Demand then using the expected value method what is your recommendation?*

*(c) Calculate the value of Perfect Information.*

## *SAQ 2*

*A vineyard is considering two varieties of grape Chardonnay and Riesling, for planting either singly or as a combination to get the best return. The return received will be based on the strength of the demand. The following probability assessments have been made.*

| | *Riesling Demand* | |
|---|---|---|
| *Chardonnay Demand* | *Strong* | *Weak* |
| *Strong* | *0.20* | *0.25* |
| *Weak* | *0.50* | *0.05* |

*If only one type of grape is planted the return based on the strength of the demand is estimated to be €20,000 for Chardonnay if demand is weak and €70,000 if demand is strong. For Riesling it is estimated to be €25,000 if demand is weak and €45,000 if demand is strong.*

*If both types of grape are planted then the expected return is as in the following table.*

| | *Riesling Demand* | |
|---|---|---|
| *Chardonnay Demand* | *Strong* | *Weak* |
| *Strong* | *€60,000* | *€26,000* |
| *Weak* | *€40,000* | *€22,000* |

*(a) Draw a decision tree for the problem.*

*(b) If the probability is confirmed as 0.7 for Strong Demand and 0.3 for Weak Demand then using the expected value method what is your recommendation?*

*(c) If the probability for strong demand for Chardonnay and weak demand for Riesling is 0.05 and that the probability of strong demand for Riesling and strong demand for Chardonnay is 0.40. Does this change the recommended decision in b? If so what is the new decision? Assume that the probabilities when Chardonnay demand is weak are still 0.05 and 0.50.*

*(d) Suppose the annual profit projections fall to €50,000 when demand for Chardonnay is strong and Chardonnay grapes only are planted. Using the original probability assessments, determine how this change would affect the optimum decision.*

## ANSWERS TO SAQS

### SAQ 1

(a) *Maximax gives Large Plane at €960, Maximin gives Small Plane at €320 ,*

*Regret table:*

| | *Strong* | *Weak* | *Maximum Regret* |
|---|---|---|---|
| *Large Plane* | *0* | *810* | *810* |
| *Small Plane* | *290* | *0* | *290* |

*Therefore the minimum regret is associated with the Small Plane.*

(b) *Expected Value for Large Plane = 0.7(960) + 0.3(-490) = 525*
*Expected Value for Small Plane = 0.7(670) + 0.3(320) = 565*

*Therefore select Small Plane*

(c) *Value of Perfect Information.*
*If Strong Demand then select Large Plane*
*If Weak Demand then select Small Plane*

*Expected value of this strategy is 0.7(960) + 0.3(320) = 672 +96 = 768*

*Value of Perfect Information = 768 – 565 = 203*

### SAQ 2

(a) *Decision Tree as in solutions for chapter 14 page 12*

(b) *Expected Values:*
*Expected Value for Chardonnay = 0.55(20) + 0.45 (70) = 42.5*
*Expected Value for Riesling = 0.30(25) + 0.70(45) = 39.0*
*Expected for both grapes = 0.05(22) + 0.50(40) + 0.25(26) = 39.6*

*Therefore Chardonnay best alternative*

(c) *This changes the expected value in the case where both grapes are planted and when Riesling only is planted.*

*Expected Value for Riesling is not = 0.10(25) + 0.90(45) = 43.0*
*Expected Value for both grapes = 0.05(22) + 0.50(40) + 0.05(26) + 0.40(60) = 46.4*

*Therefore best alternative is now to plant both grapes.*

(d) *Only Chardonnay is affected by this change.*

*Expected Value for Chardonnay = 0.55(20) + 0.45(50) = 33.5*

*This makes the planting of Chardonnay less attractive. It is best to plant both grapes.*

# UNIT 11

## MARKOV PROCESSES AND APPLICATIONS

### AIM

The aim of this unit is to introduce the concept of a Markov process and to show how the process may be applied to a range of commercial, industrial and financial problems.

### OBJECTIVES

- To show how to construct a Markov matrix containing transition probabilities.
- To demonstrate how these transition matrices may be manipulated using the methods of matrix algebra.
- To investigate the properties of special types of transition matrices.

### REQUIRED READING

Read the following chapters of the mandatory text book in conjunction with this unit: Chapter 16

### 11.1 PROPERTIES OF MARKOV PROCESS

There is a story that the Russian mathematician A.A. Markov was having a drink in his favourite watering hole. After a while he got up to leave but was beaten to the door by another customer who was very much "under the weather". Markov, following this drunken person, noticed that he was staggering from left to right and right to left randomly. As he staggered right he hit a wall and bouncing off it, he staggered left. Staggering left, he eventually hit a fence and the process continued. Markov began to compute the likelihood of where the drunk would next meet the wall and then the fence and thus began the theory of *Random Walks*, which is a part of a more general theory now known as *Markov Processes*. A Markov Process is a stochastic process: *the current state of the system depends only upon the immediately preceding state of the system*. In the case of the drunk, his next move depends only on his current position and not on how he arrived at that position.

A Markov Process has the following characteristics:

- The process operates over time, and is examined at every time period
- In each period, the process will be in one of a number of states.
- The states are both mutually exclusive and collectively exhaustive.
- The system moves between states from time period to time period, and these movements can be described by transition probabilities, which remain constant.
- The probability of the process being in a given state in a particular period depends only on its state in the preceding period. It is independent of all earlier periods.

Thus, there are two elements central to constructing a Markov model of a system, namely:

- **The number of States of the system**
- **The probabilities of moving between these states**

Markov analysis has been used in a variety of situations which include:

- The migration of people from farming to other sectors of the economy
- Consumer brand switching
- Certain classes of queuing problems
- Inspection and replacement strategies
- Finance
- Telecommunications

Fundamental to the Markov Process is the concept that at a particular point in time, the system under review is in a *certain* state. For example, if we examine a machine at a point in time, and the machine is faulty, then the Markov Process that is used to describe the machine for that point in time is in a state called 'faulty'. Part of the analysis required for Markov Processes is to identify what are the possible states that the process could be in. Returning to the machine example, the possible states for a machine could be as follows: {Working Well, Needs Adjustment, Faulty}.

In general, the Markov Process attempts to model the movement of the process over time among the various states. In the machine example, a question that the service department might have would relate to the probability of the machine being in the "Working Well" state at the last inspection, and then at the next inspection being in the "Faulty" state. The probability of a system moving from one state to another in a defined period is known as a *Transition Probability*. Transition probabilities are dependent only on the current state of the system.

A Markov chain is a sequence $x_1, x_2, x_3, \ldots$ of random variables. The set of possible values for these variables is called the *state space*, the value of $x_n$ being the state of the process at time *n*. The conditional value of $x_{n+1}$, given all past states, is equal to the conditional value of $x_{n+1}$, given only the state in the current time period, $x_n$

This can be expressed mathematically:

$P\{x_{n+1}=s_{n+1} | x_n=s_n, x_{n-1}=s_{n-1}, x_{n-2}=s_{n-2},\ldots, x_0=s_0\}=P\{x_{n+1}=s_{n+1} | x_n = s_n\}$.

Here, the set of random variables x represents a stochastic process and $s_n$ represents the state of the system at time n.

Such a Markov process, where the probability of being in a particular state at any one time only depends on the state in the immediately preceding time period, is often called a *Markov Chain*.

The probability $\mathbf{P\{x_{n+1} = s_{n+1} | x_n = s_n\}}$ is known as the *First Order* transition probability and has the characteristic that it does not change with time. Hence, it is known as a *Stationary* transition probability.

Since the number of possible states in a Markov Chain is finite, a square matrix **P**, known as the ***Transition Probability Matrix*** is commonly used in Markov analysis.
Consider:

| | | **To State** | | | | |
|---|---|---|---|---|---|---|
| | | **1** | **2** | **3** | **...........** | **n** |
| | **1** | $P_{11}$ | $P_{12}$ | $P_{13}$ | **...........** | $P_{1n}$ |
| | **2** | $P_{21}$ | $P_{22}$ | $P_{23}$ | **...........** | $P_{2n}$ |
| **From State** | **3** | $P_{31}$ | $P_{32}$ | $P_{33}$ | **...........** | $P_{3n}$ |
| | **:** | **:** | **:** | **:** | **...........** | **:** |
| | **n** | $P_{n1}$ | $P_{n2}$ | $P_{n3}$ | **...........** | $P_{nn}$ |

where n represents the number of possible states and $P_{ij}$ is the probability of making a transition from state i in a given time period to state j in the next period. Hence $P_{12}$ is the probability of making a transition from state 1 in a given time period to state 2 in the next time period. Thus the Transition Probability Matrix is

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & \dots & P_{1n} \\ P_{21} & P_{22} & P_{23} & \dots & P_{2n} \\ P_{31} & P_{32} & P_{33} & \dots & P_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ P_{n1} & P_{n2} & P_{n3} & \dots & P_{nn} \end{bmatrix}$$

One of the properties of the Transition Probability Matrix is that the sum of the probabilities in each row must be 1:

$P_{11} + P_{12} + P_{13} + \dots\dots\dots + P_{1n} = 1$
$P_{21} + P_{22} + P_{23} + \dots\dots\dots + P_{2n} = 1$ etc.

$$\sum_j P_{ij} = 1 \text{ for } i = 1 \dots n$$

## 11.2 COMPUTING TRANSITIONAL PROBABILITIES

A popular way to describe how to compute transitional probabilities is through the following brand switching example. Suppose a market research firm has carried out a survey which shows the buying behaviour of 7000 households in a medium-sized rural town which has three family-run small general stores. The study was carried out at two distinct points in time and the results are shown in Figure 11.1.

| **GENERAL STORE** | **TIME PERIOD 1 No. of Customers** | **TIME PERIOD 2 No. of Customers** |
|---|---|---|
| **WRIGHTS(A)** | **1800** | **2400** |
| **LAWLORS(B)** | **4000** | **3000** |
| **AHERNES(C)** | **1200** | **1600** |

**Figure 11.1:**
**Buying Behaviour in a Rural Town**

Figure 11.1 would seem to indicate that significant shifts have occurred in general store preferences. However, the market research report also noted the movements of customers from one store to another and this information is shown in Figure 11.2.

| | | TO STORE | | |
|---|---|---|---|---|
| | FROM | (2400) | (3000) | (1600) |
| | STORE | A | B | C |
| (1800) | A | 1600 | 100 | 100 |
| (4000) | B | 500 | 2800 | 700 |
| (1200) | C | 300 | 100 | 800 |

**Figure 11.2:**
**Customer Movements from Store to Store**

If we assume that the same **relative** movements amongst stores will continue, that is it does not change over time (i.e. it is *stationary)*, we can develop the transition probability matrix in the following way: The process is to divide the numbers in each row by the corresponding row total i.e. 1800 for row A, 4000 for row B, and 1200 for row C. Rounding to two decimal places (more needed if the probabilities in each row are to sum to 1.0) results in Figure 11.3.

| | | TO STORE | | |
|---|---|---|---|---|
| | FROM STORE | A | B | C |
| | A | 0.89 | 0.06 | 0.06 |
| P = | B | 0.13 | 0.70 | 0.18 |
| | C | 0.25 | 0.08 | 0.67 |

**Figure 11.3:**
**The Transition Probability Matrix**

The significance of obtaining the transition matrix for a Markov process is that it permits the prediction of future states of the system. The prediction of future states requires only that you know the starting state of the system and the transition matrix.

## *SAQ 1*

*A town has three supermarkets – A, B and C and 1000 families. Each family in the town visits one of the supermarkets once each week. A survey of shoppers in a particular week gives the following results:*

- *50 % of families went to supermarket A and 30% went to supermarket B.*
- *70% of those who shop at supermarket A will do so again the following week; 10 % will shop at supermarket B and 20% will shop at supermarket C*
- *Of those who shop at supermarket B, 10% will switch to supermarket A in the following week and 20 % to supermarket C.*
- *Of those who shop at supermarket C, 5% will switch to supermarket A in the following week and 5% to supermarket B.*

*How many families will visit each of the supermarkets in the week after the survey?*

## *SAQ 2*

*For the problem given in SAQ1, what will the distributions be in weeks 3 and 4?*

### 11.3 THE TRANSITION PROBABILITY MATRIX AND FUTURE STATES

The initial state of a system can be represented as a one-dimensional matrix (or row-vector) of probabilities which sum to 1.0. The state of a system at time t = 0, for instance, is:
**p (0)** = [ $p_1(0)$ , $p_2(0)$ , $p_3(0)$]

Where:

**p (0)** = a **vector** of $p_i(0)$ values

$p_1(0)$ = the probability of being in state 1 at time t = 0.
$p_2(0)$ = the probability of being in state 2 at time t = 0.
$p_3(0)$ = the probability of being in state 3 at time t = 0.

In order to compute the probability of being in state *k* at time 1, if we are currently at time t = 0, we compute:

**p (1) = p(0) P**
Where:

**p (1)** = the vector of state probabilities at time t = 1.
**p (0)** = the vector of state probabilities at time t = 0.

**P** = the transition probability matrix

In terms of our store-switching model we can then write that for time t=1,

$$\mathbf{p(1)} = [1800, 4000, 1200]\begin{bmatrix} 0.89 & 0.06 & 0.06 \\ 0.13 & 0.70 & 0.18 \\ 0.25 & 0.08 & 0.67 \end{bmatrix}$$

We use matrix multiplication methods to find p(1):

1800 (0.89) + 4000 (0.13) + 1200 (0.25) = 2422
1800 (0.06) + 4000 (0.70) + 1200 (0.08) = 3004
1800 (0.06) + 4000 (0.18) + 1200 (0.67) = 1632

Hence p (1) = [2422, 3004, 1632]

For later time periods, we have
**p (2)** = **p (1) P** = **p (0) P.P** = **p (0)** $\mathbf{P^2}$
**p (3)** = **p (2) P** = **p (1) P.P** = **p (0)** $\mathbf{P^3}$
and so on.

Thus **in general** we can say:

$$\mathbf{p(n) = p(0)\ P^n}$$

In terms of our store-switching model we can then write, for say time t = 3,

$$\mathbf{P}(3) = [1800, 4000, 1200]\begin{bmatrix} 0.89 & 0.06 & 0.06 \\ 0.13 & 0.70 & 0.18 \\ 0.25 & 0.08 & 0.67 \end{bmatrix}^3$$

## *SAQ 3*

*Consider a car rental agency that has rental offices at each of a city's two airports. Customers are allowed to return a rented car to either airport, regardless of which airport they rented from. In this first level of analysis, it can be assumed that all rentals will be for only one day, and at the end of the day, every car will be returned to one of the two rental offices.*

*An analysis by the manager has yielded the following information: 70% of cars rented from Airport A tend to be returned to that airport, and 30% of the Airport A cars tend to be returned to Airport B; 10% of the cars rented from Airport B are returned to Airport A, and 90% of the cars are returned to Airport B.*

*The manager currently has 100 cars at Airport A, and 80 cars at Airport B. It can be assumed that demand for cars greatly exceeds availability, and thus all cars available at the airports are rented.*

*The manager has asked you to estimate the number of cars he will have at each airport at the end of each of the next 3 days.*

## *SAQ 4*

*Assume that tomorrow's weather depends only on today's weather and not on past weather conditions. The meteorological office has provided the following transition matrix:*

| | | ***Tomorrow*** | |
|---|---|---|---|
| | | **Rain** | **Fine** |
| ***Today*** | **Rain** | **0.7** | **0.3** |
| | **Fine** | **0.4** | **0.6** |

*We are playing tennis in two days time, and it is currently fine outside. Calculate the probability that it will be fine in two days time.*

## *SAQ 5*

*A student of sociology suggests that all families can be classified into one of three categories, Upper Class, Middle Class, and Lower Class. From his analysis of the movement of families between these classes from one generation to the next, he suggests that the transition between the social classes of successive generations in a family can be regarded as a Markov Process. Specifically, in the case of sons, it is presumed that the class of a son will depend on his father's class and not on his grandfather's class. The student's analysis of several families suggests that the transition matrix is as follows:*

| | | ***Next Generation*** | | |
|---|---|---|---|---|
| | | **Upper** | **Middle** | **Lower** |
| | **Upper Class** | **0.45** | **0.48** | **0.07** |
| ***This Generation*** | **Middle Class** | **0.05** | **0.70** | **0.25** |
| | **Lower Class** | **0.01** | **0.50** | **0.49** |

*Calculate the probability that the great-grandson of a Lower Class person reaches the Upper Class.*

## 11.4 THE STEADY STATE AND ITS COMPUTATION

In very many cases the Markov Process will converge to a *Steady State,* or equilibrium, condition. This means that the transition probabilities will be unaltered from period to period once a critical point has been reached and the system is independent of the initial state that it started in.

The steady state can be computed by observing that as it is reached, multiplication of the state vector by the transition probabilities leaves the state vector unaltered. Thus we have for some sufficiently large n:

$$\mathbf{p}(n) = \mathbf{p}(n-1)\mathbf{P}$$

Let us assume that at steady state the state vector can be expressed as:

$$S = [s_1, s_2, s_3]$$

Where:
S = the vector of steady state probabilities
$s_1$ = Value of state condition 1
$s_2$ = Value of state condition 2
$s_3$ = Value of state condition 3

Then,

$$\mathbf{s} = \mathbf{s}.\mathbf{P}$$

defines the steady state condition and solving this matrix condition for s yields the steady state if it exists. This may be expressed also as:

$$[s_1, s_2, s_3] = [s_1, s_2, s_3] \begin{bmatrix} P_{11} P_{12} P_{13} \\ P_{21} P_{22} P_{23} \\ P_{31} P_{32} P_{33} \end{bmatrix}$$

and, since the sum of the steady state probabilities must be 1, we also have the equation
$s_1 + s_2 + s_3 = 1$.

Returning to our brand-switching model we have:

$$[s_1, s_2, s_3] = [s_1, s_2, s_3]\begin{bmatrix} 0.89 & 0.06 & 0.06 \\ 0.13 & 0.70 & 0.18 \\ 0.25 & 0.08 & 0.67 \end{bmatrix}$$

and

$s_1 + s_2 + s_3 = 1$.

Hence to solve for the three unknown values $s_1$, $s_2$, $s_3$ we have four equations/conditions and so one equation will be redundant.

Using matrix multiplication this yields:

$$s_1 = 0.89\, s_1 + 0.13\, s_2 + 0.25\, s_3$$
$$s_2 = 0.06\, s_1 + 0.70\, s_2 + 0.08\, s_3$$
$$s_3 = 0.06\, s_1 + 0.18\, s_2 + 0.67\, s_3$$

and

$$s_1 + s_2 + s_3 = 1.$$

On tidying up we have:

$$0 = -\,0.11\, s_1 + 0.13\, s_2 + 0.25\, s_3$$
$$0 = 0.06\, s_1 - 0.30\, s_2 + 0.08\, s_3$$
$$0 = 0.06\, s_1 + 0.18\, s_2 - 0.33\, s_3$$
$$1 = s_1 + s_2 + s_3$$

Here we have a system of linear equations, which we can solve for $s_1$, $s_2$ and $s_3$. Since there are three unknowns and four equations, we can drop one of the equations from the system. Any one of the first three, but not the fourth, can be dropped. The reason for this is that the first three equations are dependent, that is, one of them can be derived from the other two. (Observe that, if we add any two of these three equations and change the sign, we get the third.)

Eliminating the third of the equations gives us the three simultaneous equations in three unknowns:

$$0 = -\,0.11\, s_1 + 0.13\, s_2 + 0.25\, s_3$$
$$0 = 0.06\, s_1 - 0.30\, s_2 + 0.08\, s_3$$
$$1 = s_1 + s_2 + s_3$$

Solving these equations yields the steady-state probabilities:

$$s_1 = 0.635417,\ s_2 = 0.177083$$
$$s_3 = 0.1875$$

**Remark**: note that the computation of the steady state made/needed no reference to the initial state of the system!

To arrive at the number of households buying from the stores at steady-state we multiply the probabilities by 7000 to give for stores A, B and C respectively:

4448, 1240, 1312

Finally Figure 11.4 illustrates the shift in buying behaviour from the time the survey was conducted to the steady state situation.

| STORE | TIME PERIOD | |
|---|---|---|
| | AT SURVEY | AT STEADY-STATE |
| A | 1800 | 4448 |
| B | 4000 | 1240 |
| C | 1200 | 1312 |

**Figure 11.4:**
**The Shift in Buying Behaviour**

Store B needs to take action now! Was this evident from the original transition matrix?

▶ See the video **V37_Markov_1**, which demonstrates the calculation of steady state probabilities for a Markov process. https://goo.gl/wFHQwa

## *SAQ 6*

Let $\mathbf{P} = \begin{bmatrix} 0.4 & 0.2 & 0.4 \\ 0.3 & 0.4 & 0.3 \\ 0.3 & 0.3 & 0.4 \end{bmatrix}$

*be the transition matrix of a Markov process*

*Find:*

*(i)* $\mathbf{P}^2$ *(ii)* $\mathbf{P}^4$

***Deduce the steady state vector.***

## *SAQ 7*

*Consider the 1-step transition matrix for the families shopping at supermarkets in SAQ 1:*

| | | *To* | | |
|---|---|---|---|---|
| | | **A** | **B** | **C** |
| | **A** | **0.7** | **0.1** | **0.2** |
| ***From*** | **B** | **0.1** | **0.7** | **0.2** |
| | **C** | **0.05** | **0.05** | **0.9** |

*Use simultaneous equations to calculate the steady state probabilities.*

## *SAQ 8*

*A woman either cycles or walks to work. If she cycles to work on one day, then the probability that she will cycle on the following day is 0.8. If she walks to work, then the probability that she will walk on the following day is 0.7.*

*In the long run, how often does she walk to work?*

## *SAQ 9*

*The social mobility transition matrix used in SAQ 5 was as follows:*

| | | ***Next Generation*** | | |
|---|---|---|---|---|
| | | **Upper** | **Middle** | **Lower** |
| | **Upper Class** | **0.45** | **0.48** | **0.07** |
| ***This Generation*** | **Middle Class** | **0.05** | **0.70** | **0.25** |
| | **Lower Class** | **0.01** | **0.50** | **0.49** |

*Calculate the proportion of society that will be in each of the three classes in the long run.*

## 11.5 Absorbing States

A state is known as *Absorbing or trapping* if it is impossible to leave that state once it is reached and it is usually possible to reach it from other states in a finite number of transitions. This occurs or is signalled if any transition probability on the main diagonal, from upper left to lower right of the matrix, is equal to 1.0. Examples of trapping states might include machine breakdown, death, bad debt etc.

A very well-known example of Markov Chains with absorbing states is the *Accounts Receivable* or *Bad Debt* example. Suppose a firm has developed the following transition matrix, shown in Figure 11.5:

$$
\begin{array}{c c}
 & \begin{array}{ccccc} \mathbf{p} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{b} \end{array} \\
\begin{array}{c} \mathbf{p} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{b} \end{array} &
\left[\begin{array}{ccccc}
\mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0.55} & \mathbf{0} & \mathbf{0.45} & \mathbf{0} & \mathbf{0} \\
\mathbf{0.70} & \mathbf{0} & \mathbf{0} & \mathbf{0.30} & \mathbf{0} \\
\mathbf{0.50} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0.50} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}
\end{array}\right]
\end{array}
$$

**Figure 11.5:**
**The Transition Matrix of Outstanding Accounts**

There are five states. The state p represents an account paid. The states 1, 2 and 3 represent an account outstanding for that number of months. The state b represents a bad debt, which occurs after three months if the account hasn't been paid. The absorbing states are at the top left and bottom right of the main diagonal. Thus, once an account has been paid, it can never leave that state, and it is also considered that a bad debt will always remain in that state. A lot of useful information can be obtained from this type of matrix by use of fairly simple matrix operations. Firstly, however, the matrix must be put in to the general form as follows:

$$
\begin{array}{c|ccccc|}
 & p & b & 1 & 2 & 3 \\
\hline
p & 1 & 0 & 0 & 0 & 0 \\
b & 0 & 1 & 0 & 0 & 0 \\
1 & 0.55 & 0 & 0 & 0.45 & 0 \\
2 & 0.70 & 0 & 0 & 0 & 0.30 \\
3 & 0.50 & 0.50 & 0 & 0 & 0
\end{array}
= P = \left[\begin{array}{c|c} I & O \\ \hline R & Q \end{array}\right]
$$

Where we are able to identify the following submatrices within our original matrix:

I = an identity matrix
O = a null matrix (where all entries are zero)
R = a matrix of the transition probabilities of being absorbed in the next period.
Q = a square matrix containing transition probability for movement between all non-absorbing states.

Here

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ O = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ R = \begin{bmatrix} 0.55 & 0 \\ 0.70 & 0 \\ 0.50 & 0.50 \end{bmatrix},\ \text{and } Q = \begin{bmatrix} 0 & 0.45 & 0 \\ 0 & 0 & 0.3 \\ 0 & 0 & 0 \end{bmatrix}$$

We are now able to calculate useful information relating to the problem in hand, through matrix operations:

a. The *Fundamental Matrix* F gives the expected number of times the system would be in any of the non-absorbing states before absorption occurs:

$F = (I - Q)^{-1}$

Hence, for our example above:

$$F = \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0.45 & 0 \\ 0 & 0 & 0.3 \\ 0 & 0 & 0 \end{bmatrix}\right]^{-1} = \begin{bmatrix} 1 & -0.45 & 0 \\ 0 & 1 & -0.3 \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0.45 & 0.135 \\ 0 & 1 & 0.3 \\ 0 & 0 & 1 \end{bmatrix}$$

For this example, if an account is in state 1, the expected number of times it would occupy state 2 would be 0.45 before the debt is either paid off or turns into a bad debt.

(Recall that $A^{-1}$ is the inverse of matrix A.)

b. The *E Matrix* gives the probability of eventually being absorbed given any starting state:

$$E = FR = \begin{bmatrix} 0.932 & 0.068 \\ 0.85 & 0.15 \\ 0.5 & 0.5 \end{bmatrix}$$

For example, if the account is in the first month, there is a 0.932 probability that it will eventually be paid, and a 0.068 probability that it will become a bad debt.

c. The **P, P$^2$ and P$^3$** matrices permit the calculation of the change in the flow of funds from month to month. To do this we extract the R matrix from each P matrix. If these are called R1, R2 and R3, then we calculate:

ΔR1 = R1 – R0

ΔR2 = R2 – R1

ΔR3 = R3 – R2

For our example,

$$\Delta R2 = \begin{bmatrix} 0.315 & 0 \\ 0.15 & 0.15 \\ 0 & 0 \end{bmatrix}$$

This represents the change in cash flow during the second month. From this matrix we observe that if the account is in month 2, then 0.15 (or 15%) of the funds will be paid, while 0.15 will become bad debts.

d. Suppose the account books show accounts receivable as:

One Month: €1,000,000

Two Months: €800,000

Three Months: €200,000

We can put this in the form of a *Status Vector*:

V = [1,000,000: 800,000: 200,000].

We can now find out the portion of the €2m flowing in during each of the three months. This is done by multiplying the status vector by each ΔR matrix:

$$V\,\Delta R1 = [1{,}000{,}000{:}\ 800{,}000{:}\ 200{,}000] \begin{bmatrix} 0.55 & 0 \\ 0.70 & 0 \\ 0.50 & 0.5 \end{bmatrix}$$

From which we find:

V ΔR1 = (1,210,000 100,000)

Similar calculations will give:

V ΔR2 = (435,000 120,000)

V ΔR3 = (67,500 67,500).

These show for each time period the amount paid and the amount becoming bad debts. Of the €2m outstanding, €1,210,000 is paid in the first month while €100,000 becomes bad debt; in the second month, €435,000 is paid and €120,000 becomes bad debt; and in the third month, €67,500 is paid while €67,500 becomes bad debt. Observe that these six figures fully account for the €2m.

▶ See the video **V38_Markov_2**, which concerns a Markov process with absorbing states. https://goo.gl/O1Qrxf

## *SAQ 10*

a. Consider the following transition matrix:

$$\begin{array}{cc} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} 3/4 & 1/4 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1/5 & 4/5 \end{bmatrix} \end{array}$$

Compute the steady state of this matrix and comment on the solution.

b. Suppose the matrix is altered to:

$$\begin{array}{cc} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} 3/4 & 1/4 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 2/3 & 1/3 \\ 0 & 0 & 1/5 & 4/5 \end{bmatrix} \end{array}$$

Does this configuration impact on the steady state? Why?

## SUMMARY

In this unit the notion of a Markov chain was introduced and some basic examples explored. The area of Markov analysis is one that is broad and founded on mathematical and probability theory and whilst technical it has enjoyed many applications in areas as diverse as telecommunications, finance, marketing etc.

The system of interest is characterised by a finite number of possible states and evolves over (discrete version) time and can be indexed by time period n=0,1,2… The concept of the state vector $\mathbf{p}^{(n)}$ and the transition matrix **P** are central to the theoretical development and description of the systems evolution. For a system with n possible states the state vector consists of n elements which sum to 1.0 and the transition matrix is nxn with each row summing to 1.0 ($\sum_j \mathbf{P}_{ij} = 1$ for i =1…n). The elements of **P** are denoted by $P_{ij}$ where $P_{ij}$ is the probability of making a transition from state i in a given time period to state j in the next period and we assume this is a constant probability.

Certain simplifying assumptions regarding the system were adopted and lead in turn to key results.

One defining feature is that of the probability of being in a particular state at any one time depending only on the state in the immediately preceding time period. This more formally was expressed as

$P\{x_{n+1}=s_{n+1} | x_n=s_n, x_{n-1}=s_{n-1}, x_{n-2}=s_{n-2},…, x_0=s_0\}=P\{x_{n+1}=s_{n+1} | x_n = s_n\}$.

With the probability P $\{x_{n+1} = s_{n+1} \mid x_n = s_n\}$ being the *First Order* transition probability and has the characteristic that it does not change with time (it is a *Stationary* transition probability).

We saw also that as the system evolved according to:

$$\mathbf{p(n) = p(n\text{-}1)\ P}$$

which can also be linked (a chain) back to the initial state as

$$\mathbf{p(n) = p(0)\ P^{n}}$$

A fundamental concept was that of the system converging to a steady state whereby the state vector remains unchanged from one time period to the next. This steady state **s** if it exists is a long run outcome for the system and is characterised by the matrix equation:

$$\mathbf{S = S.P}$$

This set of simultaneous equations can then be solved for **s** together with the condition that the elements of **s** sum to 1.0. This steady state is independent of the initial state of the system.

Other special cases of interest are the presence of absorbing or trapping states. These are characterised by an entry of 1.0 on a retention cell (main diagonal from left to right) and hence zeros elsewhere in that row. A transition matrix may have several trapping states and ultimately all other states will empty/transition to these and it is of interest to estimate the proportions falling into each. Examples might be bad debt exposure, hospital treatments with cure/death outcomes, examination attempts with pass/fail outcomes in education etc.

In summary upon adopting some simplifying assumptions the evolution of a system over time can be modelled once the transition matrix is known. All future states can be computed using successive application of the transition matrix, and a steady state can be calculated (if it exists).

## FINAL COMMENTS

The general field of Markov analysis is very mathematical and outside the scope of this unit. However, the basic treatment herein is sufficient to demonstrate the usefulness of the analysis in examining systems which vary randomly over time both discretely and in continuous time. The student should consider using for example the Excel spreadsheet programming for the often onerous task of the matrix multiplications together with the interactive charting facilities to graph the system evolution over time. It is also worth exploring other example of the application of Markov analysis using web or text sources.

## SOME USEFUL REFERENCES AND RESOURCES

http://people.brunel.ac.uk/~mastjjb/jeb/or/contents.html
http://www.itemuk.com/markov.html
http://www.relex.com/products/markov.asp

## ANSWERS TO SAQS

### *SAQ 1*

Firstly, set up the transition matrix:

| | TO SUPERMARKET | | |
|---|---|---|---|
| **FROM SUPERMARKET** | **A** | **B** | **C** |
| **A** | **0.7** | **0.1** | **0.2** |
| **B** | **0.1** | **0.70** | **0.2** |
| **C** | **0.05** | **0.05** | **0.9** |

Supermarket A starts in week 1 with 50% of the market. In week 2 A will have 70% of its own share - 35% of the population. In addition, supermarket A will gain 10% of supermarket B's share (30%) - 3 % of the population. Also, supermarket A will gain 5% of supermarket C's share (20%) - 1% of the population. In total, supermarket A will have 39% of the market – 390 families will use supermarket A, down from 500 in the week of the survey.

Similarly, it can be shown that 270 families will shop in supermarket B, down from 300, and 340 families will shop in supermarket C, up from 200.

### *SAQ 2*

***Week 2:***
*From SAQ 1, the distribution in Week 2 is:*
*Supermarket A: 39%; Supermarket B: 27%; Supermarket C: 34%*

***Week 3:***
*Supermarket A:*
$39 \times 0.7 + 27 \times 0.1 + 34 \times 0.05 = 31.7$ *i.e. 317 families*

*Supermarket B:*
$39 \times 0.1 + 27 \times 0.7 + 34 \times 0.05 = 24.5$ *i.e. 245 families*

*Supermarket C:*
$39 \times 0.2 + 27 \times 0.2 + 34 \times 0.9 = 43.8$ *i.e. 438 families*

*Therefore the distribution in week 3 is:*
*Supermarket A: 31.7%; Supermarket B: 24.5%; Supermarket C: 43.8%*

***Week 4:***
*Continuing the process gives:*
*Supermarket A: 26.8%; Supermarket B: 22.5%; Supermarket C: 50.7%*

## *SAQ 3*

The transition probability matrix is as follows:

| | A | B |
|---|---|---|
| A | 0.7 | 0.3 |
| B | 0.4 | 0.6 |

At the start the state vector is [100, 80]

At the end of day 1 the state vector is:

$$[100, 80] \begin{bmatrix} 0.7 & 0.3 \\ 0.1 & 0.9 \end{bmatrix}$$

$$= [78, 102]$$

At the end of day 2 the state vector is:

$$[78, 102] \begin{bmatrix} 0.7 & 0.3 \\ 0.1 & 0.9 \end{bmatrix}$$

$$= [64.8, 115.2]$$

At the end of day 3 the state vector is:

$$[64.8, 115.2] \begin{bmatrix} 0.7 & 0.3 \\ 0.1 & 0.9 \end{bmatrix}$$

$$= [56.88, 123.12]$$

## *SAQ 4*

The initial state vector is (0, 1)

Two step transition matrix, $P^2$,

$$= P \times P = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$$

$$= \begin{bmatrix} 0.61 & 0.39 \\ 0.52 & 0.48 \end{bmatrix}$$

Therefore the state vector for two days from now is (0.52, 0.48), and thus the probability that it will be fine in two days, given that it is fine today is 0.48

### *SAQ 5*

The son is one step into the future, the grandson is two steps into the future, and the great grandson is three steps. The appropriate transition matrix is the three step matrix, $P^3$.

$P^3 = P \times P \times P$

For the transition matrix in this problem

$$P^2 = \begin{bmatrix} 0.2272 & 0.5870 & 0.1858 \\ 0.0600 & 0.6390 & 0.3010 \\ 0.0344 & 0.5998 & 0.3658 \end{bmatrix}$$

$$P^3 = \begin{bmatrix} 0.1334 & 0.6129 & 0.2537 \\ 0.0620 & 0.6266 & 0.3114 \\ 0.0491 & 0.6193 & 0.3316 \end{bmatrix}$$

The 3-step transition probability from Lower Class to Upper Class has probability 0.0491.

### *SAQ 6*

$$P = \begin{bmatrix} 0.4 & 0.2 & 0.4 \\ 0.3 & 0.4 & 0.3 \\ 0.3 & 0.3 & 0.4 \end{bmatrix}$$

$$P^2 = P \times P = \begin{bmatrix} 0.34 & 0.28 & 0.38 \\ 0.33 & 0.31 & 0.36 \\ 0.33 & 0.30 & 0.37 \end{bmatrix}$$

$$P^4 = P^2 \times P^2 = \begin{bmatrix} 0.3334 & 0.2960 & 0.3706 \\ 0.3333 & 0.2965 & 0.3702 \\ 0.3333 & 0.2964 & 0.3703 \end{bmatrix}$$

Steady state vector = **[**0.333, 0.296, 0.370**]**

correct to 3 significant figures.

## *SAQ 7*

Let the steady state probabilities be **[**x, y, z**]**

Now x + y + z = 1, thus replace z by 1 - x - y.

The condition for the steady state probabilities are that:

**[**x, y, 1- x – y**]** $\times$ P = **[**x, y, 1- x- y**]**

Using the P matrix from the problem gives three equations:

$0.7x + 0.1y + 0.05(1-x-y) = x$
$0.1x + 0.7y + 0.05(1-x-y) = y$
$0.2x + 0.2y + 0.9(1-x-y) = 1 - x - y.$

Using any two of these equations, we can solve for x and y. Using equations 1 and 2 gives:

$0.35x - 0.05y = 0.05$
$-0.05x + 0.35y = 0.05$

Solving these two equations gives x = .1667 and y = .1667

Since x + y + z = 1, then z must equal .6667.

The limiting probabilities are thus **[**0.1667 0.1667 0.6667**]**.

## *SAQ 8*

The transition matrix is:

| | **Cycle** | **Walk** |
|---|---|---|
| **Cycle** | **0.8** | **0.2** |
| **Walk** | **0.3** | **0.7** |

Let **[**x, 1-x**]** be steady state vector, where x is the probability of cycling to work.

Therefore $0.8x + 0.3(1 - x) = x$

$8x + 3 - 3x = 10x$
$-5x = -3$

Thus x = 3/5, and y = 1 - x = 2/5

Therefore the steady state vector is **[**3/5, 2/5**]**

In the long term, the man walks to work 2/5 of the time.

## *SAQ 9*

Let the steady state probabilities be **[**x, y, 1 - x – y**].**

Multiplying by the transition probability matrix yields the three equations:
$0.45x + 0.05y + 0.01(1-x-y) = x$
$0.48x + 0.7y + 0.5(1-x-y) = y$
$0.07x + 0.25y + 0.49(1-x-y) = 1-x-y.$

Take the first two and simplify:
$0.56x - 0.04y = 0.01$
$0.02x + 0.8y = 0.5$

Solving gives x = 0.07 and y = 0.62. Therefore z = 0.31.

Thus this society will be 7% Upper Class, 62% Middle Class, and 31% Lower Class.

Notice that the $P^3$ matrix in SAQ 5 is converging to these numbers.

## *SAQ 10*

a. *For the steady state:*

$$[s_1, s_2, s_3, s_4] = [s_1, s_2, s_3, s_4]\begin{bmatrix} 3/4 & 1/4 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1/5 & 4/5 \end{bmatrix}$$

*and*

$$s_1 + s_2 + s_3 + s_4 = 1$$

*Which gives a steady state of:*

$$\begin{bmatrix} 2/3 & 1/3 & 0 & 0 \\ 2/3 & 1/3 & 0 & 0 \\ 2/3 & 1/3 & 0 & 0 \\ 2/3 & 1/3 & 0 & 0 \end{bmatrix}$$

*Once the system reaches A or B it remains in one or other of those states. Note that the first two states can be reached from states 3 and 4 which will eventually empty as state 1 and 2 form a generalised trapping set i.e. once in state 1 or two one cannot exit. Hence the problem is essentially two dimensional and one can assume* $s_3$, $s_4 = 0$ and solve for $s_1$ and $s_2$ using the 2x2 matrix in the top left corner of the transition matrix.

*b.*

Let (x, y, z, w) represent the steady state vector. Applying the transition probability matrix from the question (and multiplying by 15 to eliminate the fractions) gives

$3x + 2y = 4x$
$x + 2y = 4y$

10z + 3w = 15z
5z + 12w = 15w.

The first two of these result in the same equation, x = 2y. Similarly, the last two give 3w = 5z. Together with x + y + z + w = 1, we now only have three equations in four unknowns. Therefore there is no unique solution. This can be demonstrated using the Gaussian Elimination Method, which shows the general solution to be

x = (10-16t)/15
y = (5-8t)/15
z = 3t/5
w = t,

where t is any number. There are many values of t that give valid probabilities, such as t=½ or t=¼.

Another approach is to raise the matrix to higher powers (perhaps using the MMULT function in Microsoft® Excel) until equilibrium is reached. This gives

$$\begin{bmatrix} 2/3 & 1/3 & 0 & 0 \\ 2/3 & 1/3 & 0 & 0 \\ 0 & 0 & 3/8 & 5/8 \\ 0 & 0 & 3/8 & 5/8 \end{bmatrix}$$

This shows that the system is trapped in states A and B, or in stated C and D. If for instance the system starts in State A, then it remains in state A for two thirds of the time and in state B for a third of the time.

# Unit 12: Introduction to Machine Learning

## Index

# Introduction to Machine Learning

# Conducting Machine Learning Experiments



# Unit Summary

## 1. Aims

The purpose of this unit is to introduce some of the basic concepts around machine learning, including key terminology. The unit will also introduce the software package that will be used to conduct some example experiments, which is called Weka.

## 2. Objectives

On completion of this module, you should be able to:

1. Define and recognise a machine learning algorithm
2. Distinguish between supervised and unsupervised learning
3. Define and distinguish between the different types of learning task
4. Understand the basics of preparing data for machine learning experiments
5. Be able to describe how a machine learning experiment should be planned and conducted

## 3. What is Machine Learning?

This section will introduce the concept of *Machine Learning* with computers. The background will briefly discuss the design of computer programs, the concept of an algorithm and the fundamental difference between Machine Learning and traditional programming.

### 3.1 Introduction to Computer Programs

Modern computers have their theoretical and practical origins in the work undertaken during the Second World War. At the time, there was a need to move beyond the use of calculators, devices which could perform simple arithmetic operations, towards a more automated approach to solving complex calculations such as code-breaking or artillery trajectories. This need arose both from a desire to improve accuracy, and to increase the speed at which results could be obtained.

Since the early, highly-secret, valve-driven days, computer technology has been subject to a near-exponential growth in complexity as characterised by Moore's Law. The processor at the heart of a modern device such as a mobile phone is many millions, indeed billions, of times more powerful than earlier, much larger devices.

However, despite this extraordinary rate of transformation, the fundamental aspects of computing remain: the desire to perform complex calculations with speed and accuracy beyond human ability. Computers are tools for extending human mental capacity by automating some procedural aspects. This is achieved by using a *Computer Program*.

In simple terms, computers require instructions on what to do. A computer program is a sequence of instructions, written in a formal machine code, that cause the system to behave in a particular way. The iconic example of this is the trivial program designed to print a greeting on the screen, the 'Hello World' program. In the python programming language, this is as simple as:

```
print "Hello World!"
```

The sophistication of programs is that they can be sequences of many millions of instructions, and the order of the instructions can vary conditionally based on input. It is therefore a very small extension of the above program for it to greet a person by name.

## 3.2 Algorithms and Software Engineering

With efficiency and correctness as key goals, the job of the programmer gets significantly more difficult as the specifications and behaviour which they wish to create expands. There are two key concepts which enter into the process. The first is the concept of *Software Engineering*, which is akin to treating a program as a complex building: components and their interactions are modularised and planned formally. Code is checked for correctness in a systematic manner, and certain conventions are adopted.

The second aspect is the use of algorithms. All programming languages are ultimately equivalent to one another, and so it is theoretically possible to write any program in any language. A result of this is that efficient operation can arise in the general case. There are common tasks such as searching for a sequence of characters, generating the values of a mathematical series, or drawing graphics on a screen which can be described in *pseudocode*. These procedures are known as algorithms. They are recipes for programs, which can then be engineered to complete a task with a computer.

Traditional programs specify completely the behaviour of the system, or return an error. This means that the programmer must have a formal model of the behaviour of both the expected input and the expected output. For example, in a tool to convert from Fahrenheit to Celsius, the computer would be expected to produce the output 100 for the input 212.

There are many challenges associated with this approach. Though it is effective it depends on the algorithm providing a suitable abstraction for the desired behaviour. It is necessary to consider exceptions, edge cases and variation. It is often the case that for certain tasks, the algorithm itself is difficult or impossible to specify in the general case.

For classes of problem where it is hard to formulate an appropriate general algorithm, computer science has turned to the notion of machine learning. The idea is an elegant one: rather than approximating the process directly in an algorithm, it considers how a program can be written which can simulate the process of *learning to complete the task*.

For example, recognising the gender of a person in a photograph is a difficult task to specify in general: there are numerous, often highly relative, factors to consider. On the other hand it is relatively easy for a human to generate a set of categorised examples, and these can be used in a machine learning approach to train the system to recognise gender.

Machine learning algorithms differ from traditional algorithms because their behaviour is incomplete until they are given some data to learn from. There are countless variations, and a very wide number of different types of algorithm which can be used for different tasks. It is convenient to think of each one as a way of simulating, in a primitive fashion, the kind of decision process which a human can learn from examples.

### 3.3 Artificial Intelligence vs. Machine Learning

Once it became apparent the power and speed of computers, people have thought of the notion of 'mechanical brains' or Artificial Intelligences. There has been work since the 1950s at least to create such intelligences: to create a general, thinking computer in our own cognitive image. This work has yielded some fascinating results, but has remained a staggeringly complex challenge. This so-called 'hard-AI' problem also led to some consideration as to whether it would be possible for us to even recognise if we had created intelligence, or merely simulated it.

Machine Learning can be regarded as a more pragmatic discipline. Rather than trying to 'solve' general intelligence, Machine Learning researchers and practitioners seek to employ the dynamic nature of data-driven approaches to specific, important problems. In that sense, Machine Learning is closely related to statistics, and to data mining, which are both involved with numerical reasoning and knowledge discovery. Significant success has been obtained by focusing the task which the system needs to learn to do. Recent advances demonstrate that this approach, while still narrow can yield extraordinarily effective results. Examples include Machine Translation, Automatic Speech Recognition, Image Analysis and Behavioural Analysis, all of which have become powerfully effective commercial applications thanks to this approach. This 'soft-AI' Approach has yielded very promising results in many commercial and research applications.

## 4. Machine Learning Fundamentals

Since a Machine Learning algorithm is defined by its need to *learn* before it can be used, their use differs from conventional programs because they need example data to work from. There are numerous applications for machine learning algorithms, such as hand-writing recognition, email Spam classification, or movie recommendation. In each case, there is a **training dataset**, which is used to allow the algorithm to learn the specific parameters of the problem. The algorithm, once trained, can then be used for prediction. For example, to predict which letter-class a particular image of a number should be associated with; or deciding what movie a particular user might like to see next.

### 4.1 Data in Machine Learning

A **Dataset** is defined as a set of *n* instances of data, collected together. An algorithm can only learn from the training examples which it has seen, this means that a training set should ideally be *representative*, that is, it should be a sample of data which has similar properties to the general population. This is similar to the general sampling problems in statistics. Common issues include:

- *Sample Bias* The data collected can over- or under-represent some mix of factors. For example, a dataset only covering teenagers would likely have a very different song distribution to one consisting only of middle-aged men, and neither would be representative of the overall population's listening habits.
- *Noise* Because a Data Set is a finite sub-set of a (theoretically) infinite population, there is random variation which is unaccounted for, or not possible to account for. This means that minor variations which would even out over the whole population are exaggerated. For example, it might randomly occur that a Christmas song is labelled as a Pop song.

- *Missing values* The data collected must represent the full range of possible values. For numerical information, this means that there should be examples that run from realistic maxima to realistic minima. For example, training data sets for daily temperatures in Ireland should include a small number of negative values (to represent very cold days). In categorical data, the full range of possible values should exist: for example, in a system for classification of the age of a person the range of possible values is the full age range.
- *Errors in the data*

## 4.2 Supervised vs. Unsupervised Approaches

There are several types of Machine Learning algorithm, and the main differences are determined by the goal, and the amount of existing training data. In simple terms, *Supervised* learning starts with the task. The Supervised approach uses a tailored set of training data and labels (or values) to fit the behaviour of the algorithm to the designer's needs. For example, this might include labelling pictures of animals either as cats or dogs; or calculating the value of a diamond based on its colour, size and weight. In this situation, there needs to be pre-labelled training data, and a model for the output and inputs. The data is prepared and used to train the algorithm to fit to the behaviour observed in the training data.

On the other hand, if the designer does not have a pre-existing model to impose, or behaviour to create, they might instead choose an *unsupervised* or data-driven approach. In this case, the system is used to discover patterns and associations in the data, which can be used afterwards. This might include, for example, finding documents which are similar to each other, or determining the likely next purchase by a consumer given what's already in their basket. In this case, the data used to train the system is unlabelled and the designer has limited control over what patterns the system might detect. This can lead to unexpected results, which can either be good or bad.

A third form of learning is reinforcement learning, which is commonly used in robotics, for navigation and exploration. In this case the system has a reward function, which allows it to calculate how well it is predicting results, and alters the behaviour to maximise the score. Reinforcement learning is not covered in this course, but there are many resources online if it is of interest.

### Advantages and Disadvantages of Supervised and Unsupervised Learning

Supervised Learning

Requires:

- Labelled, representative, training data

Advantages:

- Control over how the system fits its behaviour
- Easy to evaluate performance against expected behaviour
- Can take advantage of pre-existing model to refine data

Disadvantages:

- No guarantee data will fit the model
- Requires the creation of large amounts of training data
- Errors in training data can have very serious consequences

Unsupervised Learning

Requires:

- (Unlabelled) Training Data

Advantages:

- Learn from the data: new insights possible
- Don't need labelled training data
- Can learn (generally) more complex models

Disadvantages:

- Can't tell in advance what will be learned
- Can be difficult to evaluate
- Need large amounts of training data and for it to be representative

## 5. Categories of Machine Learning Problems

In order to use machine learning effectively, it's necessary to be able to apply the correct algorithm to the data, in the correct way. This requires us to have data, but also to have a question which we want to answer. We can think of a question about our data as being a specific *problem* which we need to solve. For example, given a collection of photographs, we might ask 'how do we tell if this is a photo of a dog or a cat?' (classification), 'can we group these news articles by similar topics?' (clustering). The features of the data, as well as the type of task, determine the choice of approach, *Supervised* or *Unsupervised*, and how the data is prepared. More on this in a later section.

### 5.1 Clustering

Clustering is a common data mining technique. The underlying assumption of clustering data points is that there are patterns of similarity to be discovered in the data that can be exploited. In clustering problems, the reasons for clustering are discovered by the operation of the algorithm, rather than the user. This is the key difference with classification problems.

Clustering can be *Hierarchical*, *Partition*, *Density*, or *Distribution*, depending on the nature of the data.

6. A Hierarchical Clustering algorithm builds a tree-like structure from the data points using a distance metric (such as the Euclidean Distance) and a linkage criterion (such as *single linkage*, which groups small clusters together sequentially by linking the two closest unlinked clusters in each round).
7. A Partition-based Clustering algorithm attempts to partition the space into a certain number of clusters, based on an average value, which can be randomly chosen in the space. This mean is then sequentially recalculated until the clusters converge.

8. Density based clustering assigns clusters to areas of a certain level of density, above a threshold. The data points outside these density concentrations are usually discarded as noise.
9. Distribution-based clustering applies a statistical distribution to the data. For example, a fixed number of Gaussian distributions with different parameters are sequentially fit to the data to form distinct clusters.

One key thing to remember is that clustering depends on the assumption that there is *some* natural partition within the data, and that those partitions can be expressed by the cluster model. For example, if the data is not easily linearly separable, then the resulting clusters might overlap. For example, a linear partition will struggle to properly cluster data which has overlapping curves.

Since clustering is computationally extremely expensive, algorithms typically have a fixed number of starting clusters. It's also up to the human at the end to decide if the clusters make sense, and to try and intuit why they are grouped in a particular way. This can make evaluating clustering objectively a difficult task.

Examples: Grouping medical images to discover patterns such as the presence of cancerous cells 'What similarities exist in these images of these cancerous cells?', grouping different website visitors by their behaviour in order to segment them for advertising purposes 'What patterns exist in the behaviour of our website visitors?'.

### 5.2 Classification

Classification is similar to clustering, but in this case the separations in the data are labelled and known in advance. The problem is therefore to learn the patterns in the variables which will result in one label or another, given an input instance to the classifier. Classification is typically divided into binary ('yes or no'), or multi-class problems. In a binary classification problem, the question is simply to ask whether a particular new data point fits the label criteria or not; in a multi-class scenario the question is to ask which of the labels should be applied to the new data.

Classification problems closely resemble statistical problems, and many approaches resemble that domain. Some algorithms can return a confidence score indicating how strong the correspondence is between the label class and the data point.

Examples: Spam Detection (binary classification) 'Is this email Spam?'; Handwriting Recognition (multi-class) 'Which number or letter is this an image of?'.

### 5.3 Regression

The principal difference between regression problems and classification problems is that they output a continuous value, rather than a categorical class. This closely resembles the use of statistics in regression. In this case, the data might be used to predict a response or output value for a new data point, having learned a formula for existing data. As with the above tasks, the data must be representative, and the mathematical model must fit the assumptions (for example, linear regression cannot accurately predict non-linear behaviour without transforming the data)

Example: Estimate the yield of a particular field of wheat based on the size, soil composition and climate, given data about other fields.

### 5.4 Density Estimation & Dimensionality Reduction

These two problems are typically used to simplify complex data in preparation for some other task. Density Estimation is used to attempt to discover the underlying probability distribution of a (typically large) data set, and dimensionality reduction is typically used to identify the most important features in the feature vector, and to reduce covariance to improve fitting.

Example of Dimensionality Reduction: discarding BMI (which is derived from height and weight) from health data that also includes height and weight.

Example of Density Estimation: determining the parameters of the normal distribution associated with a set of height data for men.

### 5.5 Self-Assessment Questions

Which is the appropriate problem class for each of the following Data Sets and questions?

10. 'How do I group this set of books so that readers can find what they are looking for?' [Ans: Hierarchical Clustering]
11. 'Which of these tweets are threatening and abusive?' [Ans: Binary Classification]
12. 'How much will this baby girl likely weigh given health data on her mother, and previous mothers and their babies?' [Ans: Regression]
13. 'How do I split these students into honours, ordinary and foundation level groups, given their transcripts?' [Ans: Partition-based Clustering]
14. 'What letter grade should I assign this essay, given previous ones?' [Ans: Classification]
15. 'What percentage score should I assign this essay, given previous ones?' [Ans: regression]
16. 'Given a class's assignments, how do I split them into six grades?' [Ans: clustering]
17. 'How can I reduce the variance in this data on photons?' [Ans: Dimensionality Reduction]
18. 'Which regression model best suits this data on plant populations?' [Ans: Density Estimation]

## 6. Summary

To Summarise, we start designing a Machine Learning experiment with a question. This question fits into a category of machine learning problems, based on the data set which already exists, and the desired output. If we want to discover structure in the data, we typically choose clustering. If we have a set of labels (one or more), and we want to predict the label for new data, we typically use classification. If the prediction is for a continuous value rather than a label, regression is most appropriate. Finally, it might be necessary or desirable to attempt to infer the distribution of the data (density estimation) or to choose only certain combinations of features for input (dimensionality reduction).

# Conducting Machine Learning Experiments

## Index



## Unit Summary

# Conducting Machine Learning Experiments

Given a question, a data set and a chosen algorithm, we are now ready to begin the actual process of training and testing our solution. It is very important to realise that the machine learning algorithm is not intelligent—it does not know how to interpret results, it is merely a mathematical model. This means that it is necessary to be aware that feeding incoherent data into the system, or basing the modelling on incorrect assumptions, will generally not produce an error: instead the results may be meaningless, and this might not be immediately obvious.

Most machine learning algorithms have one or more parameters which needs to be set for the system to work. For example, in a clustering task it is often necessary to specify the initial number of clusters. Some algorithms have a learning rate, or other set of parameters which, if incorrectly set, will produce incorrect results.

## 1. Experimental Procedure

The overall procedure breaks down as shown in the diagram of Figure 1.

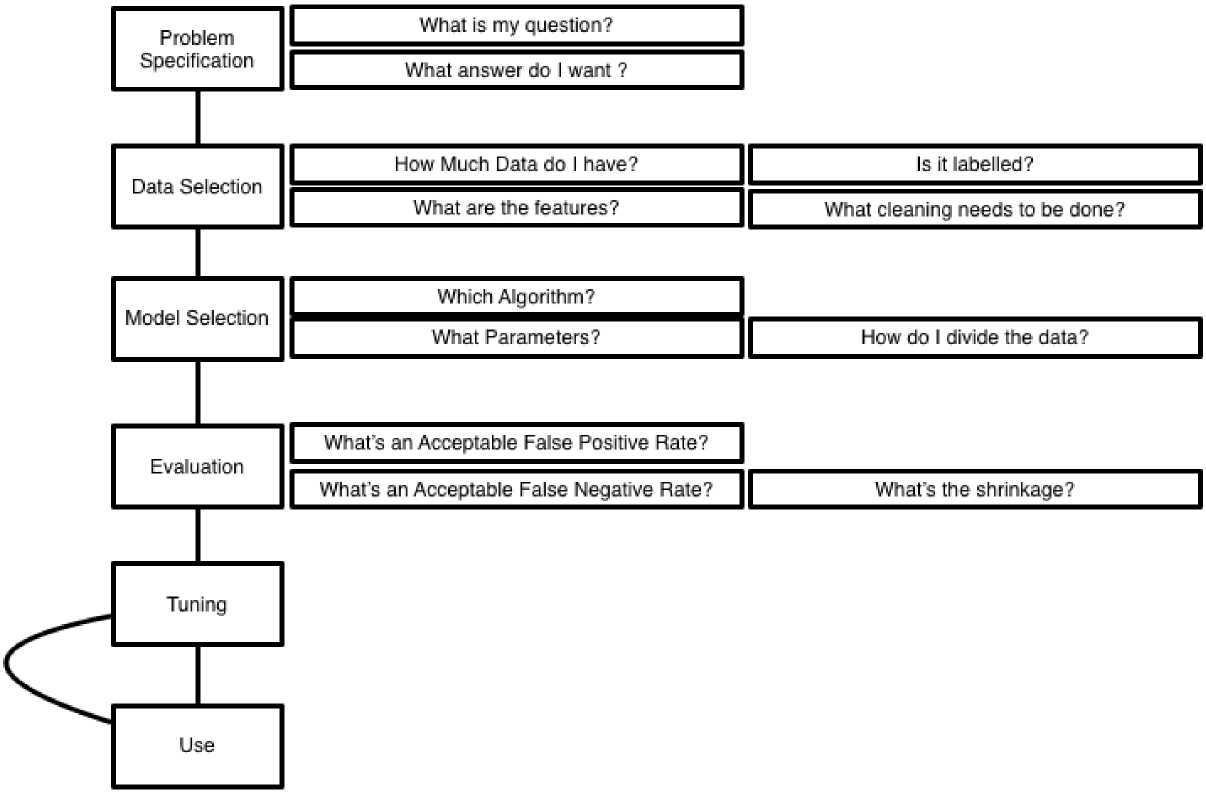



*Figure 1 - Diagram of the Experimental Procedure*

## 2. Problem Definition

The first step is to define the question in terms of inputs and outputs: what is the response (the output) we want from a given set of data? For example, we might want to know whether an email is spam or not, given labelled examples.

## 3. Data preparation

This can involve sampling the data if it is very large. If a sampling is done, it should be random and balanced. The data must then be cleaned, removing errors or incomplete data points. Then the data must be normalised: for numerical data this might include scaling all values between 0 and 1, for some other data it might include a process of bucketing: creating categorical labels from ranges of values. Other data might be split or joined: for example it might be useful to split dates from times.

Not all data has to be retained. For example, on a house price database, it might not be useful to include the colour of the front door for predicting the final value.

Data Analysis is the final preparation step: this can involve visualisation of samples of the data, pilot studies with different algorithms and the creation of subsets.

# 4. Model Selection

## 4.1 Algorithm Choice

The choice of supervised or unsupervised algorithm is determined by whether there is labelled data available. If there is no existing example data, then it is either necessary to generate some, or alternatively to use an unsupervised approach.

The second consideration is the size of the available data set, and its underlying distribution. The number of features can also be a consideration. For unknown distributions, the distribution can either be assumed or estimated. For feature simplification, dimensionality reduction can be pursued manually or automatically. Finally, *resampling* can be used to help with small data sets, but with increased risk of bias.

## 4.2 Validation

A fundamental challenge associated with machine learning, and statistics in general, is the recognition that *correlation is not causation*, that means that the behaviour observed in the data set might only be present in that data, and might not appear in a larger data set of the same population.

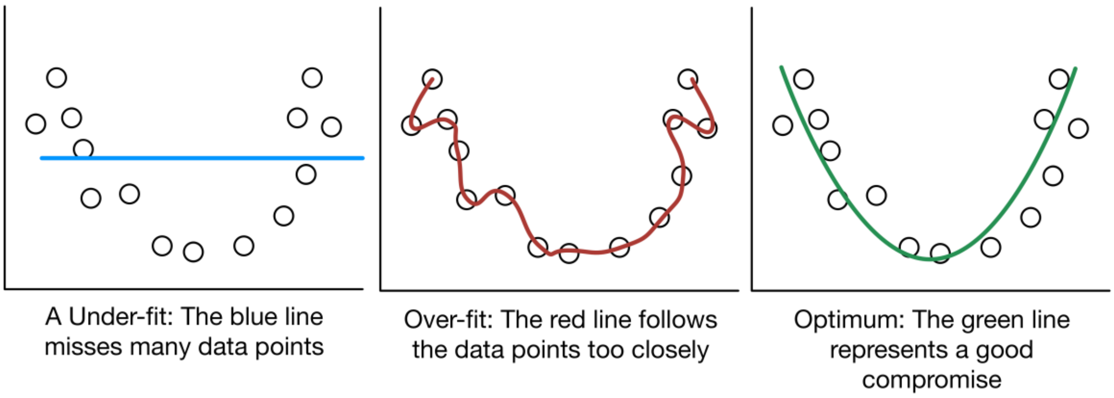



*Figure 2 - Visualised examples of over- and under- fitting*

The goal of machine learning is for the system to accurately represent the general data (the population) from which the data set is drawn. It is rarely possible to obtain the optimum fit (remember the bias/variance trade-off). Figure 2 shows a diagram that illustrates the difference between under- and over- fitting. Under-fitting is when the system does not represent features of the population, and so variance in the input does not produce change in the output. Over-fitting is when the system represents too closely the features of the data sample, but which are not generally present in the population. This means that a variance of input that should not change the output in general creates a change. The ideal is therefore a compromise: Occam's razor suggests that it should be the simplest model that represents the important features in the data set.

### Hold-out Validation

The simplest approach to validation is to simply divide the data set into two parts: one for training, and one for validation. There is an obvious cost to this in having less data to train on, which implies a likely increase in over-fitting. For that reason, it is common for the training set to be 80 or 90% of the overall data. The set should be divided in such a way as to minimise the chance of bias: this means that is it often desirable to randomly sample *witout replacement* to generate the validation set.

### k-fold Cross-Validation

Where hold-out validation is not desirable, for example with relatively small data sets, or where there is a significant chance of over-fitting, then it can be desirable to use all the data for validation. However, it is not sensible simply to use the data in one go. Instead, the preferred approach is to employ a sequence of validation experiments, where the data is trained on a different set of validation and training data in each experiment, and the results are averaged. This is commonly known as *k-fold cross validation*. The *k* is the number of different portions of the data, and it is cross-validation because all of the data is used for both training and validation, but across a sequence of combinations.

The easiest way to visualise cross validation like this is as a pizza: the data set is subdivided randomly into k slices, and in each experiment, one slice is held out for validation while the rest are used for training. Figure 3 shows iteration examples of a 10-fold cross validation process. K is often set to 10, which gives a 90% training / 10% validation ratio for each step. However, the choice of k depends on the data and experiment. For large data sets, or low over-fit risk, a k of three or five might be valid.

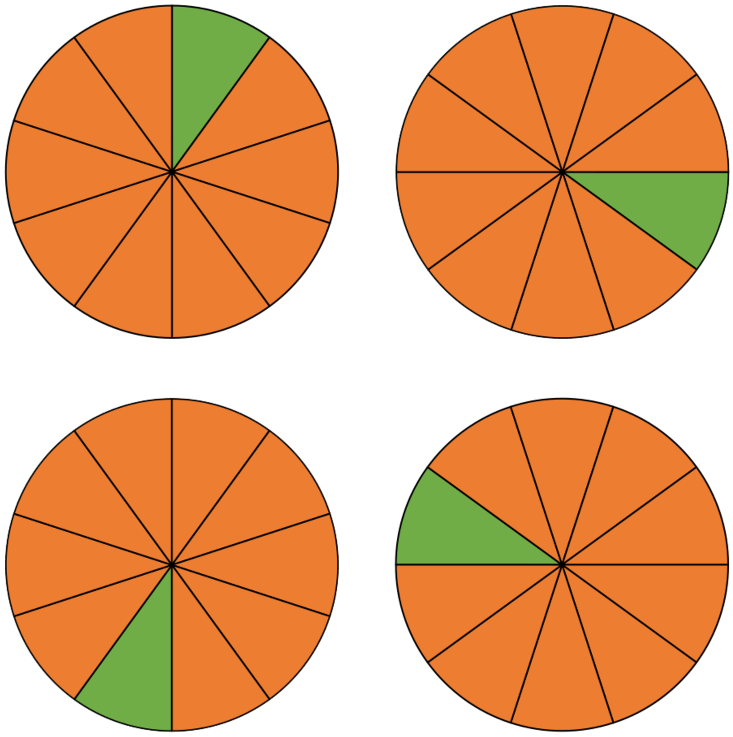


*Figure 3 - Four example iterations of the k-fold cross-validation process, where green is the held-out sub-set in each case.*

### 4.3 Measuring Classification Performance

Given a particular response from a system to a given input, there are four possible results:

19. The True Positive Result: This indicates that the response was positive, as expected
20. The True Negative Result: Indicating that the response was negative, as expected
21. The False Positive Result: Indicating that the response was positive, and incorrect: it should have been negative
22. The False Negative Result: Indicating that the response was negative, and incorrect: it should have been positive

Ideally, all of the responses would be true positive and true negative. In practice, this is not the case in most realistic machine learning scenarios. The purpose of both validation and evaluation is to estimate these rates.

The above set of responses can be tabulated in the *contingency table*. This is a common way of beginning to analyse the performance of a machine learning system.

The Contingency table is as follows:

| | Positive Result | Negative Result |
|---|---|---|
| Predicted Positive | True Positive | False Positive |
| Predicted Negative | False Negative | True Negative |

### 4.4 Measuring Regression Performance

Measuring specific performance depends on the nature of the response. The full array of statistical tools (for example the mean-squared error) can also be used. There may be different costs to balance between the different error rates: for example, medical diagnostics might want to err on the side of a higher false positive rate (marking some cases which are benign as needing further investigation rather than missing sick people); while a spam filter might want to err on false negative rate (avoiding marking real emails as spam but occasionally letting spam through).

## 5. Evaluation

We distinguish *Evaluation* from *Validation* as follows: validation is the process of estimating and selecting the best model and parameters for the data. It answers the question ‘which algorithm, with which values for its parameters is the optimum balance between under- and over-fitting?’. The selection of the best model on the data set is achieved by validation strategies. However, despite all the care and attention to avoiding over-fitting, the evaluation question remains: ‘Is our best model actually any good in more general data?’. There are two ways to approach model evaluation, but they both come down to the same strategy: testing on previously-unseen data. In effect, evaluation is a hold-out validation of the overall experiment.

The only difference between the two evaluation questions is the nature of the data to be used. In some situations, especially in unsupervised cases, it is necessary to have independently-established estimates of the predictions. This is often known as a *gold standard*. This gold standard data can have the advantage of having been used in several experiments with different techniques, giving an indication of the relative performance of the system compared to other approaches. If a gold standard does not exist, then the original data set must be divided into three:

23. Training data: to fit the model
24. Validation data: to choose the optimum model
25. Evaluation data: to test the overall result

Evaluation therefore allows us to estimate the *shrinkage*, that is, the reduction in performance of the algorithm when exposed to new data. There is usually some degree of over- or under-fitting, and the Evaluation step allows us to estimate how great that might be.

## 6. Tuning & Use

Once the system has been evaluated, it is ready for use. This means that new data has become available, and predictions / classifications can be made as needed. The performance of the system can continue to be evaluated, by collecting the new data for a new data set. This can lead to further tuning, or retraining of the system. However, this requires a complete validation/evaluation cycle, and cannot be done incrementally without an Active learning algorithm.

## 7. Self-Assessment Questions

26. Explain the difference between validation and evaluation. [Ans: Validation is about tuning the best parameters for the algorithm, evaluation is estimating the performance overall]
27. What is over-fitting? [Ans: Where the system has modelled the training data too closely, and increased the false positive rate]
28. Given a data set as follows, which features and approach might you choose for each question?
    - Data is 100,000 cars with the following features: Brand, Model, Price, Colour, Weight, Fuel Efficiency, Owner's Income
    - Question 1: 'Which are the most popular combinations of car features?' [Ans: All data, unsupervised clustering]
    - Question 2: 'How much does a particular person earn, based on them owning a BMW M3?' [Ans: Brand, Model, Price (maybe), Owner's Income]

## 8. Summary

The data-driven nature of machine learning means that we need to constantly test that our approach is correct. This means that we have a set of stages to establish a meaningful training data set and criteria for evaluation and validation. Once these performance metrics have been chosen, we run a sequence of experiments to validate the best set of parameters for the algorithm. This 'best' is still only a compromise, and so we evaluate again to determine how good we think the system is in the general case.

# Unit Summary

In this unit we have introduced the concept of machine learning, which is about using a computer to automate decision making. Machine learning is a data-driven approach, where the behaviour of the system is determined in part by examples from the data. If the data is labelled in advance, the approach is said to be supervised learning; if there is no labelled example data, it is unsupervised. There are different problems in machine learning: clustering to discover patterns, classification to label from a set of categories, and regression to predict a continuous value are the main ones which we will examine in this course.

The method for using machine learning is to first pose a question, decide on the type of problem, then choose the data set, its features and clean them. Once the data is prepared, a validation strategy must be chosen. This allows model selection, which finds the best parameters for the algorithm, balancing true- and false-positive rates (over and under fitting). Finally, the system is evaluated to estimate its error rate over the general data population.

# Unit 13: Supervised Learning

# Index

# 6. Evaluating Classifiers

## 6.1 The Confusion Matrix: Precision, Recall, F1 Score

## 6.2 Self-Assessment Exercises

# Supervised Learning

## Index

# Supervised Learning

## 1. Aims

The purpose of this unit is to describe *Supervised* Machine Learning tasks. By the end of this module, you should be familiar with the key properties of labelled data, the common tasks associated with training and evaluating supervised learning algorithms, and some common specific algorithms.

# 2. Supervision

The key feature of supervised approach to learning is that is the process of learning a *function* that describes pre-labelled data. This means that the behaviour of the algorithm and its performance are dependent on a *training data set that includes examples*. The *target* or output of the algorithm is often a label or value.

## 2.1 The Data

A key advantage of this approach is the ability to control what the algorithm should learn to predict or model for unseen data. This means that it is possible to specify more exactly the kinds of predictions which the algorithm will produce. The input data is in the form of a *training dataset*, which incorporates chosen *features*. The data set is used by the machine learning algorithm to fit the parameters of a mathematical function or statistical model, separating different target values.

An easy way to imagine this is to visualise a scenario where you draw a graph of each of the training values on a page, then draw a line that separates each class. Different techniques apply depending on the 'shapes' which separate the individual points in the classes, and whether the desired use of the system is for prediction (placing a previously-unseen value in the correct target label), or modelling (determining the important features which influence output). While a chart is 2-dimensional on paper, the input data to real machine learning algorithms are typically collections of values over many, potentially thousands, of different dimensions.

### Example: Property Values of Houses

The supervised approach works by choosing and labelling particular attributes, or features of each item in the data set. Consider the following example: imagine we want a tool which will use data gathered on previous transactions to predict or estimate the value of a house in a particular area. The housing data is presented as follows:

- Dataset: Private Property Register Sales for Dublin, January 2015
- Copyright Information: COPYRIGHT Property Services Regulatory Authority 2015 http://www.psr.ie/website/npsra/npsraweb.nsf/page/copyright-en

| Field Name | Data Type |
|---|---|
| Date of Sale | Date (dd/mm/yyyy) |
| Address | Text Field |
| Postal Code | Category Label: Postal Code, or Dublin |
| County | Dublin |

| | |
|---|---|
| Price (€) | €price |
| Not Full Market Price | Yes/No |
| VAT Exclusive | Yes/No |
| Description of Property | Category Label: Second-Hand or New Dwelling house/Apartment |
| Property Size Description | Category Label: empty, or size descriptor |

An example item from the data set shows what is featured in each item.

## 2.2 Steps Towards Deploying a Supervised Learning Algorithm

1. Choose the goal. The first question is: *what is the system trying to do? What output is expected from an input value?* For example: a system which can decide between a new house and a second-hand one.
   - In the example data set of the previous section, we might ask *'Can we tell a new house from a second-hand one?'*
   - This is a *classification problem*: we need to distinguish between new dwellings and dwellings which are not new.
2. Choose and clean features. We need to decide what features might or might not be relevant, and whether we need to clean data.
   - In the example, there are several rows with incomplete data, we would likely filter them out. Similarly, we would remove in advance any apparent errors.
   - We then engineer the features: The County field is always Dublin, so it is of little use. As we have only got data for one month, this is probably not of great information either.
   - The *address* field is a complex string including house number. It might be possible to get some information by processing this text to extract the street name, or the area, which could help.
3. Establish a Training Data Set. The next step is to collect, clean and prepare training data. The data needs to be of sufficient volume (enough individual items) to represent the 'real world' of possible values and their distribution.
   - In total, there are 1,244 rows in the file.
   - If we have no access to further data, we would randomly remove 10% of the rows from the file, as a hold-out *evaluation* set. For convenience, we reserve 144 evaluation rows, leaving 1100 items for validation and training.
4. Choose a validation strategy. It is common to use a *validation* approach to test whether the training has been successful. The two main methods are cross-validation and hold-out validation. Both of them sub-divide the training dataset in order to test whether the system has learned a useful function from the input. For example, setting the splitting factor *k* to 10% gives us a training set of 990 and validation of 110 items.
5. Choose one or more supervised learning algorithms and train them with the prepared data. For this kind of classification task, we might choose a *k-Nearest Neighbour*

*classifier*, as an example. Note that the $k$ in this classifier's name is unrelated to the cross fold count.

6. Validate the results and decide if the performance is adequate. If the performance needs to be improved, we can do one of the following:
   - change the parameters of the algorithm and re-train.
   - gather more training data and re-train.
   - clean the training data further and re-train.
7. Evaluate the System on the evaluation data, which is separate from the training/validation data set.
8. Once the best-performing algorithm has been chosen, the system can be deployed. In off-line training (where all the training is done before any predictions are made), the system's behaviour is static after training. More advanced techniques, beyond the scope of this course, allow for active or online learning to continue, incorporating feedback.

The result of this sequence of steps is a system that will give an estimated answer to an input. For example, the system could be used to predict the value of a house based on size and post-code (regression), or which digit an image of a number corresponds to (classification of classes 'zero' to 'nine'). In our example for the housing data set, the input of the postcode, price, VAT status, Full price status and Description will yield a classification of 'New' or 'second hand'.

Suggested reading for more detail on data noise types (class noise and attribute noise) and the effect of noise in machine learning: "Class Noise vs. Attribute Noise: A Quantitative Study," doi: 10.1007/s10462-004-0751-8

## 2.3 Trade-offs in Supervised Machine Learning

There are several things to consider in the choice of whether to use supervised approaches or not.

### Advantages:

9. Predictable behaviour: control over the target labels and the features means that the system can be designed to answer with specific labels.
10. Data Cleaning: analysis of outliers (unusually extreme values), erroneous values (for example, a negative or zero price).
11. Confounding Factor Removal: not all features in the data will be totally independent. For example, in a database of patient statistics, Body Mass Index (BMI) is calculated from sex, weight and height, meaning that including all three features might not improve performance compared with less features.
12. Model Simplification: On the other side of the coin, it can be useful to choose derived features, such as BMI, to reduce the number of features needed in the model. This can permit the designer of the experiment to discard several high-variance features in favour of a smaller number of more "regularly-behaved" features.

### Disadvantages:

13. Need for Data: the biggest obstacle to supervised learning is often the data curation problem. Gathering adequate amounts of training data, organising, cleaning and analysing it is a significant effort. Even more difficult or expensive can be the challenge of creating the labelled training data. This means that it is usually not a good

strategy to simply 'gather as much as possible'. Often, the data is obtained before the learning experiment is designed, which can constrain decisions.

14. Model complexity: there are significant costs in time and effort to training supervised algorithms, which can be seen as two disadvantages. One is that it makes tuning the algorithm slow. The second is the high risk of shrinkage that is the reduction in performance of the system when tested on more data not drawn from the test set. This is related to the notion of *overfitting*, where the system has been trained in response to random, statistically insignificant variation in the test data. Because algorithms often include a random (stochastic element) it is even possible to have overfitting in unbiased data.

## 2.4 Design Considerations

15. Data Gathering: The gathering of the data must be done with awareness of a realistic, representative sample of the true variance and range of the data. It is not usually possible to obtain perfectly representative data, especially when labelling and pre-processing is required. An increasingly-popular approach is to crowd-source data, either through paid interfaces such as Amazon's Mechanical Turk, or via tools such as ReCAPTCHA, or with direct campaigns.
16. Algorithm Choice: different algorithms are sensitive to different kinds of bias, which results in different variance in the response. This means that for small changes in data, large changes in the resulting function may be observed depending on the algorithm. The 'No Free Lunch' principle can be applied in the context of algorithm choice: some restaurants (learning algorithms) suit some eaters (problems), but every meal has some trade-off of cost between speed, complexity and accuracy.
17. Performance Targets: what error levels are acceptable? What speed of response is required, and how much computing power is available? In particular, questions that are important include deciding on prioritising false positive versus false negative rates.

## 2.5 Summary

In Supervised Learning, the experiment designer controls the input and output of the machine learning algorithm. The algorithm is trained on a pre-prepared sample of labelled training data, which is chosen to represent the real-world data. Then the algorithm is employed to make decisions about data of the same kind.

## 2.6 Self-Assessment Exercises

18. In a Supervised context, what is a *target*?
19. For the following targets, what type of classification problem is pose?
    a. Student grade prediction as a percentage
    b. Gender detection from photographs of faces
    c. Movie rating prediction by stars
20. Why does the 'No Free Lunch' problem arise?

# 3. Decision Trees

Decision tree is one of the most reliable and robust learning methods. It is a very practical and effective way of implementing “if-then” rules and to approximate discrete-valued functions.

## 3.1 Decision Tree Representation

Decision trees are represented by nodes, branches and the leave nodes at the bottom. They store situations and their outcomes within its nodes, allowing to remember the best cause of action in case a similar situation is encountered in the future. For example, they can be used to predict future outcomes or classify situations. Figure 1 shows an example of a decision tree to decide if we play tennis depending on the weather. Each node tests one attribute, each branch from a node selects one value for that node, each leaf node predicts the hypothesis of playing tennis. Decision trees grow as they learn information so they can become very complex and there is also the problem of overfitting or examples which are not in the training data.

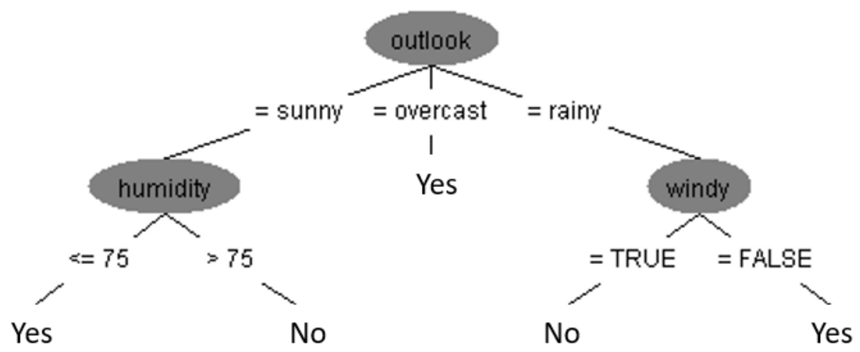



*Figure 1 – Diagram of a decision tree to predict the concept of playing tennis depending on the weather. It is represented by nodes, branches and the leave nodes at the bottom.*

## 3.2 Learning Algorithm

The algorithms for learning decision trees are based on a top-down approach that searches the space of possible decision trees for the optimal tree. The algorithm starts by finding the best attribute to classify the training examples at the root of the tree. In the example of Figure 1, the best attribute is “outlook”. Then the training examples are separated according with the possible values of this attribute, that is, down each branch from the root. The process of finding the best attribute is repeated using the training examples associated with each descendent node. For the decision tree of Figure 1, the best attributes to test at the following nodes are “humidity” and “windy”.

In general, the algorithms perform a greedy search (top-down search without backtracking) that measures how well an attribute splits the training data according with the target classification while building the tree. This is done by using statistical measures from information theory, such as entropy. At each point in the tree, the algorithm calculates the information gain for each attribute and selects the one with highest information gain.

## 3.3 Overfitting in Decision Trees

An important question in training a decision tree is how deeply to grow the tree. If the algorithm grows the tree deep enough to correctly classify all the training data, the resulting tree may overfit the training examples. Overfitting is a problem that may be difficult to solve especially when the data has noise or is too small. There are several techniques to avoid this effect. One is to define a stopping criterion by using an heuristic measure (based on the Minimum Description Length principle) to avoid the tree growing to a point it perfectly classifies the training data. Another one is to use "post-pruning" to reduce the size of the tree after it overfits the data. A validation data set can also be used to decide on the approach that determines the final tree size.

## 3.4 Random Forest

Random forest is a way of of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. It starts with a decision tree, which corresponds to the "weak learner". Then, the training algorithm chooses small subsets of variables at random and finds a variable and value for that variable taking into account the target classification output. This special type of decision tree is able to deal well with unbalanced and missing data.

## 3.5 Additional Material

For more information on decision trees, please see:

- Chapter 3 of "Machine Learning", Tom M. Mitchell.
- YouTube videos:

  https://www.youtube.com/watch?v=p17C9q2M00Q

  https://www.youtube.com/watch?v=o7iDkcpOr_g

## 3.6 Self-Assessment Exercises

For the decision tree represented in Figure 1:

1. Indicate the classification output for the following input instance: humidity=90; outlook=sunny; temperature=70; windy=true; [Ans: play tennis = No]
2. What attribute has the highest information gain? [Ans: outlook].

# 4. Support Vector Machines

Support Vector Machines (SVMs) are a type of Supervised Machine Learning Algorithm, which can be used for either classification or regression problems. The mathematics of SVMs can sometimes be complex. However, the principles underlying them can be understood through some relatively simple examples. However, one thing to keep in mind is the ‘curse of dimensionality’ principle, and especially the fact that intuition fails at high dimensions: this means that a formal understanding is necessary as the number of features increases.

## 4.1 Linear Separability

A classifier is a system for making a decision: which label should be applied to a particular item, based on its features? The components of the decision are experience (previous examples in the training data) and a way of relating the features of the new item to that experience (the algorithm).

A common approach is to consider the features of the data as a set of axes, like drawing a graph. This results in an n-dimensional space of the different labels. We can think of training the machine learning algorithm as being the same as trying to draw a border around the different examples of each label, with validation sequentially creating better borders. When asked to classify new data, the system is essentially required to decide which side of the border a new item should occupy, and hence what label it belongs to.

The challenges are to decide:

a) How to draw the border, or decision boundary, based on the distribution of the training data.

b) How to decide if the boundary is a good fit or not, and what compromises are needed.

c) How to place new data so as to get it on the correct side of the boundary, or boundaries.

If we assume (as is often the case in simple situations) that we want straight boundaries, this affects the types of shapes we can draw. Looking at the examples in Figure 1, it is possible to draw a straight-line (linear) decision boundary between the classes or labels in only two of the four cases (A and B). For the other two cases, the overlap prohibits any straight-line from passing through and separating the values.

The SVM is an example of an approach to drawing decision boundaries, based on the idea of having a clear gap between the edges of each group (the support vectors).

We will discuss the simple form of the SVM and its underlying principles. These can then be extended for other cases.

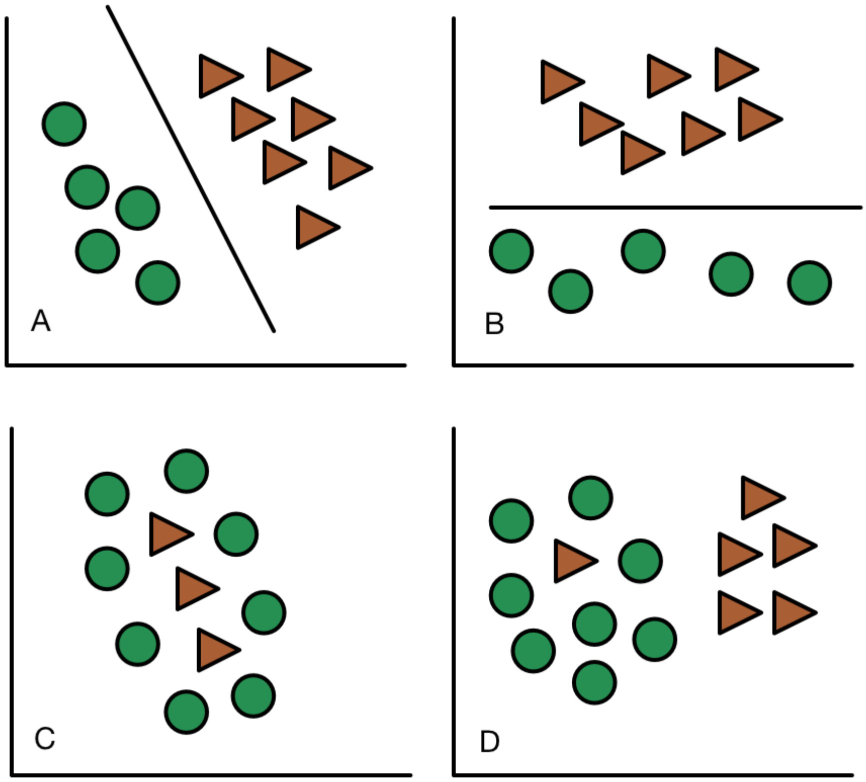



*Figure 2 - Linear separations between the labels as represented by different coloured shapes.*

## 4.2 Single-Class Linear SVMs

The single-class, linear support vector machine answers questions which are formulated as 'Does this item belong to the class or not?'. An example might be 'Is this a picture of a dog or not?'.

The SVM is based on the idea that the decision boundary should be drawn to leave the maximum margin between the classes, leaving the most room for new items on each side of the boundary. Illustrating this in Figure 2, we can see that the idea is that the black line (decision boundary) is in the middle between the edges of each group (the positive and negative examples). The regions for each group are called *hyperplanes*, because they are shapes of one dimension less than their surroundings. For example, the hyperplane is a one dimension (1-D) line on a 2-D plane, or a 3-D volume in a 4-D space. Each dimension corresponds to a feature in the feature vector representing the data.

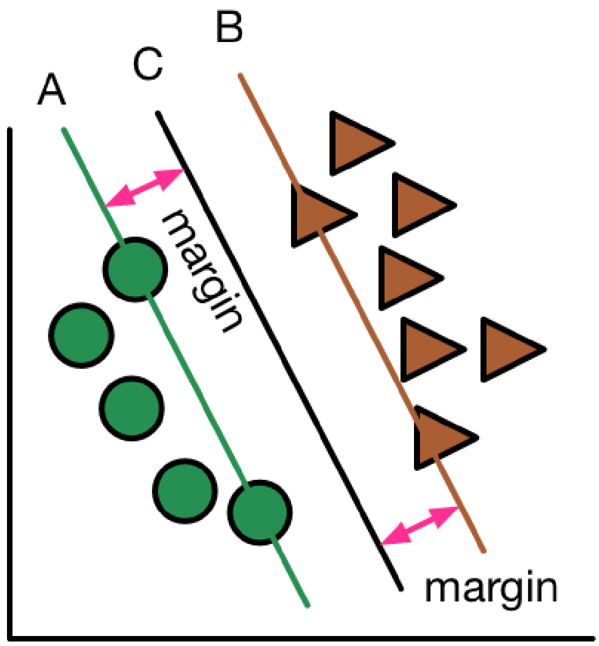



*Figure 3 – Decision boundary (black line) and margins of a linear SVM.*

The training task is therefore to learn from the example data the line that maximises the distance between the edges of the hyperplanes.

Features in this type of SVM are expressed as a vector of Real-valued numbers, and the resulting decision is either 1 (for positive) or -1 for negative.

### 4.3 Non-linearity and 'the Kernel trick'

A typical example of a non-linear separation is where a central group of examples is surrounded by another. This is not linearly separable by default. However, because of the mathematical properties of the SVM's decision space, it can be possible to transform the features using a mathematical function. For example, to project them into an higher-dimensional space, or to represent them using polar co-ordinates. In doing so, it is then possible to draw a linear separation in that space, and use that as a decision boundary. The use of the kernel trick allows SVMs to work in a very wide variety of problems which are not by default linearly separable. Figure 3 illustrates the use of a kernel to separate two classes which are not linearly separable.

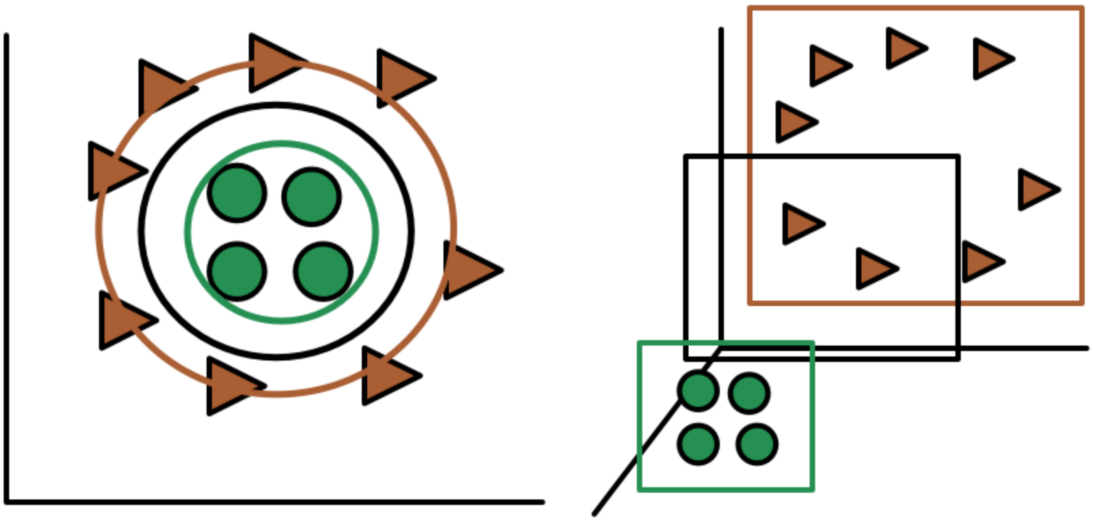


*Figure 4 - By Transforming the 2-D feature space on the left into the 3-D space on the right, we can draw a boundary that maximises the margins between the hyperplanes in the latter.*

## 4.4 Multi-class SVMs

SVMs are inherently binary classifiers: they answer the question 'Is X in Class A or B?', where often B is simply 'not A'. However, many classification situations require distinguishing between multiple labels. Without a direct way to separate the decision space into multiple regions, the usual approach is to train *multiple different SVMs*, and use their output to decide on the resulting label.

Two strategies for training the SVMs are: one versus the rest and pairwise. These will be described next.

In the one vs. rest strategy, we train each individual class against the rest. For three classes, this means three SVMs. If the classes are 'Red, Green, Blue', then the SVMs would be 'Red vs. [Green, Blue]', 'Green vs. [Red, Blue]', and 'Blue vs. [Red, Green]'. In this situation, we choose the label which has the *highest score* of the three. This means that the winner takes all, and care must be taken to ensure that the SVMs are able to produce the same score. Intuitively, we can think of this as being a situation where we draw three different boundaries and choose the one that is furthest inside that class.

In the pairwise strategy, we directly compare the different classes. For our colour example, this means the following SVMs: Red vs. Green, Red vs. Blue, and Green vs. Blue. The difference here is that we use majority voting. For each new item, we choose the one which has the most votes amongst the SVMs. For example, if the result is 2 for Green and 1 for Blue, then we choose the label Green. Note that the number of SVMs to train can grow very quickly, as it relates to the binomial coefficient. For instance, 10 classes yield 45 SVMs.

## 4.5 Additional Material

For more on the mathematics underlying SVMs, please see:

- Robert Burbidge, Bernard Buxton, "An Introduction to Support Vector Machines for Data Mining", http://www.cc.gatech.edu/classes/AY2008/cs7641_spring/handouts/yor12-introsvm.pdf

- For details about the Kernel Trick, Jordan & Thibaux, "The Kernel Trick",

  http://www.cs.berkeley.edu/~jordan/courses/281B-spring04/lectures/lec3.pdf

- Tutorial on "The Kernel Trick" for SVM, Eric Kim:

  http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

- Brandon Rohrer's YouTube channel:

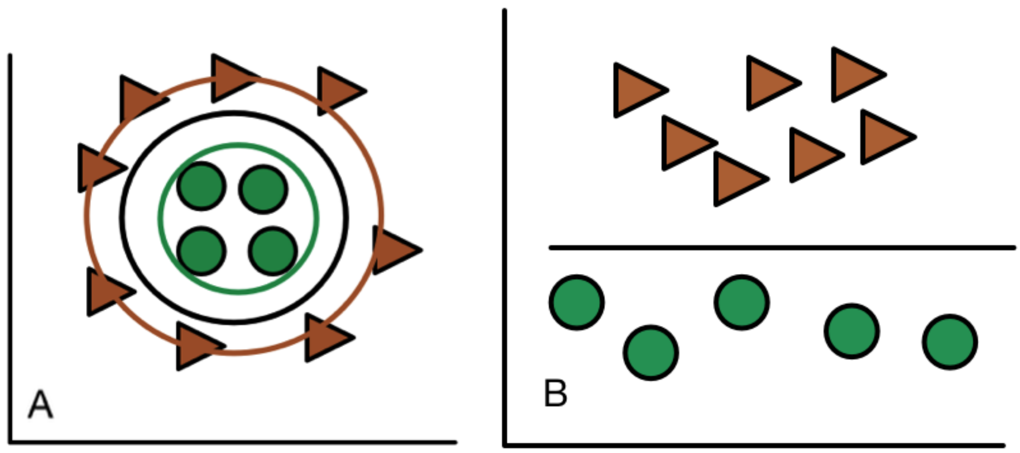
  https://www.youtube.com/watch?v=-Z4aojJ-pdg

## 4.6 Summary

An SVM establishes one or more decision boundaries between labels be learning a hyperplane that encompasses all the training examples of each class. The hyperplane is a shape with one dimension less than the projected space (e.g. a square in a cubic 3D space). The idea is to maximise the margins between the support vectors that describe the edges of the classes. In order to handle irregular separation, the 'kernel trick' is used to transform the data from a non-linear separation in one space, to a separable one in another.

## 4.7 Self-Assessment Exercises

21. Which of the following is a linearly separable set? [Ans: B]




22. What is the intuition behind maximising the margin? [Ans: To maximise the distance between the edges of the two classes]

23. How would you use a single-class SVM to create a system for deciding if a photo were of a man or a woman? [Ans: Rather than assigning the labels 'A' or 'not A', the labels

would be ‘Woman’ or ‘Man [not woman]’. It is then necessary to ensure that the data only has photos that have people in them.

24. Which of these is likely to be the chosen decision boundary for the SVM? [Ans: 2]

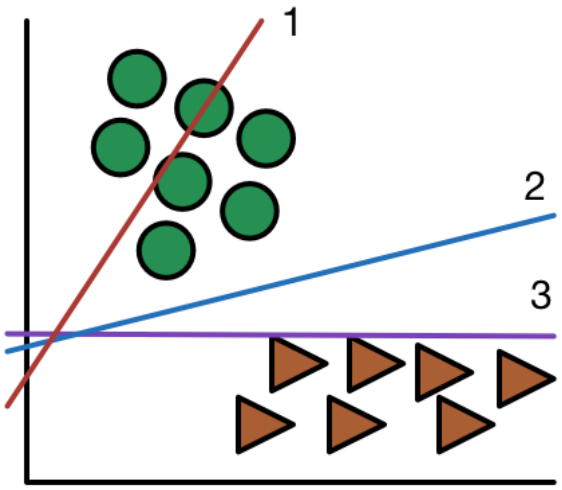



*Figure 5 – Three different options for the decision boundary of an SVM.*

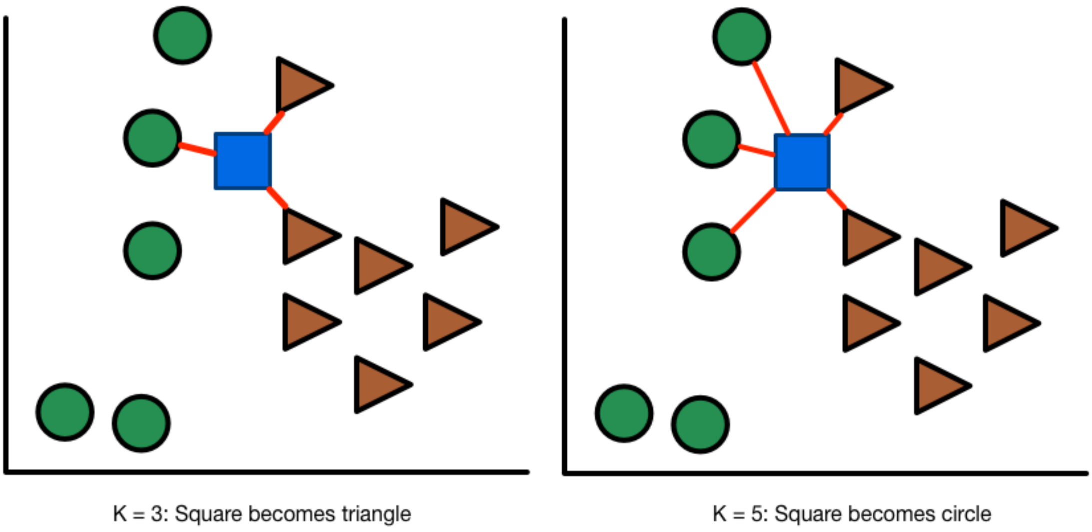
25. If a multi-class question is as follows: ‘Classify birds into parrot, eagle, seagull, robin’, what SVMs would need to be trained in the pairwise situation? [Ans: parrot vs eagle, parrot vs seagull, parrot vs robin, eagle vs seagull, eagle vs robin, seagull vs robin; Then, the one with the most votes wins, e.g. if 3 votes for seagull is the highest, then it’s a seagull]

## 5. K-Nearest Neighbour Classifiers

The k-Nearest Neighbour (k-NN) classifier is often regarded as the simplest form of classifier. It is an instance-based learning approach, which directly compares new items to those in the training data (this differs from the SVM, which computes a boundary for comparison based on the support vectors).

Put simply, k-NN classifiers assume that ‘birds of a feather flock together’, in other words that close points in the feature space should share the same class.

The k-NN classifier needs labelled example data, in the form of feature vectors, and it needs a *distance metric*, such as the Euclidean Distance, to calculate how far two points are from each other. The system works by calculating the distance between the new item and its *k* nearest neighbours. For k=3, we find the labels of the three nearest points, and choose the label of the majority of those points. For k=5, we would choose the majority of the five nearest points. k-NN classifiers can be used for multiple classes directly, given enough data.




## Figure 6 - The effect of different k on a two-class k-NN Classifier.

## 5.1 Choosing *k*

The parameter *k* is often chosen to be 3 or 5, but there is no specific reason for this. Other, much higher values can also be chosen. The advantage of the supervised approach is that it is possible to choose *k* systematically using validation. Re-training the system for different values of *k* and choosing the one with the highest score is a valid approach, but may take a long time for large *k* (in the region of 50 to 100).

## 5.2 Dealing with Ties

As the number of classes increases, and depending on the number of values for each feature, there is an increasing risk of ties. Ties arise when two or more neighbours share the same distance metric to the new value. There is no absolute answer to this situation. Some algorithms simply choose from the candidate points based on the order in the training data, while others will introduce a small random variance. For a very large number of ties, it might be desirable to increase *k*, so that the ties can be accounted for.

## 5.3 Additional Material

- Lecture notes by Amos Storkey, from "Learning from Data" course, University of Edinburgh:

http://www.inf.ed.ac.uk/teaching/courses/lfd/lectures/lfd_2005_nearest_neighbour.pdf

- Suggested video tutorial on k-NN classifier:

https://www.youtube.com/watch?v=ZBHStob8lhc

## 5.4 Self-Assessment Exercises

26. Why is it wise to choose an odd value for *k*? [Ans: reduces the chances of needing to deal with tied votes]

27. Which result would be chosen in each of the cases below (red lines indicate nearest neighbours) (ANS: Circle, Triangle, Triangle)

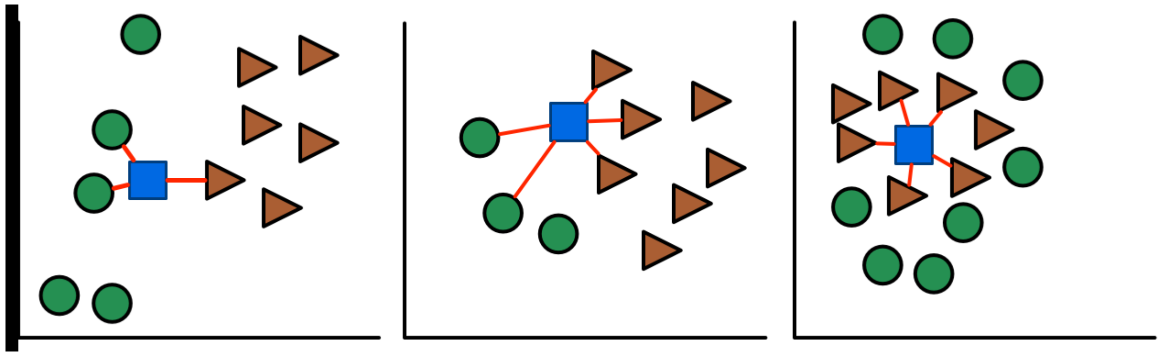


*Figure 7 – Three examples of k-NN classification.*

## 6. Neural Networks

Artificial Neural Networks (ANNs) are inspired by a simplified model of the biological structure of the brain. The idea is to connect individual decision functions (nodes) in a network. Each node activates or 'fires' in response to different input.

A recurring problem with the learning algorithms discussed so far is how to deal with solving complex problems, where the data does not behave in a conveniently linear and separable manner. One approach which is adopted in ANNs is to divide and conquer: instead of training a single function to classify information, we divide the problem space up and assign different functions to different regions. This is the intuitive model of the ANN: an interconnected set of nodes and each decides part of the overall answer.

### 6.1 Individual Neurons

Neural Networks, as the name implies, are built up of individual nodes, the artificial neurons. These are designed to be simple functions that take inputs and emit an output, based on an *activation function*.

Each neuron takes inputs, which are real-valued numbers (labelled $x_1$ to $x_n$) and generates an output $y$ based on the activation function $f(x)$. A single neuron very much resembles a conventional logical circuit. The key feature that differentiates the neural net from a single neuron is that the inputs to each neuron are *weighted* (multiplied by a particular value), so individual neurons take account of different inputs differently before applying the respective activation functions.

The ANN is therefore trained by a process of learning an optimal set of weights, each neuron applying its own weight values to each input.

#### The perceptron algorithm

Figure 7 shows the diagram of a perceptron. For this example perceptron with:

- two inputs: $x_1$, $x_2$, weighted $w_1$, $w_2$ respectively.

- activation function: $f(x) = 0$ if the sum of the weighted inputs is less than a threshold; or $f(x) = 1$ if the sum is greater than that threshold.
- output: y

the output is calculated as:

$y = f(w1 * x1, w2 * x2)$

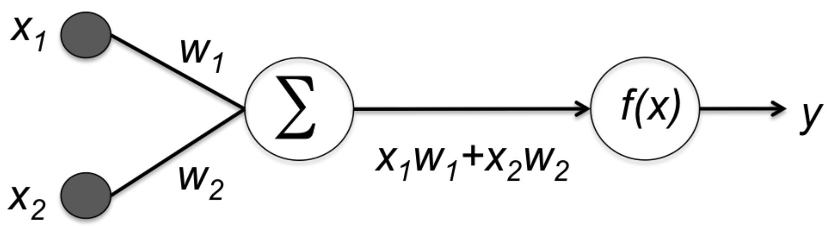



Figure 8 – Diagram of a perceptron with two input units ($x_1$ and $x_2$), respective weights ($w_1$ and $w_2$), and activation function $f(x)$.

The process for learning the weights is by a process of iteration:

28. Initialise the weights to small random values
29. For each input example *i*, set $d_i$ equal to the desired, or target, output value of $y_i$
30. Set the gain step $\eta$: the amount the weights are changed in each training step
31. For each input example *i*:
    a) $y_i = f(w_i * x_i)$, where $i = 1, 2, \ldots, n$ and $n$ is the total number of examples.
    b) Update the weights (in iteration $t+1$): $w_i(t+1) = w_i(t) + \eta(d_i - y_i)x_i$
    c) Repeat until the overall error is lower than a desired value, or after a fixed number of iterations, since the training is not guaranteed to converge (stop).

## 6.2 Layering

The model described above works reasonably well for simple problems. However, one key issue is that, due to the hard limit associated with the binary classification, the activation function can create situations where the training process is disruptive. This is especially true when it comes to creating *neural networks* that feed the output from one *layer* of nodes into another layer.

One example of a simple neural network is the combination of two input nodes fully connected to an output node. In this case, the output of the final node is the weighted sum of the outputs from the input nodes. This can be taken further, introducing a *fully connected hidden layer* (it's called hidden because it's not visible as input or output). Figure 8 shows a digram of a neural network with four input units, one hidden layer and one unit in the output layer. The use of the hidden layer allows the system to learn a wide variety of different input cases. The weights in each layer can be learned using an algorithm called *back-propagation*. Intuitively, this works by calculating the error at the output node, and *feeding back* a proportion of that error to the hidden nodes.

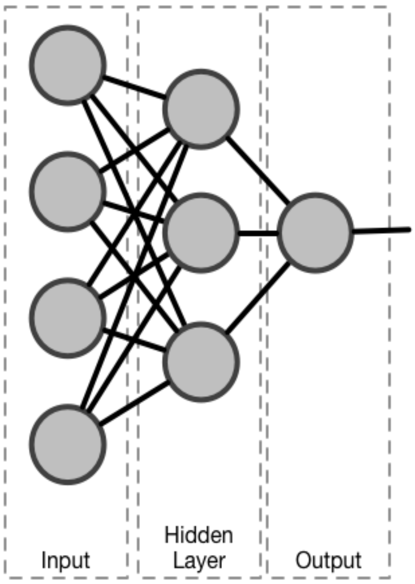



*Figure 9 - Multilayer Neural Network with one layer*

There is no strict rule on how many layers a system can have, indeed recent research has focused on *deep learning* with several layers with thousands of nodes. The exact details of how to design a neural networks is an advanced area of research, and depends heavily on using different statistical models for error propagation.

However, the perceptron's binary classifier above is too simple for use in these networks. This is for two reasons: First, the binary classification is too sensitive meaning that learning one case can disrupt the behaviour of the network in other input cases; The second issue is that linear functions (even in complex combinations) are limited in what they can solve. The solution to this is to use an activation function that returns a real value between 0 and 1, rather than a hard, binary value. In doing so, the system can account for smaller changes overall. To be useful, the hidden layers generally encompass a non-linear activation function. In fact, one of the advantages of creating multi-layer neural networks is the ability to employ different algorithmic functions at each layer. Because the weights are learned by training the networks from data, ANNs can often learn quite complex structures (but at the cost of a very high computational complexity).

Suggested reading and additional material for more details on ANNs:

- Chapter 4 of "Machine Learning", Tom M. Mitchell.
- "Neural Networks and Deep Learning", Michael Nielsen, http://neuralnetworksanddeeplearning.com/
- "Deep Learning: Methods and Applications" by Microsoft Research (2014):

  http://research.microsoft.com/pubs/209355/DeepLearning-NowPublishing-Vol7-SIG-039.pdf
- Brandon Rohrer's YouTube channel:

  https://www.youtube.com/watch?v=ILsA4nyG7I0
- Leo Isikdogan's YouTube channel:

  https://www.youtube.com/playlist?list=PLWKotBjTDoLj3rXBL-nEIPRN9V3a9Cx07

### 6.3 Summary

Neural Networks follow a rough model of how organic brains process information. The key feature of the net is that it is composed of layers of nodes, each of which emits an output that is the function of weighted inputs. In a supervised context, the training is achieved by *learning* the weights of the different nodes in each layer. By clever design of the net, it is possible to have highly flexible classifiers that can learn patterns from input with a wide variety of configurations, such as handwriting samples, images, or speech.

### 6.4 Self-Assessment Exercises

32. How is a Neural Network different from a multiclass SVM? [Ans: The second is a 'committee' of independently trained classifiers, while the first is one classifier divided into with multiple parts]
33. Why not initialise the weights to 0? [Ans: It would result in an output for no input, which would have unexpected effects]
34. What happens to the weights in the perceptron learning when the expected input matches the output? [Ans: $y$-$d$ = $0$, so no change. This means the perceptron only learns from mistakes]
35. It's possible to use unsupervised approaches to training neural networks. How might the approach differ? [Ans: $d$ cannot be found from the training data, instead it is necessary to decide on a different cost function, some way of describing the patterns which you want to find with the network.]

## 7. Evaluating Classifiers

Throughout this unit, we have discussed the idea of validation and evaluation of classifiers, but it's important to have a clear idea of what a successful result is. Their principal approach is to look at how well the system is doing in terms of the number of right and wrong answers that it generates.

The Table of Confusion described in the previous Section is useful, but it is dependent on the data: if there is an imbalance in the class memberships, then the numbers will appear skewed in the table. The solution to avoid this problem is to look at a number of ratios to determine the accuracy and error of the system.

We can see this in the following example of a two-class classifier, for recognising pictures of cars from pictures of trains.

For n = 100 samples (90 trains, 10 cars), the contingency table is as follows:

| | Positive Result | Negative Result |
|---|---|---|
| Predicted Positive | TP=90 | FP=10 |
| Predicted Negative | FN=0 | TN=10 |

This would appear to show a highly-effective (90% TP rate) classifier. In fact, the classifier is predicting trains perfectly, and failing entirely on cars. However, because 90% of the samples were trains, the system appears better than it is. In the real world, this will mean that no pictures of cars will be properly recognised, undermining the system's value.

It would be useful to be able to take into account imbalances in the data when assessing the performance of the classifier. It's also important to distinguish the different ways in which a system can be right or wrong. How many of the possible results did the system correctly classify? And of the guesses it made, how many were correctly labelled? We use the concepts of Precision and Recall (also called specifity and sensitivity respectively).

## 7.1 The Confusion Matrix: Precision, Recall, F1 Score

Precision and Recall can be calculated as:

Precision = TP/(TP + FP)

Recall = TP/(TP + FN),

where, for n:

- TP is the number of True Positives,
- TN is the number of True Negatives,
- FP is the number of False Positives,
- FN is the number of False Negatives.

Precision tells us how many times the true result from the classifier was the correct guess from all the guesses it made. Recall tells us how many times the classifier got the right answer from all the possible right answers in the data. In other words, the first tells us how reliable the guess was in terms of being right, the other tells us how often it recognised that right answer when asked.

It's common to calculate a balanced view of the precision and recall because these may provide limited information, depending on the application. For example, a classifier that's always right, but only finds 10% of cases (i.e. high-precision, but low recall) might be less useful in certain cases than one that is right half the time, but for 70% of cases.

One way to this is to take the *harmonic mean* of the precision and recall, called the F-score. The general F-score equation is:

$$F_b = (1 + b)\frac{Precision * Recall}{(b * Precision) + Recall}$$

where the F-score range is from 0 to 1, with 1 being the best result.

The most common measure balances Precision and Recall evenly, and is called the F1-Score. Its formula is:

$$F_1 = 2\frac{Precision * Recall}{Precision + Recall}$$

Another common measure is the Accuracy, which is defined as:

$$Accuracy = (TP + TN) / n$$

For more on the confusion matrix see:

http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/
https://en.wikipedia.org/wiki/Confusion_matrix

## 7.2 Self-Assessment Exercises

36. Calculate the complete confusion matrix for the following data:
    - Classes: Happy Face, Sad Face
    - n = 20, (10 Happy, 10 Sad)
    - Classifier results = 15 Happy, 5 Sad
2. What is the F(0.5) score for the above data? What is the difference between F(0.5) and the F1-score?

## Index

### 4.4 Self-Assessment Questions

# Unsupervised Learning

This unit covers *unsupervised* learning, and some key techniques associated with training and evaluating unsupervised systems. By the end of this unit you should be able to distinguish different types of unsupervised tasks, and understand how to apply different example algorithms. You should also have an awareness of the consequences of an unsupervised approach for evaluation and validation.

## 1. Defining the Unsupervised Learning Task

Machine Learning has been defined in previous units to be about learning behaviour from data. A particular algorithm is chosen and labelled data is modelled as a set of input attributes for the algorithm. The data input, along with the parameters learned by the algorithm, determine the output predictions. In an unsupervised system, we have *unlabelled* data. That means that the data presented to the system is fed in 'as it is', and it is up to the algorithm to discover some order or pattern.

Typical Unsupervised Learning tasks include:

1. **Clustering:** grouping together data points which have similar attributes
2. **Signal Separation:** extracting the uncorrelated variables from a set of input attributes
3. **Sequence Learning:** learning patterns of sequences from data

This unit will cover an example from each of these areas. There are many other approaches and algorithms for unsupervised learning and for more complex tasks.

## 1.1 Supervised vs. Unsupervised

For Supervised approaches, we want to fit up to some extent the behaviour of the algorithm to the labels in the example training data. Note that it is important not to fit the algorithm to the data too much so that it can generalise well to do correct predictions for unseen input data. This means that the supervised task is about trying to reproduce, as general as possible, an existing desired set of outputs for a given input. *Unsupervised* approaches do not benefit from an existing view of how the system should behave: no example data means that the purpose is not to fit existing behaviour, but to discover patterns in the data that were previously unknown. For this reason, unsupervised learning is often closely related to the notion of 'Data Mining': digging through piles of unlabelled data for the gold of insights.

It's important to understand this key difference extends beyond whether or not labelled or unlabelled data is available: Supervised learning should be chosen to model existing behaviour, while unsupervised learning is chosen when new insights are required. Initially it might seem like this is an impossible task: What value is there in random patterns in data? After all, it's a proven principle of statistics (and therefore of machine learning) that correlation is not the same as causation. Patterns in a particular data set might just be present in that data set, and not in larger ones. Also, without the ability to give examples, the behaviour emerges *from the data*, which means that the result may not be what is expected. This is both good and bad, depending on the needs of the user.

## 1.2 Data

The data determines a very great deal in the context of unsupervised learning. There are a number of consequences of the lack of examples, and the lack of pre-determined behaviour. The first is that there is no guarantee that what the user expects to see is what will emerge: the dominant signal within the "pool of signals" including noise (for example, unexpected or random signals) might be different when using an algorithm compared to what the user observes when looking at the data. The second thing is that, generally, far more data is required. Since the objective is to find patterns that represent well the *general* case from the specific data, the broadest practical set of examples (including very similar instances) for pattern search are necessary.

Unsupervised learning is often used therefore to find potential behaviours or patterns for further exploration, kicking off a more complex or specific analysis. A second common use is in finding similar examples to an input instance, for example in recommendation of movies, books, or other items. A third common application is to learn a sequence from data to generate new sequences, for example generating new 'music'.

## 1.3 Evaluation

In the evaluation of supervised learning, a portion of the data is retained for evaluation. It is also possible to check the intermediate performance of the system through validation, for example using *cross-validation*, as discussed previously. In the purely unsupervised scenario, the interpretation of the results is a question for the user: do the clusters, signals and sequences discovered by the system make sense?

The specifics of how to evaluate depend on the task and the algorithm, as with other machine learning approaches. However, in general, there are some ways to approach the question 'has my system actually found a meaningful signal in the data?' or, borrowing from the mining analogy, 'have I struck machine learning gold?'

### Direct Evaluation

The most obvious approach to evaluating the performance of a system is to inspect the output. A common approach is to ask an independent, expert user if the patterns which are shown in the data make sense to them. This approach works well if the behaviour can be explained, and if there is a good way for the expert to judge the quality. In common tasks like clustering, or sequences detection, the evaluators are asked to decide if the clusters are coherent (i.e. if the members of the cluster are visibly related). It is common to employ several expert evaluators, and to measure the correlation of their agreement or disagreement.

A second approach to direct evaluation is to compare the results of the algorithm to a *Gold Standard*. This is similar to the supervised task. In this case, a pre-existing, labelled data set is created with examples that a system *should* be able to find. This approach is also known as external evaluation. It is common to apply a statistical measure, such as the F-measure, to the comparison.

There are trade-offs to both approaches: in the first instance, it can be laborious for experts to have to read large amounts of output and human judgement can be subjective. It can also be the case that the reason for the cluster or pattern is sensible, but the expert misses it. On the other hand, for the Gold Standard, the use of a pre-existing set of answers limits the *knowledge discovery* aspects of the task.

### Performance Metrics and Error Criteria

Another approach is to establish a requirement for the pattern to be *stable* and *internally consistent*. Two approaches exist for this: the first is to run the clustering algorithm multiple times on the input data, and look for stable patterns across the runs (effectively a kind of cross-validation). However, this is not guaranteed to converge (i.e. to form any useful agreement). The second approach is to set a criterion internally, such as an error measure, maximum distance metric or other internal property of the resulting output.

### Application Evaluation

Finally, it is possible to evaluate an unsupervised algorithm by making use of it in a particular application. For example, it might be possible to use the algorithm to improve the performance of a search engine, automatic speech recognition or recommendation tool. The performance gain in those systems might be partially or fully due to the performance of the unsupervised system depending on the particular application. A component of this approach could be to build a *supervised* system from the labelled patterns generated by the unsupervised approach, and to check the performance of the system against new data.

## 1.4 Summary

Unsupervised learning is about *discovering* signals in noise-like data: patterns in the data. It takes as its input unlabelled data, and the algorithm itself is responsible for making order from the unstructured data. Evaluating unsupervised learning is difficult because there is not necessarily a single 'right' answer.

## 1.5 SAQ

1. A simple way of deciding between supervised and unsupervised approaches is whether labelled data exists. Why this might not be correct? [Ans:1. no existing model might exist, 2. the objective might be to discover knowledge, rather than reproduce it]
2. How might you use direct evaluation to decide if your performance metrics are valid? [Ans: use expert judgement or gold standard criteria, and look for correlation with metrics]

# 2. Principal Component Analysis

This section will introduce and discuss one approach to signal detection with unsupervised learning. When the goal of unsupervised learning is to discover knowledge, or discover patterns in data, a key requirement is to be able to get the simplest description of the data points.

- For an interactive introduction to Principal Component Analysis (PCA) with more details, see: http://setosa.io/ev/principal-component-analysis/

- Suggested video tutorial, from Siraj Raval's YouTube channel: https://www.youtube.com/watch?v=jPmV3j1dAv4

## 2.1 Data Complexity

In machine learning, data is represented by a set of different attributes, each with a range of possible values. For example, a dataset about medical patients might include their sex, their eye colour, their weight, and their body mass index, or hundreds of other attributes. The challenge in machine learning is to make sense of this data in a statistically sound manner: how can the system best describe the behaviour of different data points for each attribute?

In the supervised learning unit, the SVM kernel trick was shown to be an approach for dealing with non-linear relationships. The intuition was that we wanted to transform the data to maximise the decision boundary in a linear, lower dimension space. In that case, we did it by applying a more complex function to the algorithm. Another approach is to find **principal components**: the most informative combinations of attributes of the data.

The real world of data is full of complexity that cannot be accounted for directly in the mathematical models. It's often necessary to reduce and separate the different attributes from one another. For example, in the medical case, a person's sex and their height are partially related. Similarly, their BMI is a direct combination of their height, weight and sex. Including all of these values in a model might result in confusion or lower predictive power. This is the reason for using PCA: how can we maximise the *variance* in the data? This means reducing the redundancy and the noise.

## 2.2 The Principal Components

Maximising the variance of the data can be achieved by transforming the attributes used to describe it. The relative position of the data points is unchanged, but the degree to which they differ is increased. PCA can achieve this by:

1. Translating the data to the origin
2. Scaling the attribute values
3. Combining attributes in different proportions

It is important to realise that any change is applied to the whole data: the transformation must be uniform.

Graphically, this is like scaling, translating and rotating points. Figure 1 shows an example of PCA applies to a set of data points in a 2D space.

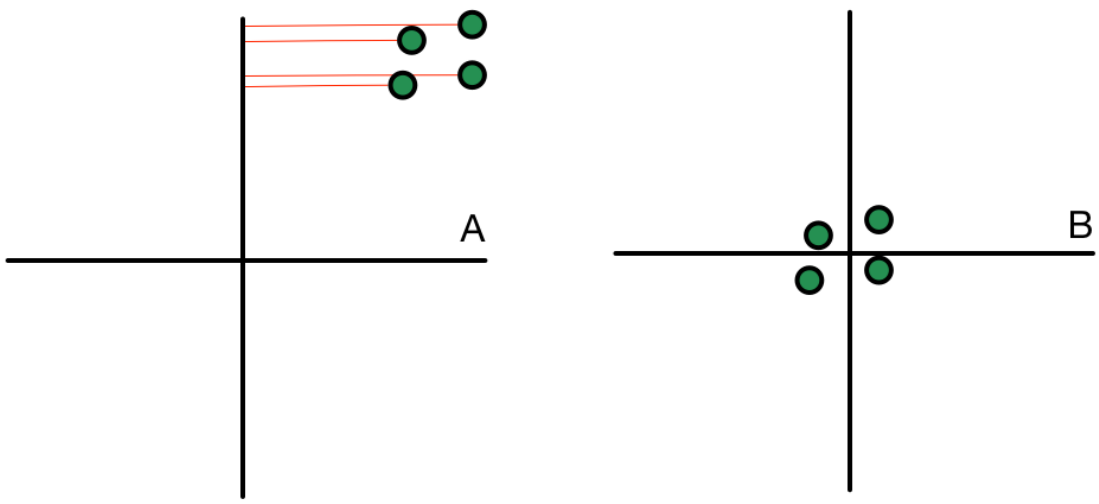



*Figure 1 – Example of PCA Transformation*

## 2.3 Dimensionality Reduction

In addition to transforming the data to a new basis, it might be desirable to reduce the number of attributes. This can be for two reasons: first to reduce redundancy, but also for performance reasons. Performance depends on the number of attributes because they can be costly in terms of processing and memory use. PCA allows the user to decide whether they can afford to remove attributes that are less informative, and thus helping to avoid the overfitting problem.

## 2.4 Mathematical Approach

PCA represents the data as an $m$ x $n$ matrix, where $n$ is the number of samples, and $m$ is the number of attributes.

A simple approach is to use eigenvector decomposition, but more advanced approaches exist, such as applying singular value decomposition of the matrix. In either case, the result is the same set of data with a changed base and increased variance.

The result effectively is an ordered list of the attributes of the data, so that the first component accounts for the highest information in the data, and so on.

PCA imposes some assumptions:

1. Linearity: As it is a set of linear transformations, the change of base assumes the linear relationships in the data.
2. The larger the variance, the more important the signal: The assumption is made that amplifying the variance does indeed represent the important changes in the data more obviously. This is not necessarily true: small variances might be highly important.
3. Continuous values: PCA in the simple case only applies to numerical values.

There are strategies to generalise and deal with non-linear behaviours and discrete values (categories).

For more detail on the mathematics, see: http://arxiv.org/pdf/1404.1100.pdf and on discrete data see: http://www.unc.edu/~skolenik/talks/Gustavo-Stas-PCA-generic.pdf

## 2.5 Summary

Unsupervised learning is focused on knowledge discovery: how to separate the signal from the noise in the variation of data. PCA is a technique for refining data: reducing redundancy, removing errors and simplifying the information represented by the data. It achieves this by applying different linear transformations to the data. The resulting representation is expected to be simpler and perhaps more useful. However, in order to achieve this, there are several assumptions made about the nature of variance, the distribution of the data and the values of the attributes. Any transformation such as this implies some loss of fidelity, so it is of course a trade-off.

## 2.6 SAQ

1. What are the three things that PCA seeks to reduce? [Ans: Covariance, Redundancy, Noise]
2. What is a Principal Component? [Ans: Graphically, an axis of the new base for the data, which preserves the most information possible]
3. Why must the data be linearly related? [Ans: the transformations are linear, and must preserve linear relationships in the data. Non-linear transformations would change the 'shape' formed by the data points]

# 3. K-Means Clustering

This section discusses one of the simplest algorithms in the unsupervised domain. The k-Means clustering algorithm can yield good results for many problems. In addition, it is often used as a first attempt to decide how to proceed with more advanced techniques later.

## 3.1 The Clustering Problem

One way to attempt to discover, or mine, knowledge from unlabelled data is to find groups, or clusters, in that data. This is called the clustering problem in machine learning, which is an unsupervised learning task. The reason that it is unsupervised is that it is not known in advance what the groups are and, thus, the groups in the data must be 'discovered'.

This is in contrast to a classification problem in supervised learning, where the groups are known, and the questions is to assign group membership to a particular data point. The two problems are linked: the difference is whether there exists already a model of groups (in the supervised case, there is) or not (as in the unsupervised case).

Suggested video tutorial, from Siraj Raval's YouTube channel:

https://www.youtube.com/watch?v=9991JlKnFmk

### K-Means Clustering

The intuition behind k-Means clustering is that the attributes in the data for each data point describe the data well, and on average the data points which are similar to each other should be grouped together. However, because there is no knowledge in advance of which groups might exist, it is necessary to arrange the data points in groups and then check the resulting groups for their correctness. Clustering therefore can be seen as an iterative process.

In summary, k-means clustering is based on the idea that 'things which have similar average values across their attributes should be grouped together'.

### 'Natural' Clusters and Data patterns

The clustering process assumes that there is a linear, separable cluster space that can be found by this process. This is similar to the challenge of finding class boundaries in the supervised learning problem. For example, in the diagram of Figure 2, the close, linearly separable groups are observable. This is the ideal situation for clustering. These clusters are sometimes referred to as being 'globular'.

Where the data is not this clearly separable, several complications can emerge, which can lead to sub-optimal division of the space. For example, a single natural cluster might be divided artificially into numerous smaller clusters, which are linearly separate.

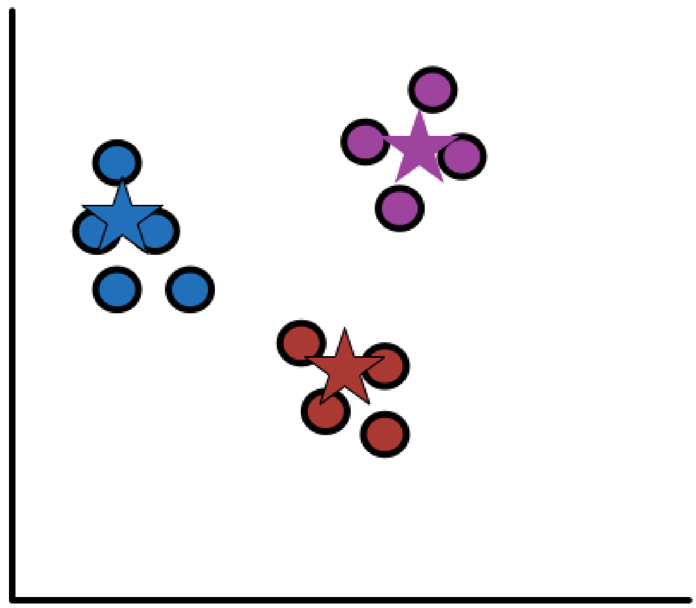


*Figure 2 - Example Globular Clusters with Centres*

For a discussion of the issues of cluster shape and the behaviour of k-Means in general, see http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

## 3.2 Algorithm

Given a set of data points, a distance measure and a value *k* for the number of clusters, the k-Means algorithm is as follows:

1. Initialise the centre of each cluster at random
2. Assign each data point to its nearest cluster based on the distance measure
3. Re-calculate the centre of each cluster based on the assigned data points
4. Repeat steps 2 and 3 until convergence (no change or small change in centres of clusters)

The formula for the centre of cluster $v_i$ is:

$$v_i = (1/N_i)\ \Sigma\ x_j$$

where $v_i$ is the centre of the cluster *i*, $N_i$ is the number of data points in that cluster, $x_j$ is the set of values for the attribute *j* of the data points in the cluster *i*. In simple terms, the cluster is centred on the arithmetic average value of each of the separate attributes for the data points in that cluster.

The process is roughly illustrated in Figure 3: The clusters are initially chosen at random (1), then the data points are assigned (2), and the centres are recalculated (3). The data points are then reassigned as necessary (4). Because a point was reassigned, the cluster centres are once again recalculated (5). This process repeats until no change of cluster is observed.

## 3.3 Choosing k

Cluster determination is highly data-dependent, and not always predictable. It can also be very uncertain whether the clustering process will converge and stop. Choosing k is therefore an iterative process, and requires some evaluation metric, either a gold standard or some measure of the coherence of the clusters.

One feature of this approach is that because the initial positions are chosen randomly, each time the algorithm is run a different answer might emerge.

### Variations

The basic algorithm has been extended in many ways, to reduce the effect of some of the weaknesses and assumptions. One problem is that the clusters are very sensitive to outliers, because one extreme value can drag the centre of the clusters away from the other points disproportionately. This can be somewhat relieved by using a different centre-point calculation, such as k-*Medioid* approaches. A second problem is that of handling large or non-globular data patterns. This can sometimes be solved by creating *hierarchical* clusters, where clusters over a certain size are sub-divided internally. Finally, the single-assignment issue can arise in some data: when two cluster centres are very close, a data point could belong to either set. This can be solved in part by the notion of *fuzzy* sets, which allow for different degrees of membership and overlap.

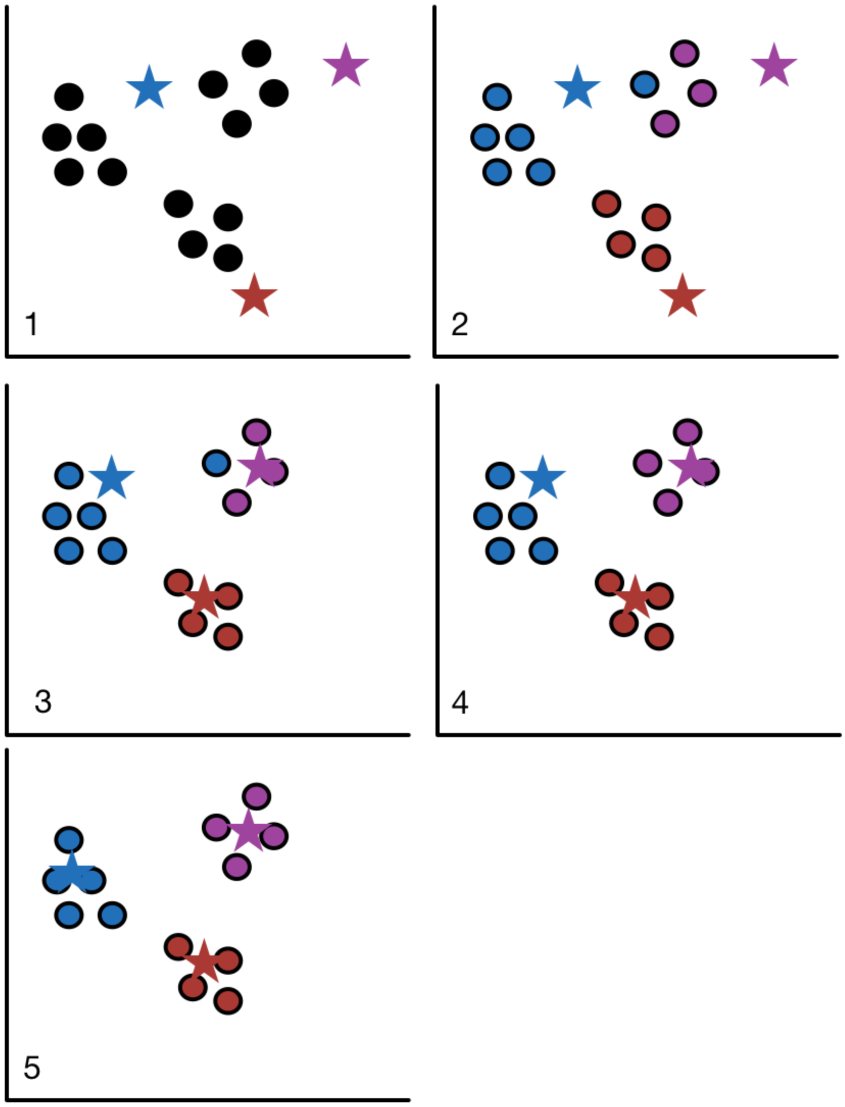



*Figure 3 - Example k-Means clustering: (1) the centres represented by starts are randomly assigned, (2) The data points are assigned to the clusters (represented with the same colour), (3) the centres of the clusters are recalculated. The last two points are repeated iteratively until a given stop criteria is reached. In this example, (4) and (5) represent the second iteration.*

## 3.4 Summary

K-Means clustering is an unsupervised approach to clustering, or grouping data points for which no labels exist. The choice of k is arbitrary and there are different ways of doing it based on the nature of the data. The data points are grouped together using (most commonly) a Euclidean distance metric. Initially, the cluster centres are chosen at random (this can have a significant impact). The data points are assigned to the nearest cluster and then the cluster centres move to the average of the values of these data points. Then, the distances are all recalculated, and the process repeats until no points change their cluster membership.

### 3.5 K-Means Clustering vs. K-Nearest Neighbour vs. K-fold cross validation

It's important not to confuse the different approaches discussed in this course that also use the k prefix:

- **k-Means clustering** is an *unsupervised machine learning* algorithm to solve the *clustering* task. This means it is designed to find groups of data where no labels are known.
- **k-Nearest Neighbour** is a *supervised machine learning algorithm* to solve the *classification* task. It is designed to find the boundaries in data between different *known* labels.
- **k-fold cross-validation** is a technique applied to validating machine learning experiments. It is not an algorithm. Instead, it is designed to allow all the data to be used to train a supervised learning algorithm in sequence.

### 3.6 SAQ

1. How is the centre of a cluster calculated? [Ans: Random (at the start) or average of the values of the dimensions of each of the data points in the cluster]
2. In the k-Means clustering process, what is convergence? [Ans: the point at which after recalculating the centres no change of assignment happens]
3. What assumptions are made about the type of data? [Ans: continuous, numerical values]
4. How to handle sub-optimal results? [Ans: re-run, change k, remove outliers, PCA]

## 4. Hidden Markov Models

This section discusses unsupervised learning with Hidden Markov Models (HMMs). The section will introduce the notion of HMMs and discuss the three types of problems which can be solved with HMMs. A key difference with the previous approaches described before is that HMMs work on data that is not independent. Some examples of HMM applications will also be briefly discussed.

### 4.1 Markov Models and Data

HMMs are widely used in areas such as bioinformatics, geographical information systems and *Computational Linguistics*, including Automatic Speech Recognition (ASR), grammatical analysis (Part of Speech Tagging), etc.

One feature common to all the systems which have been presented so far is that they were based on data sets comprising of single, independent values. This means that for the data we have looked at so far, there has been no relationship between one input and another. The fundamental assumption of independence is central to many statistical and mathematical models, especially those underlying machine learning. However, this notion of independence makes it impossible to solve many important problems in the real world. Two of the most common examples are ASR (recognising words from a sequence of speech sounds) and genetic analysis (recognising proteins from a sequence of amino acids).

A Markov Model is a mathematical model of the states and transition probabilities of a Markov process. The model can therefore be thought of as a set of states, with different probabilities for the next state from the current one.

A key requirement of the Markov model is that it must have a finite, countable number of states. This means that the data is not continuous as in other examples.

### The Markov Assumptions

While HMMs do permit data to be inter-related, they do not generalise to all kinds of inter-relations. In particular, two strong assumptions are made in the Markov Model:

1. **The Markov Assumption**: The next state is dependent only on the current state. This defines a *first-order* Markov Model. If the current state depends on *x* previous states, it would be a model of order *x*. Higher order models are computationally complex, but the fundamental principles are similar.
2. **The Stationary Assumption**: The behaviour of the transition probabilities is independent of *real* time. That means that the transition probabilities remain the same for the life of the model.
3. **Output independence**: The current output is independent of previous outputs.

## 4.2 Hidden Markov Models: Three Types of Problem

Take an example of a ski resort. Assume that the conditions at the ski resort depend only on those of the previous day. The resort can have two states: Safe to ski, or Unsafe. A direct Markov process would take the observations from a sequence of days, and compute the likelihood of each state given the current state. The top diagram in Figure 4 shows the Markov Model of the sky resort for a sequence of days.

However, it is often the case that the states of the system are not directly observable. In our ski example, imagine that it is not possible to observe the ski slope from the bottom of the mountain, and it is necessary to rely on an atmospheric reading. The slope's state would be *hidden* and would have to be deduced from the *observed* state of the atmospheric pressure: high or low. The bottom diagram of Figure 4 illustrates a Hidden Markov Model for the sky resort example.

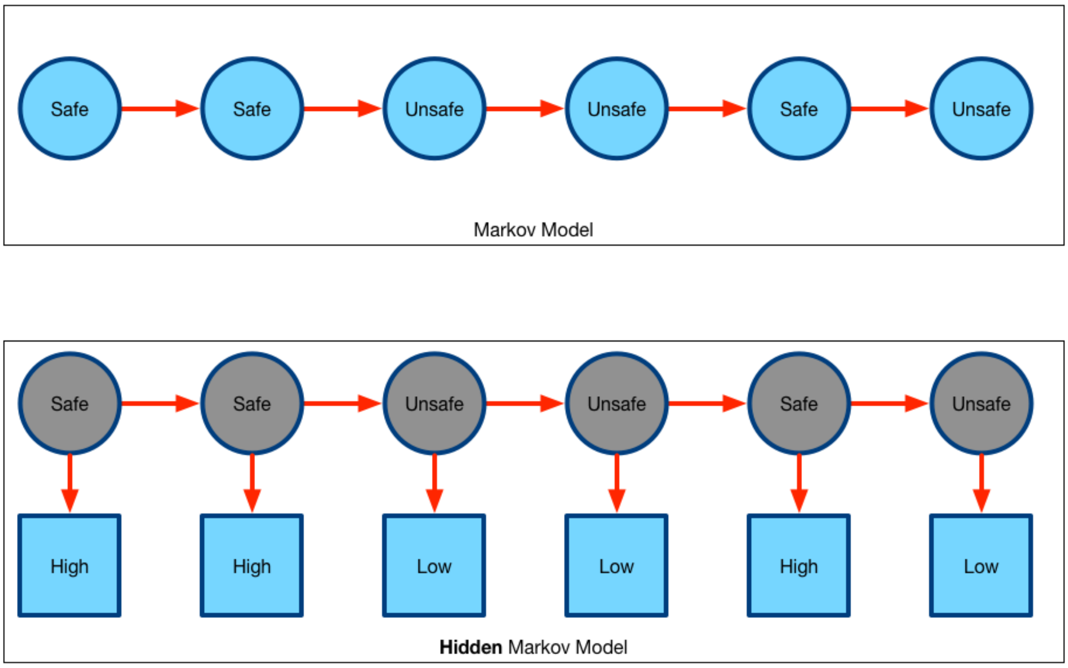



*Figure 4 - State transitions of the HMM representing the conditions of the Ski Resort*

This means that in a Hidden Markov Model the objective is to understand the sequence of observations based on an underlying, hidden, set of probabilistic transitions between states. The diagrams in Figure 5 represent the possible transitions between states for the case of a Markov Model and a HMM.

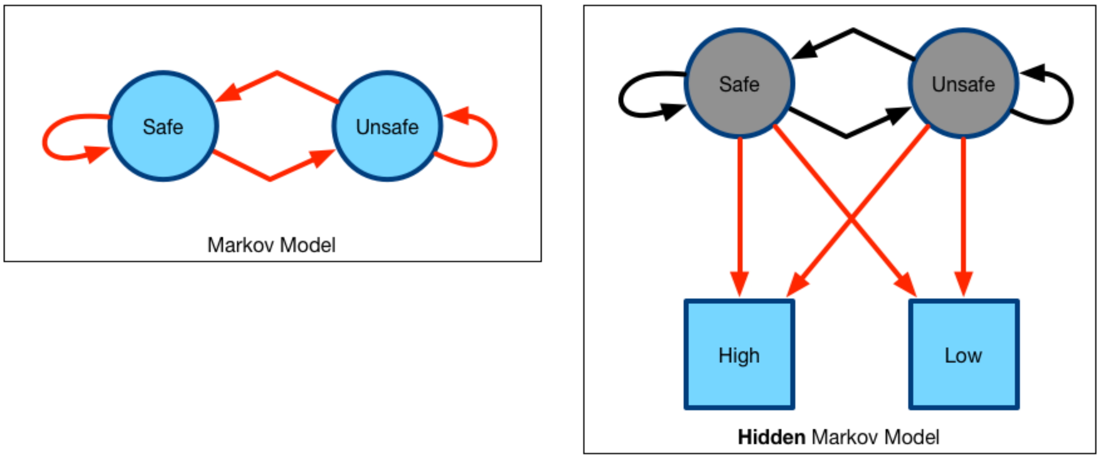



*Figure 5 - Ski Resort Graphical Models*

The HMM is therefore composed of Five parts:

1. The list of possible States
2. The list of possible Observations
3. The State Transition Probabilities
4. The Emission Probabilities

5. The Initial State Probability

Depending on the goal of the user, different problems can be solved with HMMs. These are categorised into three types:

### Problem 1: Evaluation

The evaluation problem is characterised by the question 'Given the HMM parameters, and a set of observations, how likely is the sequence observed?'

A classic example of this is the dishonest casino. We observe the results of a dice roll as the emission. The dice can either be fair (equally likely results) or loaded (far more likely to roll low). We estimate that the dealer is switching between fair and unfair dice with a certain probability. Each of these types of dice is a state in the Markov Model, and the sequence of observed results will let us judge how probable it is that the dealer is cheating according to our model.

In the ski example, we could use the sequence of observed atmospheric models to decide if the weather is unusual. This outcome might prompt us to close the slope.

Evaluation is achieved using the *Forward Algorithm*, which considers every possible path through the transition states and calculates the probability of the observed sequence.

### Problem 2: Decoding

The second problem is to infer the most likely *path* through the states for a particular set of observations. In other words, what is the most likely *state sequence* for a particular *observed sequence*?

This is usually solved using the *Viterbi Algorithm*. This algorithm sequentially computes the most likely (highest probability) path through all the possible transitions of the states, given the observations.

For more detail of the Viterbi Algorithm, see:

https://en.wikipedia.org/wiki/Viterbi_algorithm

### Problem 3: Estimation

In the third case, we do not know the transition or emission probabilities. We know how many states and how many observation values are possible. The objective here is to estimate the model. This is achieved using the *Baum-Welch* algorithm. Briefly, the algorithm begins with initial probabilities (random), and proceeds *Backwards and Forwards* through the different observations, estimating the state transition and emission probabilities until convergence.

## 4.3 Additional Resources

Some additional Resources:

- Xing, Singh "Hidden Markov Models"
  http://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture17.pdf

- Rabiner “Tutorial on Hidden Markov Models”
  https://www.robots.ox.ac.uk/~vgg/rg/slides/hmm.pdf
- Craven “Hidden Markov Models”
  https://cw.fel.cvut.cz/wiki/_media/courses/a6m33bin/hmms-1.pdf
- Video on Bayes Theorem, from Brandon Rohrer’s YouTube blog:
  https://www.youtube.com/watch?v=5NMxiOGL39M
- Suggested videos:
  https://www.youtube.com/watch?v=kqSzLo9fenk
  https://www.youtube.com/watch?v=EqUfuT3CC8s
  https://www.youtube.com/watch?v=4XqWadvEj2k

## 4.4 SAQ

1. In a HMM, what is ‘hidden’? [Ans: the underlying probabilistic model consisting of the state of the system. Each state has a probability of emitting a particular observation and transition probabilities]
2. What is the difference between Evaluation and Decoding [Ans: the first estimates the likelihood of a sequence, while the second estimates the most likely sequence]