

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

**THE OPTIMAL PLACEMENT OF SWITCHING
DEVICES ON RURAL MEDIUM-VOLTAGE
SYSTEMS**

**NEEMESH PITAMBER
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITY OF CAPE TOWN**

**THESIS PREPARED IN FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MSC IN ELECTRICAL ENGINEERING**

JUNE 2001

'I do not know what I may appear to the world, but to myself I seem to have been only like a boy playing on the sea-shore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me'.

Issac Newton

University of Cape Town

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to the following people without whom this contribution would not have been possible:

Charles Dingley for his guidance and persistent motivation throughout this entire thesis.

The following people from Eskom for their willingness to help:

Sakkie van Aarde - Protection department in Eskom Worcester.

Izak van der Merwe - Protection department in Eskom Brackenfell.

Riaan Smit - Planning department in Eskom Brackenfell.

Brendan Jackson - Customer service department in Eskom Bellville.

Richard Ahlschlager - Protection department in Eskom East London.

Tony Hunt for taking the time to give me a course on reticulation protection.

Finally, I would like to give a special thanks to my family and my friends namely Krishnen, Hitesh, Shahil, Nilan, Emile, Kalpana, Amit, Shireen, Fatima and Laila for their constant support and encouragement without which I would definitely have not succeeded so far.

SUMMARY

Electricity is supplied to rural areas by radial networks operating for example at 11 or 22 kV. A problem with radial networks is that faults on the feeders result in the loss of electricity supply to all customers if they are not sectionalised. Some radial networks emanating from one feeder from the sub-station may consist of up to 300 km of line, taking into account spurs, which makes fault detection difficult and increases customer outage time.

Protective devices such as reclosers, sectionalisers and fuses are placed on distribution systems to reduce the number of customers that will not have service at one time or another due to any fault on the system. There do not appear to be clear rules with regard to the placement of these protective devices. With the installed cost of devices such as sectionalisers being over fifty thousand rand, this is naturally of concern to a power utility.

The aim of my study is to contribute ideas and develop a computer model, based on reducing the cost of energy not supplied on a system, in order to determine the optimum number and location of sectionalisers. This model can act as a tool to guide planners in their decision making for the placement of sectionalisers on a distribution system.

A genetic algorithm-based method has been developed by Levitin, Mazal-Tov and Elmakis (1994) to determine switching device locations. Although Levitin, Mazal-Tov and Elmakis successfully implemented a genetic algorithm version, my work aims to explore the more recently developed Population Based Incremental Learning (PBIL) algorithm to solve the same problem. The PBIL algorithm is a stochastic search technique that combines characteristics of both the genetic algorithm and competitive learning (Baluja, 1994). It has been shown

to match and even outperform the genetic algorithm (Baluja, 1994). I decided to solve the same problem using the PBIL algorithm not only because it might be a superior tool in this case, but because it might be a useful tool in solving other problems.

The computer model that I designed was tested on two demonstration networks. These networks consisted of 53 and 96 sections respectively. Sectionalisers were placed in suitable locations, on the networks, during the testing of the computer model. Even though Levitin, Mazal-Tov and Elmakis (1994) used an advanced form of the genetic algorithm, it took still more evaluations to attain the optimum result than the PBIL version that I have used.

The model shows that the optimal number and placement of sectionalisers is very sensitive to parameters such as failure rate, cost of unsupplied energy, discount rate and the average repair time of a fault. This suggests that the model could perhaps be useful as a planning tool to complement a network planner's experience and judgement.

TABLE OF CONTENTS

	PAGE
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 PROTECTION DEVICES USED ON RURAL MV NETWORKS	5
2.1 Temporary and permanent faults	5
2.2 Application of protection devices on a system	6
2.3 Reclosers	7
2.4 Fuses	8
2.5 Sectionalisers	12
2.6 Conclusion	14
CHAPTER 3 THEORY OF POWER SYSTEM DISTRIBUTION RELIABILITY	15
3.1 Introduction to power system reliability	15
3.2 Reliability cost/worth assessment	17
3.3 Reliability Indices	19
3.4 Conclusion	24
CHAPTER 4 PROTECTION PHILOSOPHIES DEVELOPED FOR SYSTEM IMPROVEMENT	25
4.1 Load Exposure Index developed for the placement of protective devices	25
4.2 Economic optimisation of an Australian network	31
4.3 Conclusion	37
CHAPTER 5 INTRODUCTION TO EVOLUTIONARY ALGORITHMS	38
5.1 What are evolutionary algorithms?	38
5.2 Biological background	39
5.3 Terminology used in Genetic Algorithms	40

5.4	Introduction to Genetic Algorithms	42
5.5	Introduction to the PBIL	51
5.6	Conclusion	57
CHAPTER 6 DESIGN OF THE COMPUTER MODEL		58
6.1	Purpose and overview of the model	58
6.2	Input data	59
6.3	Output data	60
6.4	Data structures used	61
6.5	Application of the PBIL to this problem	63
6.6	Implementation of the model	75
6.7	Testing	75
6.8	Conclusion	79
CHAPTER 7 TEST RESULTS USING THE PBIL ALGORITHM		80
7.1	Sensitivity of the PBIL to its parameter settings	80
7.2	Effect of the unit cost of ENS on the annual savings in the total cost for a utility	85
7.3	Testing the model on two MV networks	89
7.4	Future enhancements to the computer model	101
7.5	Conclusion	102
CHAPTER 8 CONCLUSIONS		103
REFERENCES		104
APPENDIX A	DISTRIBUTION SWITCHGEAR	
APPENDIX B	INPUT DATA FOR THE PROGRAM	
APPENDIX C	MATLAB CODE FOR THE PROGRAM	
APPENDIX D	TESTS PERFORMED ON LEBANON NETWORK	
APPENDIX E	TESTS PERFORMED ON LEVITIN NETWORK	

***Chapter 1* INTRODUCTION**

Protection plays two major roles in power systems. One role is the protection of power system equipment in the event of a fault and the other is the reduction of inconvenience to customers due to power interruptions. Improvement in protection systems will result in a more reliable service to customers.

Customer failure statistics show that MV distribution systems contribute as much as eighty percent of the duration of outages (Billinton and Allan, 1996:221). Recent years have seen an increased emphasis on the power quality of MV distribution systems (these voltages are typically 11 kV and 22 kV in South Africa). Power quality covers harmonic content, voltage stability and the frequency and duration of outages. It is the last aspect, frequency and duration of outages, that this thesis is concerned with. A good protection system reduces the frequency and duration of outages ensuring better service to customers.

Sectionalising devices are normally placed on distribution systems to isolate faulted sections from the remainder of the system. This reduces the number of customers affected by faults on the system. There do not appear to be any clear rules for the placement of sectionalising devices such as reclosers, fuses and sectionalisers on rural MV systems. With the installed cost of a device such as a sectionaliser being of the order of R50 000, this is naturally a cause for concern.

The aim of my thesis is to contribute ideas on this question by developing a computer model that will determine the optimum number and placement of sectionalisers on a rural MV feeder by calculating the economic optimum in terms of the cost of unsupplied energy.

Without the use of computers, protection engineers rely on their experience for the placing of these protective devices. Selecting the optimum number of sectionalising devices and determining the best location for them is not an easy task. With the increased interest in the application of evolutionary computation in power systems, I have decided to develop a computer model that uses an algorithm that could be employed as a tool to aid network planners in the optimal placement of sectionalisers.

Genetic algorithms have been found to be suitable for a wide variety of problems in power systems, for example load forecasting, optimal capacitor placement and service restoration. As an example, Sundhararajan and Pahwa (1994) developed a computer model for a capacitor problem that uses a genetic algorithm to determine:

- 1) The most adequate locations to install capacitors.
- 2) The types and sizes of capacitors to be installed.
- 3) The setting of capacitors at different loading conditions.

This optimisation was solved using the GA by formulating the problem as a multi-objective, constrained one.

A genetic algorithm-based method has already been developed for the optimal allocation of sectionalisers (Levitin, Mazal-Tov and Elmakis, 1994). I decided to tackle this same problem using the so-called Population Based Incremental Learning algorithm (PBIL). The PBIL integrates properties of a genetic algorithm with competitive learning. This results in a simple yet powerful algorithm, which has outstripped the genetic algorithm both in terms of speed and in the final solution obtained for solving problems that have carefully been devised with the intention of favouring the genetic algorithm described by Baluja (1994). I decided to implement my computer model using the PBIL algorithm as this work could

have a useful spin-off in showing up other possible uses of PBILs in the power system field.

Research for this thesis was carried out at UCT over a period of 15 months between March 1999 and June 2000. A literature search was performed to obtain the necessary background information to tackle the problem. Many textbooks, journal articles, conference proceedings and research reports were consulted to obtain information.

I searched throughout the "Inspec" and "Applied Science and Technology Abstracts" electronic databases as well as through the IEEE power journals and electrical power textbooks that are available in UCT's library. The literature research that I carried out produced twelve papers, two technical reports and two websites that proved to be very helpful. I discussed the problem with people who work in the protection field and I also visited Eskom's Worcester, Brackenfell and Bellville offices.

The following people were consulted during the study:

- Sakkie van Aarde - Protection department, Eskom Worcester.
- Izak van der Merwe - Protection department, Eskom Brackenfell.
- Riaan Smit - Planning department, Eskom Brackenfell.
- Brendan Jackson - Customer service department, Eskom Bellville.
- Richard Ahlschlager - Protection department, Eskom East London.
- Tony Hunt - Retired Eskom protection engineer.

I made a trip to Eskom Worcester and there Sakkie van Aarde explained the protection equipment and philosophies adopted by Eskom. He showed me a spreadsheet model that he has developed, which calculates the system performance indices of feeders in Worcester.

I attended a course offered by Mr Tony Hunt. He kindly condensed and compressed a four-day reticulation protection course into one day's duration for me. This course has been acknowledged as being a very comprehensive, practical and technically useful one concerning reticulation protection up to 22kV.

This thesis is made up of eight chapters. Chapter two gives a description of the protection devices used on rural MV systems. The basic operation of sectionalising devices, namely reclosers, fuses and sectionalisers, is discussed.

Chapter three discusses the importance of improving the reliability of distribution systems from both the utility's and customer's points of view. Performance indices used to measure the reliability of a system are described.

The fourth chapter discusses protection philosophies that have been developed and implemented for improving the reliability of distribution systems.

Chapter five gives an introduction and a general understanding of a genetic algorithm. An introduction to the Population Based Incremental Learning (PBIL) algorithm is then given.

The design of a computer model based on the PBIL algorithm is given in chapter six. The development of this model is the main aspect of this research work.

Chapter seven presents the results obtained from testing the model on two demonstration networks. A discussion of the results obtained is integrated into this chapter.

Chapter eight presents the main conclusions arising out of this work.

Chapter 2 PROTECTION DEVICES USED ON RURAL MV NETWORKS

The purpose of this chapter is to describe the most commonly used protection devices on a distribution system for the isolation of faulted sections. Isolation of faulty sections limits customer service interruptions on the network. This chapter begins with a brief discussion of temporary and permanent faults. It then describes protection devices such as reclosers, sectionalisers and fuses.

2.1 TEMPORARY AND PERMANENT FAULTS

Approximately 80% of faults occurring on a distribution system are temporary in nature (Billinton and Allan, 1996:221). Temporary faults, also known as transient faults, are faults that can be cleared by the operation of a circuit breaker or recloser and do not recur when the line is re-energised (GEC, 1987:235). Examples of these temporary faults would include (Coopers, 1990:7):

- Arcing from a live terminal to ground caused by a bird or snake.
- Clashing of two conductors.
- A tree branch falling across two conductors, which eventually burns off.
- Arcs resulting from lightning over-voltages.

The remaining 20% of faults that occur on distribution systems are known as permanent faults. These faults require repair work before the supply can be restored. Examples of causes of such faults include (Coopers, 1990:7):

- A fallen conductor.
- A broken insulator.

- A vehicle crashing into a pole.

2.2 APPLICATION OF PROTECTIVE DEVICES ON A SYSTEM

A rural distribution system usually consists of an extensive system of overhead radial feeders. A circuit breaker is installed at the main substation because circuit breakers have a high fault current interrupting capacity.

The overall problem caused by faults is always one of restricting the area of outage to a minimum and ensuring rapid restoration of supply. This can be accomplished by the application of overcurrent protective devices such as sectionalisers, reclosers and fuses (see Appendix A). Figure 2.1 shows the application of these protection devices on a rural radial MV distribution system. A detailed description of these devices will be given in the following sections.

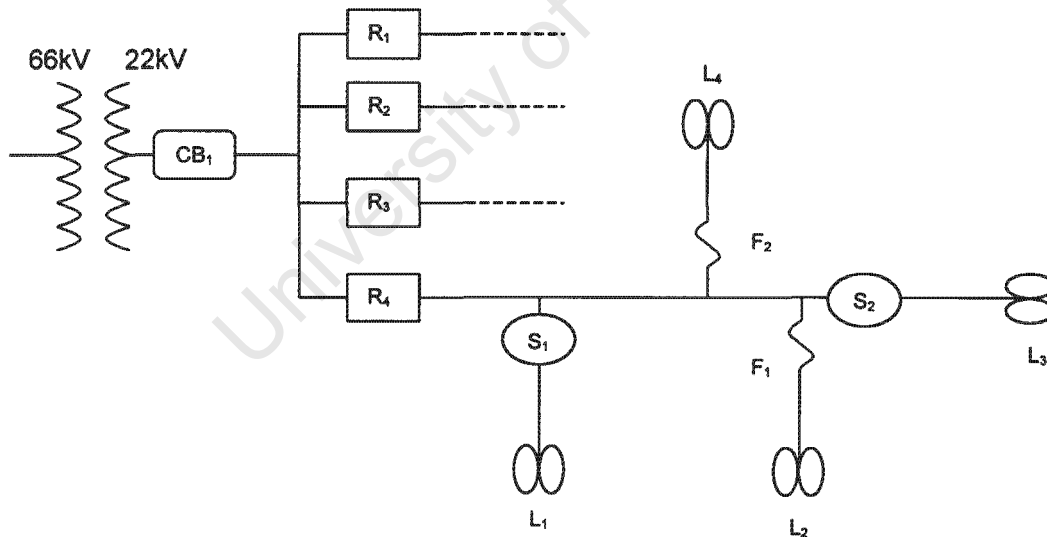


FIGURE 2.1 DISTRIBUTION SYSTEM SHOWING THE APPLICATION OF PROTECTIVE DEVICES

The distribution system shown in Figure 2.1 contains a substation transformer with a 66kV primary and a 22kV secondary. The transformer feeds four 22kV main feeders, each protected by a recloser, R_1 to R_4 . The circuit breaker, represented by CB_1 , serves as backup protection for the four main feeder reclosers. The feeder containing recloser R_4 supplies electricity to four loads L_1 , L_2 , L_3 and L_4 . The sending end of each lateral, connected to the primary line of this feeder, has a fuse (F_1 and F_2) or sectionaliser (S_1 and S_2) installed to restrict an outage to the smallest practical area of the system.

2.3 RECLOSERS

Reclosers are compact, pole-mounted circuit interrupters with automatic reclosing capabilities. They are smaller, lighter and cheaper than circuit breakers and do not have complicated and costly installation procedures. They may be single- or three-phase units that are hydraulically or electronically controlled. Reclosers are coordinated with upstream substation circuit breakers and with either or both downstream sectionalisers and fuses.

Reclosers permit temporary faults to clear without de-energising any section of the distribution system permanently. They also interrupt permanent faults in sufficient time to protect against conductor burn down or equipment damage. Reclosers allow devices in coordinated protective systems to isolate faulted sections. Reclosers can be placed in series on lengthy feeders to prevent the loss of supply to the entire feeder due to faults occurring towards the end of it.

Figure 2.2 shows the typical operation sequence of a recloser used in conjunction with a downstream sectionaliser and fuse. The first closure is almost instantaneous in clearing any temporary faults that exist before the downstream fuse has a chance to blow. The sectionaliser registers this first closure as its first count. The fuse blows during the first delayed trip in the recloser's sequence and

this registers as the sectionaliser's second count. The recloser and sectionaliser reset if the blown fuse clears the fault. If the fault is still present, the sectionaliser opens after the second delayed trip which it registers as its third count. If the recloser reaches a "lockout" stage, it must be manually reset to resume operation (Coopers, 1990:67). A recloser would "lockout" if, for example, a fault that is caused by a tree branch falling across the conductors persists after the third delayed trip.

The preset number of operations before lockout is normally set to four reclosures. If the recloser does not lockout and no further overcurrents are detected after a time of approximately two minutes, then the recloser automatically resets.

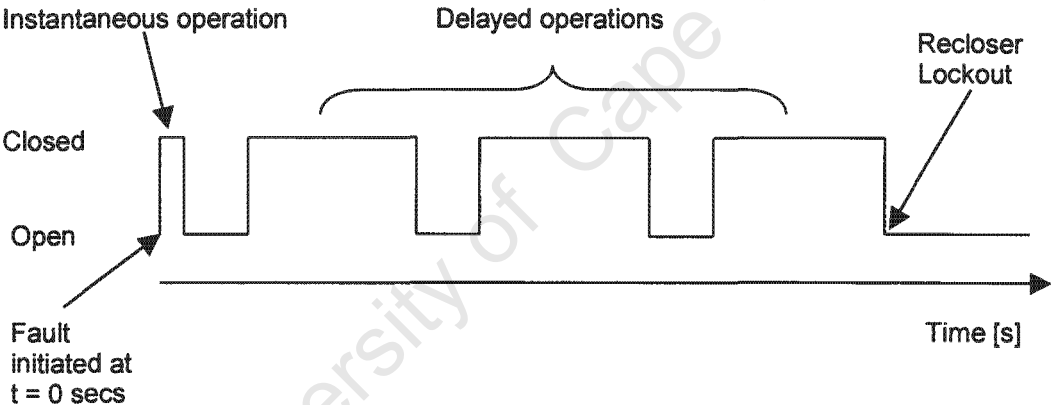


FIGURE 2.2 RECLOSER OPERATION TO LOCK-OUT

2.4 FUSES

"A fuse is a device that, by the fusion of one or more of its specially designed and proportioned components, opens the circuit in which it is inserted and breaks the current when this exceeds a given value for a sufficient time. The fuse consists of all the parts that form the complete device" (BS, 1975:270).

A fuse is the simplest form of overcurrent protection that can be used on a distribution system. It consists basically of a piece of conducting material known as a fuselink that melts when an overload current passes through it. A fuselink is the “replaceable part or assembly, comprised entirely or principally of the conducting element, required to be replaced after each circuit interruption to restore the fuse to operating condition” (ANSI C37, 1994:40). When the fuselink “breaks”, the faulted section is isolated from the rest of the system.

The fuselink acts as a conductor when the current that flows through it is less than the continuous current rating of the fuse. The fuse behaves differently depending on the magnitude of the overcurrent. If a fault current that is up to six times the continuous current rating of the fuse occurs, a certain segment or breakpoint of the fuselink is designed to melt due to the increase in temperature. A gap forms at the fuselink’s breakpoint. An arc will then form across this gap. Then, the width of the gap increases because the arc causes the link to burn away from the breakpoint. The gap increases to such an extent that the arc can no longer be sustained and is extinguished. If the fault current is greater than six times the continuous current rating, many of the breakpoints along the fuselink melt simultaneously, causing rapid arc extinction. (Glover and Sarma, 1994:329)

Fuses are manufactured with the following specifications according to Glover and Sarma (1994:330):

1. *Voltage rating* – This is the rms voltage that determines the fuse’s ability to suppress the arc that forms when the fuselink melts. Medium voltage fuses are available for system voltages up to 38kV.
2. *Continuous current rating* – This is the maximum rms current that flows through the fuselink without causing it to melt.
3. *Interrupting current rating* – This is the maximum current that the fuse can safely interrupt. Examples of standard interrupting currents for medium voltage fuses are 65, 80 and 100kA respectively.

4. *Time response* – The melting and clearing time of a fuse is obtained from a 'time-current' curve and depends on the magnitude of the fault current. Fuselinks are normally manufactured from metals such as tin, lead or silver and combinations of such metals in order to obtain a specific time-current characteristic curve. Typical time-current characteristic curves for a 10K fuselink are shown in Figure 2.3. The 10 represents the interrupting current rating in KA for the fuse. In this case, the maximum current that the fuse can safely interrupt without damaging any other part of the fuse besides the fuselink is 10KA. K is a standard or recognized specification used to describe 'fast' fuse types, that is, fuses that have a speed ratio of 6 to 8. The speed ratio is the ratio of the minimum-melt current at 0.1s to the minimum-melt current at 300s (Coopers, 1990:52). The minimum melting time current characteristic, or TCC, is the time when the fuselink begins to melt. The maximum clearing TCC curve is the maximum expected time for the fuselink to 'blow'. Figure 2.3 shows that the fuselink begins to melt at 0.8s and clears within 1.5s for a fault current of 50A.

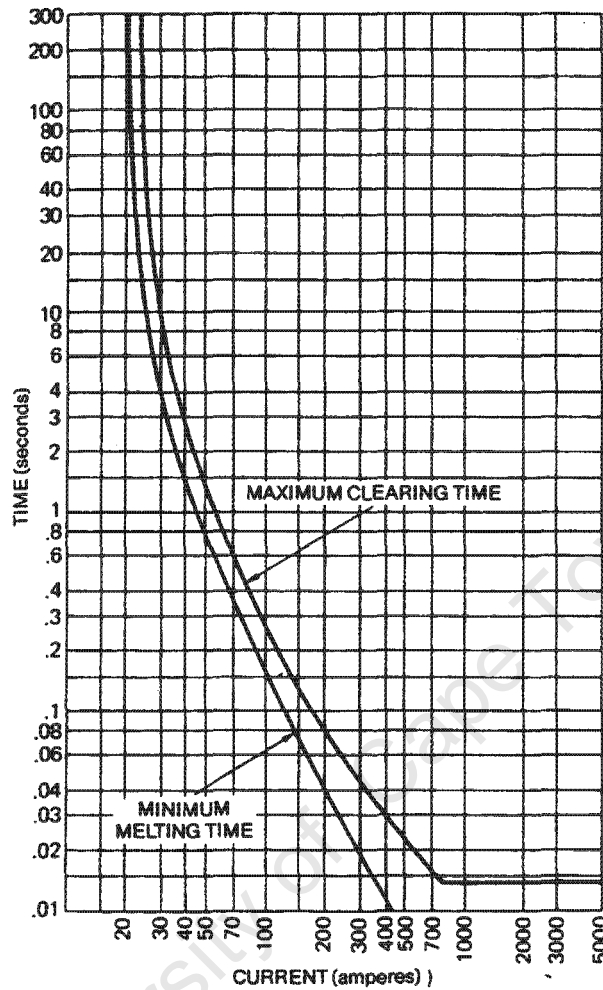


FIGURE 2.3 TCC CURVE FOR 10K FUSELINK (COOPERS,1990:53)

Fuses are reliable devices. They are not expensive and are relatively small in size when compared to devices such as reclosers and sectionalisers. They do not require a backup protection device such as a recloser in order to operate. The main disadvantage of a fuse is that the fuselink has to be manually replaced after it melts.

The following points should be taken into consideration for the application of fuses on a distribution system: (Eskom, 2000)

1. Fuses do not operate fast enough to clear lightning surges as they travel at nearly half the speed of light. Fuses with a low interrupting current rating (or rupturing capacity) will be damaged by lightning surges.
2. There is the possibility of human error in the placement of fuses on a distribution system for example, placing a 15K fuse instead of a 25K fuse.
3. With time, the fuselinks chemically degenerate and can blow needlessly.
4. System disturbances can cause fuses to blow unnecessarily if the fuses are not coordinated properly. For example, fuses must withstand inrush currents caused by transformers being energised.
5. As it is not economically viable to manufacture a multitude rating of fuses, only certain ratings are manufactured for networks. As an example, South African distribution systems keep 30K, 20K, 15K, 10K and 6K fuses in stock.

2.5 SECTIONALISERS

Sectionalisers are automatically controlled devices that isolate faulted sections of line from the rest of the distribution system. They are pole-mounted devices and are not usually placed on the primary line in order to confine outages to the smallest segment of the distribution system possible. They are more commonly used for protecting the primary line from faults occurring in lateral sections of the distribution system.

A sectionaliser should not be confused with a recloser because it does not have the capacity to interrupt a fault current. As a result, they are placed in conjunction with an upstream back-up device, such as a recloser or reclosing circuit breaker that possesses fault current breaking capacity. They can be used in the place of fuses or between a recloser and a fuse.

Sectionalisers count the number of recloser operations during fault conditions. After a predetermined or preset number of recloser operations, while the recloser is still open, the sectionaliser also opens and “locks out” in order to disconnect the faulty section (Coopers, 1990:68). The recloser is then reclosed, restoring service to customers in fault-free areas of the distribution system.

For temporary faults, the sectionaliser does not complete counting the set sequence of fault current pulses required for it to lock out. As a result, the sectionaliser remains closed and will reset itself to start the counting sequence for the next fault that occurs.

Sectionalisers are manufactured in single- and three-phase units, with hydraulic or electronic operating mechanisms. Although they are far more expensive than fuses, the benefits that a utility can derive from placing sectionalisers make them a good investment. These benefits include “application flexibility, convenience and safety” (Coopers, 1990:68) and examples of the benefits are given in the next paragraph.

Testing a circuit after a permanent fault occurs is simplified because of the fault-closing capability of the sectionaliser. Should the fault still be present after the sectionaliser is reclosed, the back-up recloser will protect the circuit. Since sectionalisers do not require any replaceable parts after a lockout, for instance a fuselink in the case of a fuse, service restoration is far more rapid and convenient. The problem of choosing the wrong size and type of fuselink is also avoided. They do not open accidentally under load due to a damaged link. Sectionalisers do not have time-current characteristic curves and can be used between two protective devices whose operating curves are very close (Coopers, 1990:68).

Basic coordination between the sectionaliser and the recloser requires two steps:

1. The actuating level of the sectionaliser needs to be established. A count is registered each time the current sensed is greater than this activating level.
2. The sectionaliser needs to be programmed to count one less than the number of trips to the lockout of the backup recloser. If the recloser is set to lockout after four trips then, the sectionaliser should be set to count to a maximum of three to open.

2.6 CONCLUSION

Most faults on a distribution system are temporary in nature. The loss of electricity supply to customers due to faults on radial distribution systems can be reduced by the application of overcurrent protective devices such as reclosers, sectionalisers and fuses. The use of these devices enables rapid restoration of electrical supply to customers.

Chapter 3 THEORY OF POWER SYSTEM DISTRIBUTION RELIABILITY

This chapter discusses various measures of distribution system reliability. It then discusses the reliability cost/worth model that has been developed to determine the 'worth' of the change in reliability in a system. The chapter concludes with a discussion on a selection of performance indices used for evaluating the reliability of distribution systems.

3.1 INTRODUCTION TO POWER SYSTEM RELIABILITY

"The word 'reliability' has a wide range of meanings in the power industry. At the moment, 'adequacy' and 'reliability' could be used synonymously in general utility parlance. Adequacy evaluation encompasses all aspects of assessing the ability of a component, subsystem or system to perform its intended function. Many varied activities such as relay co-ordination, load flow analysis, and equipment sizing rightly fall on this broad definition of reliability evaluation." (EPRI, 1981)

This thesis views reliability from a much more restricted perspective: "Reliability evaluation involves observing outages on the distribution system, noting their causes and effects, compressing this historical data into relevant summary information, and using this information to improve the performance of existing systems." (EPRI, 1981)

The consequences of an outage on a distribution system are generally less serious than those on the generation side because they have a localised effect rather than a system-wide effect.

Table 3.1 shows customer supply unavailability statistics for a particular distribution utility in the UK. (The name of the utility and the date when these statistics were compiled are not given). The middle column shows the contribution of each 'sector' of a power system (named in the first column) towards the annual average number of minutes of the unavailability of supply to a customer. The last column shows each of these contributions as a percentage of the total annual number of minutes of supply unavailability per customer (96.8 minutes for this utility). From these statistics, it can be seen that distribution systems (2nd to 5th contributors in the first column) constitute over 80% of the unavailability of supply to a customer. This indicates that there is a need to be concerned with evaluating the reliability of distribution systems.

CONTRIBUTOR	AVERAGE UNAVAILABILITY PER CUSTOMER YEAR	
	(minutes)	(%)
Generation/Transmission	0.5	0.5
132kV	2.3	2.4
66kV and 33kV	8.0	8.3
11kV and 6.6kV	58.8	60.7
Low voltage	11.5	11.9
Arranged shutdowns	15.7	16.2
Total	96.8	100

TABLE 3.1 CUSTOMER SUPPLY UNAVAILABILITY STATISTICS (BILLINTON AND ALLAN, 1996:221)

3.2 RELIABILITY COST/WORTH ASSESSMENT

This section shows that economics play an important role in determining the appropriate investment cost to improve a system's reliability. The key question asked is "Where or on what should the next pound, dollar or franc be invested in the system to achieve the maximum reliability benefit?" (Billinton and Allan, 1996:13)

Figure 3.1 illustrates how the reliability of a power system is related to an investment cost. It shows that the system investment must be increased to improve reliability of the supply. As the reliability of a system is increased, there is a decreasing increment in reliability ΔR for a given increase in investment ΔC . This means that system reliability becomes increasingly expensive for a utility as it is pushed towards a target level of 100% reliability.

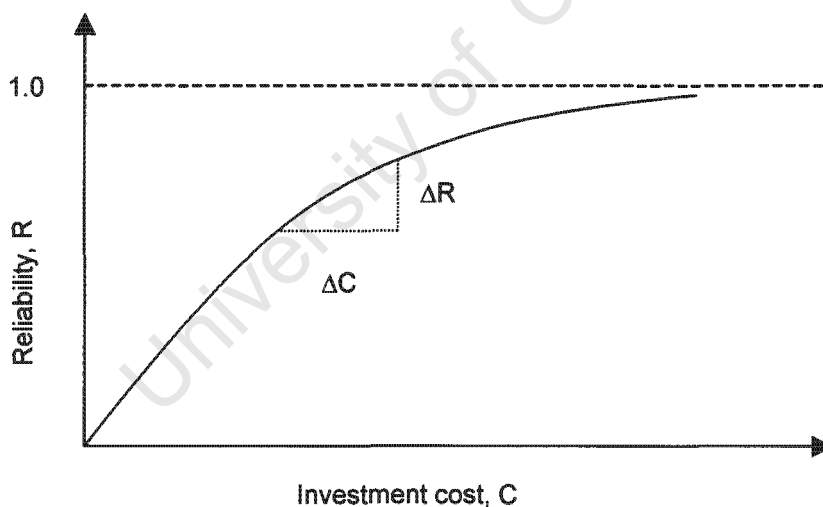


FIGURE 3.1 INCREMENTAL COST OF RELIABILITY (AFTER BILLINTON AND ALLAN, 1996:13)

The model that has been described does not show the financial value of reliability to the customer. This has led to the development of the reliability cost/worth model.

The total costs of a system must also include its value to the customers in the form of fewer power interruption costs (Billinton and Allan, 1996:13). When making a reliability investment, the worth to the customer due to this increase in reliability should be greater than the cost to the utility. The lowest total cost, which is the optimum reliability level, occurs when the incremental benefit to the customer and the incremental cost to the utility are equal.

Figure 3.2 illustrates the reliability cost/worth concept.

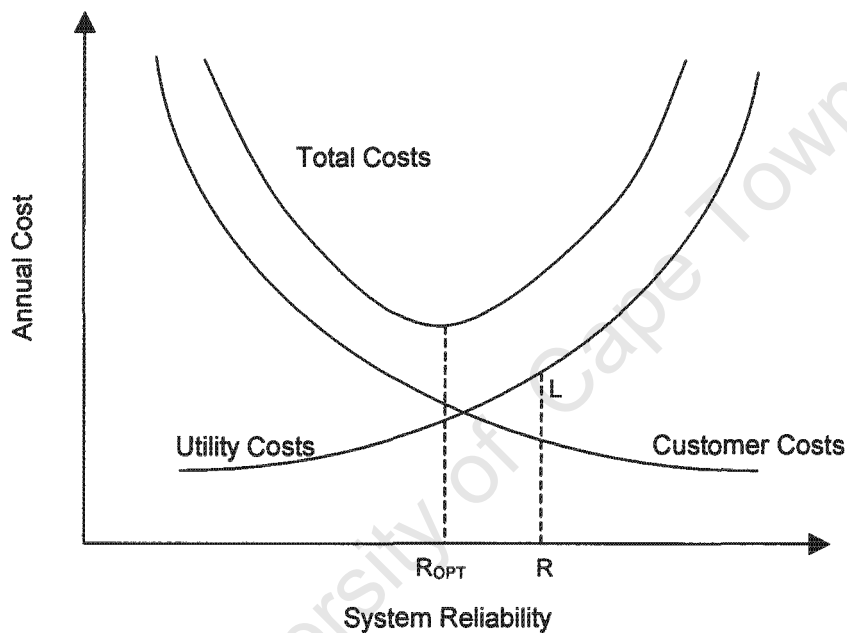


FIGURE 3.2 OPTIMAL RELIABILITY LEVEL (AFTER BILLINTON AND LAKHANPAL, 1996)

The utility cost curve shows that an increase in the system reliability level results in an increase in the system cost. The customer cost curve decreases as the system reliability level is increased. The total cost curve is the sum of these two curves and has a minimum value that represents the optimal reliability level.

Looking at Figure 3.2, it can be seen that the more detailed cost/worth approach is superior to a simpler approach which would aim only at achieving a certain

level of reliability R at the lowest possible cost L (Billinton and Lakhanpal, 1996). If the customer costs are taken into account it then results in a reliability level R_{OPT} as shown in Figure 3.2. For the case shown in Figure 3.2, the utility costs are lower using the reliability cost/worth approach because investment costs are less at this optimal reliability level than at the original reliability level R .

3.3 RELIABILITY INDICES

Reliability indices may be classified as follows (EPRI, 1981):

1. Indices of component reliability. These provide a measure of the continuity of service provided by components.
2. Indices of load-point reliability. These provide a measure of the continuity of service to individual load points.
3. System indices. These provide a measure of the continuity of service to the entire distribution system.

This thesis is concerned with system indices because the optimal system configuration, in terms of the number and location of sectionalisers, is being determined. The remainder of this chapter describes the most common performance indices that are used to evaluate the reliability of distribution systems.

3.3.1 CUSTOMER-ORIENTATED INDICES

The indices presented in this section were obtained from Billinton and Allan (1996:224).

1. System Average Interruption Frequency Index

The System Average Interruption Frequency Index (SAIFI) gives the average number of interruptions per *customer served* or supplied per year. It is the total number of customer interruptions divided by the total number of customers served on the distribution system. A customer interruption is defined as one interruption to one customer.

$$SAIFI = \frac{\text{Total Number of Customer Interruptions}}{\text{Total Number of Customers Served}} \quad (3.1)$$

2. Customer Average Interruption Frequency Index

The Customer Average Interruption Frequency Index (CAIFI) is the average number of interruptions per *customer interrupted* per year. It is the total number of customer interruptions divided by the total number of customers that are affected by at least one interruption annually.

$$CAIFI = \frac{\text{Total Number of Customer Interruptions}}{\text{Total Number of Customers Affected}} \quad (3.2)$$

3. System Average Interruption Duration Index

This System Average Interruption Duration Index (SAIDI) is the average interruption duration for *customers served* during a year. It is determined by dividing the sum of all customer interruption durations in a year by the number of customers served.

$$SAIDI = \frac{\text{Sum of Customer Interruption Durations}}{\text{Total Number of Customers}} \quad (3.3)$$

4. Customer Average Interruption Duration Index

The Customer Average Interruption Duration Index (CAIDI) is the average interruption duration for *customers* that are *interrupted* during a year. It is determined by dividing the sum of all customer interruption durations in a year by the total number of customer interruptions.

$$CAIDI = \frac{\text{Sum of Customer Interruption Durations}}{\text{Total Number of Customer Interruptions}} \quad (3.4)$$

3.3.2 LOAD- AND ENERGY-ORIENTED INDICES

The calculated values of the customer-oriented indices will not change if the average load at a load point is 5kW or 5MW. As a result, load- and energy-oriented indices were developed to consider the effect of small and large load interruptions. The indices that are presented in this section were obtained from Billinton and Allan (1986:225).

Average Load

For the evaluation of these indices, it is important to know the average load at each load-point busbar. The average load L_a over a given period is the maximum demand L_p , multiplied by the load factor f .

$$L_a = L_p f \quad (3.5)$$

Energy Not Supplied (ENS)

This is the total energy not supplied by the system due to outages. The units for ENS are kWh/yr if the loads are in kW but it is preferable to use MWh/yr for larger loads.

$$ENS = \sum L_{a(i)} U_i \quad (3.6)$$

where $L_{a(i)}$ is the average load connected to load point i .
 U_i is the average outage time at load point i .

It should be noted that an assumption is made that it is sufficiently accurate to use average loads for calculating ENS. This might not reflect the severity of a system outage if the system tends to fail mainly due to heavy load conditions. It is likely that the average outage time could be higher under heavy load conditions (for example, because of bad weather conditions).

Average Energy Not Supplied (AENS)

This is the total energy not supplied (ENS) divided by the total number of customers served. The units for AENS are kWh/y/customer.

$$AENS = \frac{\sum L_{a(i)} U_i}{N} \quad (3.7)$$

where N is the total number of customers.

3.3.3 CALCULATION OF RELIABILITY INDICES

This section illustrates how the reliability indices are calculated for a distribution system with six load-point busbars. Table 3.2 contains the customer data for the distribution system and Table 3.3 gives the customer outage data over a specified year.

LOAD POINT	NUMBER OF CUSTOMERS	AVERAGE LOAD CONNECTED
	N	L _a [kW]
1	1000	4000
2	600	2000
3	700	3500
4	1000	5000
5	800	3200
6	900	3300
TOTAL	5000	21000

TABLE 3.2 CUSTOMER DATA FOR A DISTRIBUTION SYSTEM (AFTER BILLINTON AND ALLAN, 1996:227)

OUTAGE CASE	LOAD POINT AFFECTED	NUMBER OF CUSTOMERS DISCONNECTED	CONNECTED LOAD INTERRUPTIONS	INTERRUPTION DURATION	CUSTOMER HOURS DURATIONS	ENERGY NOT SUPPLIED
		N	L _c [kW]	d[hrs]	Nd[cust hrs]	L _c d[kWh]
1	1	1000	4000	2.0	2000	8000
	5	400	1600	0.5	200	800
2	3	350	2500	1.0	350	2500
3	5	800	3200	1.5	1200	4800

TABLE 3.3 CUSTOMER OUTAGE DATA (AFTER BILLINTON AND ALLAN, 1996:227)

- The total number of customer interruptions = 1000+400+350+800 = 2550.
- The total number of customers affected = 1000+400+350+800-400 = 2150.
The figure of 400 is subtracted at the end because the 800 customer interruptions in Case 3 include the same 400 customers affected in Case 1.
- The sum of customer hour durations = 2000+200+350+1200 = 3750.

$$SAIFI = \frac{2550}{5000} = 0.510 \text{ interruptions/customer}$$

$$CAIFI = \frac{2550}{2150} = 1.19 \text{ interruptions/customer affected}$$

$$SAIDI = \frac{3750}{5000} = 0.75 \text{ hrs/system's customer}$$

$$CAIDI = \frac{3750}{2550} = 1.47 \text{ hrs/customer interruption}$$

$$ENS = 8000 + 800 + 2500 + 4800 = 16100 \text{ kWh}$$

$$AENS = \frac{16100}{5000} = 3.22 \text{ kWh/customer}$$

3.4 CONCLUSION

This chapter gives an introduction to distribution system reliability. It has described the reliability cost/worth concept used for determining the appropriate investment costs to improve system reliability. A range of performance indices used to evaluate the reliability of distribution systems have been presented.

Chapter 4 PROTECTION PHILOSOPHIES DEVELOPED FOR SYSTEM IMPROVEMENT

The purpose of this chapter is to show the benefit that a utility can derive from the placement of sectionalising devices on a system. This chapter begins with a description of a reliability index known as the Load Exposure Index that was developed to help planners to determine the location of protection devices. It then describes protection philosophies that have been developed and implemented by an Australian utility for the placement of reclosers.

4.1 LOAD EXPOSURE INDEX DEVELOPED FOR THE PLACEMENT OF PROTECTIVE DEVICES

This section describes a reliability index that was developed by Luth (1991). It is known as the Load Exposure Index or LEI. The LEI helps network planners to determine the optimum number of sectionalising devices such as reclosers, fuses and sectionalisers. The correct placement of these devices improves the reliability of the feeder because the interruptions in power supply experienced by the customers are reduced.

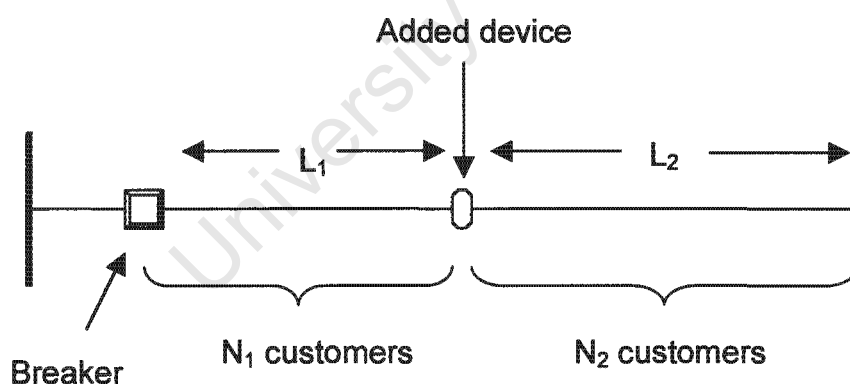
According to Luth (1991) there are two main factors that determine the average duration of the interruptions to a customer on a feeder. These are the number of customers on the feeder and the exposed line length. Luth does not take into consideration other factors among which could be the response time of the maintenance crew. This could be attributed to the fact that the response time does not vary as much as the other two parameters that have just been mentioned above. For example, some utilities assume a response time of approximately two hours for each interruption.

The basis of Luth's method in using the LEI is that the added sectionalising device protects customers upstream from it from faults that occur downstream. Thus, the LEI is calculated by "multiplying the exposed line length by the number of customers *downstream of the device*" (Luth, 1991). The exposed line length is the distance from that device to the *next downstream* device. The LEI for a feeder is equal to the sum of the LEI's for each of the sectionalising devices on the feeder.

Luth (1991) developed the following rules to maximise the benefit for the placement of new sectionalising devices on a feeder.

Rule 1. *When a protective device is added to a feeder, the improvement in Luth's reliability index is directly proportional to the length of circuit newly protected multiplied by the number of customers between the new device and the next upstream protective device.*

This rule is illustrated in Figure 4.1



Load-exposure index before addition of device = $(L_1 + L_2) * (N_1 + N_2)$

Load-exposure index with device = $L_1 * (N_1 + N_2) + L_2 * N_2$

Reduction in feeder load exposure index = $L_2 * N_1$

FIGURE 4.1 DETERMINING LOAD-EXPOSURE INDEX FOR A DEVICE PLACED ON A FEEDER (LUTH, 1991)

According to Luth's paper, there is a linear relationship between SAIDI and the LEI, but the detail provided by Luth is not sufficient to explain this linear correlation. Based on this relationship, I would think that it would be fair to assume that a decrease in the LEI would correlate with a decrease in the system interruption duration as measured by SAIDI. Hence, the reduction in LEI for the feeder reflects an increase in feeder reliability. In this case, the increase in the LEI is represented by $L_2 \cdot N_1$ in Figure 4.1.

A problem with using Rule 1 is that customers with high power consumptions (hospitals) will be treated on an equal basis with customers of smaller power consumptions (domestic use), that is, as a single customer. Under such circumstances, Luth says that it would seem more appropriate to substitute the number of customers in Rule 1 with the total load. The rule is then modified to (Luth, 1991):

Rule 1A. *When a protective device is added to a feeder, the improvement in Luth's reliability index is directly proportional to the length of circuit newly protected multiplied by the load connected between the new device and the next upstream device.*

The second rule used for the placement of sectionalising devices is as follows:

Rule 2. *The optimum location for a protective device is the geographic midpoint (assuming that the load and outages are evenly distributed along the circuit). When the device is installed at this location, the value of SAIDI is reduced by 25%.*

I decided to test this rule by performing a calculation to determine the reduction in SAIDI as a function of x , where x is the proportion of the line length from the sending end. The following assumption were made:

- Customers are evenly distributed along the feeder and use similar loads.
- Outage duration is the same for each interruption.
- If the device is placed at any point along the feeder, then the faults will be evenly distributed according to the distance ratio before and after the sectionalising device. For example, if there are four faults and the device is placed at $\frac{1}{4}$ of the length down the line, then one fault will occur before the device and three will occur downstream of it.

Section 3.3.1 shows that SAIDI is represented as follows:

$$SAIDI = \frac{\text{Sum of Customer Interruption Durations}}{\text{Total Number of Customers}} \text{ hrs / customer} \quad (4.1)$$

I developed the following equation based on equation 4.1 and the three assumptions that have been made.

$$SAIDI = \frac{[xT_{NI} \times N_c \times d] + [(1-x)T_{NI} \times (1-x)N_c \times d]}{N_c} \quad (4.2)$$

where x is the proportion of the line length from the sending end.

T_{NI} is the total number of faults that occur.

N_c is the number of customers disconnected due to an interruption.

d is the duration of the interruption.

The value of SAIDI in equation 4.2 was calculated for different values of x as shown in Table 4.1. N_c was taken to represent 12 customers, d was 2 hours and T_{NI} represented 6 faults.

X	0	1/3	1/2	2/3
SAIDI	12.00	9.33	9.00	9.33

TABLE 4.1 VALUES OF SAIDI OBTAINED WHICH CORRESPOND TO DIFFERENT VALUES OF X

These results show that the greatest reduction in SAIDI takes place at the mid-point of the feeder because the value for SAIDI is lowest at this point. It also shows that when the device is placed at the midpoint, the value of SAIDI is reduced by 25%.

In cases where the load is not evenly distributed but is concentrated in a section along the feeder, the following Rule 3 should be used.

Rule 3. *When there is a heavy load concentration at one point in the circuit, reliability improvement is maximised when a protective device is added immediately downstream of the load concentration.*

Suppose that we have a feeder with loads as shown in Figure 4.2. The numbers 1-5 are the possible locations where a sectionalising device could be placed.

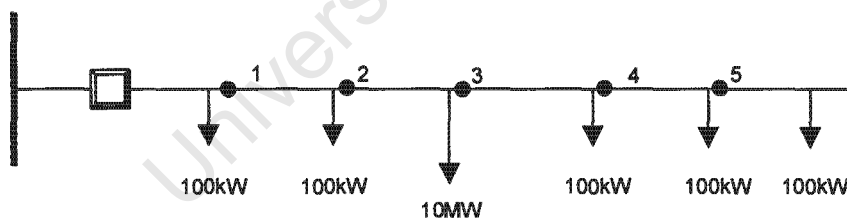


FIGURE 4.2 LARGE CUSTOMER PROTECTED FROM SMALLER CUSTOMERS DOWNSTREAM

According to Lakervi and Holmes (1989:188), Equation 4.3 represents the saving in the total cost of a network, in terms of the reduction of the cost of unsupplied energy, due to the installation of a sectionaliser.

$$C_i = KP_i\lambda_i r d - M \quad (4.3)$$

where i = section number.

K = annuity factor – factor by which first-year outage costs are multiplied to get discounted costs over the whole review period.

P_i = demand upstream of i .

λ = failure rate per line length.

l_i = line length downstream of i .

r = average outage time for an interruption.

d = cost per unit of energy not supplied.

M = cost of a sectionaliser.

C_i = saving in the total cost if a sectionaliser is placed at location i .

In Equation 4.3, $KP_i\lambda_i r d$ is the cost of unsupplied energy (ENS is the failure rate x average repair time for an interruption x average power disconnected) that results when a sectionaliser, that costs M , is installed. Therefore, the savings in the total cost can be calculated if the cost of the sectionaliser is subtracted from the cost of unsupplied energy resulting from placing the device.

The saving in the total cost of the network that results from placing a sectionalising device in each of the locations 1-5 in Figure 4.2 was calculated using the following values for the parameters:

$K = 13.8$ (K is approximately 13.8 at a discount rate of 6% over 30 years).

$\lambda = 0.125$ faults/km/annum.

l_i = line length between each location is 5 km.

$r = 2$ hours.

$d = 20$ c/kWh.

$M = R50\ 000$.

The results are tabulated in Table 4.2.

LOCATION	1	2	3	4	5
SAVINGS IN COST OF ENS	-R48 275	-R47 240	R55 570	R21 070	-R14 120

TABLE 4.2 SAVINGS IN TOTAL COST DUE TO THE PLACEMENT OF A DEVICE AT DIFFERENT LOCATIONS

From these results, it can be seen that the greatest saving in the total cost occurs at the point along the feeder if the sectionalising device is placed immediately downstream of the greatest load (as stated in Rule 3). Any fault that occurs downstream of the device will cause the device to operate and prevents a loss in power supply for the large load.

Deciding on the optimal placement of sectionalising devices is not a simple task. The rules show that placing the devices in unsuitable locations can greatly reduce their effectiveness.

4.2 ECONOMIC OPTIMISATION OF AN AUSTRALIAN RURAL NETWORK

Great Southern Energy (GSE) is a gas and electricity utility located in New South Wales. It provides electricity to about 225 000 customers in a service area of 176 000 square kilometres (Leith and Sullivan, 1999). Feeders at 11, 22 and 33 kV are used for rural electricity distribution. Travelling distances between switching points can be as much as 80 km (Leith and Sullivan, 1999). Switching points are places on the network where sectionalising devices such as reclosers and sectionalisers can be placed.

RULES FOR THE APPLICATION OF ELECTRONIC RECLOSERS

Leith and Sullivan (1999) from GSE have developed the following two rules for the installation of reclosers.

1. The One-Second Rule

Reclosers are located so that so that they take no longer than one second to clear faults in their protection zone so as to minimize fire risk from hot particles (Leith and Sullivan, 1999).

This helps to reduce the risk of bushfires that are very prevalent. This 'time' rule is not always fixed at one second and can be altered to suit the environmental conditions, for example it can be lengthened in damp areas because the risk of bushfire is much less than in the hot, dry areas (Leith and Sullivan, 1999). It would be interesting to know what range of times are used in South Africa, but I have not been able to establish this. A question that may arise with regard to the one-second rule could be - what is the effect of the application of this rule in terms of the number of reclosers installed? The one-second rule is not related to the number of reclosers installed. The purpose of the one-second rule is to ensure that a recloser isolates a faulty section quickly to extinguish any arc before the arc has a chance to ignite a fire.

This is economic optimisation of a very different kind from that prevalent in South Africa where the rules that are used for the placement of reclosers are based on their cost and on the resulting savings of unsupplied energy rather than on the reduction of the expenses incurred due to bushfires. These costs could include damage to the power system equipment and even legal claims should a death occur as a result of the fire.

2. The Twice-Times Rule

The pickup current of upstream reclosers should be at least two times the pickup current of downstream reclosers (Leith and Sullivan, 1999).

Pickup current is the minimum fault current that causes the recloser to operate. This rule should ensure that there are no unnecessary operations of the backup recloser in connection with faults that can be cleared by the downstream recloser.

Figure 4.3 shows an example where the two rules mentioned above are applied to eight reclosers placed in series on a feeder. This is a hypothetical case - in practice a pickup current of 1280A on a 22 kV feeder would imply that the total fault impedance is about 10 ohms, which is very low. Reach is the maximum current a recloser will withstand for correct operation without being damaged.

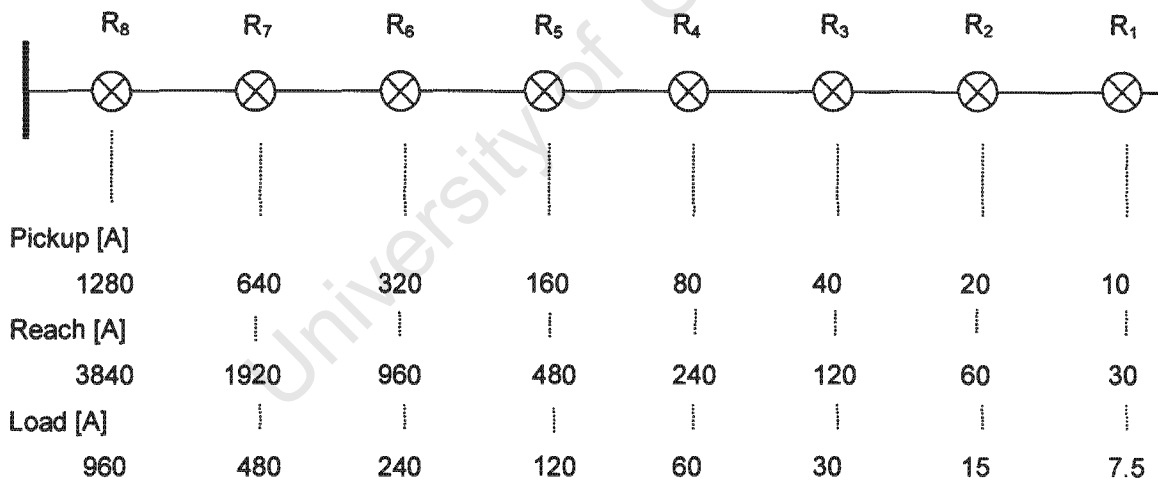


FIGURE 4.3 PICKUP AND LOAD CURRENT FOR EIGHT RECLOSERS PLACED IN SERIES (AFTER LEITH AND SULLIVAN, 1999)

The three-phase fault current at the end of the feeder in Figure 4.3 is 30A. The most distant recloser R1 has to have a pickup of 10A in order to clear a 30A fault

in one second. This is illustrated in Figure 4.4. From this graph, it can be seen that the recloser opens once the 30A level is reached.

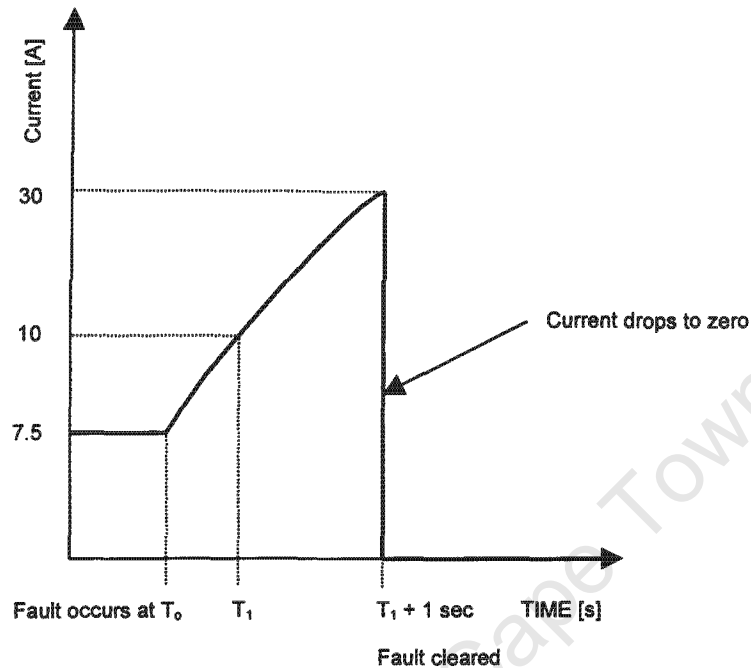


FIGURE 4.4 POSSIBLE TCC CURVE FOR RECLOSER R1

7.5A is the maximum load to avoid trip due to transformer magnetising inrush current. The occurrence of 'magnetising inrush current' is a transient condition that occurs when a transformer is energised. It is not a fault condition, therefore, the protection system should not operate under this transient condition.

From the twice-times rule, the next upstream recloser R2 will have to have a 20A pickup. According to the one-second rule, this recloser must clear a 60A fault current in one second. This procedure is repeated for each of the reclosers up to R8.

Leith and Sullivan performed a study on one of GSE's feeders using four reclosers in series as shown in Figure 4.5.

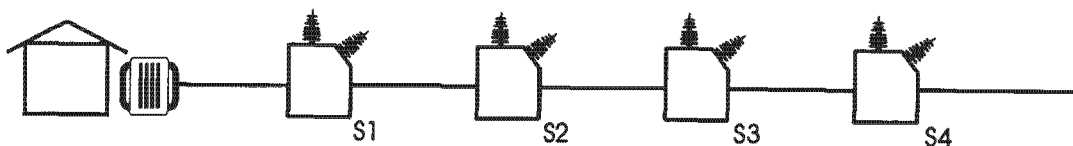


FIGURE 4.5 DIAGRAM OF FOUR RECLOSERS PLACED IN SERIES (LEITH AND SULLIVAN, 1999)

The capital invested in the installation of these reclosers amounted to AUS\$100 000. This study was carried out to show the economic advantages of using reclosers on a feeder. Table 4.3 shows a summary of the results obtained for an economic evaluation of these four reclosers. When this study was performed, it was assumed that 80% of all faults would be temporary and could be cleared by the reclosers.

Recloser	S1	S2	S3	S4	TOTAL	UNITS
Customers connected	200	100	50	50	400	
Length	4	8	16	32	60	miles
Failure rate	1	1	2	2		Faults/mi/yr
Repair rate	2.00	3.00	4.00	5.00		hrs/fault
<u>ECONOMIC EVALUATION</u>						
Lost revenue	22	43	108	324	497	A\$pa
Repair cost	160	480	2 560	6 400	9 600	A\$pa
Social cost	864	1 728	4 320	12 960	19 872	A\$pa
Total annual cost	1 046	2 251	6 988	19 684	29 969	A\$pa
Capitalised cost	9 460	20 368	63 224	178 090	271 141	A\$

TABLE 4.3 ECONOMIC EVALUATION OF RECLOSER PLACEMENT (LEITH AND SULLIVAN, 1999)

Table 4.3 shows the number of customers on the feeder. It gives the feeder section lengths protected by each recloser as well as the failure rate and repair rate of each of these sections. The lost revenue value is the lost revenue due to

outages. The social cost figure refers to costs associated with “death and injury claims from electrical causes, fines under occupational safety and health legislation and damage claims for energy not supplied” (Leith and Sullivan, 1999). The repair cost figure is the cost to send a maintenance crew to repair a fault. The total annual cost for each recloser is obtained by totalling the values obtained for the lost revenue, the repair cost and the social cost figure.

The lost revenue cost figures seem to be fairly low compared to the repair cost and social cost figures and it would appear that the cost of unsupplied energy seems insignificant. Leith and Sullivan do not explain how the lost revenue figure is calculated, but it is interesting that one of the components that make up the social costs figure are damage claims for energy not supplied. Therefore, it still seems valid to perform an optimisation to reduce the cost of unsupplied energy.

From the table it can be seen that the cost figures for the economic evaluation increase at a rapid rate from S_1 to S_4 . This can be attributed to the increased feeder section lengths protected by each recloser. The repair rate also increases because of the increased travelling distances to repair faults that occur further from the supply end of the feeder. Flooding can also add to the travel time since the floods make certain areas inaccessible to maintenance crews.

GSE have found that the use of reclosers can produce substantial savings in costs to utilities that far outweigh the high initial costs of installing them reclosers. It was found that although an initial amount of AUS\$100 000 was invested for installing four reclosers, a total of AUS\$120 000 was saved annually. From the paper, it is not clear how this value of AUS\$120 000 was obtained. According to the figures in Table 4.3 it should be $\text{AUS\$271 000} - \text{AUS\$100 000} = \text{AUS\$171 000}$. This study shows that the use of sectionalising devices results in a considerable saving to utilities in the cost of restoring power to its customers.

4.3 CONCLUSION

Although the investment for installing sectionalising devices may be high initially, the use of these devices can produce substantial savings in costs that makes the investment beneficial, to both the utilities and the customers, in the long run.

University of Cape Town

Chapter 5 INTRODUCTION TO EVOLUTIONARY ALGORITHMS

This chapter begins with an introduction to evolutionary algorithms. It then gives a description of two types of evolutionary algorithms. These are the Genetic Algorithm and the Population Based Incremental Learning (PBIL) algorithm.

5.1 WHAT ARE EVOLUTIONARY ALGORITHMS?

Evolutionary algorithms (commonly known as EAs) are computer-based problem-solving systems based on the principles underlying the theory of evolution. They provide a powerful and practical alternative to traditional mathematical and engineering problem solving techniques that have been devised (EvoNet, 1999).

They have been applied to many problems. For example, they have been used to improve (EvoNet, 1999):

- Supply chain management.
- Distribution networks - telephone and electricity networks.
- Fraud detection.
- Layout and location of retail outlets.

Evolutionary algorithms all share a common conceptual base in that they simulate the evolution of individual structures using three processes called Selection, Mutation and Recombination (EvoNet, 1999). These processes favour reproduction between individuals of higher 'quality' rather than between those of a lower 'quality' (Miranda, Srinivasan and Proenca, 1996). A detailed description of these processes is given later.

Research work based on EAs have been widely documented. One of the most popular EAs that have been developed is the so-called 'Genetic Algorithm'. A description of the genetic algorithm is given in section 5.4.

5.2 BIOLOGICAL BACKGROUND

The theory of evolution was formulated by Charles Darwin and has been recognised as the most important theory in biology (Ridley, 1996:1). Evolution means 'change' (Ridley, 1996:1). 'Change' is an alteration in the form and behaviour of organisms between generations. Evolution is the process used by all living organisms in their development and quest for improvement to adapt to a consistently changing environment. Evolution takes place through a process called 'natural selection' to ensure the preservation of qualities in a species that are best suited for its survival (Ridley, 1996:1). It also ensures that these qualities are passed on to succeeding generations. Natural selection in its most general form consists of the following three processes (Ridley, 1996:73):

- *Reproduction* – Individuals must reproduce in order to form a new generation.
- *Mutation* – This introduces random changes to genetic information resulting in individuals with properties that are new to the existing population.
- *Fitness* – In the theory of evolution, fitness means that some individuals in the population possess some characteristics that are an advantage and make them more likely than others to reproduce. Therefore, we can see that fitness is an advantage to a species even if the original mutation would normally have been a disadvantage. For example, a water bird that has a wry beak would be more able to scoop up its prey from shallow waters than a bird with a straight beak. The advantageous mutation that resulted in the development of the crooked beak was then passed down to the progeny (Ridley, 1996:108).

When any of these processes are not met then natural selection does not occur. When all three processes are met, individuals with characteristics conferring higher fitness values will produce more offspring. The frequency of occurrence of that type of individual will increase in the population. These processes ensure advancement of a species to a more developed state of existence (Ridley, 1996:74).

5.3 TERMINOLOGY USED IN GENETIC ALGORITHMS

Table 5.1 gives the 'natural' (biological) names of terms used in genetics and their corresponding 'artificial' names used in evolutionary algorithms.

NATURAL SYSTEM	ARTIFICIAL SYSTEM
Chromosome	String
Gene	Feature or Detector
Allele	Feature value
Locus	String position
Phenotype	Parameter set or a decoded structure
Epistasis	Nonlinearity

TABLE 5.1 NATURAL AND ARTIFICIAL TERMINOLOGIES USED. (GOLDBERG, 1989: 22)

Chromosomes in biological systems are analogous to strings in artificial systems. Chromosomes contain the hereditary information of an individual that is passed on from generation to generation. Strings contain data that represent potential solutions to the objective function being optimised. For example, the variable $x=7$ could be a potential solution for some function $f(x)$. A binary string representation of the variable x could be [0111].

Variations between individuals of the same species are caused by the genetic construction of the individual. This genetic construction is known as the *genotype*. In artificial systems, the total 'package' of strings is called a *structure* (Goldberg, 1989:22). For the scope of this thesis, a structure consists of a single string.

In natural systems the phenotype refers to the type of creature that is formed due to the influence of the environment on its genotype. In artificial systems, an evaluation routine *decodes* the structures to form a particular *parameter set* (alternative solution). (Goldberg, 1989:22)

Chromosomes are made up of *genes*. Each gene has a particular *locus* or position in the chromosome. A gene contains a characteristic known as an allele that determines the expression of the gene. For example, there is a specific gene that is responsible for the colour of a person's hair. The locus of the gene could be position 3 in the chromosome and its allele value is the actual colour (blonde for example). Likewise, in artificial systems, each bit or group of bits that belongs to a string represents a value for some variable of the problem (gene). The value of each bit ('0' or '1' – allele value) is known as the *feature* or *detector* value of the string. Figure 5.1 gives a physical representation of a few of the terms mentioned above. (Goldberg, 1989:22)

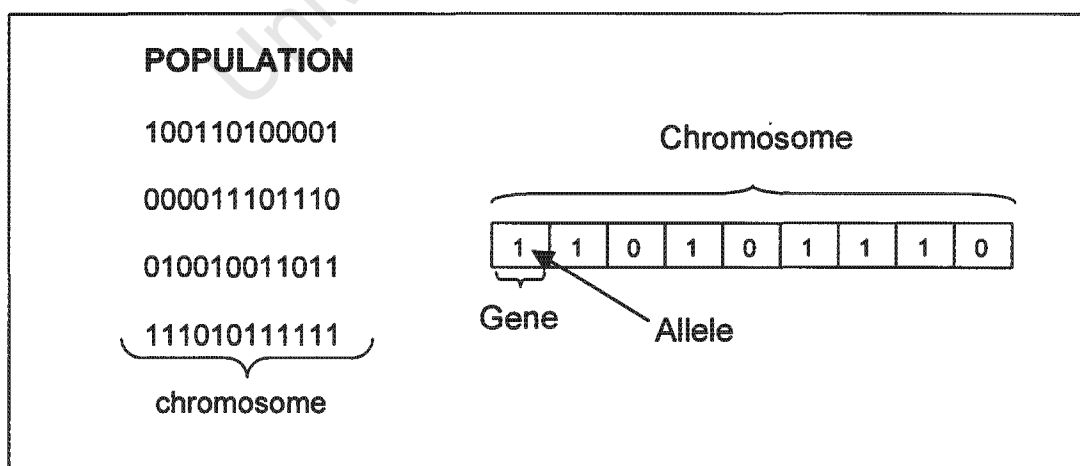


FIGURE 5.1 PHYSICAL ILLUSTRATIONS OF TERMS USED IN A GA (KHYLATASH, 1995)

5.4 INTRODUCTION TO GENETIC ALGORITHMS

John Holland and his students at the University of Michigan developed the genetic algorithm in the 1960's. Genetic algorithms were not designed to solve any specific problems such as sorting, tree traversal or function optimisation. Instead, a genetic algorithm was designed to be a high level simulation of an adaptive system that is biologically motivated. It is a representation of evolution. "The central theme of research on genetic algorithms has been robustness, the balance between efficiency and efficacy necessary for survival in many different environments" (Goldberg, 1989:1). This statement makes sense if efficiency refers to computation time. Some programs require more iterations to produce better results, however, more iterations mean that the computation time will increase. If efficiency does not relate to computation time then it would seem that the two qualities, efficiency and efficacy, should go hand in hand.

Designers of artificial systems admire biological systems for their ability for self-repair, self-guidance and reproduction that "barely exist in the most sophisticated artificial systems" (Goldberg, 1989:2).

5.4.1 DIFFERENCES BETWEEN GA AND TRADITIONAL METHODS

A genetic algorithm is different from the usual optimisation and search methods such as linear programming for at least three of the following reasons (Goldberg, 1989:7):

- 1) "Genetic Algorithms work with a coding of the parameter set, not the parameters themselves".
- 2) "Genetic Algorithms search from a population of points, not a single point".
- 3) "Genetic Algorithms use payoff or objective function information, not derivatives or other auxiliary knowledge".

4) "Genetic Algorithms use probabilistic transition rules, not deterministic rules".

The remainder of this section explains the last four statements.

Genetic algorithms require very little information about a problem. Suppose that we have a black box that can take a set of variables (solution string) and return a function value as shown in Figure 5.2.

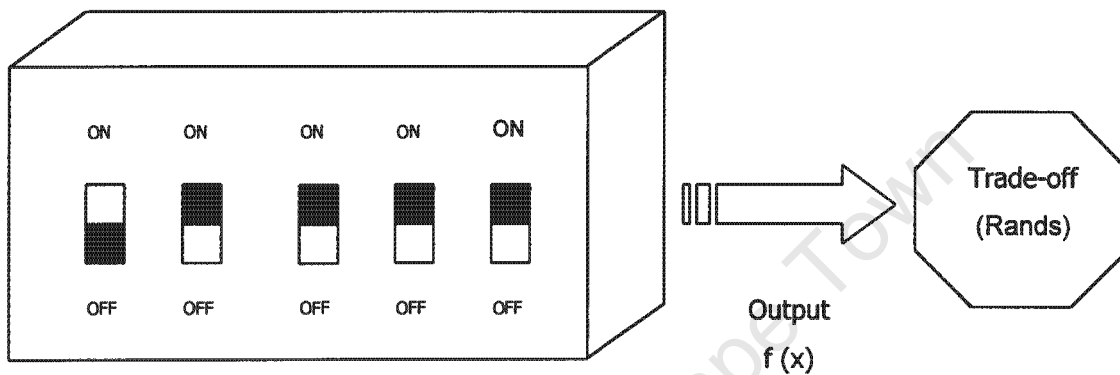


FIGURE 5.2 BLACK BOX OPTIMISATION PROBLEM (GOLDBERG, 1989:8)

The black box device consists of five switches. The object is to find the best arrangement of the switches in their on or off states to obtain the maximum function value for $f(x)$. Best solutions are allowed to evolve subject to some fitness criteria, while internally the mechanics are left largely as a black box.

Suppose the function $f(x) = x^2 + 6$ needs to be maximised within the interval $(0 \leq x \leq 50)$. Traditional optimisation methods would manipulate x in the same way that one would turn a control knob to tune into a radio station. The optimisation algorithm would 'fiddle' with the parameter x ('turn the knob' using techniques specific to that particular optimisation algorithm) and search the function space until the maximum value of the function has been obtained. With a genetic algorithm, the chromosome (string) should in some way store information about the solution (encode the string) that it represents. The best and most

commonly used method of encoding is a binary string. Each bit in the string represents some characteristic of the solution. For example, the string can be used to represent the coefficients in an equation. In Figure 5.2, parameter x can be coded using a string of binary numbers to represent the arrangement of switches, for example 0 1 1 1 1 means that the first switch is off and the remaining are on. Thus, genetic algorithms work with a coding of the parameter set using a finite length string.

Most optimising algorithms use a single point in a function space as the basis for a search (Baluja, 1994). These algorithms can converge to the wrong peak in search spaces that have a multitude of peaks (Goldberg, 1989:9). An interesting feature of the genetic algorithm is its ability to 'climb' many peaks in parallel because it can conduct a search from a population of points. This point can be explained by using the black box optimisation problem as an example. Most algorithms would use a single set of switch settings. They would then perform some operation that results in a new switch setting. In contrast, a genetic algorithm would generate a population of strings (multiple number of switch settings) and then produce a successive population of strings. The method used for generating a population of strings is described in section 5.4.2. Although parallel searching does not guarantee that the optimum solution is found, it does reduce the probability of premature convergence.

For a search algorithm to solve a problem, a lot of auxiliary information is needed, for example gradient techniques need derivatives (Goldberg, 1989:9). However, genetic algorithms do not require this knowledge. All that is needed for a genetic algorithm to obtain an improved solution are the trade-off values or objective function values associated with individual strings.

Genetic Algorithms use probabilistic transition rules instead of deterministic ones to guide their search (Goldberg, 1989:10). This may seem strange to people who use deterministic methods because they consider probabilistic methods as being

random searches (Goldberg, 1989:10). Since the initial starting points of a genetic algorithm (searching in parallel) are random, it searches regions in the search space that would otherwise be overlooked by other algorithms.

5.4.2 OPERATION OF A GENETIC ALGORITHM

This section demonstrates how a genetic algorithm works. The black box optimisation problem is used as an example to explain some of the concepts that are presented.

Figure 5.3 is a block diagram representation of the operation of a simple genetic algorithm. A detailed explanation of it is given after the diagram.

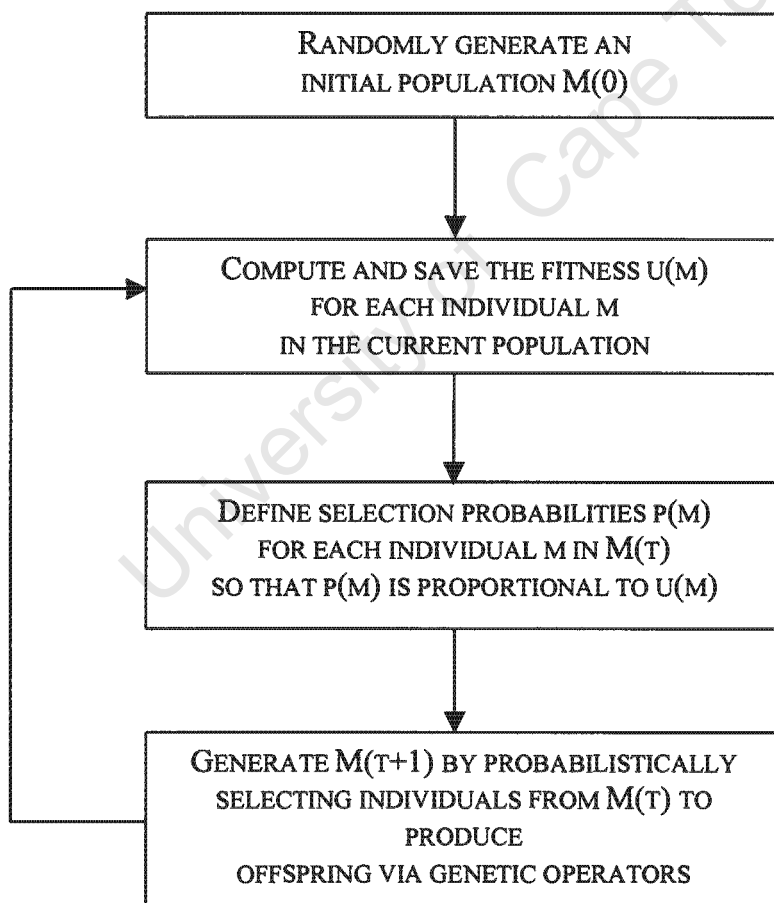


FIGURE 5.3 FLOWCHART REPRESENTATION OF A GENETIC ALGORITHM (WHITLEY, 1993:7)

A genetic algorithm commences by generating a population of strings that are solutions to the function being optimised. Initially, this group of strings is generated randomly. An example of the random generation of a population of four potential solutions for the black box problem could be as follows:

```
0 1 0 1 1
1 1 0 0 1
0 0 1 1 1
1 0 0 0 1
```

Special genetic operators continuously improve the solution strings obtained over a number of generations. The special genetic operators are (Goldberg, 1989:10):

- Reproduction
- Crossover
- Mutation

REPRODUCTION

Reproduction (also known as selection) is the process whereby a new population (generation) is created by selecting strings from the previous population according to their *fitness*. The fitness can be considered to be a measure of the effectiveness of the string as a potential solution for the objective function being optimised (Baluja, 1994). Strings are selected for recombination according to their fitness values. Strings that have a high fitness value will be selected with a greater probability for producing 'offspring' or successors in the next generation. This is an artificial translation of 'natural selection'.

Table 5.2 shows the fitness values for a population of strings, that have been randomly generated, for the black box optimisation problem.

	STRING	DECIMAL VALUE	FITNESS	% OF FITNESS TOTAL
1	0 1 0 1 1	11	127	11.4
2	1 1 0 0 1	25	631	57.0
3	0 0 1 1 1	7	55	5.0
4	1 0 0 0 1	17	295	26.6
Total			1108	100

TABLE 5.2 SAMPLE STRINGS AND THEIR FITNESS VALUES. (AFTER GOLDBERG, 1989:11)

A description will be given now to show how the values in the fourth and fifth columns in Table 5.2 were obtained for the four sample strings that were randomly generated. These string values are decoded to their decimal equivalent shown in the third column. Taking string 3 as an example, the decimal equivalent of 00111_2 is 7. The decimal value (7) is then substituted into the objective function ($f(x)=x^2+6$) to obtain the fitness value 55. The fitness value of each string is obtained in the same way. These fitness values are summed up to give the total fitness value of 1108. Each fitness value is then divided by the total fitness value and multiplied by 100 to give the percentage total fitness value shown in column five (For string 3, this would be $55/1108 \times 100 = 5\%$).

A 'reproduction operator' is used to select strings from the population for mating. The so-called 'roulette wheel method' is commonly used to create the reproduction operator in algorithmic form. This wheel is biased so that each string is allocated a space on the roulette wheel that is proportional to the percentage of the total fitness that it possesses (Goldberg, 1989:12). The purpose of this is to ensure that the genetic algorithm gives the strings that produce high fitness values a better chance of being selected for reproducing.

Figure 5.4 shows the biased roulette wheel that is obtained from fitness values of the strings that were generated randomly.

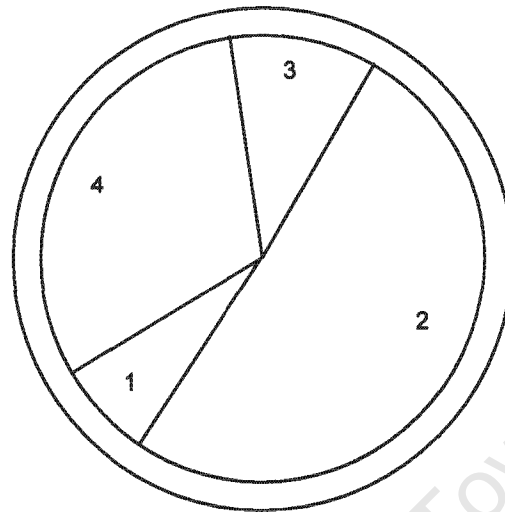


FIGURE 5.4 BIASED ROULETTE WHEEL (AFTER GOLDBERG, 1989:10)

This wheel is spun four times so that four 'parent' strings are selected. These four 'parent' strings are selected for mating to produce four 'offspring' strings to maintain the population size for the next generation.

String 2 has a fitness value of 631 that represents approximately 57 percent of the total fitness. This is represented by portion 2 on the roulette wheel (57% of the roulette wheel). Therefore, string 2 has a probability of 0.57 for reproduction selection each time the roulette wheel is spun.

A duplicate of each of the four strings that is selected for reproduction is made. These strings are placed into a mating pool (or buffer) for further genetic operator action (Goldberg, 1989:12).

CROSSOVER

Crossover is the genetic operator applied to 'parent' strings to produce 'offspring' strings. It consists of swapping string parts between individual strings in the mating buffer (Goldberg, 1989:12). Exchanging parts between a pair of strings can produce strings with high fitness values. This is because the advantaged offspring strings can receive better genes or bits from the two strings that are mating. Figure 5.5 gives an illustration of the crossover concept. Crossover takes place at a point or points along the string length called the crossover points or crossover sites. The string bits that lie to the right of the crossover site are swapped. The methods used for selecting crossover sites are described next.

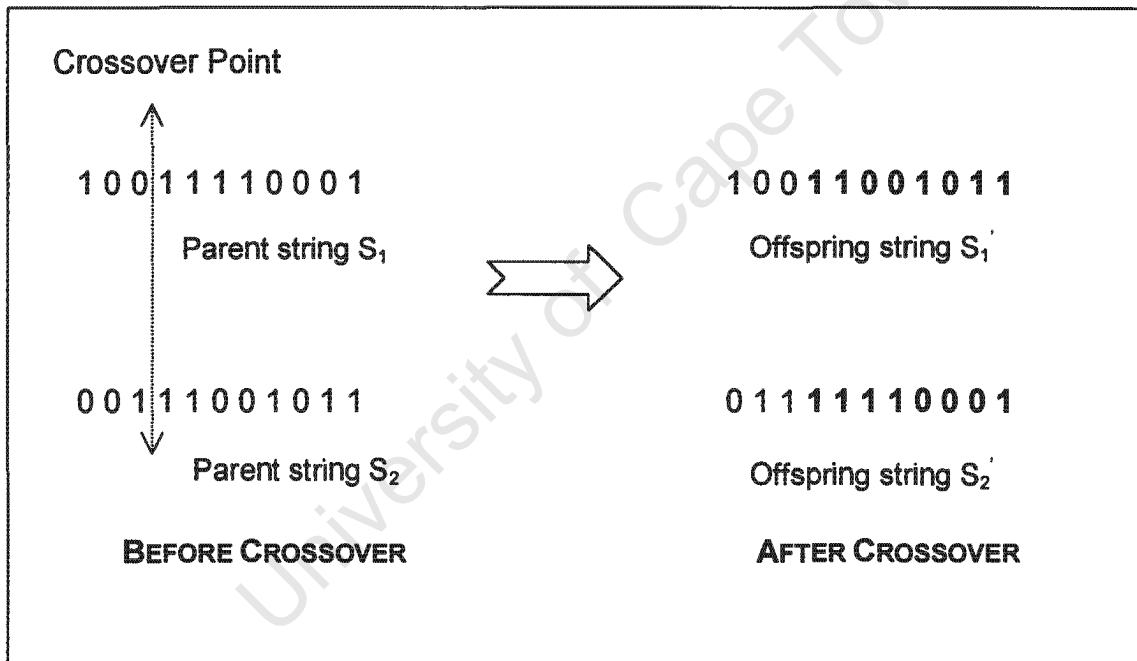


FIGURE 5.5 CROSSOVER DEMONSTRATION (GOLDBERG, 1989: 11)

The following three crossover operators are commonly used in most genetic algorithms today (Baluja, 1994):

1. One-Point Crossover

A string that contains h bits will have a string length of h . In one-point crossover, an integer position g (crossover site) is selected randomly between bit position 1 and $h-1$ (Goldberg, 1989:12). In the example shown below, g is randomly selected as 12. The parts that lie beyond the crossover site (after the twelfth bit) are swapped. A gap has been left at the crossover site to illustrate the bits that are swapped.

Parent S_1 : 100001100011 01110

Parent S_2 : 000111001100 11111

Child S_1' : 100001100011 11111

Child S_2' : 000111001100 01110

2. Two-Point Crossover

In two-point crossover, two crossover sites are selected randomly. In this case, the section that lies between the two crossover sites is swapped. In the example shown below, the bits between bit positions 5 and 11 (selected randomly) are swapped.

Parent S_1 : 10001 000011 00111

Parent S_2 : 00011 111000 10011

Child S_1' : 10001 111000 00111

Child S_2' : 00011 000011 10011

3. Uniform Crossover

In uniform crossover, bits at the same bit positions in each string are swapped at random. The particular case where the parent strings consist of a string 1's and 0's has been used to make it easier to demonstrate uniform crossover.

Parent S₁: 0000000000000000

Parent S₂: 1111111111111111

Parent S₁: 1000111011001010

Parent S₂: 0111000100110101

MUTATION

The last operator used is the mutation operator. Mutation involves changing a bit in a random position along the string. Mutation may be considered to be a random walk through the search space. It behaves as a type of 'life insurance'. Some bit values that are considered to be valuable may be lost during selection. Mutation is a way of recovering these lost bits. Mutation must be used with great care. It is normally used with a low probability and its value is taken to be about 0.0001 for most populations. Excess mutation can be harmful because the genetic algorithm will then use a random search. (Goldberg, 1989:14)

The three operators that have been described in this chapter have been used to make genetic algorithms solve many difficult optimisation problems. A genetic algorithm uses a simple process to optimise a function. It involves the generation of a random initial population. This population is then evaluated by calculating the fitness of the strings. Then the three genetic operators (reproduction, crossover and mutation) are used to create a new population.

5.5 INTRODUCTION TO THE PBIL

The Population Based Incremental Learning (PBIL) algorithm is a stochastic search technique that combines characteristics of both the genetic algorithm and competitive learning (Baluja, 1994). It has been shown to match and even

outperform the genetic algorithm (Baluja, 1994). A simple description of competitive learning networks follows.

5.5.1 COMPETITIVE LEARNING NETWORKS

Competitive learning is a method that has been devised to "cluster a number of unlabelled points into distinct groups" (Baluja, 1994). This means that patterns are identified from a set of input data and are organised into groups (clusters) that exhibit a certain degree of similarity. For example, different tree species can be identified from a group of trees by considering their height and trunk thickness simultaneously.

Competitive learning occurs without supervision (Baluja, 1994). This means that the network must decide what features it will use to group the input data without any external help. A learning technique (described later) is used for clustering the input data.

Figure 5.6 illustrates the competitive learning concept. Data is fed to the network through input nodes 1-5. The nodes communicate to each other through the 'inhibitory' and 'excitatory' connections. Nodes use inhibitory connections to 'inhibit' other nodes in the same layer. Consider an example where the gender of a person is being determined (ANN, 1999). If the probability of the gender of a person being male is 0.8 and the probability is 0.2 for being female, the network will choose the higher probability and inhibit the lower one. Each input node outputs a value to each node in the output layer, that is, it 'excites' nodes in the layer above them. The net input to each output node is done via the excitatory connections.

Each connection has a weight or strength assigned to it. The weights show how strongly each node influences the nodes connected to it. The network 'learns' by adapting the weights associated with these connections using a training rule

(described later). The information that the network receives is supplied as activation values, that is, each node is assigned a number (ANN, 1999). Nodes with higher numbers represent greater activation. Connection strengths of the inhibitory and excitatory connections determine how much of the activation value is transferred to the next node (ANN, 1999). The activation values received by each node are summed and passed to the next node connected to it. In this way, the activation is passed through the input nodes to the output nodes. The weights become finely tuned as more input patterns are identified based on the experience gained from the previous training patterns. A useful output should be produced from training the network. Using the gender example, an output of '1' could indicate that the gender of a person is male and a '0' output could indicate that the person is female.

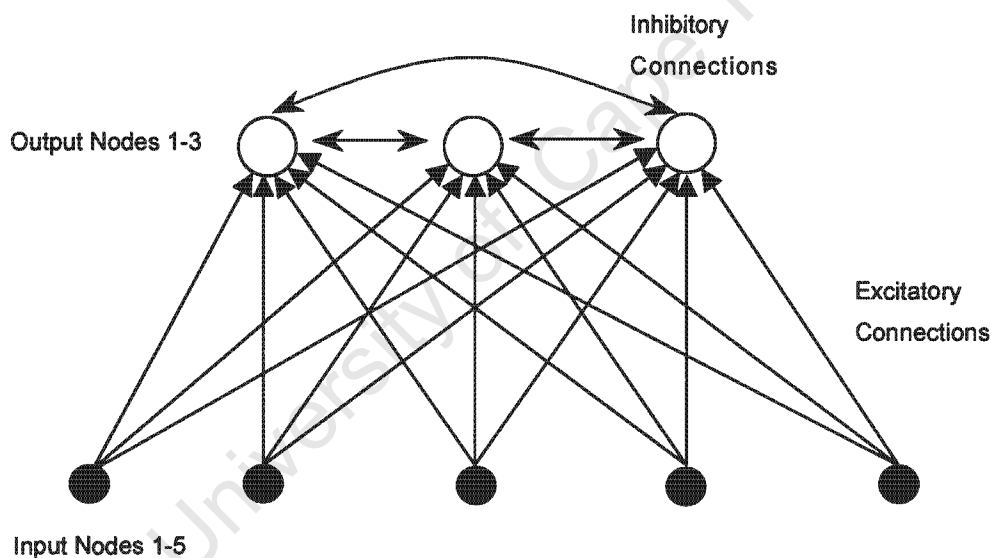


FIGURE 5.6 A COMPETITIVE LEARNING NETWORK (BALUJA, 1994)

The initial weights of the network are chosen randomly. The formula shown below is used to calculate the activation of the output nodes (Hertz, Krogh and Palmer, 1993).

$$output_i = \sum_j w_{ij} \times input_j \tag{5.1}$$

In this equation w_{ij} is the connection strength between input node i and output node j . The output node with the greatest activation is called 'the winner'. The following equation, commonly known as the 'weight update rule', is used to adjust the weights of the winner (Hertz, Krogh and Palmer, 1993).

$$\Delta w_{ij} = LR \times (input_j - w_{ij}) \quad (5.2)$$

Here LR is the Learning Rate. A learning rate of 0.1 is commonly used. In order to train the network, the winning output is continually presented to the network until stabilisation occurs. "After the network training is complete, weight vectors for each of the output units can be considered as prototype vectors for one of the discovered classes" (Baluja, 1994). Weights of the prototype vector emphasise the distinguishing features of that particular class (Baluja, 1994). The creation of a prototype vector is an important feature of the PBIL algorithm.

5.5.2 OPERATION OF THE PBIL

The PBIL creates a probability vector that can be regarded as a prototype vector for producing strings of high fitness for the next generation (Baluja, 1994). The probability vector is the same length as the solution string. Like the genetic algorithm that has been described, the PBIL uses binary representation for the solution strings.

The values of the bits in the probability vector represent the likelihood of a "1" or "0" being generated at that location. Initially, all the bit values in the probability vector are set to 0.5 (Baluja, 1994). This implies that each bit in the string stands an equal chance of generating either a "1" or "0" in that position. A population of solution strings is generated using this probability vector. The PBIL is similar to the GA at this stage, as they both create solutions that improve with successive generations and they explore a large search space in a parallel manner. The difference between the two occurs in the formation of the next population. In a

simple PBIL algorithm the solution strings that produce the best fitness values in each generation are chosen. The probability vectors are then adjusted according to these strings using the 'weight update rule'. This adjustment results in the next generation producing sample strings that move towards the best solution.

Figure 5.7 demonstrates the concepts involved in the generation of a new population using the PBIL. Four solution strings are produced with the probability vector initialised to 0.5. The third solution string generated produces the highest fitness value in this example.

Initialised Probability Vector: $[0.5 \ 0.5 \ 0.5 \ 0.5]$

Generated Samples: (1) $\left(\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{array} \right)$

Best Sample from above: $[1 \ 0 \ 1 \ 0]$

New Probability vector: $[0.6 \ 0.4 \ 0.6 \ 0.4]$

FIGURE 5.7 UPDATING A PROBABILITY VECTOR (AFTER BALUJA, 1994)

The probability vector is adjusted according to the best sample solution generated (third string). The next generation is biased so that there is a sixty percent chance of a 1 being placed in the first and third locations. In this way, the probabilities are moved to either 0 or 1. The rate at which the probability vector is updated is controlled by the learning rate. Equation 5.3 is applied to alter the probability vector and is the same one that is used in the weight update rule in competitive learning.

$$P[i] = (P[i] \times (1 - LR)) + (LR \times \text{bestsamp}[i]) \quad (5.3)$$

In this instance [i] represents each element in the vector, P is the probability vector, LR is the learning rate and *bestsamp* is the best solution string in each generation. Solution strings that are produced from the updated probability vector are now more similar to the best one in the previous generation (as in a competitive learning network where the output is moved towards a particular sample point). This process of updating the probability vector, continues until the bit values in the probability vector move away from 0.5 to favour a "1" or "0" (Baluja, 1994).

Figure 5.8 shows the operation of the PBIL algorithm.

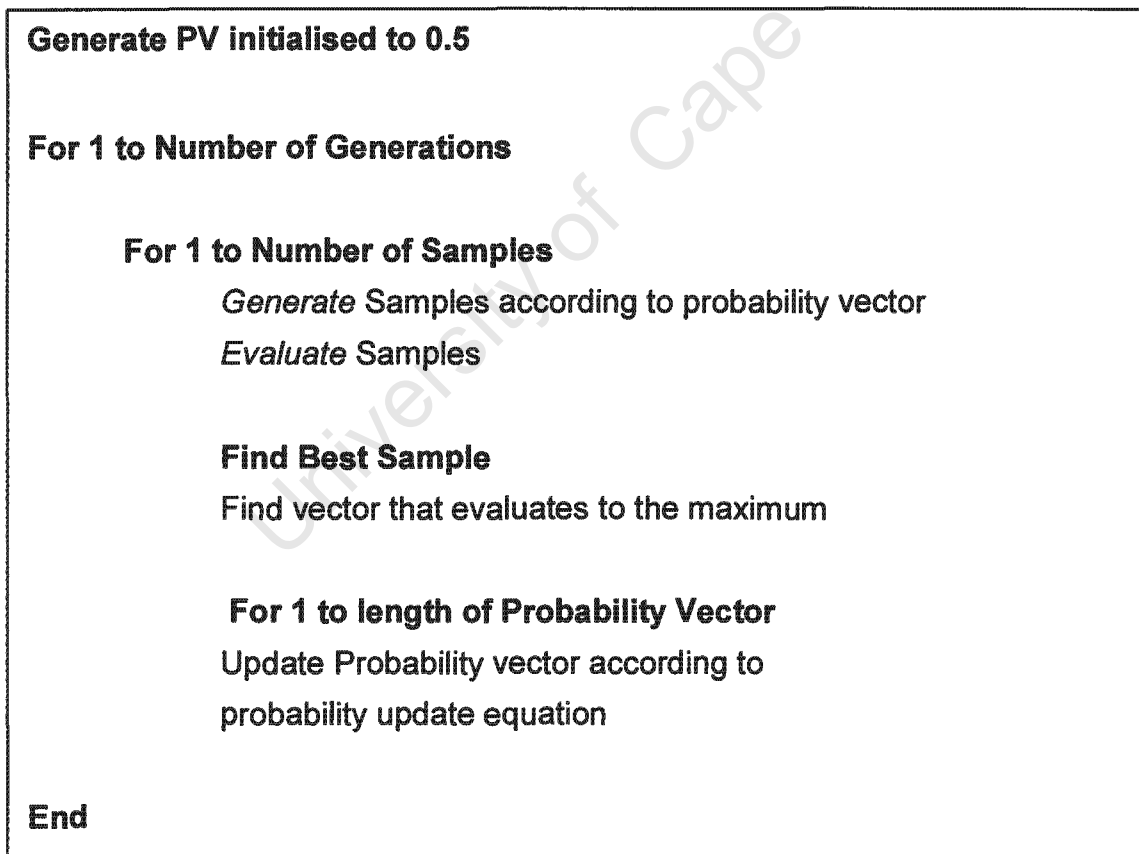


FIGURE 5.8 BASIC ALGORITHM FOR THE PBIL (BALUJA, 1994)

5.6 CONCLUSION

In this chapter, an introduction to the basic genetic algorithm has been given. Over the last few years the application of genetic algorithms has broadened to include many scientific fields and has resulted in the modification of the algorithm to solve many more complex problems

An introduction to the PBIL has also been given. This is a very simple and powerful algorithm that may be better suited for the application to certain problems that have been solved by genetic algorithms.

University of Cape Town

Chapter 6 DESIGN OF THE COMPUTER MODEL

This chapter describes the design of a computer model for determining the optimal number and placement of sectionalisers on a network. The purpose for designing the model is given. Then, the input and output data and data structures required for the model are described. The application of the PBIL algorithm in the computer model is presented. A brief description of the implementation of the model is given. Finally, tests that were performed on the model are presented.

6.1 PURPOSE AND OVERVIEW OF THE MODEL

The purpose of the model is to provide a tool to complement the system planner's experience in positioning sectionalisers on a rural MV network. The model is designed to provide an economically optimum placing of these devices. This optimum is in terms of the balance between the cost of installing these sectionalisers against the savings made by reducing the loss of revenue associated with energy not sold during outages. The user of the model may also take into account the cost to customers of the inconvenience caused by outages.

The model is parameter-driven. That is, it is designed to enable the user to easily change any or all of the parameters affecting the balance of costs referred to above. Each run of the model provides the user with the following options:

- 1) Determine the optimum number and placement of sectionalisers on a particular system.
- 2) Determine the optimum placement of a specified number of sectionalisers on a particular system

- 3) Determine the annual cost of unsupplied energy for sectionaliser locations, on a particular system, that are specified by the user.
- 4) Determine the optimum number of sectionalisers on a system with no financial constraints.

The third option is important in enabling the planner to see the effect of placing the sectionalisers in locations that seem more feasible than the locations obtained by the model. The model thus provides the user with a tool that assists in deciding on both the optimum number of sectionalisers and their placement.

The model is based on the PBIL algorithm described in the previous chapter.

6.2 INPUT DATA

The network parameters for each run are read in from a data file. The data file is called at the beginning of each run so that the user can change the contents of it.

In broad terms the input data for the network specifies:

- (i) the configuration of the network.
- (ii) the length for each section of the network.
- (iii) the loads at various points on the network.
- (iv) the locations of sectionalisers on the network.

The user is prompted to enter other parameters, such as the fault-rate and the cost of a sectionalising device, that are necessary for calculating the annual cost of unsupplied energy.

The full list of parameters is given in Table 6.1.

The section predecessor of each section in the network.	Sections on the network where the user would like to place sectionalisers
Length per section [km]	Installed cost of a sectionaliser [R]
Fault rate per section [faults/km/yr]	Cost of unsupplied energy [R/kWh]
Loads at various points [kW]	Discount rate for PV of costs [%]
Average repair time for a fault [hours]	Number of years to discount costs over

TABLE 6.1 INPUT DATA REQUIRED BY THE MODEL

An example of the input data is given in Appendix B.

6.3 OUTPUT DATA

Each run of the model provides the user with the following information:

1. The optimum number of sectionalisers.
2. The optimum location of the sectionalisers.
3. A financial evaluation of the project:
 - i. The annual cost of unsupplied energy for the network.
 - ii. The present value of the total cost (cost of unsupplied energy and cost of the sectionalisers) over the life cycle of the network.
 - iii. The annual cost of unsupplied energy that results if the user decides to add one more sectionaliser to a specific section on the network. The user may be prepared to install an extra sectionaliser if the capital is available to invest in the project.
4. The value of SAIDI for the network configuration.
5. A graph displaying the behaviour of the PBIL algorithm during the search for the optimal solution as in Figure 6.7.

6.4 DATA STRUCTURES USED

The main data structures used in the model are:

(i) The *network_data* array. An $a \times 7$ element array holding the network configuration information, where a is the number of sections in the network. The 7 items held for each section are as follows:

1. The section number, which is the same as the row number in the array.
2. The section number that is the immediate predecessor of the section number in the first column.
3. The length of the section in km.
4. The repair time, in hours, for a fault occurring in each section. It was decided that the user should just enter an average value for the repair time (for example, 2 hours) for any fault that occurs on the network to make the program more user friendly.
5. The annual outage time in hours/yr.
6. The presence of a sectionaliser. A '1' means that there is a sectionaliser at the sending end of this particular section and a '0' means that there is no sectionaliser.
7. The value for the average load connected to this particular section in kW.

(ii) The *annual_cost_of_unsupplied_energy* array. An $a \times 11$ element array holding values for each section of the network necessary for the calculation of the annual cost of unsupplied energy. There are 11 items held for each section. Items 4 to 9 have been included so that the user can see the effect on the optimal solution obtained if an assumption is made that there is a chance that the sectionalisers placed may fail to operate (see test in section 6.7). For all the tests carried out in Chapter 7, it was assumed that the probability of sectionaliser

failure is zero. Therefore, items 4 to 9 are not necessary and the *annual_cost_of_unsupplied_energy* array is then an $a \times 5$ element array. The 11 items held for each section, remembering that items 4 to 9 are not needed if it is assumed that there is no chance that a sectionaliser will fail to operate when required to, are as follows:

1. The load interrupted due the isolation of this particular section without any sectionalisers installed.
2. The section number containing the closest upstream sectionaliser for this section.
3. The load interrupted due to the operation of the closest upstream sectionaliser for this section.
4. The section number that holds the *next* upstream sectionaliser in the event that the first upstream sectionaliser fails to operate.
5. The load interrupted due the operation of the *next* upstream sectionaliser that has just been described.
6. The section number that holds the next upstream sectionaliser in the event that the second upstream sectionaliser fails to operate.
7. The section number that holds the next upstream sectionaliser in the event that the third upstream sectionaliser fails to operate.
8. The load interrupted due to the operation of the second upstream sectionaliser.
9. The load interrupted due to the operation of the third upstream sectionaliser.
10. The annual cost of losses caused by total isolation, that is, due to the operation of the main system breaker.
11. The annual cost of losses caused by local isolation, that is, due to the operation of the sectionalisers.

(iii) The *probability* vector. This is an a -element vector holding values for the probability vector that is needed by the PBIL algorithm to generate trial solutions. Initially, all the bits of this vector are set to 0.5.

6.5 APPLICATION OF THE PBIL ALGORITHM TO THIS PROBLEM

This section shows the structure of the program and gives a brief description of the purpose of each of the procedures used by the program. A more detailed description of four of the procedures is then given.

University of Cape Town

6.5.1 STRUCTURE OF THE PROGRAM

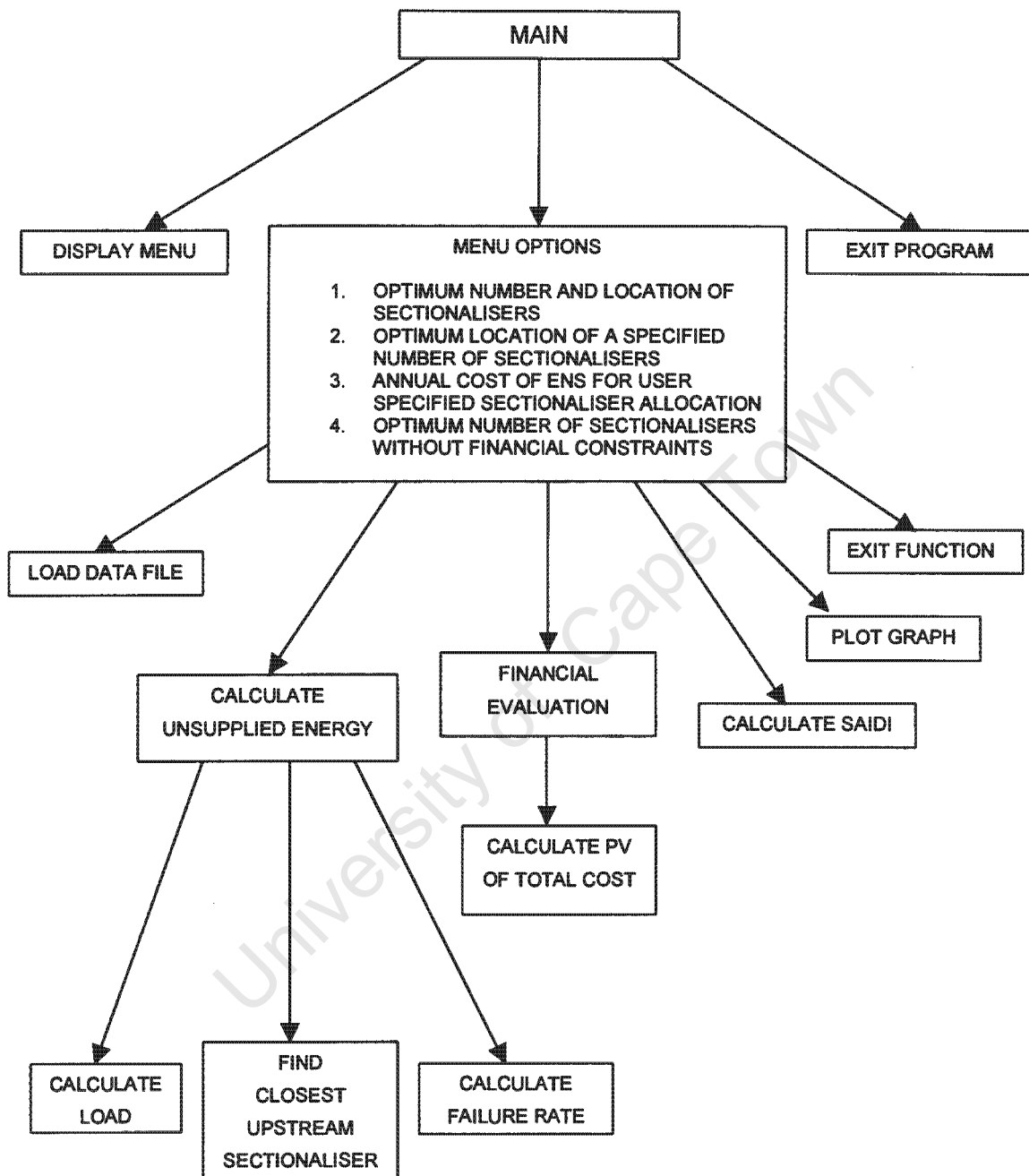


FIGURE 6.1 STRUCTURE DIAGRAM OF THE PROGRAM

1. *DISPLAY MENU* – A menu is displayed which gives the user four options to choose from.

2. *MENU OPTIONS* – The four options that the user can choose from have been mentioned in section 6.1 and are shown below once again:

- i. Determine the optimum number and placement of sectionalisers on a particular system.
- ii. Determine the optimum placement of a specified number of sectionalisers on a particular system
- iii. Determine the annual cost of unsupplied energy for sectionaliser locations, on a particular system, that are specified by the user.
- iv. Determine the optimum number of sectionalisers on a system with no financial constraints.

A different function is called that corresponds with the option selected. The names of the four functions in the program are *COSTENS_MINIMISATION*, *SPECIFIC_SWITCH*, *CALCULATE_ENS* and *ENS_MINIMISATION*. Each of these functions uses the PBIL algorithm.

3. *LOAD DATA FILE* – This procedure is called so that the user can enter the network data. This information will be stored in an array.

4. *CALCULATE ENS* – This function is called by the PBIL to calculate the annual cost of unsupplied energy for the network for each trial solution generated.

5. *PLOT GRAPH* – This function plots a graph to show the behaviour of the PBIL algorithm during the search for the optimal solution.

6. *FINANCIAL EVALUATION* – This procedure is used to calculate the annual total cost of the network.

7. *SAIDI* – Calculates the value of SAIDI for the network arrangement.

8. *FIND CLOSEST UPSTREAM SECTIONALISER* – This procedure finds the section number containing the closest upstream sectionaliser for each section.

9. *CALCULATE FAILURE RATE* – This procedure is used to calculate the total failure rate after each section.

10. *CALCULATE LOAD* – This function calculates the load that is interrupted due to the isolation of a particular section.

6.5.2 UNDERLYING LOGIC OF THE ALGORITHM

This section begins with a description of how a network can be represented using a tree diagram (Levitin, Mazal-Tov and Elmakis, 1994). It then gives a more detailed description of four of the procedures that were used.

Figure 6.2 shows a simple network that can be used by the model for determining the location of sectionalisers. Each section of the network has been given a number so that information such as the length of the section can be stored.

This system may be represented by a tree graph as shown in Figure 6.3. A tree graph representation of the network makes it easy to store the network information in an array as described in section 6.4. The heavy dots on the tree diagram correspond with the heavy dots on the system in Figure 6.2. These dots represent the beginning, or sending end, of each section.

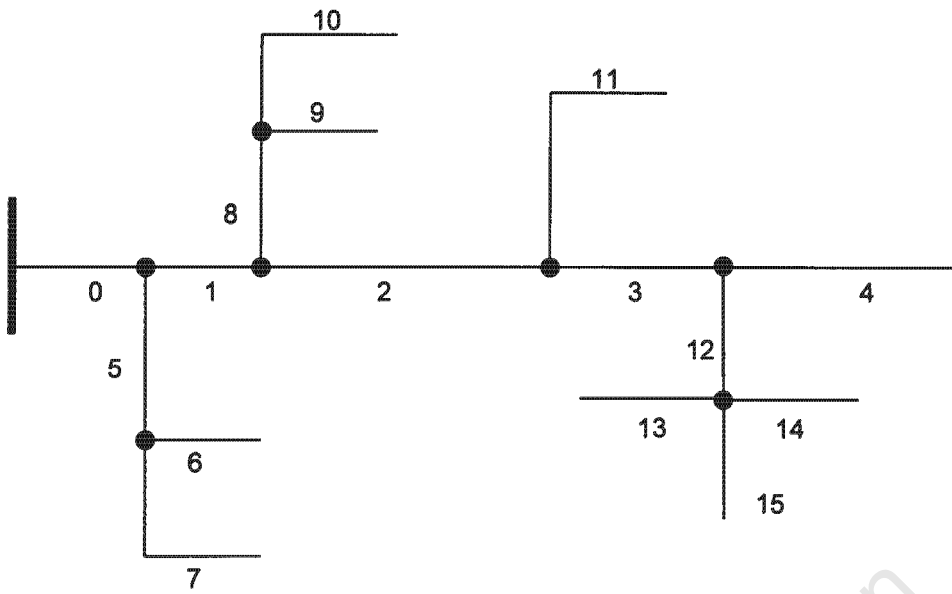


FIGURE 6.2 MODEL OF A SIMPLE NETWORK

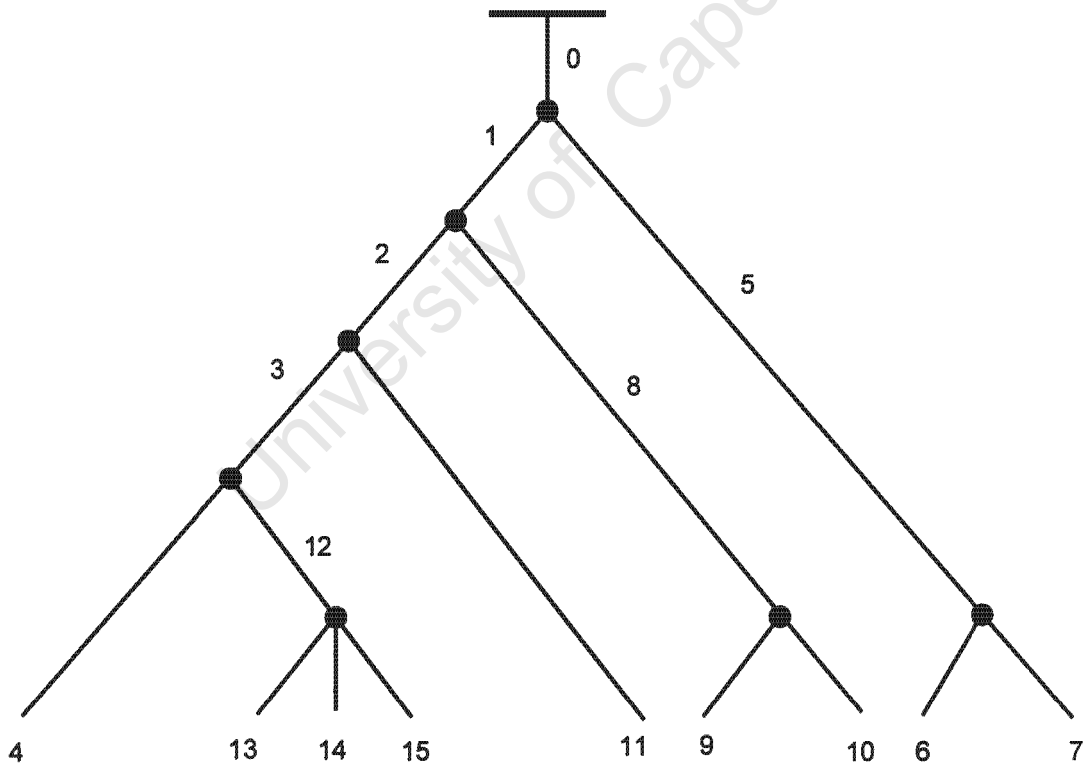


FIGURE 6.3 TREE DIAGRAM REPRESENTATION OF THE SIMPLE NETWORK

The numbers at the bottom of the tree diagram, sections 4, 13, 14, 15 and so forth, are normally referred to as terminal branches since no other sections are dependent on them. If any of the sections that are terminal branches were to be isolated, supply to the remainder of the network would not be affected.

The remainder of this section will describe four of the procedures that were developed.

1. CALCULATE FAILURE RATE

This function uses recursion to calculate the total failure rate downstream of each section of the network. The network's data file and the section number, for which we are trying to determine the failure rate, are passed as input arguments to the function.

The function locates all the numbers in the second column (containing the immediate predecessor of the section in the first column) of the *network_data* array that are equal to the section number at which the failure rate is being determined. It then moves to the adjacent number, in the first column, to the values that have been found and then stores these values in a vector called *leaves*. Then for each of the elements in the vector *leaves*, the function finds the length for that section and multiplies the length by the failure rate per km and stores the value obtained, in a variable, as the failure rate. This is performed recursively until the total failure rate downstream of a section is found. The operation of this function is demonstrated next.

Suppose that we have the simple network shown below:

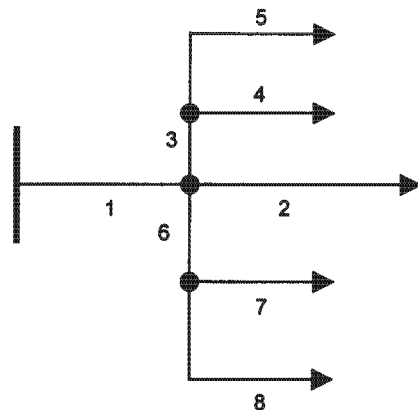


FIGURE 6.4 A SIMPLE DISTRIBUTION NETWORK

and that we are trying to find the total failure rate downstream of the sending end of section 1:

section	predecessor	length
1	0	7
2	1	4
3	1	6
4	3	3
5	3	2
6	1	5
7	6	1
8	6	1

FIGURE 6.5 TRAVERSAL OF THE NETWORK_DATA ARRAY TO DETERMINE FAILURE RATE DOWNSTREAM OF SECTION 1

The function locates section 1 and finds that the length of this section is 7 km's from the third column of the network_data array (Figure 6.5). The length, 7 km's,

is then multiplied by the failure rate per km. The value obtained for this multiplication is stored in a variable as the current failure rate downstream of section 1. After that, the function searches the second column (which contains the section number that precedes the section number in the first column) for all the values equal to 1. The function finds sections 2, 3 and 6 and stores these values into the vector *leaves*, that is *leaves* = [2,3,6]. From Figure 6.4 it can be seen that sections 2, 3 and 6 are the sections that are immediately downstream of section 1. The function picks up the values 4, 6 and 5 km's that are the lengths of sections 2, 3 and 6 respectively, and multiplies each of these lengths by the failure rate per km. The three values obtained by the multiplication are added to the previous value for the failure rate that was stored in a variable. The function then repeats the procedure, that has just been described, for each element in the vector named *leaves*. By repeating the procedure, the function finds all the sections downstream of section 1 and thus calculates the total failure rate downstream of it. Pseudo code representation for this is as follows:

Function *FAILURE_RATE* (*Network_data* array and *Section_number*)

Initialise *failure_rate* to zero

Find values in second column in network data array == *Section_Number* and store in vector *leaves*

For 1 to length of the vector *leaves*

failure_rate = *length_of_section* x *fault_rate_per_km*

 Call function *FAILURE_RATE* to calculate failure rate of next element in the vector *leaves* and add value to the variable named *failure_rate*

End For

Return *failure_rate* for *Section_Number*

2. FIND CLOSEST UPSTREAM SECTIONALISER

The name of the function for this procedure is called *Calculate_CPP*. This procedure finds the section that contains the sectionaliser that is the closest upstream one for a particular section in the network. It is important to find the closest upstream sectionaliser because a fault in a particular section will cause an interruption in power supply to all sections that are downstream of this sectionaliser. With this information, the energy not supplied due to faults that occur in each section during the year can be calculated.

This is another recursive function and passes the network data file and the section number, for which the closest upstream sectionaliser is being determined, as input arguments. The function checks if the sectionaliser value in the sixth column of the network data array corresponding to the section number, that is passed to the function as an argument, is a one. If it is a one, then that particular section contains a sectionaliser and that is the value assigned as the closest upstream sectionaliser of that particular section. If the value is a zero, the array is traversed using recursion until the first upstream section containing a sectionaliser is found. For example, if the closest upstream sectionaliser for section 7 in Figure 6.4 were being determined, the function would check if the switch value for section 7 is a one. If it were a one then the closest upstream sectionaliser for section 7 is at the sending end of section 7. If it were a zero then the function would check the second column (the predecessor column), of the network data array, to find that the section that lies immediately upstream of section 7 is section 6. If the switch value of section 6 is a one then the sending end of section 6 contains the closest upstream sectionaliser for section 7. If the switch value of section 6 is a zero, the same procedure that has been described is repeated until the closest upstream sectionaliser for section 7 is found. The pseudo code for this procedure is as follows:

Function *CALCULATE_CPP* (Network Configuration Array and Section_Number)

If value in switch column of array corresponding to Section_Number == 1

 Then closest_upstream_switch == Section_Number

Else

 Call function again and check if the next upstream section contains device

End

Return the section number containing closest upstream sectionaliser

3. CALCULATE UNSUPPLIED ENERGY

This procedure calculates the annual cost of unsupplied energy for the network. It exists in four forms, that is, *MINSYSENS1* – 4 that correspond with each of the menu options mentioned in section 6.5.1. Each form calculates the annual cost of unsupplied energy, but a few modifications are necessary to perform this calculation depending on which option is selected by the user. Therefore,

- function *ENS_MINIMISATION* calls *MINSYSENS1*.
- function *COSTENS_MINIMISATION* calls *MINSYSENS2*.
- function *SPECIFIC_SWITCH* calls *MINSYSENS3*.
- function *CALCULATE_ENS* calls *MINSYSENS4*.

The array containing the network data and the cost of unsupplied energy per kWh are passed to this function as input arguments. An array called *ENSDATA* (*annual_cost_of_unsupplied_energy* array) is formed. The annual cost of unsupplied energy, due to the operation of the main system breaker (total isolation), is calculated for each section. *CALCULATE_CPP* is called to find the closest upstream sectionaliser for each section. This is needed so that the annual cost of unsupplied energy, due to the operation of the sectionalisers (local isolation), is also calculated for each section. The annual cost of unsupplied

energy due to total and local isolation that have been calculated for all of the sections are summed up to give the annual cost of unsupplied energy for the whole network. The pseudo code for this function follows:

Function *MINSYSENS* (Network data, unit cost of ENS)

 Create array *ENSDATA* to store values for calculation of ENS

 Find the closest upstream sectionaliser for each section and store in the second column of *ENSDATA*

 The load interrupted due to the operation of the closest upstream sectionaliser in the third column of *ENSDATA*

 Calculate cost of unsupplied energy caused by total isolation for each section and store values in second last column of *ENSDATA*

 Calculate cost of unsupplied energy caused by local isolation for each section and store values in last column of *ENSDATA*

 Sum the last two columns of *ENSDATA* to determine annual cost of unsupplied energy.

Return annual cost of unsupplied energy

4. PBIL

This section describes the implementation of the PBIL algorithm in the functions named in 6.5.1. Before the PBIL can begin operation, a data file containing the network data needs to be loaded. Then other input parameters such as the unit cost of unsupplied energy and fault rate that are necessary for the calculation of the annual cost of unsupplied energy are entered. Next, a probability vector in which each bit is set to 0.5 is created. The bits are set to 0.5 so that the trial solutions generated in the first generation are created randomly. The probability vector is the same length as the number of sections in the system. Each bit value in the probability vector represents the chance of a '1' or '0' being placed in each bit position in the trial solution (see section 5.5). This binary representation of the trial solution is suitable for this optimisation problem because a '1' in the trial

solution means that there is a sectionaliser in that section and a '0' means that a sectionaliser has not been placed. These trial solutions replace the switch column in the network data array so that the annual cost of unsupplied energy with this sectionaliser arrangement can be determined. The functions *MINSYSENS1-4* (depending on which option the user has selected) are evaluated using each of these trial solutions. The trial solution that produces the highest fitness value is then used to update the probability vector. The updated probability vector is used to create a new batch of trial solutions for the next generation to produce trial solutions with higher fitness values. This process of updating the probability vector, based on the best trial solution in each generation, is continued so that the optimum solution can be found eventually after a certain number of generations have been completed. A graph is then plotted to show the behaviour of the PBIL algorithm during the search for the best trial solution.

For the function *SPECIFIC_SWITCH* (the function that finds the optimal placement of a specified number of sectionalisers), the PBIL is penalised each time it produces trial solutions that contain either more or less sectionalisers than the specified number of sectionalisers. The PBIL is penalised by multiplying the objective function (annual cost of unsupplied energy plus the annual cost of the sectionalisers placed) that is being optimised by a factor of ten million and thus forces the PBIL to generate trial solutions with the specified number of sectionalisers. The pseudo code for the operation of the PBIL is as follows:

```
Load data file containing network configuration
Enter other parameters for calculating cost of unsupplied energy
Create probability vector initialised to 0.5
  For 1 to maximum_number_generations
    For 1 to number_trial_solutions
      Create trial solution based on probability vector
      Calculate fitness value of this trial solution
      Test to see if current trial solution is best one so far
    End For
    Update probability vector using best trial solution from each
    generation
  End For
Return trial solution that gives the best fitness value
Plot graph to show behaviour of the algorithm during the search
```

6.6 IMPLEMENTATION OF THE MODEL

The computer model was implemented using MATLAB. It is the language with which I am most proficient and would be suitable for this problem. Although Matlab programs may take longer to execute than other programming languages such as C++ or Java, the emphasis for the implementation of the model was on functionality rather than efficiency.

A listing of the program can be found in Appendix C.

6.7 TESTING

This section describes a test that was done on the model to see if it is working correctly.

DEMONSTRATION OF THE PBIL ON A SIMPLE NETWORK

Figure 6.6 shows a simple network that has been used to test the program. The network consists of a main system breaker with 25 sections. Fifteen of these sections are loaded. Levitin, Mazal-Tov and Elmakis (1994) used this network to test the genetic algorithm-based method that they designed for determining the optimal location and number of sectionalisers. The numbering of the sections is the same as those by Levitin, Mazal-Tov and Elmakis.

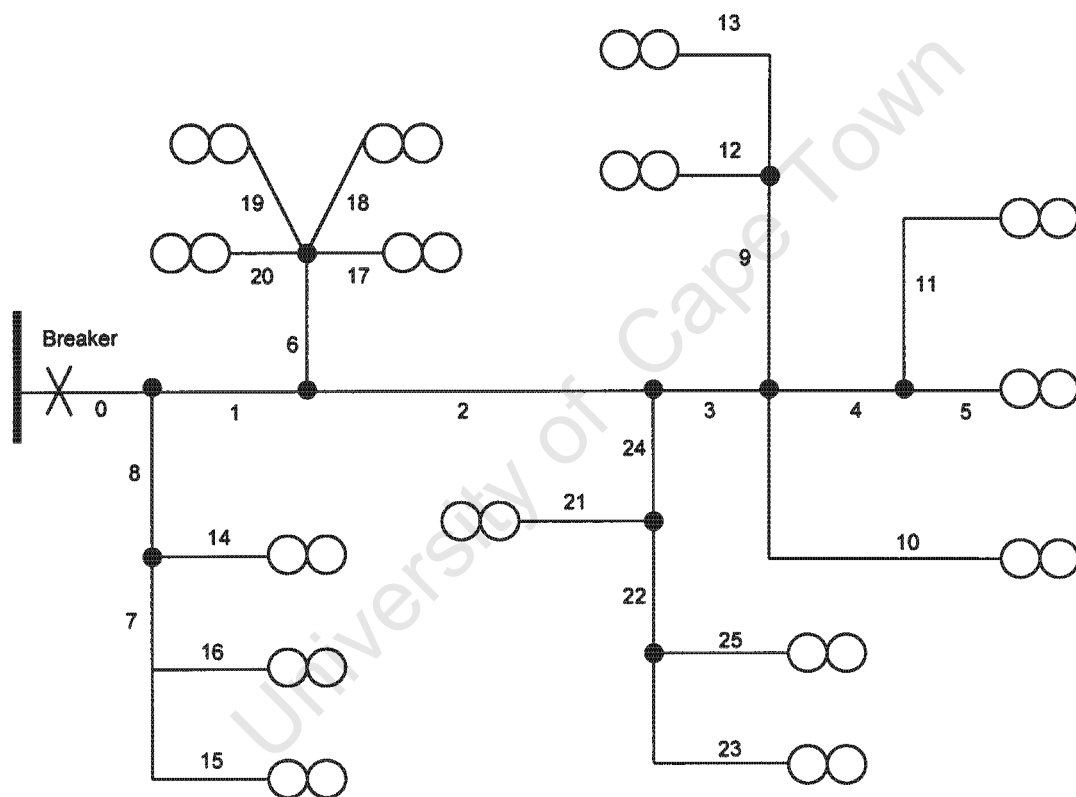


FIGURE 6.6 A SIMPLE TEST DISTRIBUTION SYSTEM (LEVITIN, MAZAL-TOV AND ELMAKIS, 1994)

In (Levitin, Mazal-Tov and Elmakis, 1994) the annual cost of a sectionaliser h , has been taken as US\$1000. Repair time τ , for the sectionalisers was taken as one hour and T_s , the main breaker switching time, was taken as 0.03 hours.

The probability of sectionaliser failure p and the outage time of a sectionaliser γ , due to sectionaliser failure, were varied to see the effect that this has on the output of the program.

The program was run using one hundred generations and thirty trial solutions in each generation. Table 6.2 shows the results that were obtained by varying the two parameters (p and γ) using my program. Figure 6.7 shows the behaviour of the PBIL algorithm in its search for the optimal solution.

TEST	H	γ	p	NUMBER OF SECTIONALISERS	TOTAL ANNUAL COST
	[US\$]	[hrs]			[US\$]
A	1000	0.02	0.000	6	14 869
B	1000	0.10	0.010	5	19 544
C	1000	0.25	0.005	3	27 007

TABLE 6.2 EFFECTS OF SECTIONALISER p AND γ ON ANNUAL ENS

The total annual cost figure represents the annual cost of unsupplied energy plus the annual cost of the sectionalisers installed.

The reason for performing this test was to compare the results that the PBIL obtained with the results that Levitin, Mazal-Tov and Elmakis (1994) obtained using their genetic algorithm-based method. The results that were obtained for tests A, B and C were also obtained by Levitin, Mazal-Tov and Elmakis (1994).

The number of evaluations that were necessary to obtain the optimum solution is not mentioned in (Levitin, Mazal-Tov and Elmakis, 1994). However, a graph was plotted showing the progress of the genetic algorithm during its search up to only 3000 evaluations (the genetic algorithm had not yet reached the optimum after 3000 evaluations). My program, which uses the PBIL algorithm, obtained the

optimum solution after approximately 1200 evaluations (see Figure 6.7 – the optimal solution is reached after approximately 40 generations x 30 trial solutions). This demonstrates the searching power of the PBIL algorithm.

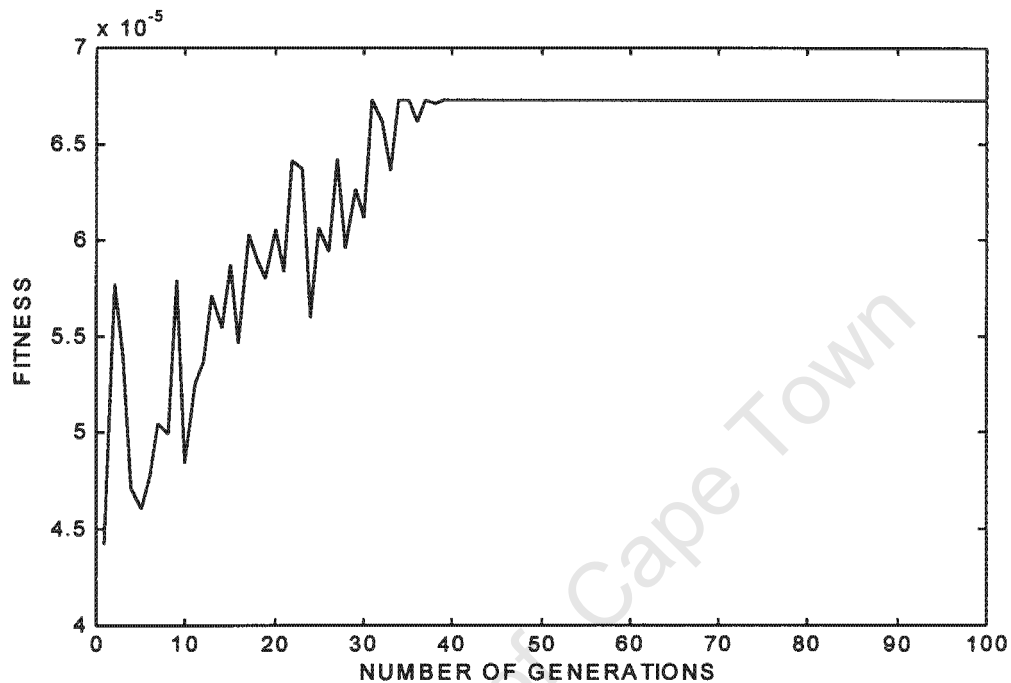


FIGURE 6.7 FITNESS OBTAINED USING A PBIL

Sectionaliser breakdown is not a serious problem and many people would regard it as a second-order consideration. Other factors that are more important, such as the fact that an extra sectionaliser will introduce extra downtime or the value used for the unit cost of unsupplied energy, are presented in the next chapter to demonstrate the effect that they have on the results obtained by the PBIL.

6.8 CONCLUSION

A computer model that has been designed to obtain the optimum location and placement of sectionalisers has been presented here. Tests that have been carried out show that the program, that has been implemented, is producing results that seem feasible. The tests also showed that the PBIL algorithm obtained the results with considerably fewer numbers of evaluations than the genetic algorithm method used by Levitin, Mazal-Tov and Elmakis (1994).

University of Cape Town

Chapter 7 TEST RESULTS USING THE PBIL ALGORITHM

Chapter seven shows how the parameter-setting values chosen for the PBIL affect the performance of it. The model is then tested on two networks and the sensitivity of the model to parameters such as failure rate and discount rate is demonstrated. The results obtained when using the PBIL on these networks and the ensuing discussions are presented in this chapter.

7.1 SENSITIVITY OF THE PBIL TO ITS PARAMETER SETTINGS

This section shows the importance of selecting the correct parameter-setting values for the PBIL to ensure the optimal performance from it. The PBIL's learning rate and the number of its trial solutions were varied to obtain the best PBIL parameter-setting values. These tests were carried out using Test A that was described in section 6.7.

7.1.1 SENSITIVITY OF THE PBIL TO NUMBER OF TRIAL SOLUTIONS

The number of trial solutions used for each run of the program was increased to see the effect that this had on the output of the program. The program was run using 100 generations and the results of this test are shown in Table 7.1.

NUMBER OF TRIAL SOLUTIONS	NUMBER OF GENERATIONS TO REACH OPTIMAL SOLUTION	TIME TO REACH OPTIMAL SOLUTION [min:sec]
5	81	00:18
10	70	00:27
15	54	00:32
20	45	00:40
25	42	00:41
30	38	00:44
35	25	00:33
40	30	00:47

TABLE 7.1 EFFECT OF THE NUMBER OF TRIAL SOLUTIONS SELECTED

From Table 7.1, it can be seen that the program runs rapidly, approximately 2.5 times faster, using five trial solutions instead of forty. This is because there are fewer solutions to evaluate, as there are 500 evaluations (5x100) using 5 trial solutions per generation compared to 4000 evaluations (40x100) using 40 trial solutions per generation.

Table 7.1 also shows that it takes more generations to determine the optimum number of sectionalisers for placement on a network if a small number of trial solutions is used. For this test, only 30 generations using 40 trial solutions are needed, compared to 81 generations using 5 trial solutions. The reason for the reduced number of generations required to obtain the optimal solution is due to the increase in the number of evaluations performed per generation before the probability vector is updated in order to generate the next population. In general, increasing the number of trial solutions in a generation allows the algorithm to search more regions in the function space and thus reduces the number of generations required to find the optimal solution.

From Table 7.1, it may appear that the optimal solution is reached much more quickly if 35 trial solutions are used instead of 40. However, it must be remembered that the trial solutions obtained in the first generation are obtained by initialising the bits in the probability vector to 0.5. This means that the trial solutions obtained in the first generation are generated randomly. The probability vector learns or is updated according to the best trial solution in each generation. Therefore, the closer the best solution obtained from the first generation is to the optimal solution, the faster the PBIL will converge to the optimum solution.

A compromise between speed and the number of trial solutions that are used needs to be reached. It can clearly be seen in Table 7.1 that there is no significant difference with regard to the number of generations needed in obtaining the optimal solution when using any number of trial solutions varying between 20 and 40. For this problem, 20 to 25 trial solutions will suffice as a larger number of trial solutions only increase the run time of the program.

7.1.2 SENSITIVITY OF THE PBIL TO ITS LEARNING RATE

The value chosen for the learning rate of the PBIL is of great importance. Essentially, the learning rate can take on any value between one and zero. A compromise must be reached between the *exploration* and *exploitation* of the function space. *Exploration* of the function space refers to the algorithm's capacity to explore the function space fully, whilst *exploitation* is the knowledge gained by the algorithm about the function space in order to reduce the number of regions in which it should search (Baluja, 1994). A value needs to be found for the learning rate, so that a balance between the amount of *exploration* and *exploitation* of the function space can be found, to enable the PBIL to find the optimal solution as rapidly as possible.

The PBIL's learning rate was varied between zero and one to demonstrate the effect of this on the PBIL's performance. The program was run using two hundred generations with twenty trial solutions and the results are shown in Table 7.2.

LEARNING RATE	NUMBER OF SECTIONALISERS	ANNUAL COST OF ENS [US\$]	NUMBER OF GENERATIONS TO REACH THE OPTIMUM
0.001	12	198 866	Optimum not reached
0.005	9	187 990	Optimum not reached
0.01	9	192 161	Optimum not reached
0.05	6	181 294	63
0.1	6	181 294	30
0.2	6	181 294	25
0.25	7	183 854	Optimum not reached
0.3	7	183 854	Optimum not reached
0.4	8	185 710	Optimum not reached
0.5	9	188 756	Optimum not reached
0.6	10	195 339	Optimum not reached
0.7	12	207 548	Optimum not reached
0.8	13	204 474	Optimum not reached
0.9	15	208 576	Optimum not reached

TABLE 7.2 EFFECT OF VARYING THE LEARNING RATE ON THE PBIL

From Table 7.2, it can be seen that the optimal solution will be found if the learning rate is set at any value between 0.05 and 0.2. It can also be seen that it takes more generations to find the optimal solution with a low learning rate, for example, it takes 63 generations with a learning rate of 0.05 compared to 25 generations with a learning rate of 0.2. The reason for this will be explained later in this section.

A graph demonstrating the effect of varying the learning rate, between 0.001 and 0.6, on the performance of the PBIL is given in Figure 7.1.

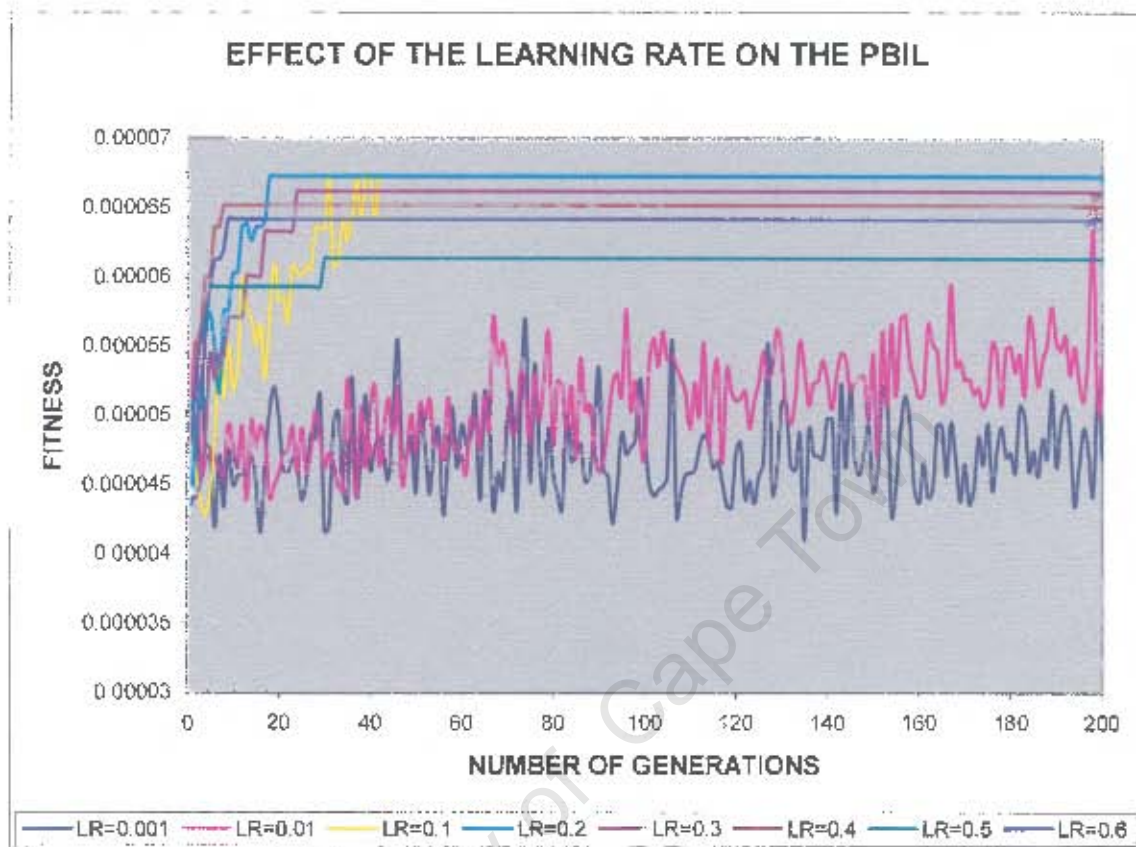


FIGURE 7.1 EFFECT OF THE LEARNING RATE ON THE PBIL

Figure 7.1 shows that for low learning rates between 0.001 and 0.01, the search is quite random and the solutions obtained are far from the best one. The equation for updating the probability vector was presented in section 5.5.2 as shown below (*bestsamp* is the trial solution with the highest fitness value obtained from each generation):

$$P[i] = (P[i] \times (1 - LR)) + (LR \times \text{bestsamp}[i])$$

If the learning rate is set to zero then the equation for updating the probability vector becomes:

$$P[i] = P[i]$$

This is another way of saying that there is no *exploitation* of the function space and the algorithm is merely performing a random search.

From Figure 7.1, it can be seen that the PBIL converges too quickly to the fitness values 6.65×10^{-6} , 6.5×10^{-6} , 6.4×10^{-6} , 6.2×10^{-6} as the learning rate is increased from 0.3 to 0.6. These fitness values are further away from the optimal solution, which in this instance is 6.72×10^{-6} . If the learning rate is set to one then the equation for updating the probability vector becomes:

$$P[i] = \text{bestsamp}[i]$$

This means that the function space is not explored at all and the probability vector instantly takes on the value of the best solution from the *first* generation and does not shift from that value.

As the learning rate is increased from zero to one, the amount of exploration decreases and the exploitation increases (Baluja, 1994). It can be seen that the best value for the learning rate is between 0.05 and 0.2. I decided to use a learning rate of 0.1 for most of the tests even though a learning rate of 0.2 acquired the optimum solution more rapidly. This was done to guard against the possibility of premature convergence by the PBIL when it is trying to find the optimal solution.

7.2 EFFECT OF UNIT COST OF ENS ON THE ANNUAL SAVINGS IN THE TOTAL COST FOR A UTILITY

This section shows the importance of the value selected for the cost of a unit of unsupplied energy C_e . Table 7.3 shows the annual savings in the total cost for

the distribution system in Figure 6.6 (previous chapter) that result when a sectionaliser is placed in each of the possible locations in turn (locations 1 to 25) for different ranges of values of C_e between US\$0.25/kWh and US\$2/kWh. The annual savings in the total cost are obtained by subtracting the annual cost of a sectionaliser from the reduction of the annual cost of unsupplied energy that results from placing the sectionaliser in that particular location.

LOCATION	SAVINGS IN THE	SAVINGS IN THE	SAVINGS IN THE	SAVINGS IN THE	SAVINGS IN THE
	ANNUAL TOTAL	ANNUAL TOTAL	ANNUAL TOTAL	ANNUAL TOTAL	ANNUAL TOTAL
	COST	COST	COST	COST	COST
	[US\$]	[US\$]	[US\$]	[US\$]	[US\$]
	C_e =US\$0.25/kWh	C_e =US\$0.5/kWh	C_e =US\$1/kWh	C_e =US\$1.5/kWh	C_e =US\$2/kWh
1	814	2 728	6 556	10 384	14 212
2	1272	3 644	8 387	13 131	17 874
3	659	2 418	5 935	9 453	12 970
4	-179	743	2 585	4 428	6 270
5	-939	-778	-457	-135	187
6	460	2 020	5 140	8 260	11 380
7	-530	40	1 180	2 319	3 459
8	-76	948	2 995	5 042	7 090
9	-715	-331	439	1 208	1 978
10	-936	-773	-446	-118	209
11	-945	-789	-479	-168	143
12	-967	-833	-567	-300	33
13	-957	-814	-528	-242	44
14	-943	-787	-473	-159	154
15	-963	-825	-550	-275	0
16	-939	-778	-457	-135	187
17	-946	-792	-484	-176	132
18	-939	-778	-457	-135	187
19	-943	-787	-473	-159	154
20	-945	-789	-479	-168	143
21	-941	-781	-462	-143	176
22	-284	532	2 164	3 795	5 427
23	-942	-784	-468	-151	165
24	535	2 170	5 441	8 711	11 981
25	-946	-792	-484	-176	132

TABLE 7.3 EFFECT OF C_e ON ANNUAL SAVINGS IN THE ANNUAL TOTAL COST OF THE SYSTEM

From Table 7.3, it can be seen that the value selected for C_e is very important. For location 7 with $C_e=US\$0.25/kWh$, the reduction in the annual cost of ENS is -US\$530 and with $C_e=US\$2/kWh$, the reduction in the annual cost of ENS is US\$3459. The negative value means losses are incurred instead of savings being effected because the reduction in ENS caused by the placement of a sectionaliser is less than the cost of the sectionaliser. The placement of a sectionaliser on the system can be justified using $C_e=US\$2/kWh$ because the reduction in the annual cost of ENS is so much higher than the cost of a sectionaliser which is taken as US\$1000 (Levitin, Mazal-Tov and Elmakis, 1994).

It can also be seen that the placement of a sectionaliser on certain sections does not result in a significant saving in the annual total cost of the distribution system. This is because the load isolated by sectionalisers on these sections is very small. In fact, most of these sections are sections that were described as being terminal branches in section 6.5.2.

It is very difficult to assign a value for the unit cost of unsupplied energy, as there are many issues to be taken into account. Such considerations include the loss of power supply to the consumer and his accompanying inconvenience. There have been many discussions to determine the appropriate value that should be used for the cost of unsupplied energy. For this reason, the model was designed so that the user can specify a value for the cost of unsupplied energy. Dutkiewicz (1991) published a paper on a study that was performed in Sweden by Anderson (1986) with regard to this topic. The results of the study are presented in Table 7.4.

CONSUMER OUTAGE COSTS [US\$/kWh]	DURATION OF OUTAGE		
	0.5h	2h	8h
Industry	2.41	5.55	14.75
Households	0.24	1.27	14.75
Agriculture	0.48	2.19	13.20
Offices	6.14	25.25	106.44
Commerce	5.43	14.60	51.45
Railroads	1.94	5.78	21.21
Entire Country	2.08	6.42	26.38

TABLE 7.4 TYPICAL VALUE OF C_e FOR DIFFERENT CONSUMERS IN SWEDEN (DUTKIEWICZ, 1991)

Dutkiewicz took into account the economic differences that exist between South Africa and Sweden as well as the varying exchange rates. He formulated Table 7.5 that gives the rand equivalent of the cost of outages for different periods of time. From the paper it is not clear what type of customer these results are valid for, but it is clear that the value selected for the cost of unsupplied energy will have a significant effect on the output of the program since this value varies greatly for different outage durations.

OUTAGE DURATION [hrs]	R/kWh (1980 Values)
0.5	1.70
1.0	2.60
1.5	5.10
2.0	6.90
4.0	8.90
8.0	19.00

TABLE 7.5 C_e VALUES FOR DIFFERENT OUTAGE DURATIONS IN SOUTH AFRICA (DUTKIEWICZ, 1991)

7.3 TESTING THE MODEL ON TWO MV NETWORKS

The program was tested on two demonstration networks used by Vladimiro (1998) and Levitin, Mazal-Tov and Elmakis (1994). Diagrams showing the lengths and average loads of the first network, that were used by Vladimiro, can be found in Appendix D. Connection points that were used for each section in the program are labelled on the diagram. This network comprises 53 sections and 27 load points. The total load installed is 8.25MW and the total line length is 24.3km. This network will be referred to as the Lebanon network.

The second test system was published by Levitin, Mazal-Tov and Elmakis (1995). It will be referred to as the Levitin network from this point onward. A diagram of this network showing the lengths and loads for each section are shown in Appendix E. This system consists of 96 sections with 52 load points. The total load installed is 15.76MW and the total line length is 40.9km.

Ahlschlager (2000) suggested the following figures for the costs of a sectionaliser:

- Cost of a sectionaliser = R39 000.
- Installation cost = R10 000.
- 14% overhead charge.

The total installed cost of a sectionaliser is thus R55 860. All tables and graphs shown in this section are those obtained for the Lebanon network. The results of the tests performed on the Levitin network are shown in Appendix E. The results for the Levitin network follow the same pattern as those obtained from the Lebanon network. Therefore, discussions about the Lebanon network apply to the Levitin network as well.

7.3.1 PERFORMANCE OF THE PBIL ON THE TWO NETWORKS

The two networks were tested using the PBIL. A failure rate of 0.2 faults/km/yr, an average repair time of 2 hours for each fault and the cost of unsupplied energy of 50 c/kWh were used. A value of 50 c/kWh was used instead of a lower value such as 25 c/kWh to take into consideration the inconvenience to a customer. The column heading "Annual Total Cost" that appears in the following tables is the sum the annual cost of unsupplied energy and the annual cost of the installed sectionalisers.

For the Lebanon network, the PBIL placed three sectionalisers at the sending end of sections 48, 36 and 10. For the Levitin network, the optimal number of sectionalisers was five and they were placed at the sending end of sections 72, 39, 36, 4 and 2. These sectionalisers have been placed in locations in the network that would be feasible in actual protection engineering practices.

I decided to remove the sectionaliser that had been placed by the program on section 36, for the Lebanon network, to see the effect that this would have on the annual cost of unsupplied energy. The annual cost of unsupplied energy increased from R25 141 to R30 288. This represents an increase of R5 147 in the annual cost of unsupplied energy. Removing the sectionaliser on section 36 would result in an increase, in the cost of unsupplied energy, of approximately R71 000 over a period of 30 years at a discount rate of six percent.

Figure 7.2 shows the performance of the PBIL for obtaining the optimal solution. The PBIL was run using a learning rate of 0.1, 100 generations, and 40 trial solutions. From the graph it can be seen that the optimal solution is obtained after approximately 40 generations. This is equivalent to only 1600 evaluations. This shows the strength of the PBIL. Few evaluations were required to obtain the optimal solution and no advanced decoding procedures are needed. The PBIL

performed the optimisation without the need of mutation, which if implemented, can improve the performance of the PBIL.

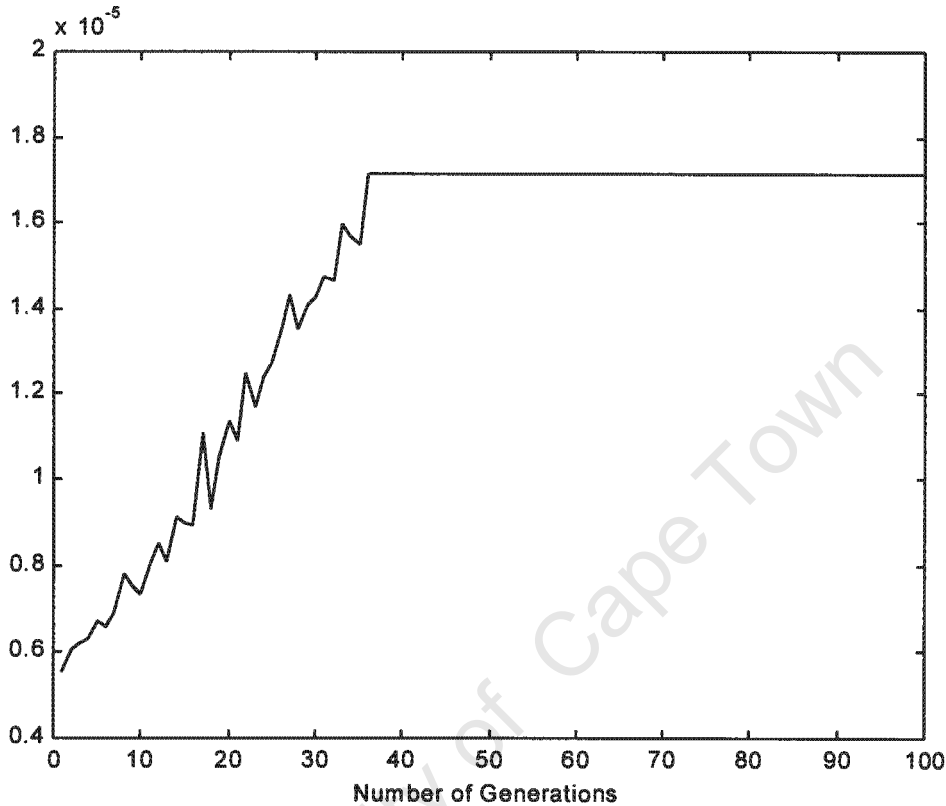


FIGURE 7.2 PBIL OBTAINING THE OPTIMAL SOLUTION FOR THE LEVITIN NETWORK

The remainder of this chapter shows the sensitivity of the model to network parameters such as failure rate, cost of unsupplied energy and discount rate.

7.3.2 EFFECT OF VARYING THE FAILURE RATE

The value for the failure rate per km for the Lebanon network was varied as shown in Table 7.6. The average repair time for a fault was kept constant at 2 hours and the cost of unsupplied energy was taken to be 50 c/kWh. The same

test was performed on the Levitin network and the results are shown in Appendix E.

FAILURE RATE	NUMBER OF SECTIONALISERS PLACED	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST OF ENS	ANNUAL TOTAL COST
[f/km/yr]		[hrs/yr/cust]	[R]	[R]	[R]
0.01	0	5.87	0	2 010	2 010
0.05	0	7.81	0	10 050	10 050
0.10	0	10.25	0	20 100	20 100
0.15	0	12.68	0	30 151	30 151
0.20	3	11.41	12 174	25 142	37 316
0.25	4	11.68	16 232	26 343	42 575
0.30	4	12.94	16 232	31 544	47 776
0.35	4	14.20	16 232	36 744	52 976

TABLE 7.6 EFFECT OF FAILURE RATE ON ANNUAL TOTAL COST AND SAIDI

From Table 7.6, it can be seen that the program suddenly places three sectionalisers when the failure rate is increased from 0.15 faults/km/yr to 0.2 faults/km/yr. I calculated the annual cost of unsupplied energy for the network without any sectionalisers using a failure rate of 0.2 faults/km/yr to be R40 201. Table 7.6 shows that the installation of three sectionalisers is justified because the annual cost of unsupplied energy is reduced by a considerable amount from R40 201 to R25 142.

The effect of the failure rate on the total annual cost of unsupplied energy was plotted and is shown in Figure 7.3.

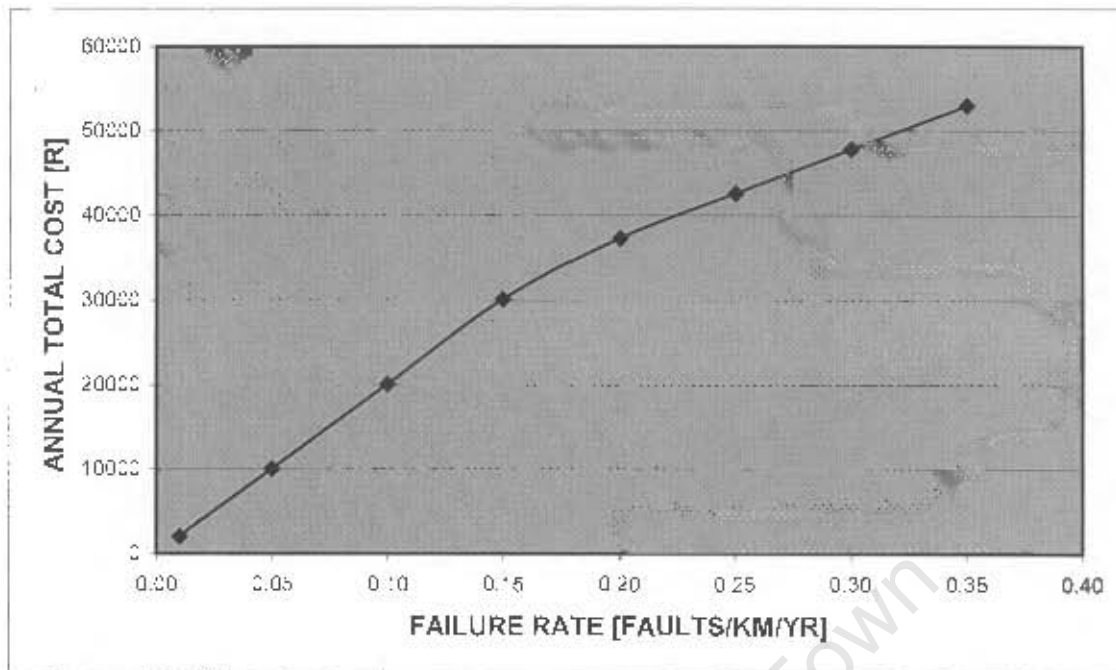


FIGURE 7.3 EFFECT OF FAILURE RATE ON THE ANNUAL TOTAL COST

As expected, the total cost increases for both the Levitin and Lebanon networks as the failure rate per km is increased because the number of customer outages has also gone up. Intuitively, it would be expected that there is a linear relationship between the total cost and the failure rate. However, Figure 7.3 shows that the total cost increases less than linearly as the failure rate is increased. This is because the placement of the three sectionalisers, after the failure rate has reached a value of 0.2 faults/km/yr, reduces the annual cost of unsupplied energy for the network.

As the failure rate increases it would be expected that the System Average Interruption Duration Index (SAIDI) would also increase for the same reason mentioned above. Figure 7.4 shows the effect of the failure rate on SAIDI in the Lebanon network. In section 3.3.1, it was shown that SAIDI is determined by dividing the sum of all customer interruption durations in a year by the number of customers served. The variation shown in the graph can be attributed to the fact

that SAIDI only applies to those customers that have actually been affected by an outage on a system. SAIDI decreases when the fault rate is increased to 0.2 faults/km/yr because the sectionalisers that have been placed reduce the number of customers that are affected by an interruption. The SAIDI value continues to increase once again as the failure rate is increased because the number of sectionalisers installed remains at 4 up to a failure rate of 0.35 faults/km/year.

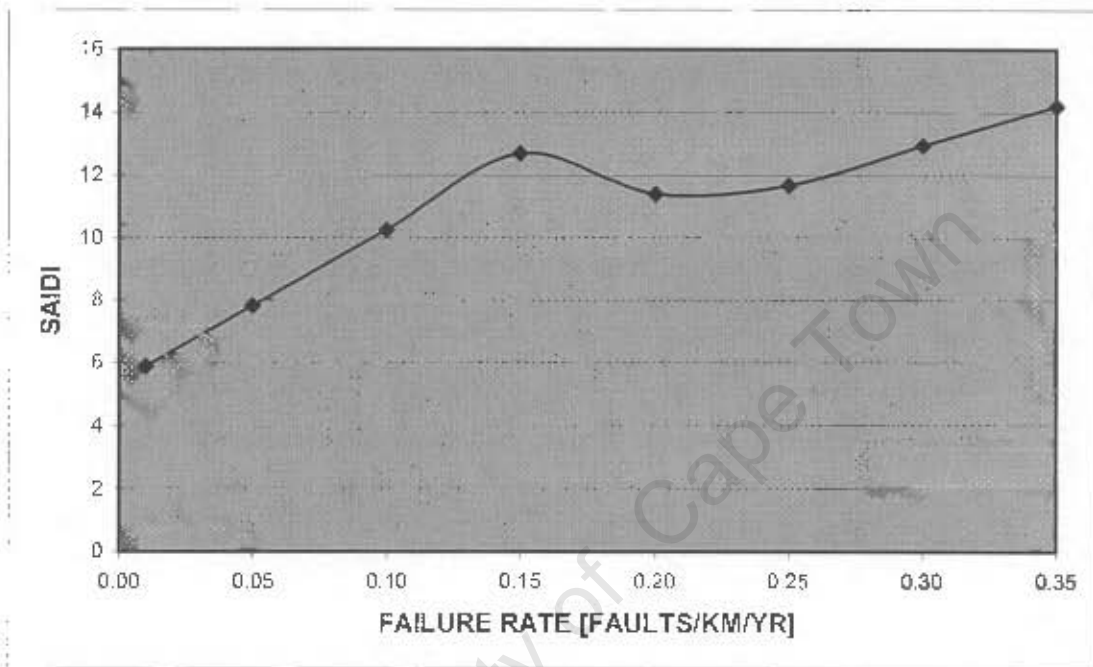


FIGURE 7.4 EFFECT OF FAILURE RATE ON SAIDI

7.3.3 EFFECT OF THE COST OF UNSUPPLIED ENERGY ON THE SYSTEM

The effect of increasing the unit cost of unsupplied energy on the total cost and SAIDI is shown in this section. The average repair time of a fault is kept constant at two hours and the failure rate is kept constant at 0.2 faults/km/year. The results obtained for the Lebanon network are shown in the Table 7.7 and those for the Levitin network can be found in Appendix E.

COST OF ENS	NUMBER OF SECTIONALISERS PLACED	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST OF ENS	ANNUAL TOTAL COST
[R/kWh]		[hrs/yr/cust]	[R]	[R]	[R]
0.25	0	20.50	0	20 100	20 100
0.50	3	11.41	12 174	25 142	37 316
1.00	4	7.73	16 232	42 285	58 517
1.50	5	6.36	20 290	57 851	78 141
2.00	5	5.91	20 290	77 134	97 424
3.00	7	5.06	28 406	106 380	134 786
4.00	7	4.83	28 406	141 839	170 245
5.00	7	4.70	28 406	176 801	205 207

TABLE 7.7 EFFECT OF CE ON THE ANNUAL TOTAL COST AND SAIDI

Graphs showing the effect of the unit cost of unsupplied energy on SAIDI and on the annual total cost of the network are shown in Figures 7.5 and 7.6.

As expected, the results show that as the unit cost of unsupplied energy increases so does the total cost of the network. The relationship shown in the graph in Figure 7.5 is not exactly linear because the increasing number of sectionalisers that are placed as the unit cost of unsupplied energy is increased cause the annual cost of unsupplied energy to increase to a value that is less than the expected value.

The value of SAIDI decreases as shown in Figure 7.6 when the cost of unsupplied energy per kWh is increased. With an increase in the number of sectionalisers, the average interruption duration for the customers must fall, hence the reduction of the value of SAIDI. The large increase in the unit cost of unsupplied energy justifies the program placing 7 sectionalisers when the cost of unsupplied energy is R5/kWh on the Lebanon network. For the Levitin network, the program places 19 sectionalisers when the unit cost of unsupplied energy is 5/kWh. More sectionalisers are placed on this network, as it is bigger than the

Lebanon network. The drop in the value of SAIDI due to an increased number of sectionalisers is demonstrated in the next section.

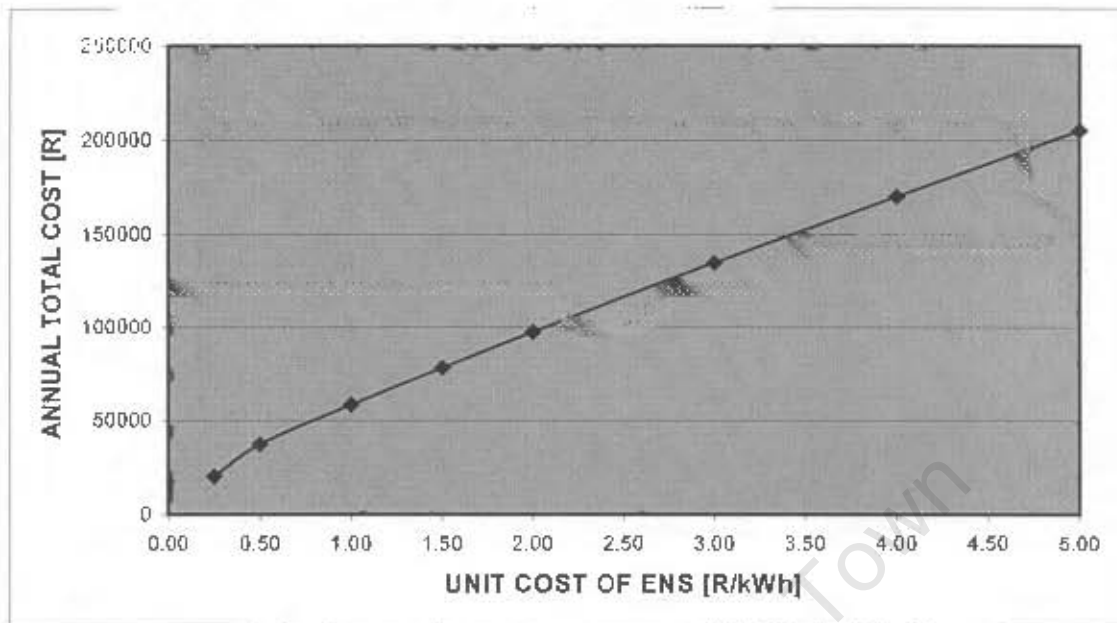


FIGURE 7.5 EFFECT OF CE ON ANNUAL TOTAL COST

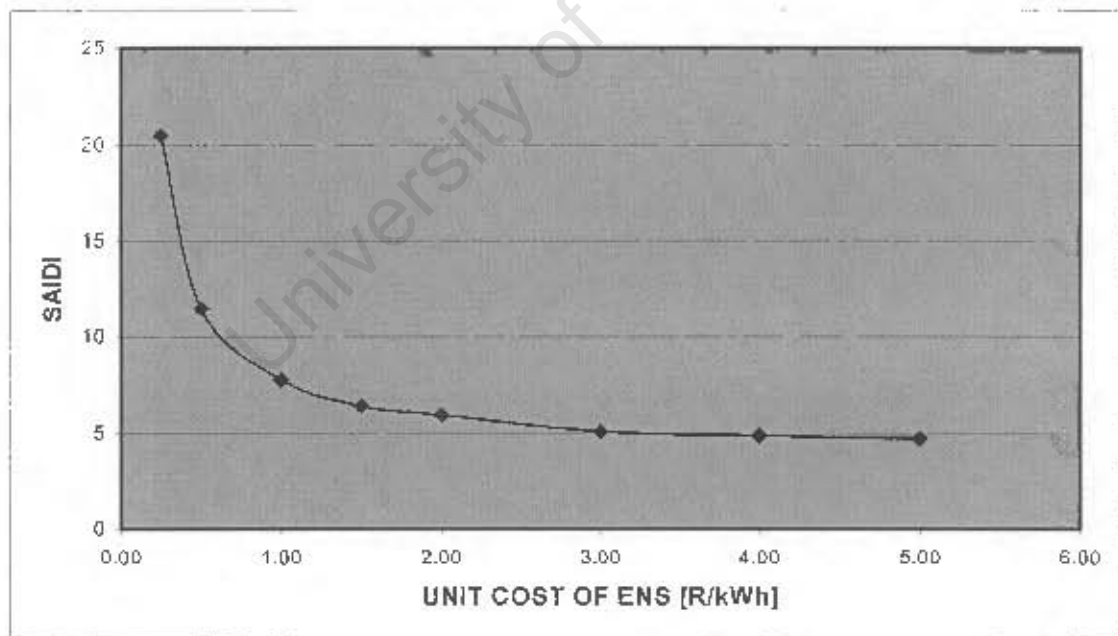


FIGURE 7.6 EFFECT OF CE ON SAIDI

7.3.4 EFFECT OF PLACING A SPECIFIED NUMBER OF SECTIONALISERS

In this section, two tests are carried out to see the effect of the user increasing the number of sectionalisers on the output of the program.

In the first test, the number of sectionalisers placed is increased and the effect it has on SAIDI and on the total cost is observed. The failure rate is kept constant at 0.2 faults/km/year, the average repair time of a fault at 2 hours and the cost of unsupplied energy at 50c/kWh. The values obtained for the Lebanon network are summarised in Table 7.8.

NUMBER OF SECTIONALISERS PLACED	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST OF ENS	ANNUAL TOTAL COST
	[hrs/yr/cust]	[R]	[R]	[R]
1	13.83	4 058	34 978	39 036
2	12.57	8 116	29 831	37 947
3	11.41	12 174	25 142	37 316
4	10.42	16 232	21 143	37 375
5	9.95	20 290	19 284	39 574
6	9.72	24 348	18 416	42 764

TABLE 7.8 EFFECT OF NUMBER OF SECTIONALISERS ON THE ANNUAL TOTAL COST AND SAIDI

Table 7.8 shows that as the number of sectionalisers on the network are increased, the annual total cost of the network initially decreases. This may be attributed to fewer outages being experienced by the customers. A critical point or absolute minimum is reached with 3 sectionalisers for the Lebanon network and 4 for the Levitin network. These are the optimum numbers of sectionalisers to place on the two networks. The addition of more sectionalisers after this critical juncture has been reached results in an increase in the annual total cost of the network. This is due to the reduction in the annual cost of unsupplied energy, that

result from placing sectionalisers, being less than the cost of installing more sectionalisers.

Table 7.8 shows that SAIDI decreases as the number of sectionalisers on the distribution system is increased. This is to be expected as the number of customers affected by outages is reduced when additional sectionalisers are placed.

In the second test the same procedure was used, but the average repair time of a fault was varied to range between 0.5 and 2 hours. This assumption was made to see the effect that this would have on the annual total cost of the network, since the time taken to locate a fault will decrease with an increasing number of sectionalisers (because it is easier to identify which area of the network is faulty from the sectionaliser that has isolated the fault). A saturation limit will be reached when an additional sectionaliser does not significantly reduce the average repair time but only causes an increase in the investment in the project.

The results obtained by using the assumption that has just been described are shown in Table 7.9. The repair time was decreased as the number of sectionalisers placed on the network by the user was increased.

NUMBER OF SECTIONALISERS PLACED	AVERAGE REPAIR TIME	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST OF ENS	ANNUAL TOTAL COST
	[hrs]	[hrs/yr/cust]	[R]	[R]	[R]
1	2.0	13.83	4 058	34 978	39 036
2	1.5	10.77	8 116	22 416	30 532
3	1.0	8.70	12 174	13 273	25 447
4	0.9	7.65	16 232	10 713	26 945
5	0.7	6.98	20 290	7 904	28 194
6	0.5	6.46	24 348	4 987	29 335

TABLE 7.9 EFFECT OF NUMBER OF SECTIONALISERS PLACED ON ANNUAL TOTAL COST AND SAIDI

If the reduction in repair time due to the increase in number of sectionalisers placed is taken into consideration, then there is a larger saving in the total annual cost of the network. This is clearly seen by comparing the values for annual cost of unsupplied energy shown in Table 7.8 with those shown in Table 7.9. As expected, reducing the average repair time, as the number of sectionalisers on the network is increased, reduces the annual cost of unsupplied energy for the network by a greater amount (than just using a constant repair time with an increasing number of sectionalisers).

If the variation in repair time is considered, then the value of SAIDI follows the same pattern as in the first test, but this value decreases more rapidly since the average outage time is reduced.

In practice, the optimal number of sectionalisers may not always be used. The utility might place one sectionaliser less than the optimum since all utilities try to avoid unnecessary upfront-costs. For the Lebanon network, the annual total cost to the utility for installing three sectionalisers, which are the optimum number obtained from the program is R37 316, whereas for two sectionalisers, the annual cost is R37 947. The difference of R631 is minimal in this case and thus a utility would consider it feasible to place two sectionalisers instead of three.

The difference in the total annual cost of the network in Table 7.8 does not seem to vary by a large amount over the entire range when the number of sectionalisers on the network is increased. The difference between placing three and four sectionalisers is only R59. This can be attributed to the values of the parameters, such as the unit cost of unsupplied energy and the annual cost of a sectionaliser, that were used when testing the network. To demonstrate this, I increased the unit cost of unsupplied energy from 50c/kWh to R3/kWh. The difference in the total annual cost between placing the optimum number of sectionalisers and one more than the optimum increased to R3831.

7.3.5 EFFECT OF DISCOUNT RATE ON TOTAL COST OF THE NETWORK

In this section the importance of choosing a suitable value for the discount rate is shown. The discount rate is the rate used to determine the present value of future cash flows. The failure rate was kept at 0.2 faults/km/yr, the repair time per fault occurrence as 2 hours and the cost of unsupplied energy as 50 c/kWh. The results obtained for the Lebanon network are illustrated in Figure 7.7. The results for the Levitin network are shown in Appendix E.

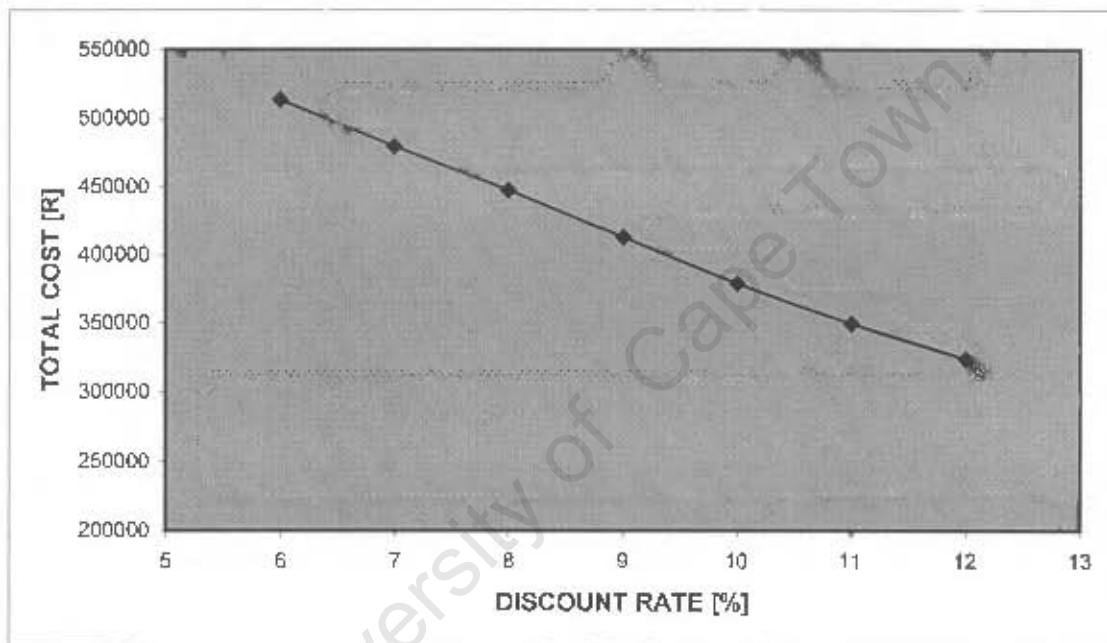


FIGURE 7.7 EFFECT OF DISCOUNT RATE ON THE TOTAL COST OF THE NETWORK

From the graph, it can be seen that as the discount rate is increased, the present value of the total cost of the system decreases. Discounting cash flows with higher discount rates always results in the present value decreasing. This can easily be seen if we look at the following equation (Khatib, 1997):

$$PV = M \left(\frac{1}{r} - \frac{1}{r(1+r)^n} \right)$$

Where PV = present value.

M = annual cash flow.

r = discount rate.

n = number of years to discount the costs over.

From this equation, it can be seen that as soon as r is increased, the present value will decrease. From Figure 7.7, it can be seen that if the value of the discount rate used is very high then the present value of the future cash flows is low (it is approximately R320 000 at a discount rate of 12% compared to R510 000 at a discount rate of 6%). Therefore, a low discount rate of about 6% needs to be used if the prediction of future cash flows is to be of use. The discount rate is also referred to as the "opportunity cost of capital" (Khatib, 1997:36) because if there were no profit from an investment then it would be better to undertake another investment that the investor will profit from.

7.4 FUTURE ENHANCEMENTS TO THE COMPUTER MODEL

A few suggestions for future work to improve the program are given in this section.

1. The model that has been implemented uses the basic PBIL. Work has been done in recent years on improving the performance of the PBIL. The extensions that have been added to the standard PBIL have made it evolve into a more powerful algorithm. Using this modified version would improve the efficiency of the program. The run time of the program would also improve if the program were to be coded in a language such as C.
2. The program would be more user friendly if a programming language such as Visual Basic were used. Visual Basic has the facility for graphical interfacing.

3. Increasing the number of sectionalisers placed results in a decrease of the overall average repair time of the system. A selection of graphs with decaying curves showing the decrease in repair time due to an increasing number of sectionalisers could be added to the program. The program would then reduce the average repair time as the number of sectionalisers was increased according to the graph selected.

7.5 CONCLUSION

These tests have shown the sensitivity of the results to parameters such as the failure rate, the unit cost of unsupplied energy and the discount rate and that this model should therefore be used with caution, and with an awareness of the effects of changes in these parameters.

Chapter 8 CONCLUSIONS

The main conclusions arising from my research and testing are that:

1. The PBIL algorithm was successfully implemented to determine the optimum number and placement of sectionalisers on a medium-voltage distribution network.
2. The PBIL algorithm always converges to an optimal solution, regardless of the size or configuration of the distribution network.
3. The PBIL algorithm is a superior tool to the genetic algorithm for determining the optimum number and placement of sectionalisers. This is demonstrated by the fact the computer model designed is based on a very simple version of the PBIL algorithm, yet it obtained the same results as an advanced genetic algorithm-based method using a fewer number of evaluations.
4. The computer model designed is sensitive to the following parameters:
 - The failure rate
 - The unit cost of unsupplied energy
 - The average repair time of a fault
 - The discount rate

The tests carried out show that small variances in these parameters lead to significant changes in the solutions obtained. Thus, it is imperative that system planners use accurate data in utilizing this method to find the optimal number and location of sectionalisers.

5. The PBIL algorithm is a simple and powerful algorithm that also has the capabilities of solving various types of other power system optimisation problems.

REFERENCES

- Ahlschlager R, 2000. Protection department, Eskom East London. Personal communication.
- Anderson PM, 1999. *Power System Protection*. IEEE.
- ANN, 1999. Website <http://psych.utoronto.ca/~reingold/courses/ai/nn.html>, accessed on 27/11/2000.
- ANSI C37, 1994. *IEEE Standard Service Conditions and Definitions for High Voltage Fuses*. American National Standards Institute, New York.
- Baluja S, 1994. Population Based Incremental Learning: A Method for Integrating Genetic Based Function Optimization and Competitive Learning. *Internal paper CMU-CS-95-193*, Carnegie Mellon University.
- Bayliss CR, 1999. *Transmission and Distribution Electrical Engineering*. 2nd Edition MPG Books Limited, Bodmin.
- Billinton R and Allan RN, 1996. *Reliability Evaluation of Power Systems*. 2nd Edition, Plenum Press.
- Billinton R and Lakhanpal D, 1996. Impacts of demand-side management on reliability cost/reliability worth analysis. *IEE Proc. GenerTransmDistrib*, Vol 143, No. 3, May.
- BS, 1975. Current-limiting Fuses. *British Standards*, 2692:Part1.
- Coopers Power Systems, 1990. *Electrical Distribution-System Protection*. 3rd edition Protection manual, USA.
- Darveniza M, 1999. Guidelines for the Installation of Auto-Reclosers and Sectionalisers. *Internal paper*, Queensland University, Australia.
- Davis L, 1996. *Handbook of Genetic Algorithms*. International Thomson Computer Press.
- Dingley C, 1998. Lecture notes from EEE489F, University of Cape Town.
- EPRI, 1981. Research into Distribution Reliability and Risk Analysis Models. *Electric Power Research Institute*, Project 1356, Final report, September.

- Eskom, 2000. Rural Reticulation Protection: Section 0 – Network Philosophies. *Eskom National Guide SCSAGAAQ0*.
- ETA, 1995. *Power System Protection*. Electricity Training Association, Vol.1 Principles and Components, UK.
- Evonet, 1999. Website <http://www.evonet.dcs.napier.ac.uk>, accessed on 25/11/2000.
- GEC, 1987. *Protection Relays Application Guide*. GEC ALSTHOM T&D Protection and control Limited, 3rd Edition, England.
- Glover JD and Sarma M, 1994. *Power System Analysis and Design*. International Thomson Publishing, 2nd Edition, USA.
- Goldberg D, 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, Reading, MA.
- Gonen T, 1988. *Modern Power System Analysis*. John Wiley And Sons, California.
- Greene J, 1998. Lecture Notes from EEE 485F, University of Cape Town.
- He Y, Anderson G and Allan RN, 1999. Determining Optimum Location and Number of Automatic Switching Devices in Distribution Systems. *Paper BPT99-24-17 accepted for presentation at IEEE Power Tech'99 Conference*, Budapest, Hungary, Aug 29-Sept 2.
- Hertz J, Krogh A and Palmer G, 1993. *Introduction to the Theory of Neutral Computation*. Addison Wesley.
- Holland J, 1992. *Adaption in Natural and Artificial Systems*. MIT Press, Cambridge, MA.
- Jonnavithula S and Billinton R, 1996. Optimal Switching Device Placement in Radial Distribution Systems. *IEEE Trans. on Power Delivery*, Vol. 11, No. 3, July.
- Khatib H, 1997. *Financial and Economic Evaluation of Projects in the Electricity Supply Industry*. IEE, Stevenage, UK.
- Khylatash G, 1995. *Population Based Incremental Learning Algorithm*. BSc thesis, Department of Electrical Engineering, UCT, November.

- Lakervi E and Holmes EJ, 1989. *Electricity distribution network design*. Peter Peregrinus Ltd, London.
- Leith AE and Sullivan B, 1999. Economic Optimisation Of An Australian Rural Electricity Network Using Automatic Circuit Reclosers With Remote Monitoring And Control. *Great Southern Energy Internal paper*, NSW, Australia.
- Levitin G, Mazal-Tov S and Elmakis D, 1994. Optimal Sectionaliser Allocation in Electric Distribution Systems by Genetic Algorithms. *Electric Power Systems Research*, Vol 31, May.
- Luth J, 1991. Four Rules to help locate circuit protective devices. *Electrical World*, Vol 205, No. 8, Aug.
- Lythall RT, 1972. *JSP Switchgear Book*. 7th Edition, Butterworth and Co Limited.
- Miller RH and Malinowski JH, 1994. *Power System Operation*. 3rd Edition, McGraw-Hill.
- Miranda V, Srinivasan D and Proenca LM, 1996. Evolutionary Computation in Power systems. *Proceedings of PSCC96*, Dresden, Germany, August.
- Reyrolle Switchgear. Pamphlet 1370/8-66.
- Ridley M, 1996. *Evolution*. 2nd Edition, Blackwell Science, USA.
- Seaman B, 1995. *Population-based Incremental Learning: A guided Stochastic search Method*. BSc thesis, Department of Electrical Engineering, UCT, November.
- Sundhararajan S and Pahwa A, 1994. Optimal Selection of Capacitors for radial distribution systems using a genetic algorithm. *IEEE Trans. on Power Systems*, Vol. 9, No. 3, August.
- Vladimiro M, 1998. Reliability and decision making even if with unreliable data. *ENVIMA conference*, Lebanon, June.
- Whitley D, 1993. *Foundations of Genetic Algorithms*. Morgan Kaufman Publishers.
- Wright A and Newberry PG, 1982. *Electric Fuses*. Peter Peregrinus Limited, London.

APPENDIX A

DISTRIBUTION SWITCHGEAR

University of Cape Town

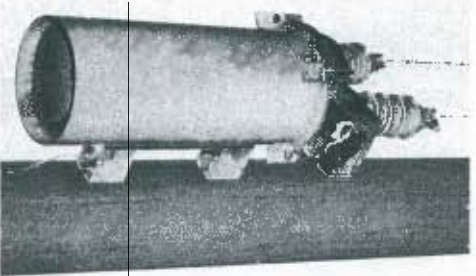


Figure 17A2.
Pole-top installation of single-phase recloser.

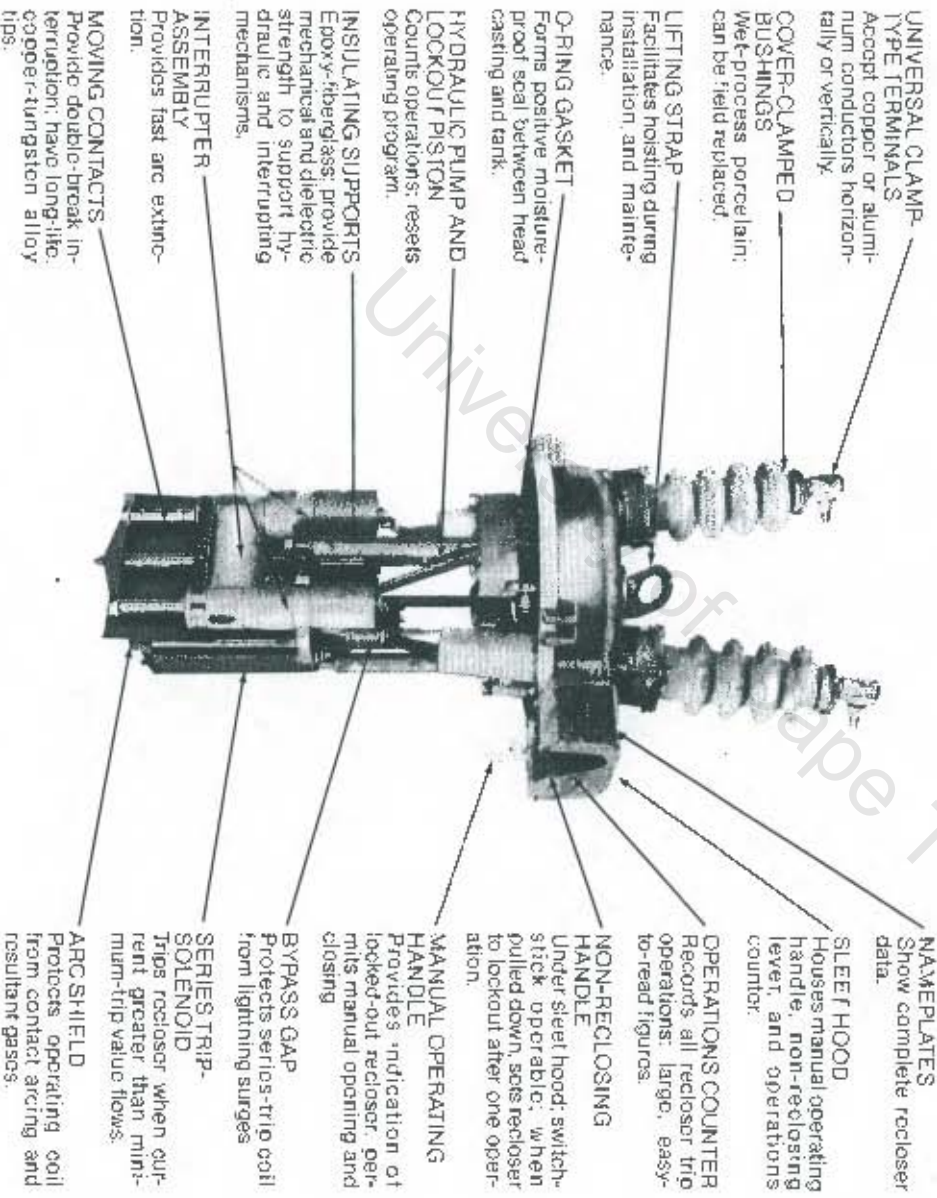


Figure 18A2.
Unlanked single-phase recloser.

Detail



Vacuum Bushes in Porcelain



Umbilical Plug and Socket



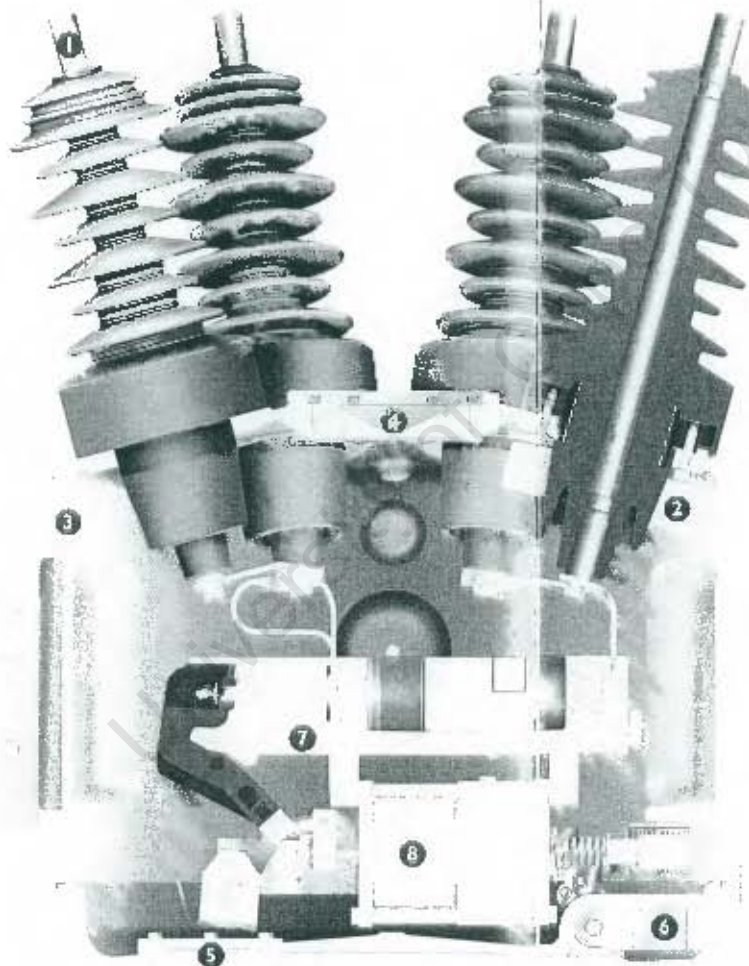
Position Indicator



Magnetic Actuator

Main Features

- 1 Single piece, aluminium or copper cored EPDM rubber bushings, with grooves to take optional wildlife guards/HV boots.
- 2 Current transformers are mounted within the tank's controlled environment, while capacitive voltage dividers moulded into the bushings on both sides of the GVR provide voltage signals for use by the Paraset.
- 3 Aluminium housing with lightweight, moulded baseplate, secured by stainless steel bolts and incorporating rubber O-ring seals.
- 4 Optional pressure-relief disc, to comply with IEC 298 Appendix AA, offers the highest levels of safety.
- 5 Mechanical ON/OFF position indication visible through clear viewing window from ground level.
- 6 Hook stick-operated manual trip and lockout control.
- 7 A single moulding supports the three phase vacuum interrupter assembly, magnetic actuator mechanism and one-piece drive beam.
- 8 The single coil magnetic actuator is based on a solenoid plunger, held in the tripped or closed position by a permanent magnet.



Single Coil Magnetic Actuator

The actuator coil is energised in one direction to 'power close' the GVR and in the opposite direction to open it by de-latching the holding force. This is a unique feature of the single coil actuator design used in the GVR and ensures reliable tripping operation under all battery conditions and even for manual trip.

Closing

The bi-stable design ensures that the plunger is held back in the open position (1) until the solenoid current rises above the level required to guarantee closure. Once the holding force is overcome (2), the circuit breaker closes positively (3), due to the stored energy in the solenoid and permanent magnets.

Tripping

The solenoid is energised in the reverse direction (4) to overcome the magnetic hold-on force and de-latch the actuator. Opening is then completed by the energy stored during the closing stroke in the contact pressure and opening springs and is completely independent of the power supply during electrical opening, and of the operator during manual opening. The energy required to trip is approximately 1/30th of that required to close.

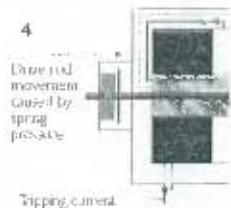
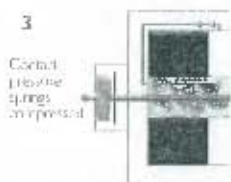
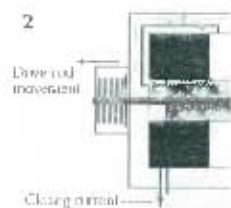
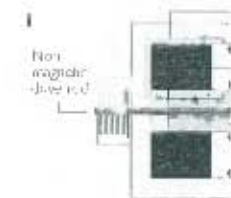


ABB vacuum reclosers: Reliability, Lower Total Costs, Maximum Flexibility

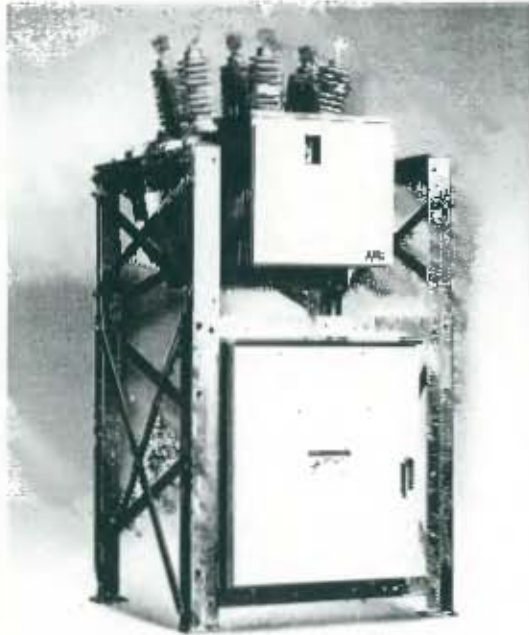
ABB vacuum reclosers offer all the benefits of ABB vacuum breakers: lower life cycle costs through lower installation, maintenance and operating costs; and safety, reliability, environmental acceptability, and extended duty cycles. In addition, ABB vacuum reclosers offer the optimum

Application flexibility, with units available in a wide range of ratings able to meet virtually any conditions on T&D systems.

Operating flexibility to reduce both initial and operating costs. ABB reclosers are operable regardless of the direction of power flow, since bushings 2-4-6 or 1-3-5 can be used as line or load. Fewer cable connections lower the initial cost of the substation, eliminating the need to interchange line and load connections reduces operating costs.

Installation flexibility, available with frames and control cabinets that enable installation on the substation pad or on a pole. Control cabinets can be installed on the rear or left side of the substation frame for even more flexibility. And frame height is adjustable to match the height of existing bus.

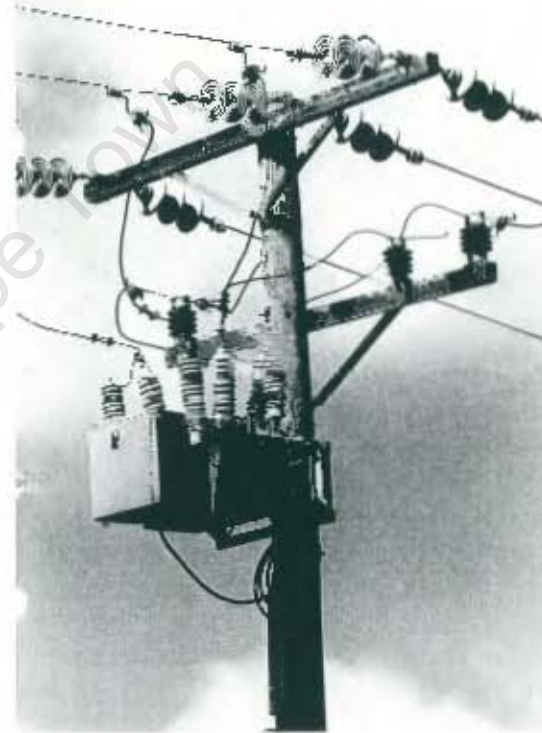
Type ESV Vacuum Recloser



Type ESV Vacuum Recloser



Type ESVA Vacuum in Air Recloser



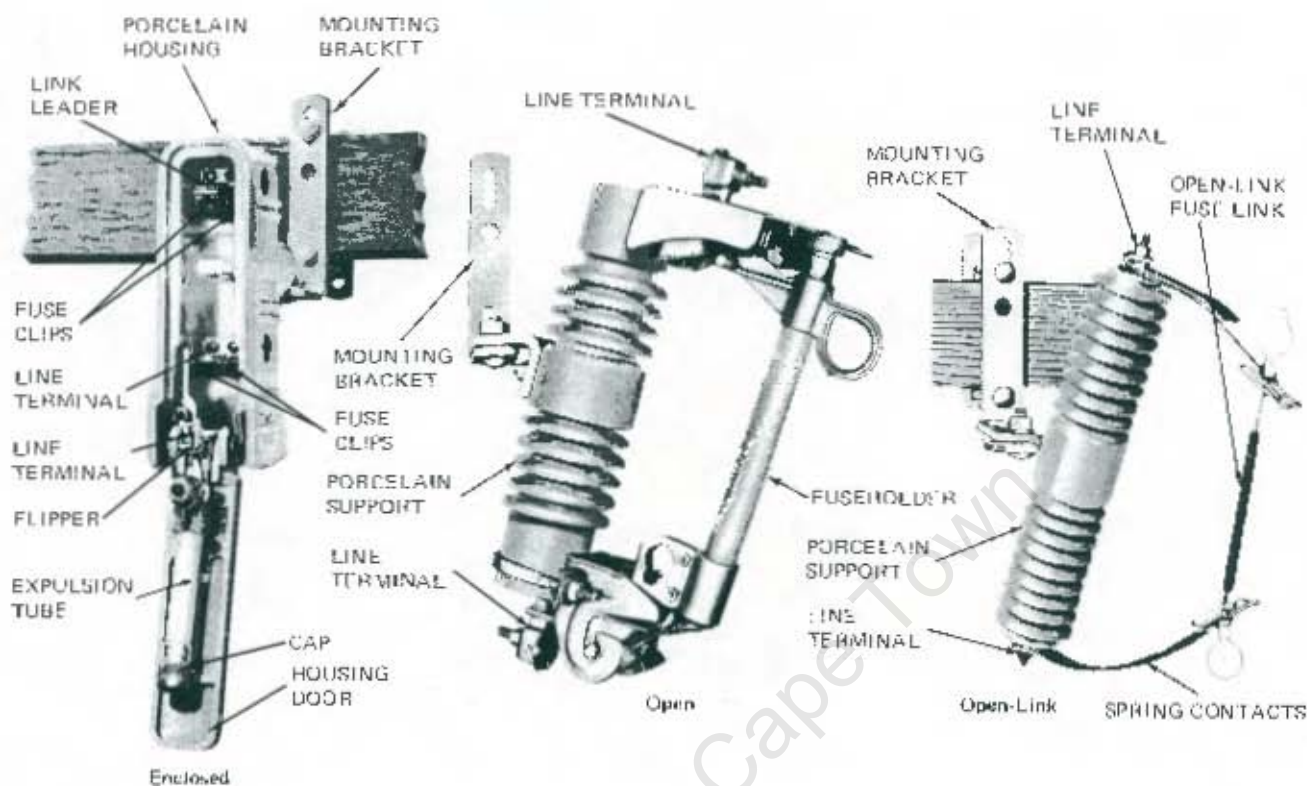


Figure 4A2. Distribution-type fuse cutouts.

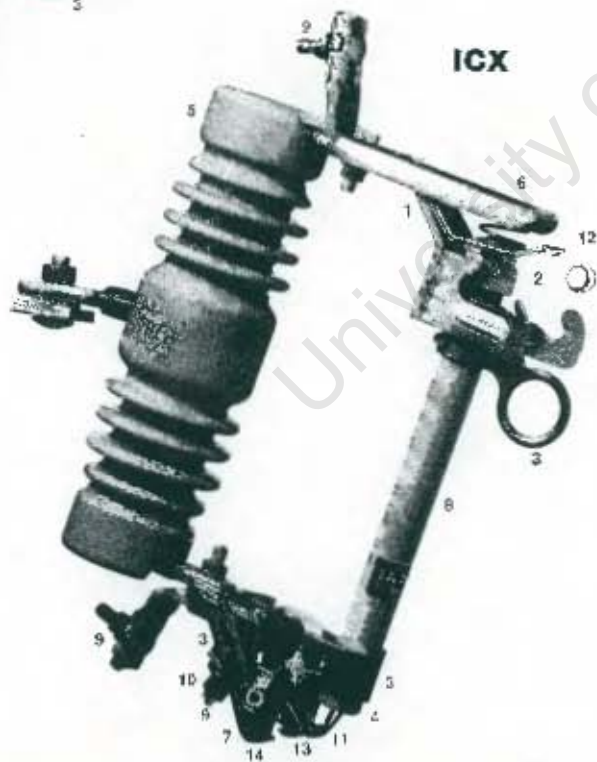


Figure 5A2. Vacuum fuse.

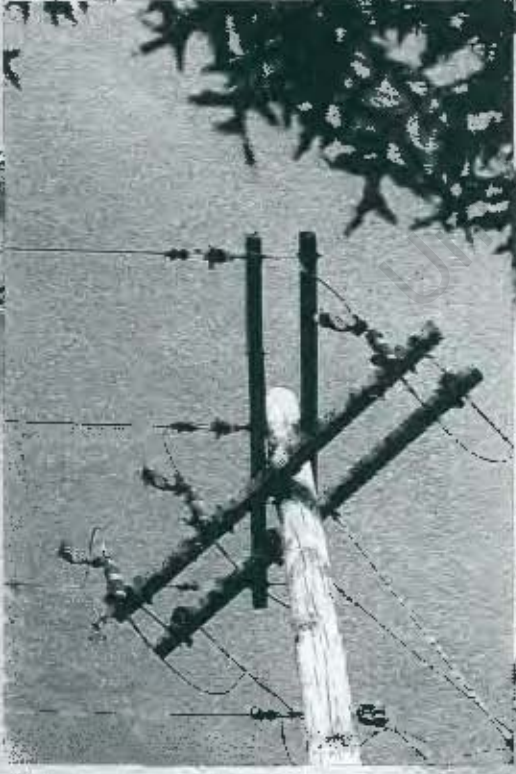


Figure 6A2. Current-limiting fuses.

**It's The Details That
Distinguish ABB Cutouts . . .
Features**



- 1 Continuous current paths
- 2 Universal contacts
- 3 Bronze castings
- 4 Stainless steel pivot pins
- 5 Porcelain Insulators
- 6 Galvanized steel sheet shield
- 7 Stainless steel flipper
- 8 Durable fuse tube
- 9 Quick connect terminals
- 10 Recessed threads on link nut
- 11 Dual direction lifting eye
- 12 Easy to operate loadbreak hooks
- 13 Fuse links with controlled tension
- 14 Locking cam



Singles and three phase, loop or lateral lines are sectionalized by ABB cutouts during maintenance or contingency operations.

ABB cutouts protect capacitor banks against overcurrents, visibly indicate when equipment is energized, and interrupt capacitive currents.



Type-OYT Recloser



Type-OYS Sectionalizer

Reyrolle type - OYT reclosers are available in two sizes, one with closing coils for systems of up to 4.4 kV and the other for systems up to 25.8 kV. The main features are as follows:

High-speed clearance of fault current: resulting in reduced damage at the point of fault.

Continuity of operation — up to 200 successful reclosures have been recorded in service without attention.

Variable duty-cycle and tripping time with up to 4 clearances in succession with any combination of instantaneous followed by delayed trips.

Sensitive instantaneous earth-fault with minimum tripping value of 20 amps.

Long maintenance intervals — 3 years or 100 operations recommended.

Reyrolle type - OYS sectionalizers are available for systems up to 14.4 kV.

The main features are as follows:

Precise mechanical co-ordination with recloser.

Variable setting for up to 3 pulses of fault current.

Line switching — load-breaking and fault making.

Sensitive earth fault protection: co-operates with that of the recloser.

Co-operation with multi-shot reclosing circuit-breakers.

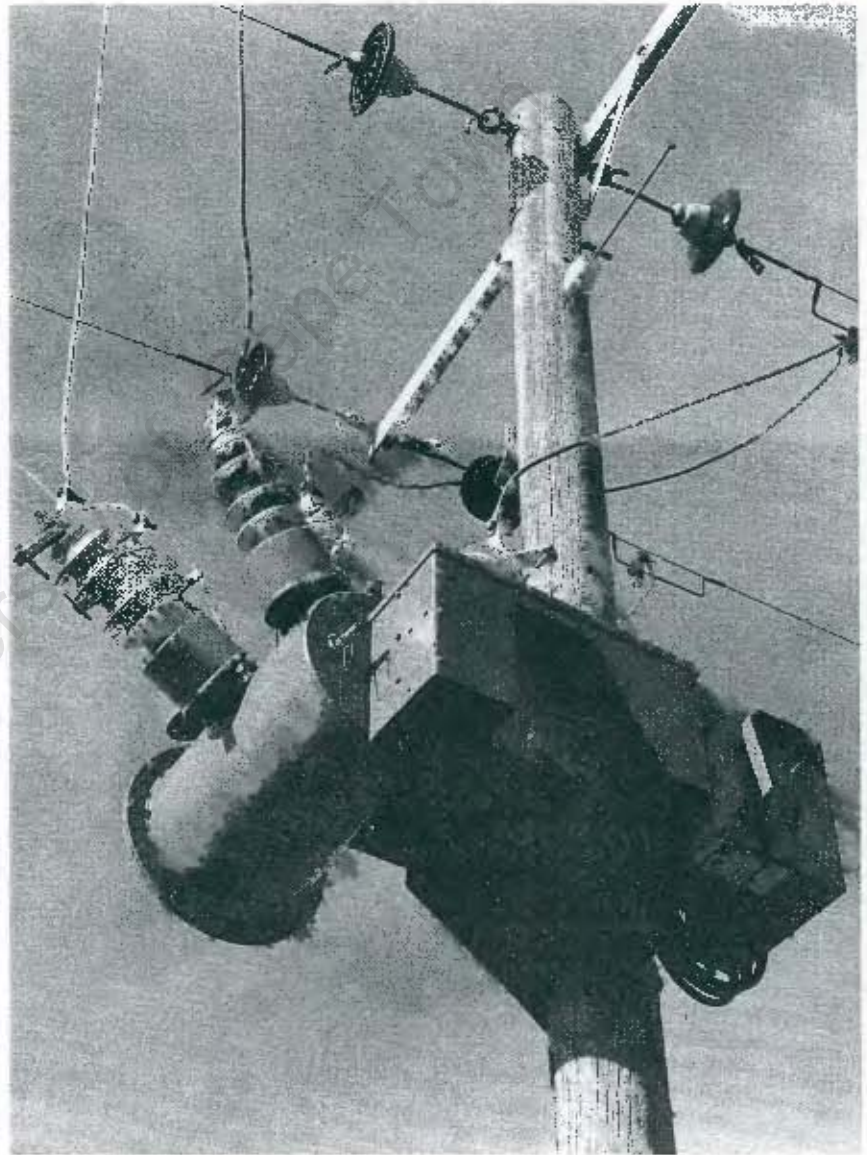
Maintenance negligible — 3 year oil check.

Type PMS Three Phase SF6 Pole Mounted Sectionalizer

Description

The Hawker Siddeley Switchgear design of SF6 Pole Mounted Sectionalizers was introduced to complement our well established range of PMR SF6 Pole Mounted Auto-Reclosers.

Sectionalizers are designed to co-ordinate with auto-reclosers such as the PMR. They are essentially Fault Make, Load Break Switches which have an ability to count pulses of fault current open during the dead time of the controlling auto-reclosers or circuit breaker. If the pulses of fault current cease before the preset number, the Sectionalizer will reset ready to start counting again should another fault occur.



PMS Sectionalizer fitted with an Optional Lock in Switch

AUTOLINK

Automatic Electronic Sectionalizer

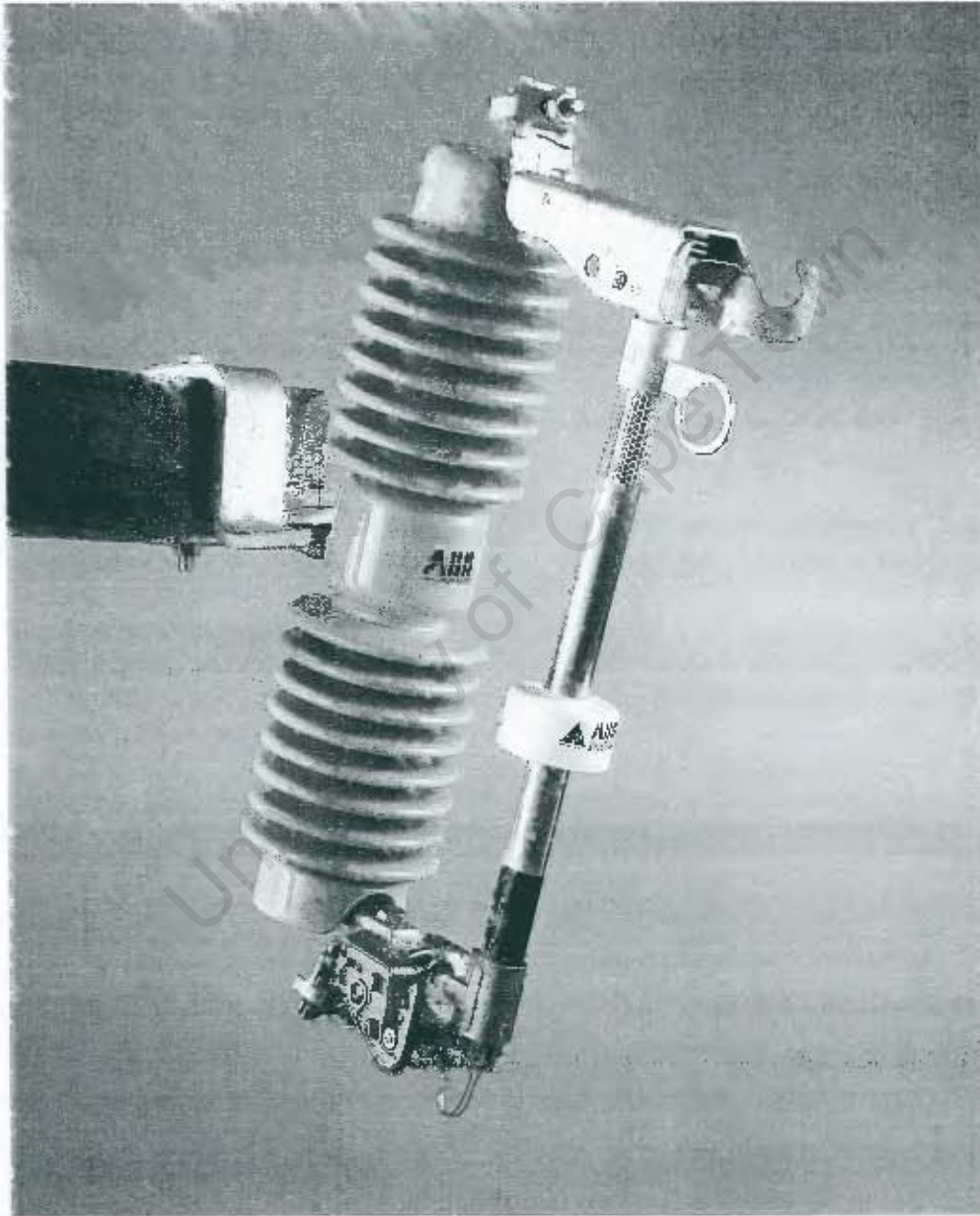
System Voltage: Up to 36kV

Lightning Impulse (BIL): 95, 125, 150 and 170kVp

Rated Current: 250A

Actuating Current: 5 to 224A

Fault Counts: 1 to 7



Manufactured in Australia by ABB T&D Limited
A.C.N. 300 189 588

Covered by Australian Patent No. 558 283

ABB

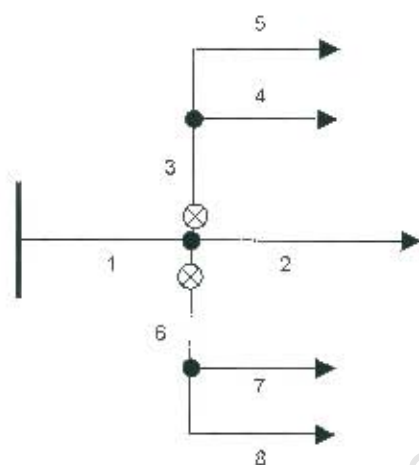
APPENDIX B

INPUT DATA FOR THE PROGRAM

University of Cape Town

EXAMPLE OF INPUT DATA REQUIRED TO RUN THE PROGRAM

The input data that is required to run the program is demonstrated using the simple network shown below. The ⊗ symbol is used to show the places on the network where sectionalisers have been installed.



The following information for the network would be entered into the network data array.

Section Number	Section Predecessor	Length [km]	Average Repair Time [hrs]	Outage Time	Sectionaliser	Load [kw]
1	0	3	2	-	0	0
2	1	4	2	-	0	100
3	1	2	2	-	1	0
4	3	1	2	-	0	150
5	3	1	2	-	0	250
6	1	2	2	-	1	0
7	6	1	2	-	0	100
8	6	1	2	-	0	150

The program automatically inserts the value entered by the user for the average repair time for a fault into the fourth column. The outage column has been left blank as the values in this column are calculated and not entered by the user. The user needs to add "1's" and "0's" in the sectionaliser column if the user wishes to find the annual cost of unsupplied energy for user specified locations, otherwise, the program will automatically place sectionalisers when searching for the optimal solution.

The user will be asked to enter:

1. The annual failure rate (faults/km/annum).
2. The average repair time for a fault.
3. The unit cost for unsupplied energy (R/kWh).
4. Installed cost of the sectionaliser (R).
5. The number of years to discount the cost over.
6. The discount rate.

APPENDIX C

MATLAB CODE FOR THE PROGRAM

University of Cape Town

```
fprintf('This program determines the optimal number and placement of sectionalisers on a power system.\n\n');
```

```
program_menu
```

```
number=input('\n\nEnter choice 0, 1, 2, 3 or 4 now: ');
```

```
while (number ~= 0 )
```

```
    switch number
```

```
    case 0
```

```
        break
```

```
    case 1
```

```
        calculate_ens
```

```
    case 2
```

```
        costens_minimisation
```

```
    case 3
```

```
        specific_switch
```

```
    case 4
```

```
        ens_minimisation
```

```
    ...
```

```
    otherwise
```

```
        disp('You have entered an invalid number')
```

```
        number=input('\n\nEnter choice 0, 1, 2, 3 or 4 now: ');
```

```
end
```

```
program_menu
```

```
number=input('\n\nEnter choice 0, 1, 2, 3 or 4 now: ');
```

```
end
```

University of Cape Town

% This is the menu for the program. User selects the desired option

```
fprintf('\nEnter 1 if you would like to calculate the ENS for a specific arrangement of
sectionalisers.');
```

```
fprintf('\nEnter 2 if you would like to find the optimum number of sectionalisers for a
system.');
```

```
fprintf('\nEnter 3 if you would like the optimal placement of a specified number of
sectionalisers for the power system.');
```

```
fprintf('\nEnter 4 if you would like to see the placement of sectionalisers on the system
with unlimited cost.');
```

```
fprintf('\nEnter 0 to exit.');
```

University of Cape Town

```
function closest_upstream_switch = calc_cpp(switch_data,node)
%closest_upstream_switch = calculate_cpp(switch_data,node)
%This function finds the closest upstream sectionaliser

closest_upstream_switch=0;
if ( switch_data(node,6)~= 0 )
    closest_upstream_switch = switch_data(node,1);
else
    if(switch_data(node,2)==0)% check so that the matrix not exceeded e.g. node 8
        closest_upstream_switch = switch_data(1,2);
    else
        leaf=find(switch_data(:,1)==switch_data(node,2));
        closest_upstream_switch = calc_cpp(switch_data,leaf);
    end
end

end;
```

University of Cape Town

```
function failure_rate = calculate_failure_rate(switch_data,node)
%failure_rate = calculate_failure_rate(switch_data,node)
%This function calculates the total failure rate downstream a section.
%switch_data is a matrix containing the distribution system data.
%node is the section number which the failure rate is being determined for.

failure_rate=0;

leaves = find(switch_data(:,2)==node);

for i= 1:length(leaves)
    %failure_rate = failure_rate + switch_data(leaves(i),3)*switch_data(leaves(i),6) +
    calculate_failure_rate(switch_data,leaves(i));
    failure_rate = failure_rate + switch_data(leaves(i),3) +
    calculate_failure_rate(switch_data,leaves(i));
end;
```

University of Cape Town

```
function load = calc_load(switch_data,node)
%load = calculate_load(switch_data,node)
%This function finds the load interrupted by the disconnection
%of a particular section

load=0;
if ( switch_data(node,4)~= 0 )
    load = switch_data(node,4);
else
    leaves = find(switch_data(:,2)==node);
    for i= 1:length(leaves)
        load = load + calc_load(switch_data,leaves(i));
    end;
end;
```

University of Cape Town

```

%This function places the optimal number of sectionlisers on the system
%with no cost restraints

fitrec=[];           %Array to store values for graph plotting
bestever=-inf;
maxgen=100;         %Number of generations for the PBIL

Loads the data file that contains system data
!notepad sysldat.dat
disp('Press any key to continue after saving the data file.....')
pause

load sysdata.dat    %Loading of data file in Notepad which the
A=sysdata;          %planner uses to enter the system data

%load sysdata.m     %Data files that already contain system data for 3 networks
%A=sysdata;         %and are used for testing the program
%load lebanon.m
%A=lebanon;
%load network.m
%A=network;

failure_rate = input('Enter a value for the failure rate per km: ');
repair_rate = input('Enter a value for the average repair time for a fault (hours): ');
Ce = input('Enter a value for the cost of supplied energy(per kWh): R ');

%Creation of matrix that holds all data required for calculator of ENS
for i=1:length(A(:,1))
    A(i,3)=A(i,3)*failure_rate;    %Failure rate per section
    A(i,4)=repair_rate;            %Average Repair time
    A(i,5)=A(i,3)*A(i,4);          %Outage time per section
    A(i,7)=A(i,7)*Ce;              %Cost energy supply interruption per section
end

%Calculation of cost of system's unsupplied energy without sectionalisers
Load = sum(A(:,7));
for i=1:length(A(:,1))
    TOTENS(i,1) = A(i,3)*A(i,4)*Load;
end
Tot_ENS=sum(TOTENS(:,1));

B = zeros(maxgen,4);           %Matrix holds all the output values
C = zeros(length(A(:,1)),3);   %Matrix that holds the final sectionaliser placement

%Calculation of energy supply interruption per hour of each section
for i=1:length(A(:,1))
    load_after = calcu_load(A,i);
    A(i,7)=load_after;
end

%Creation of probability vector initialised to 0.5
PV=0.5*ones(1,length(A(:,1)));

%Calculation of fitness values for trial solutions generated
for g=1:maxgen
    bestfit=-inf;
    for t=1:20
        ts=rand(size(PV))<PV;    %Generation of random trial solutions
        A(:,6)= ts';              %Replacement of switch column by trial solution

        %minimising ens is same as maximising 1/function
        f = 1/minsysens1(A,Load,Ce);

        %determination of the best solution in each generation
        if f>bestfit
            bestfit=f;
            bestsol=ts;
        end
    end
end

```

```

end

    %Equation to adjust the probability vector
    PV=0.9*PV+0.1*bestsol;
    PV=PV-0.005*(PV-0.5);
    fitrec=[fitrec,bestfit];

%Determination the best trial solution ever
if bestfit>bestever
    bestever=bestfit;
    besteversol=bestsol;
end

%Output to the screen
fprintf('Maxgen: %g      gen: %g      best: %g      bestever:
%g\n',maxgen,g,1/bestfit,1/bestever)
B(g,1) = maxgen;
B(g,2) = g;
B(g,3) = 1/bestfit;
B(g,4) = 1/bestever;

end

wklwrite('ens_minimisation.xls',B)      %Writes the output to an excel file.

%Plots a graph of fitness vs the number of generations
plot(fitrec)
xlabel('Number of Generations')
ylabel('Fitness')

besteversol
%Calculates the number of sectionalisers in the best solution
num_switches=length(find(besteversol));
minens=1/bestever;
C(:,1)=besteversol';
wklwrite('ens_minimisation2.xls',C)      %Writes the output to an excel file.
fprintf('The minimum total system cost is R %g with %g sectionalisers.
',minens,num_switches)

```

```

function minsens1 = minsens1(A,Tot_Load,Ce)

%minsens = minsens(A)calculates the ENS for a system with
%no financial constraints.

mu = 0.02;           %failure rate of sectionaliser
tor = 1;            %repair time of sectionaliser
gamma = mu*tor;
p = 0.000;          %Probability of sectionaliser failure
Ts = 0.03;         %time for breaker to reclose after sectionaliser opens
Ce = 1;
Cue = 7;
node_column = A(:,1);
switch_column = A(:,6);
num_switches = sum(switch_column);

ENSDATA=zeros(length(node_column),7); %matrix holds all the values for calculation of ENS

%Calculation of the closest upstream switch for each section D(i)
for j=1:length(node_column)
    closest_upstream_switch = calc_cpp(A,j);
    ENSDATA(j,2) = closest_upstream_switch; %CPP column
end

% Matrix formation for the calculation of system's annual cost of ENS
for k=1:length(node_column)

    % Calculation of ZD(i)
    if ENSDATA(k,2)==0 % check so that matrix not exceeded
        ENSDATA(k,3)=Tot_Load; %main breaker trips causing total isolation
    else
        ENSDATA(k,3) = A(ENSDATA(k,2),7); %Load interrupted due to D(i)
    end

    % Calculation of d(i)
    if A(k,6)==0 % Check to see no switch is present in the section
        ENSDATA(k,4)=ENSDATA(k,2); %If there's no switch then d(i)=D(i)
    else
        if A(k,2)==0 % check so that matrix not exceeded
            ENSDATA(k,4)=A(1,2);%Places a zero (breaker trips) if ENSDATA(k,2)==0
        else
            ENSDATA(k,4)=ENSDATA(A(k,2),2); %If there is a switch in i,find d(i).
        end
    end

    % Calculation of Zd(i)
    if ENSDATA(k,4)==0 % check so that matrix not exceeded
        ENSDATA(k,5)= Tot_Load; %main breaker trips causing total isolation
    else
        ENSDATA(k,5) = A(ENSDATA(k,4),7); %losses due to isolation of d(k)
    end

    ENSDATA(k,6) = A(k,3)*Ts*Tot_Load; %Calculation of energy losses due to main breaker
    ENSDATA(k,7) = A(k,5)*ENSDATA(k,3); %calculation of local isolation energy losses

end
minsens1=sum(ENSDATA(:,6))+sum(ENSDATA(:,7));

```

```
%This function (costens_minimisation) determines the optimal number and location of
%sectionalisers on the system.
```

```
fitrec=[]; %Array to store values for graph plotting
bestever=-inf; %Lowest value that the function can evaluate to
maxgen=100; %Number of generations
LR=0.1; %Learning rate
FF=0.005; %Forgetting factor
trialsol=20; %Number of trial solutions in each generation
```

```
Loads the data file that contains system data
```

```
!notepad sysdata.dat
```

```
disp('Press any key to continue after saving the data file.....')
```

```
pause
```

```
%load sysdata.dat %Loading of data file containing the system data
%A=sysdata; %from notepad
```

```
%load sysdata.m %Data files that already contain system data for 3 networks
%A=sysdata; %and are used for testing the program
```

```
%load lebanon.m
```

```
%A=lebanon;
```

```
%load network.m
```

```
%A=network;
```

```
fprintf('\n')
```

```
failure_rate = input('Enter a value for the failure rate per km: ');
```

```
repair_rate = input('Enter a value for the average repair time for a fault (hours): ');
```

```
Ce = input('Enter a value for the cost of supplied energy(R per kWh): ');
```

```
installed_cost=input('Enter the installed cost of the sectionaliser: R ');
```

```
n=input('Enter the number of years to discount the costs over: ');
```

```
r=0.06;
```

```
h=installed_cost*1.14/(1/r-1/((r)*((1+r)^n))); %Annualised cost of sectionaliser
```

```
%installed cost * 14% overhead divide by
```

```
annuity factor
```

```
%Creation of matrix that holds all data required for calculation of ENS
```

```
for i=1:length(A(:,1))
```

```
    A(i,3)=A(i,3)*failure_rate; %Failure rate per section
```

```
    A(i,4)=repair_rate; %Average Repair time
```

```
    A(i,5)=A(i,3)*A(i,4); %Outage time per section
```

```
    A(i,7)=A(i,7)*Ce; %Cost energy supply interruption per section
```

```
end
```

```
%Calculation of system's unsupplied energy without sectionalisers
```

```
Load = sum(A(:,7));
```

```
for i=1:length(A(:,1))
```

```
    TOTENS(i,1) = A(i,3)*A(i,4)*Load;
```

```
end
```

```
Tot_ENS=sum(TOTENS(:,1));
```

```
B = zeros(maxgen,4);
```

```
%Matrix holds all the values of the output
```

```
C = zeros(length(A(:,1)),1);
```

```
%Matrix that holds the final sectionaliser placement
```

```
%Calculation of energy supply interruption per hour of each section
```

```
for i=1:length(A(:,1))
```

```
    load_after = calcu_load(A,i);
```

```
    A(i,7)=load_after; %Load interrupted per section
```

```
end
```

```
F = A;%Matrix to store the data when an additional sectionaliser is added
```

```
%Creation of probability vector initialised to 0.5
```

```
PV=0.5*ones(1,length(A(:,1)));
```

```
%Calculation of fitness values for trial solutions generated
```

```
for g=1:maxgen
```

```

bestfit=-inf;
for t=1:trialsol
    ts=rand(size(PV))<PV;          %Generation of random trial solutions
    A(:,6)= ts';                 %Placement of trial solution into switch column

    %Calculation of function value
    f = 1/minsysens2(A,h,Load,Ce);

    %Determination of the best solution in each generation
    if f>bestfit
        bestfit=f;
        bestsol=ts;
    end
end

%Equation to adjust the probability vector
PV=(1-LR)*PV+(LR*bestsol); %
PV=PV-FF*(PV-0.5*(ones(1,length(PV)))));
fitrec=[fitrec,bestfit];

%Determination the best trial solution ever
if bestfit>bestever
    bestever=bestfit;
    besteversol=bestsol;
end
%Output to the screen
fprintf('Maxgen: %g      gen: %g      best: %g      bestever:
%g\n',maxgen,g,1/bestfit,1/bestever)
B(g,1) = maxgen;
B(g,2) = g;
B(g,3) = 1/bestfit;
B(g,4) = 1/bestever;

end

wklwrite('costens_minimisation.xls',B) %Writes the output to an excel file.

%Plots a graph of fitness vs the number of generations
plot(fitrec)
xlabel('Number of Generations')
ylabel('Fitness')

%Determines the number of sectionalisers in the best solution
num_switches=length(find(besteversol));
minens=1/bestever;
C(:,1)=besteversol';
E=[A(:,1) A(:,2) C(:,1)]
wklwrite('costens_minimisation2.xls',E) %Writes the output to an excel file.
fprintf('The minimum total cost is R %g with %g sectionalisers. \n\n',minens,num_switches)
A(:,6)=besteversol';

%Calculation of SAIDI
for i=1:length(A(:,1))          %Load interrupted by isolation of each section
    load_after = calcu_load(A,i);
    ENSDATA(i,1)=load_after;
end

%Calculation of the closest upstream sectionaliser for each section
for i=1:length(A(:,1))
    closest_upstream_switch = calc_cpp(A,i);
    ENSDATA(i,2) = closest_upstream_switch;
end

for i=1:length(A(:,1))
    if ENSDATA(i,2)==0          %Check so that matrix not exceeded
        %Main breaker trips causing total isolation
        ENSDATA(i,3)=sum(A(:,7))*A(i,3)*A(i,4);
    else

```

```

    %Cost of ENS due to local isolation
    ENSDATA(i,3) = ENSDATA(ENSDATA(i,2),1)*A(i,3)*A(i,4);% cost of losses caused by
local isolation
    end
end
SAIDI = sum(ENSDATA(:,3))/sum(A(:,7));
fprintf('The value of SAIDI for this sectionaliser arrangement is %g.\n\n',SAIDI);

%Financial evaluation of the system

number=input('Enter 1 if you would like to determine the difference in the savings in cost
by the placement \nof an additional sectionaliser in a specified section to the current
sectionaliser arrangement or\nEnter 2 if you would like the PV of the annual total cost of
the system over a specified period \nEnter 0 to exit: ');
if isempty(number)
    disp('Please enter a valid number')
    fprintf('\n\n')
    number=input('Enter 1 if you would like to determine the difference in the savings in
cost by the placement \nof an additional sectionaliser in a specified section to the
current sectionaliser arrangement or\nEnter 2 if you would like the PV of these annual
total cost of the system over a specified period \nEnter 0 to exit: ');
else
    while (number~=0)
        switch number
            case 0
                break
            case 1
                %Determines the feasibility of adding an additional sectionaliser
                node=input('\n\nEnter the section that you would like to place the additional
sectionaliser on: ');
                F(node,6)=1;
                change_cost=unsupenergy(F,h,Load)-minens;
                F(node,6)=0;
                if change_cost>0
                    fprintf('\n\nThe additional sectionaliser in section %g results in an
additional annual \ncost of R %g and is therefore not feasible.\n\n',node,change_cost);
                else
                    fprintf('\n\nThe additional sectionaliser in section %g results in a savings
of R %g.\n\n',node,(-1*change_cost));
                end
                number=input('\n\nEnter 1 if you would like to recalculate for another section
otherwise \nEnter 2 to determine the discounted costs of the energy losses \nEnter 0 to
exit: ');
            case 2
                %Discounts the system's total costs over a specified number of years
                period=input('\n\nEnter the number of years to discount the costs over: ');
                discount_rate=input('\n\nEnter the discount rate of the project as a percentage:
');
                PV=financiall(minens,period,discount_rate);
                fprintf('\n\nThe present value of the distribution systems costs over a period of
%g years \nat a discount rate of %g %% is R %g.',period,discount_rate,PV);
                number=input('\n\nEnter 1 to determine feasibility of sectionaliser addition
\nEnter 2 to determine the system total cost over a specified number of years \nEnter 0 to
exit: ');
                ...
            otherwise
                disp('You have entered an invalid number')
                number=input('Enter 1, 2 or 0 now: ');
        end
    end
end

```

```

function minsens2 = minsens2(A,h,Tot_Load,Ce)

%minsens2 = minsens(A)calculates the ENS for cost_ensminimisation

mu = 0.02;           %failure rate of sectionaliser
tor = 1;            %repair time of sectionaliser
gamma = mu*tor;
p = 0.000;          %Probability of sectionaliser failure
Ts = 0.03;          %time for breaker to reclose after sectionaliser opens
Ce = 1;
Cue = 7;
node_column = A(:,1);
switch_column = A(:,6);
num_switches = sum(switch_column);

ENSDATA=zeros(length(node_column),11); %matrix holds all the values for calculation of ENS

%Calculation of the closest upstream sectionaliser for each section D(i)
for j=1:length(node_column)
    closest_upstream_switch = calc_cpp(A,j);
    ENSDATA(j,2) = closest_upstream_switch; %CPP column
end

% Matrix formation for the calculation of system's annual cost of ENS
for k=1:length(node_column)

    % Calculation of ZD(i)
    if ENSDATA(k,2)==0 % check so that matrix not exceeded
        ENSDATA(k,3)=Tot_Load; %main breaker trips causing total isolation
    else
        ENSDATA(k,3) = A(ENSDATA(k,2),7); %Load interrupted due to D(i)
    end

    % Calculation of d(i)
    if A(k,6)==0 % Check to see no switch is present in the section
        ENSDATA(k,4)=ENSDATA(k,2); %If there's no switch then d(i)=D(i)
    else
        if A(k,2)==0 % check so that matrix not exceeded
            ENSDATA(k,4)=A(1,2);%Places a zero (breaker trips) if ENSDATA(k,2)==0
        else
            ENSDATA(k,4)=ENSDATA(A(k,2),2); %If there is a switch in i,find d(i).
        end
    end

    % Calculation of Zd(i)
    if ENSDATA(k,4)==0 % check so that matrix not exceeded
        ENSDATA(k,5)= Tot_Load; %main breaker trips causing total isolation
    else
        ENSDATA(k,5) = A(ENSDATA(k,4),7); %losses due to isolation of d(k)
    end

    % Calculation of d(D(i))
    if ENSDATA(k,2)==0 %check so matrix is not exceeded
        ENSDATA(k,6)=A(1,2); %Places a zero (breaker trips) if ENSDATA(k,2)==0
    else
        ENSDATA(k,6)=ENSDATA(ENSDATA(k,2),4); %Places d(D(i)) into ENSDATA(k,6)
    end

    % Calculation of d(d(i))
    if ENSDATA(k,4)==0 %check so matrix is not exceeded
        ENSDATA(k,7)=A(1,2); %Places a one if ENSDATA(k,2)==0
    end
end

```

```

else
    ENSDATA(k,7)=ENSDATA(ENSDATA(k,4),4);%Places d(d(i)) into ENSDATA(k,7)
end

% Calculation of Z(d(D(i)))
if ENSDATA(k,6)==0 % check so that matrix not exceeded
    ENSDATA(k,8)= Tot_Load; %main breaker trips causing total isolation
else
    ENSDATA(k,8) = A(ENSDATA(k,6),7); %losses due to isolation of d(D(i))
end

% Calculation of Z(d(d(i)))
if ENSDATA(k,7)==0 % check so that matrix not exceeded
    ENSDATA(k,9)= Tot_Load; %main breaker trips causing total isolation
else
    ENSDATA(k,9) = A(ENSDATA(k,7),7); %losses due to isolation of d(d(i))
end

ENSDATA(k,10) = A(k,5)*(((1-p)*ENSDATA(k,3))+p*(ENSDATA(k,8)));
ENSDATA(k,11) = gamma*((1-p)*ENSDATA(k,5)+(p*(ENSDATA(k,9))))*A(k,6);

end
minsysens2 = sum(ENSDATA(:,10))+sum(ENSDATA(:,11))+(num_switches*(h+(mu*Ts*Tot_Load)));

```

University of Cape Town

```

% This function (specific_switch) determines the optimum placement of a
% specified number of sectionalisers

fitrec=[];      %Array to store values for graph plotting
bestever=-inf;
maxgen=100;
LR=0.1;        %Learning rate
FF=0.005;      %Forgetting factor
trialsol=20;   %Number of trial solutions in each generation

Loads the data file that contains system data
!notepad sysldat.dat
disp('Press any key to continue after saving the data file.....')
pause

%load sysdata.dat      %Loading of data file in Notepad which the
%A=sysdata;           %planner uses to enter the system data

%load sysdata.m        %Data files that already contain system data for 3 networks
%A=sysdata;           %and are used for testing the program
%load lebanon.m
%A = lebanon;
%load network.m
%A=network;

fprintf('\n')
num_switches = input('Enter the number of sectionalisers for optimal placement: ');
fprintf('\n')

failure_rate = input('Enter a value for the failure rate per km: ');
repair_rate = input('Enter a value for the average repair time for a fault (hours): ');
Ce = input('Enter a value for the cost of supplied energy(per kWh): R ');
installed_cost=input('Enter the installed cost of the switch: R ');
n=input('Enter the number of years to discount the costs over: ');
r=input('Enter a value for the discount rate as a percentage: ');
h=installed_cost*1.14/(1/r-1/((r)*((1+r)^n))); %Annualised cost of sectionaliser
%installed cost * 14% overhead

%Creation of matrix that holds all data required for calculation of ENS
for i=1:length(A(:,1))
    A(i,3)=A(i,3)*failure_rate; %Failure rate per section
    A(i,4)=repair_rate; %Average Repair time
    A(i,5)=A(i,3)*A(i,4); %Outage time per section
    A(i,7)=A(i,7)*Ce; %Cost energy supply interruption per section
end

%Calculation of cost of system's unsupplied energy without sectionalisers
Load = sum(A(:,7));
for i=1:length(A(:,1))
    TOTENS(i,1) = A(i,3)*A(i,4)*Load;
end
Tot_ENS=sum(TOTENS(:,1));

B = zeros(maxgen,4); %Matrix holds all the values of the output
C = zeros(length(A(:,1)),3); %Matrix that holds the final switch placement

%Calculation of energy supply interruption per hour of each section
for i=1:length(A(:,1))
    load_after = calcu_load(A,i);
    A(i,7)=load_after; %Load interrupted per section
end

F = A; %Matrix to store the data when an additional sectionaliser is added

%Creation of probability vector initialised to 0.5
PV=0.5*ones(1,length(A(:,1)));

%Calculation of fitness values for trial solutions generated

```

```

for g=1:maxgen
    bestfit=-inf;

    for t=1:15
        ts=rand(size(PV))<PV;          %Generation of random trial solutions
        %ts is a function that randomly places sectionalisers
        %ts = rand_switch_placement(length(A(:,1)),num_switches);
        A(:,6)= ts';                  %Placement of trial solution into switch column
        %Calculation of function value
        f=1/minsysens3(A,h,num_switches,Load);

        %Determination of the best solution in each generation
        if f>bestfit
            bestfit=f;
            bestsol=ts;
        end
    end

    %Equation to adjust the probability vector
    PV=(1-LR)*PV+(LR*bestsol);
    PV=PV-FF*(PV-0.5*(ones(1,length(PV))));
    fitrec=[fitrec,bestfit];

    %Determination the best trial solution ever
    if bestfit>bestever
        bestever=bestfit;
        besteversol=bestsol;
    end
    %Output to the screen
    fprintf('Maxgen: %g      gen: %g      best: %g      bestever:
%g\n',maxgen,g,1/bestfit,1/bestever)
    B(g,1) = maxgen;
    B(g,2) = g;
    B(g,3) = 1/bestfit;
    B(g,4) = 1/bestever;
end

wklwrite('specific_switch1.xls',B) %Writes the output to an excel file.

%Plots a graph of fitness vs the number of generations
plot(fitrec)
xlabel('Number of Generations')
ylabel('Fitness')

bestswitchposition= besteversol';
optimum_solution = [A(:,1),      A(:,2),          besteversol'];

fprintf('\n')
disp('node      predecessor      switch');
for i=1:length(A(:,1))
    fprintf('%g      %g      %g\n',A(i,1),A(i,2),bestswitchposition(i,1));
    C(i,1) = A(i,1);
    C(i,2) = A(i,2);
    C(i,3) = bestswitchposition(i,1);
end
wklwrite('specific_switch2.xls',C) %Writes the output to an excel file.
minens=1/bestever;
fprintf('The minimum cost unsupplied energy is R %g with %g sectionalisers.
\n\n',minens,num_switches)
A(:,6)=besteversol';

%Calculation of SAIDI
for i=1:length(A(:,1))          %Load interrupted by isolation of each section
    load_after = calcu_load(A,i);
    ENSDATA(i,1)=load_after;
end

%Calculation of the closest upstream sectionaliser for each section

```

```

for i=1:length(A(:,1))
    closest_upstream_switch = calc_cpp(A,i);
    ENSDATA(i,2) = closest_upstream_switch;
end

for i=1:length(A(:,1))
    if ENSDATA(i,2)==0    %Check so that matrix not exceeded
        %Main breaker trips causing total isolation
        ENSDATA(i,3)=sum(A(:,7))*A(i,3)*A(i,4);
    else
        %Cost of ENS due to local isolation
        ENSDATA(i,3) = ENSDATA(ENSDATA(i,2),1)*A(i,3)*A(i,4);
    end
end
SAIDI = sum(ENSDATA(:,3))/sum(A(:,7));
fprintf('The value of SAIDI for this sectionaliser arrangement is %g.\n\n',SAIDI);

%Financial evaluation of the system

number=input('Enter 1 if you would like to determine the difference in the savings in cost
by the placement \nof an additional sectionaliser in a specified section to the current
sectionaliser arrangement or\nEnter 2 if you would like the PV of the annual total cost of
the system over a specified period \nEnter 0 to exit: ');
if isempty(number)
    disp('Please enter a valid number')
    fprintf('\n\n')
    number=input('Enter 1 if you would like to determine the difference in the savings in
cost by the placement \nof an additional sectionaliser in a specified section to the
current sectionaliser arrangement or\nEnter 2 if you would like the PV of these annual
total cost of the system over a specified period \nEnter 0 to exit: ');
else
    while (number~=0)
        switch number
            case 0
                break
            case 1
                %Determines the feasibility of adding an additional sectionaliser
                node=input('\n\nEnter the section that you would like to place the additional
sectionaliser on: ');
                F(node,6)=1;
                change_cost=unsupenergy(F,h,Load)-minens;
                F(node,6)=0;
                if change_cost>0
                    fprintf('\n\nThe additional sectionaliser in section %g results in an
additional annual \ncost of R %g and is therefore not feasible.\n\n',node,change_cost);
                else
                    fprintf('\n\nThe additional sectionaliser in section %g results in a savings
of R %g.\n\n',node,(-1*change_cost));
                end
                number=input('\n\nEnter 1 if you would like to recalculate for another section
otherwise \nEnter 2 to determine the discounted costs of the energy losses \nEnter 0 to
exit: ');
            case 2
                %Discounts the system's total costs over a specified number of years
                period=input('\n\nEnter the number of years to discount the costs over: ');
                discount_rate=input('\n\nEnter the discount rate of the project as a percentage:
');
                PV=financial1(minens,period,discount_rate);
                fprintf('\n\nThe present value of the distribution systems costs over a period of
%g years \nat a discount rate of %g %% is R %g.',period,discount_rate,PV);
                number=input('\n\nEnter 1 to determine feasibility of sectionaliser addition
\nEnter 2 to determine the system total cost over a specified number of years \nEnter 0 to
exit: ');
                ...
            otherwise
                disp('You have entered an invalid number')
                number=input('Enter 1, 2 or 0 now: ');
        end
    end
end

```

end
end

University of Cape Town

```

function minsens3 = minsens3(A,h,spec_switches,Tot_Load)

%minsens3 = minsens3(A)calculates the ENS for specific_switch

mu = 0.02;           %failure rate of sectionaliser
tor = 1;            %repair time of sectionaliser
gamma = mu*tor;
p = 0.00;           %Probability of sectionaliser failure
Ts = 0.03;          %time for breaker to reclose after sectionaliser opens

node_column = A(:,1);
switch_column = A(:,6);
%load_column = A(:,7);
num_switches = sum(switch_column);

ENSDATA=zeros(length(node_column),11); %matrix holds all the values for calculation of ENS

%Calculation of the closest upstream switch for each section D(i)
for j=1:length(node_column)
    closest_upstream_switch = calc_cpp(A,j);
    ENSDATA(j,2) = closest_upstream_switch; %CPP column
end

% Matrix formation for the calculation of system's annual cost of ENS
for k=1:length(node_column)

    % Calculation of ZD(i)
    if ENSDATA(k,2)==0 % check so that matrix not exceeded
        ENSDATA(k,3)=Tot_Load; %main breaker trips causing total isolation
    else
        ENSDATA(k,3) = A(ENSDATA(k,2),7); %Load interrupted due to D(i)
    end

    % Calculation of d(i)
    if A(k,6)==0 % Check to see no switch is present in the section
        ENSDATA(k,4)=ENSDATA(k,2); %If there's no switch then d(i)=D(i)
    else
        if A(k,2)==0 % check so that matrix not exceeded
            ENSDATA(k,4)=A(1,2);%Places a zero (breaker trips) if ENSDATA(k,2)==0
        else
            ENSDATA(k,4)=ENSDATA(A(k,2),2); %If there is a switch in i,find d(i).
        end
    end

    % Calculation of Zd(i)
    if ENSDATA(k,4)==0 % check so that matrix not exceeded
        ENSDATA(k,5)= Tot_Load; %main breaker trips causing total isolation
    else
        ENSDATA(k,5) = A(ENSDATA(k,4),7); %losses due to isolation of d(k)
    end

    % Calculation of d(D(i))
    if ENSDATA(k,2)==0 %check so matrix is not exceeded
        ENSDATA(k,6)=A(1,2); %Places a zero (breaker trips) if ENSDATA(k,2)==0
    else
        ENSDATA(k,6)=ENSDATA(ENSDATA(k,2),4); %Places d(D(i)) into ENSDATA(k,6)
    end

    % Calculation of d(d(i))
    if ENSDATA(k,4)==0 %check so matrix is not exceeded
        ENSDATA(k,7)=A(1,2); %Places a one if ENSDATA(k,2)==0
    end
end

```

```

else
    ENSDATA(k,7)=ENSDATA(ENSDATA(k,4),4);%Places d(d(i)) into ENSDATA(k,7)
end

% Calculation of Z(d(D(i)))
if ENSDATA(k,6)==0 % check so that matrix not exceeded
    ENSDATA(k,8)= Tot_Load; %main breaker trips causing total isolation
else
    ENSDATA(k,8) = A(ENSDATA(k,6),7); %losses due to isolation of d(D(i))
end

% Calculation of Z(d(d(i)))
if ENSDATA(k,7)==0 % check so that matrix not exceeded
    ENSDATA(k,9)= Tot_Load; %main breaker trips causing total isolation
else
    ENSDATA(k,9) = A(ENSDATA(k,7),7); %losses due to isolation of d(d(i))
end

ENSDATA(k,10) = A(k,5)*(((1-p)*ENSDATA(k,3))+(p*(ENSDATA(k,8))));
ENSDATA(k,11) = gamma*((1-p)*ENSDATA(k,5)+(p*(ENSDATA(k,9))))*A(k,6);

end
%function penalised each time solutions are produced with more or less sectionalisers than
the specified number
minsysens3 =
sum(ENSDATA(:,10))+sum(ENSDATA(:,11))+ (spec_switches*(h+(mu*Ts*Tot_Load)))+( (num_switches-
spec_switches)*10000000);

```

University of Cape Town

%This function (Calculate_ens) calculates the annual cost of ENS for a
%switch arrangement specified by the user

!notepad sysldat.dat

disp('Press any key to continue after saving the data file.....')

pause

load sysdata.dat %Loading of data file in Notepad which the

A=sysdata; %planner uses to enter the system data

%load sysdata.m %Data files that already contain system data for 3 networks

%A=sysdata; %and are used for testing the program

%load network.m

%A=network;

%load lebanon.m

%A=lebanon;

fprintf('\n')

failure_rate = input('Enter a value for the failure rate per km: ');

repair_rate = input('Enter a value for the average repair time for a fault (hours): ');

Ce = input('Enter a value for the cost of supplied energy(R per kWh): ');

num_switches=sum(A(:,6));

installed_cost=input('Enter the installed cost of the sectionaliser: R ');

n=input('Enter the number of years to discount the costs over: ');

r=input('Enter a value for the discount rate: ');

h=installed_cost*1.14/(1/r-1/((r)*((1+r)^n))); %annualised cost of sectionaliser

%Creation of matrix that holds all data required for calculation of ENS

for i=1:length(A(:,1))

A(i,3)=A(i,3)*failure_rate; %Failure rate per section

A(i,4)=repair_rate; %Average Repair time

A(i,5)=A(i,3)*A(i,4); %Outage time per section

A(i,7)=A(i,7)*Ce; %Cost energy supply interruption per section

end

%Calculation of cost of system's unsupplied energy without sectionalisers

Load = sum(A(:,7));

for i=1:length(A(:,1))

TOTENS(i,1) = A(i,3)*A(i,4)*Load;

end

Tot_ENS=sum(TOTENS(:,1));

%Calculation of cost of ENS for the specified arrangement

unsupplied_energy = minsens4(A,h,Load);

fprintf('The minimum cost unsupplied energy is R %g with %g sectionalisers.

\n\n',unsupplied_energy,num_switches)

%Calculation of SAIDI

%Load interrupted by isolation of each section

for i=1:length(A(:,1))

load_after = calcu_load(A,i);

ENSDATA(i,1)=load_after;

end

%Calculation of the closest upstream sectionaliser for each section

for i=1:length(A(:,1))

closest_upstream_switch = calc_cpp(A,i);

ENSDATA(i,2) = closest_upstream_switch;

end

for i=1:length(A(:,1))

if ENSDATA(i,2)==0 %Check so that matrix not exceeded

%Main breaker trips causing total isolation

ENSDATA(i,3)=sum(A(:,7))*A(i,5);

else

%Cost of ENS caused by local isolation

```

    ENSDATA(i,3) = ENSDATA(ENSDATA(i,2),1)*A(i,5);
end
end
SAIDI = sum(ENSDATA(:,3))/sum(A(:,7));
fprintf('The value of SAIDI for this sectionaliser arrangement is %g.\n\n',SAIDI);

%Financial evaluation of the system

number=input('Enter 1 if you would like to determine the difference in the savings in cost
by the placement \nof an additional sectionaliser in a specified section to the current
sectionaliser arrangement or\nEnter 2 if you would like the PV of the annual total cost of
the system over a specified period \nEnter 0 to exit: ');
if isempty(number)
    disp('Please enter a valid number')
    fprintf('\n\n')
    number=input('Enter 1 if you would like to determine the difference in the savings in
cost by the placement \nof an additional sectionaliser in a specified section to the
current sectionaliser arrangement or\nEnter 2 if you would like the PV of these annual
total cost of the system over a specified period \nEnter 0 to exit: ');
else

    while (number~=0)
        switch number
            case 0
                break
            case 1
                %Determines the feasibility of adding an additional sectionaliser
                node=input('\n\nEnter the section that you would like to place the additional
sectionaliser on: ');
                F(node,6)=1;
                change_cost=unsupenergy(F,h,Load)-minens;
                F(node,6)=0;
                if change_cost>0
                    fprintf('\n\nThe additional sectionaliser in section %g results in an
additional annual \ncost of R %g and is therefore not feasible.\n\n',node,change_cost);
                else
                    fprintf('\n\nThe additional sectionaliser in section %g results in a savings
of R %g.\n\n',node,(-1*change_cost));
                end
                number=input('\n\nEnter 1 if you would like to recalculate for another section
otherwise \nEnter 2 to determine the discounted costs of the energy losses \nEnter 0 to
exit: ');
            case 2
                %Discounts the system's total costs over a specified number of years
                period=input('\n\nEnter the number of years to discount the costs over: ');
                discount_rate=input('\n\nEnter the discount rate of the project as a percentage:
');
                PV=financial1(minens,period,discount_rate);
                fprintf('\n\nThe present value of the distribution systems costs over a period of
%g years \nat a discount rate of %g %% is R %g.',period,discount_rate,PV);
                number=input('\n\nEnter 1 to determine feasibility of sectionaliser addition
\nEnter 2 to determine the system total cost over a specified number of years \nEnter 0 to
exit: ');
                ...
            otherwise
                disp('You have entered an invalid number')
                number=input('Enter 1, 2 or 0 now: ');
        end
    end
end
end

```

```

function minsens4=minsens4(A,h,Tot_Load)

%minsens4 = minsens4(A)calculates the ENS for calculate_ens.

mu = 0.02;           %failure rate of sectionaliser
tor = 1;            %repair time of sectionaliser
gamma = mu*tor;
p = 0.00;           %Probability of sectionaliser failure
Ts = 0.03;          %time for breaker to reclose after sectionaliser opens

node_column = A(:,1);
switch_column = A(:,6);
num_switches = sum(switch_column);

ENSDATA=zeros(length(node_column),12); %matrix holds all the values for calculation of ENS

%Calculation of energy supply interruption per hour of each section
for i=1:length(node_column)
    load_after = calcu_load(A,i);
    ENSDATA(i,1)=load_after;%NB load interrupted per section
end

%Calculation of the closest upstream switch for each section D(i)
for j=1:length(node_column)
    closest_upstream_switch = calc_cpp(A,j);
    ENSDATA(j,2) = closest_upstream_switch; %CPP column
end

% Matrix formation for the calculation of system's annual cost of ENS
for k=1:length(node_column)

    % Calculation of ZD(i)
    if ENSDATA(k,2)==0 % check so that matrix not exceeded
        ENSDATA(k,3)=Tot_Load; %main breaker trips causing total isolation
    else
        ENSDATA(k,3) = ENSDATA(ENSDATA(k,2),1); %Load interrupted due to D(i)
    end

    % Calculation of d(i)
    if A(k,6)==0 % Check to see no switch is present in the section
        ENSDATA(k,4)=ENSDATA(k,2); %If there's no switch then d(i)=D(i)
    else
        if A(k,2)==0 % check so that matrix not exceeded
            ENSDATA(k,4)=A(1,2);%Places a zero (breaker trips) if ENSDATA(k,2)==0
        else
            ENSDATA(k,4)=ENSDATA(A(k,2),2); %If there is a switch in i,find d(i).
        end
    end

    % Calculation of Zd(i)
    if ENSDATA(k,4)==0 % check so that matrix not exceeded
        ENSDATA(k,5)= Tot_Load; %main breaker trips causing total isolation
    else
        ENSDATA(k,5) = ENSDATA(ENSDATA(k,4),1); %losses due to isolation of d(k)
    end

    % Calculation of d(D(i))
    if ENSDATA(k,2)==0 %check so matrix is not exceeded
        ENSDATA(k,6)=A(1,2); %Places a zero (breaker trips) if ENSDATA(k,2)==0
    else
        ENSDATA(k,6)=ENSDATA(ENSDATA(k,2),4); %Places d(D(i)) into ENSDATA(k,6)
    end
end

```

```

end

% Calculation of d(d(i))
if ENSDATA(k,4)==0 %check so matrix is not exceeded
    ENSDATA(k,7)=A(1,2); %Places a one if ENSDATA(k,2)==0
else
    ENSDATA(k,7)=ENSDATA(ENSDATA(k,4),4);%Places d(d(i)) into ENSDATA(k,7)
end

% Calculation of Z(d(D(i)))
if ENSDATA(k,6)==0 % check so that matrix not exceeded
    ENSDATA(k,8)= Tot_Load; %main breaker trips causing total isolation
else
    ENSDATA(k,8) = ENSDATA(ENSDATA(k,6),1); %losses due to isolation of d(D(i))
end

% Calculation of Z(d(d(i)))
if ENSDATA(k,7)==0 % check so that matrix not exceeded
    ENSDATA(k,9)= Tot_Load; %main breaker trips causing total isolation
else
    ENSDATA(k,9) = ENSDATA(ENSDATA(k,7),1); %losses due to isolation of d(d(i))
end

ENSDATA(k,10) = A(k,5)*(((1-p)*ENSDATA(k,3)+(p*(ENSDATA(k,8)))));
ENSDATA(k,11) = gamma*((1-p)*ENSDATA(k,5)+(p*(ENSDATA(k,9))))*A(k,6);
ENSDATA(k,12) = ENSDATA(k,3)*A(k,5);
end
ENSDATA;
minsysens4 = sum(ENSDATA(:,10))+sum(ENSDATA(:,11))+(num_switches*(h+(mu*Ts*Tot_Load)));

```

```
function financial1=financial1(annual_cost_ens,plant_life,discount_rate)
%financial1=financial1(annual_cost_ens,plant_life,discount_rate)
%This function calculates the discounted costs over a specified number of years
format bank %displays in dollars and cents
%fprintf('\n');
r = discount_rate/100;
financial1=annual_cost_ens*(1/r - 1/((r)*(1+r)^(plant_life)));
```

University of Cape Town

```
function
rand_switch_placement=rand_switch_placement(vector_length,num_switches,Position_probabilit
y)
%function used to generate random trial solutions
%rand_switch_placement=rand_switch_placement(vector_length,num_switches)
%input the length of vector and number of sectionalisers required for random placement

switch_vector = zeros(1,vector_length);

while num_switches>0
    position = fix(vector_length*rand)+1;
    % check to ensure that the switch isn't placed in the same place in the vector i.e
    don't get 2 switches
    %placed in position [1,1] in switch_vector
    if switch_vector(1, position) == 0
        switch_vector(1, position) = 1;
        num_switches=num_switches-1;
    end
end

end

rand_switch_placement=switch_vector;
```

University of Cape Town

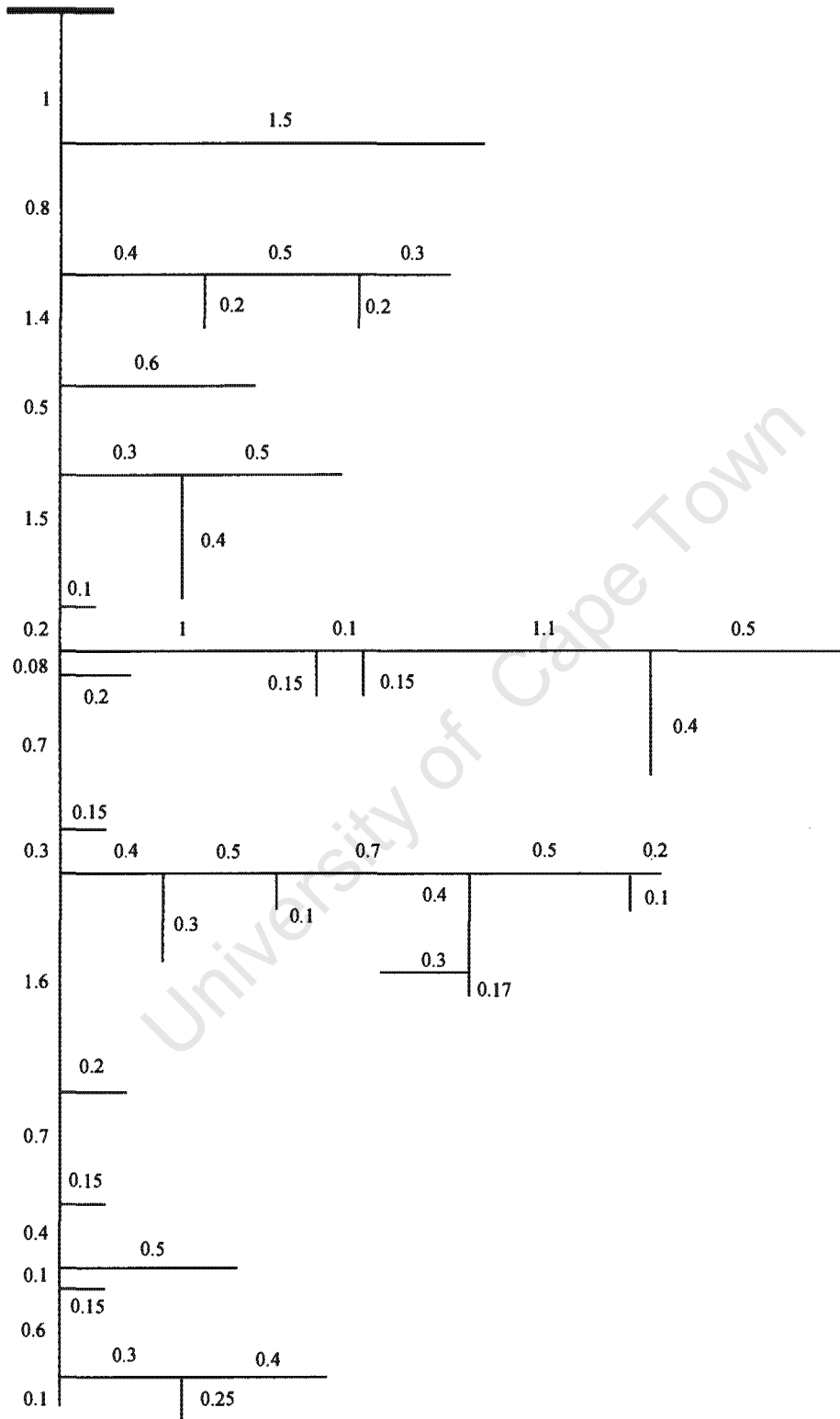
APPENDIX D

RESULTS FOR THE TESTS PERFORMED ON THE LEBANON NETWORK

University of Cape Town

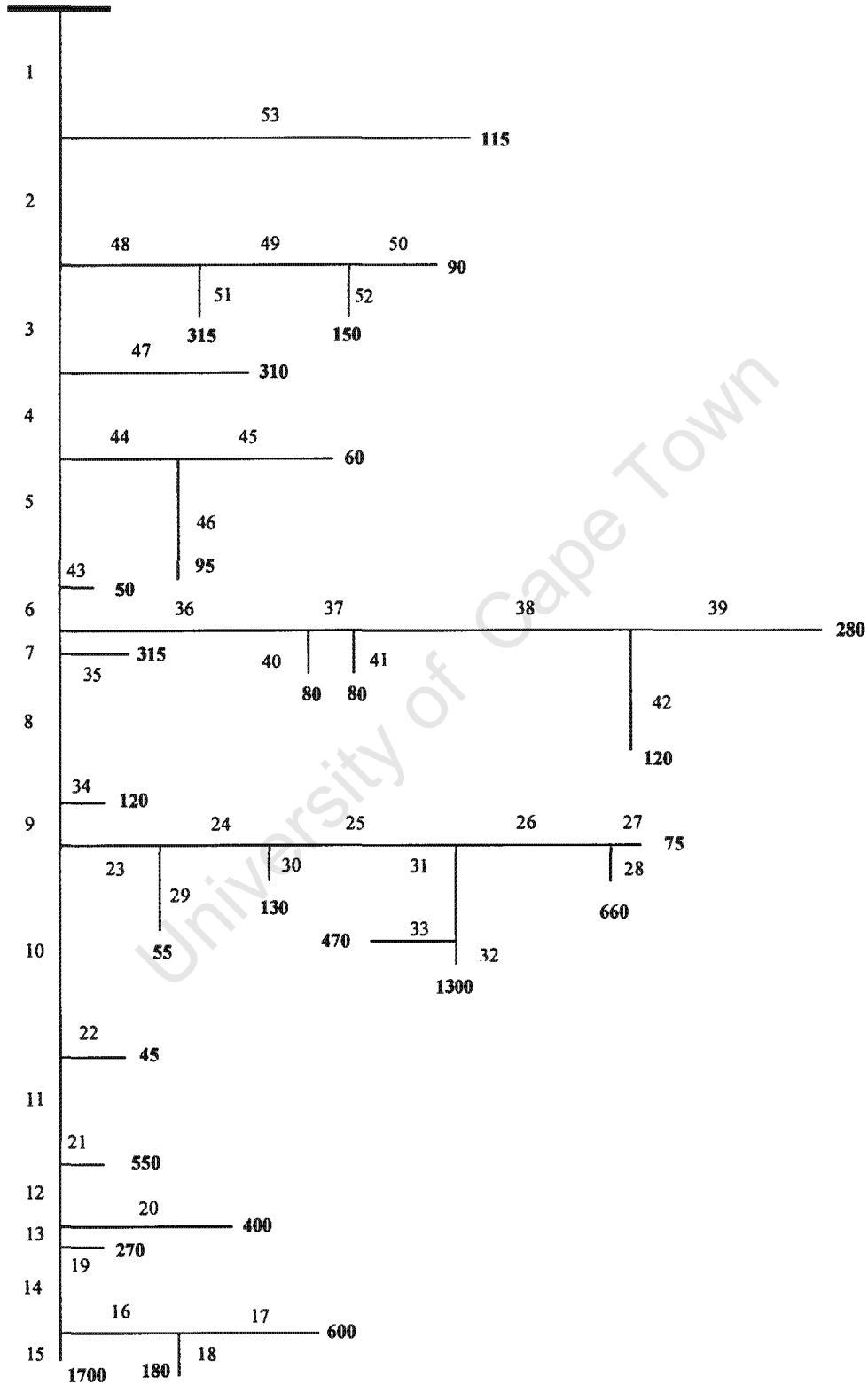
LEBANON RADIAL 15KV NETWORK

LENGTHS IN KM OF EACH SECTION



LEBANON RADIAL 15KV NETWORK

REGULAR FONT – SECTION NUMBERS
 BOLD FONT – AVERAGE LOADS IN KW



OPTIMUM PLACEMENT OF A SPECIFIED NUMBER OF SECTIONALISERS FOR THE LEBANON NETWORK

REPAIR TIME PER SECTION	NUMBER OF SECTIONALISERS PLACED	SECTIONS SECTIONALISERS PLACED ON	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST ENS	TOTAL COST
[hrs]			[hrs/yr/cust]	[R]	[R]	[R]
2	1	10	13.83	4058	34978	39036
2	2	36,10	12.57	8116	29831	37947
2	3	48,36,10	11.41	12174	25142	37316
2	4	48,36,10,23	10.42	16232	21143	37375
2	5	48,44,36,23,10	9.95	20290	19284	39574
2	6	48,47,44,36,23,10	9.72	24348	18416	42764

OPTIMUM PLACEMENT OF A SPECIFIED NUMBER OF SECTIONALISERS VARYING THE REPAIR RATE FOR LEBANON

REPAIR TIME PER SECTION	NUMBER OF SECTIONALISERS PLACED	SECTIONS SECTIONALISERS PLACED ON	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST ENS	TOTAL COST
[hrs]			[hrs/yr/cust]	[R]	[R]	[R]
2.0	1	10	13.83	4058	34978	39036
1.5	2	36,10	10.77	8116	22416	30532
1.0	3	48,36,10	8.70	12174	13943	26117
0.9	4	48,36,10,23	7.65	16232	9701	25933
0.7	5	48,44,36,23,10	6.98	20290	7026	27316
0.5	6	48,47,44,36,23,10	6.46	24348	4987	29335

MINIMISATION OF THE ANNUAL COST OF ENS BY VARYING C_e FOR THE LEBANON NETWORK

COST OF ENS	NUMBER OF SECTIONALISERS PLACED	SECTIONS SECTIONALISERS PLACED ON	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST ENS	TOTAL COST
[R/kWh]			[hrs/yr/cust]	[R]	[R]	[R]
0.25	0		20.50	0	20100	20100
0.50	3	48,36,10	11.41	12174	25142	37316
1.00	4	48,36,28,10	7.73	16232	42285	58517
1.50	5	23,48,44,36,10	6.36	20290	57851	78141
2.00	5	48,44,36,23,10	5.91	20290	77134	97424
3.00	7	48,47,44,36,23,10,4	5.06	28406	106380	134786
4.00	7	48,47,44,36,23,10,5	4.83	28406	141839	170245
5.00	7	48,47,44,36,23,10,6	4.70	28406	176801	205207

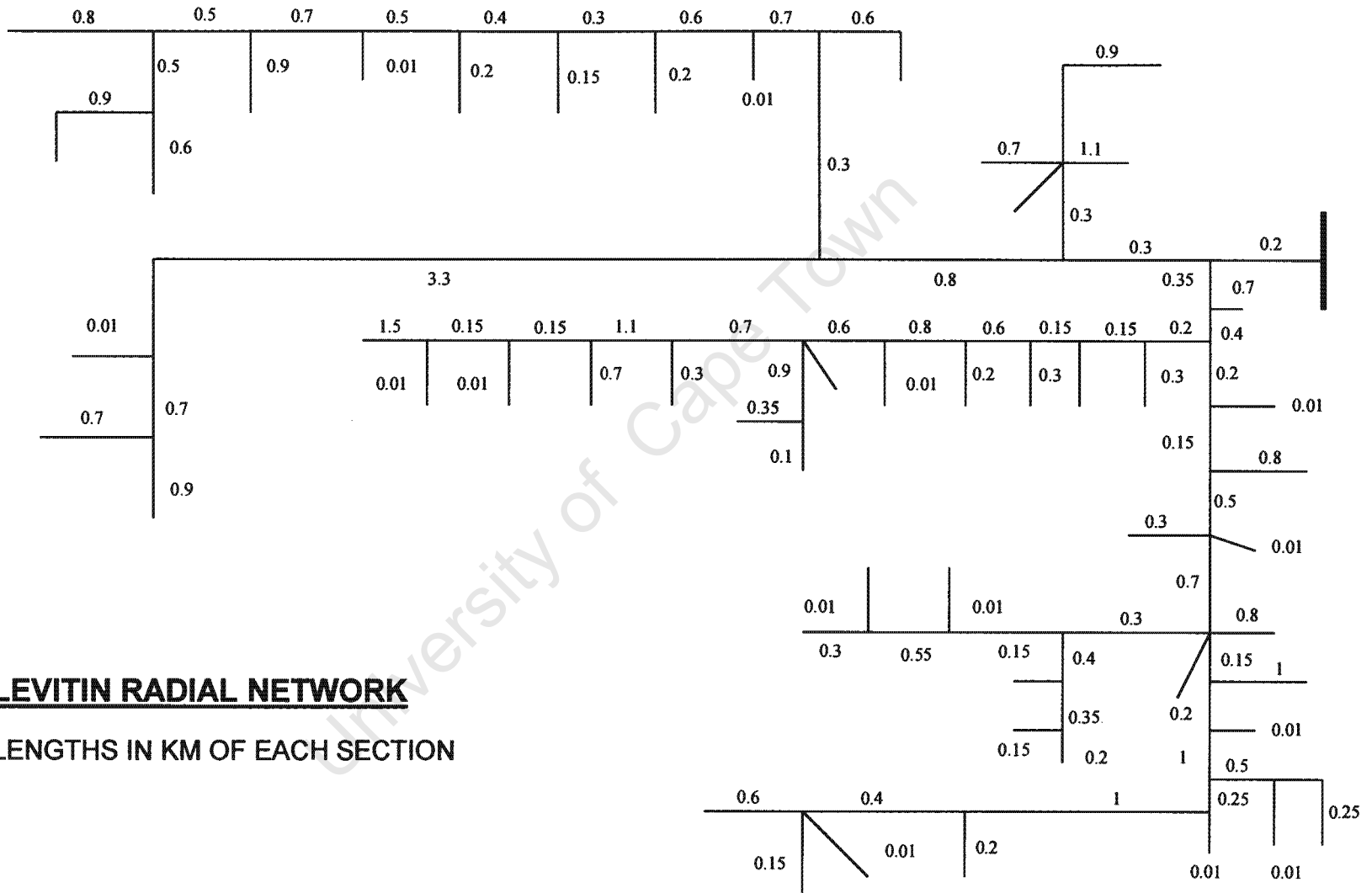
MINIMISATION OF THE ANNUAL COST OF ENS BY VARYING THE FAILURE RATE FOR THE LEBANON NETWORK

FAILURE RATE	NUMBER OF SECTIONALISERS PLACED	SECTIONS SECTIONALISERS PLACED ON	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST ENS	TOTAL COST
[f/km/yr]			[hrs/yr/cust]	[R]	[R]	[R]
0.01	0		5.87	0	2010	2010
0.05	0		7.81	0	10050	10050
0.10	0		10.25	0	20100	20100
0.15	0		12.68	0	30151	30151
0.20	3	48,36,10	11.41	12174	25142	37316
0.25	4	48,36,10,4	11.68	16232	26343	42575
0.30	4	48,36,10,5	12.94	16232	31544	47776
0.35	4	48,36,10,6	14.20	16232	36744	52976

APPENDIX E

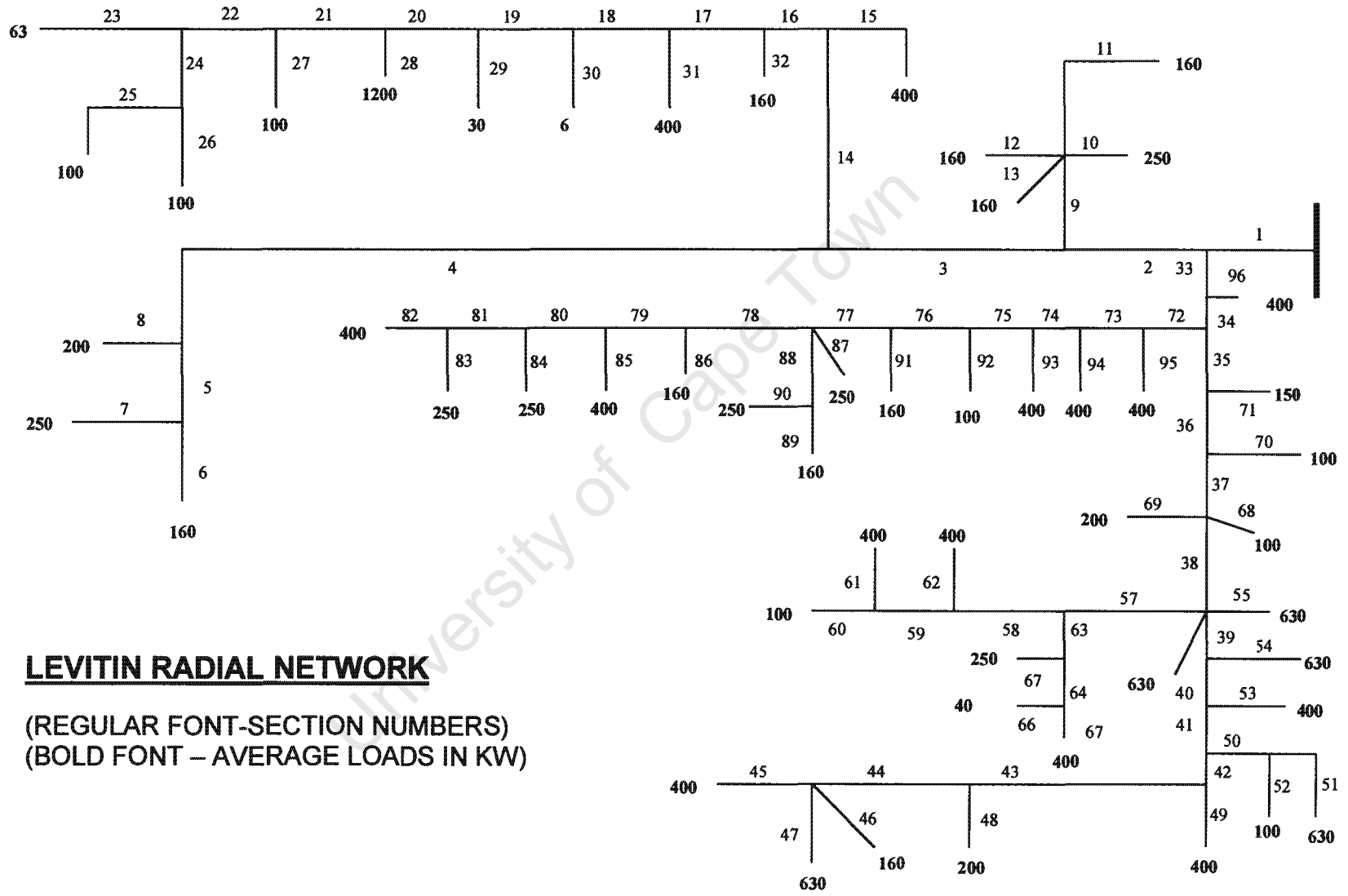
RESULTS FOR THE TESTS PERFORMED ON THE LEVITIN NETWORK

University of Cape Town



LEVITIN RADIAL NETWORK

LENGTHS IN KM OF EACH SECTION



LEVITIN RADIAL NETWORK
 (REGULAR FONT-SECTION NUMBERS)
 (BOLD FONT – AVERAGE LOADS IN KW)

OPTIMUM PLACEMENT OF A SPECIFIED NUMBER OF SECTIONALISERS FOR THE LEVITIN NETWORK

REPAIR TIME PER SECTION	NUMBER OF SECTIONALISERS PLACED	SECTIONS SECTIONALISERS PLACED ON	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST ENS	TOTAL COST
[hrs]			[int/yr/cust]	[R]	[R]	[R]
2.0	1	2	11.07	4058	87386	91444
2.0	2	72,2	8.21	8116	65023	73139
2.0	3	72,35,2	5.74	12174	45672	57846
2.0	4	72,36,9,3	5.19	16232	41403	57635
2.0	5	75,41,36,4,2	4.67	20290	37364	57654
2.0	6	72,39,21,4,2	4.23	24348	33949	58297

OPTIMUM PLACEMENT OF A SPECIFIED NUMBER OF SECTIONALISERS VARYING THE REPAIR RATE FOR THE LEVITIN NETWORK

REPAIR TIME PER SECTION	NUMBER OF SECTIONALISERS PLACED	SECTIONS SECTIONALISERS PLACED ON	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST ENS	TOTAL COST
[hrs]			[int/yr/cust]	[R]	[R]	[R]
2.0	1	2	11.07	4058	87386	91444
1.5	2	72,2	6.16	8116	48848	56964
1.0	3	72,36,2	3.15	12174	25339	37513
0.9	4	73,39,35,2	2.33	16232	18927	35159
0.7	5	74,39,35,4,2	1.63	20290	13476	33766
0.5	6	73,39,35,21,4,2	1.06	24348	8984	33332

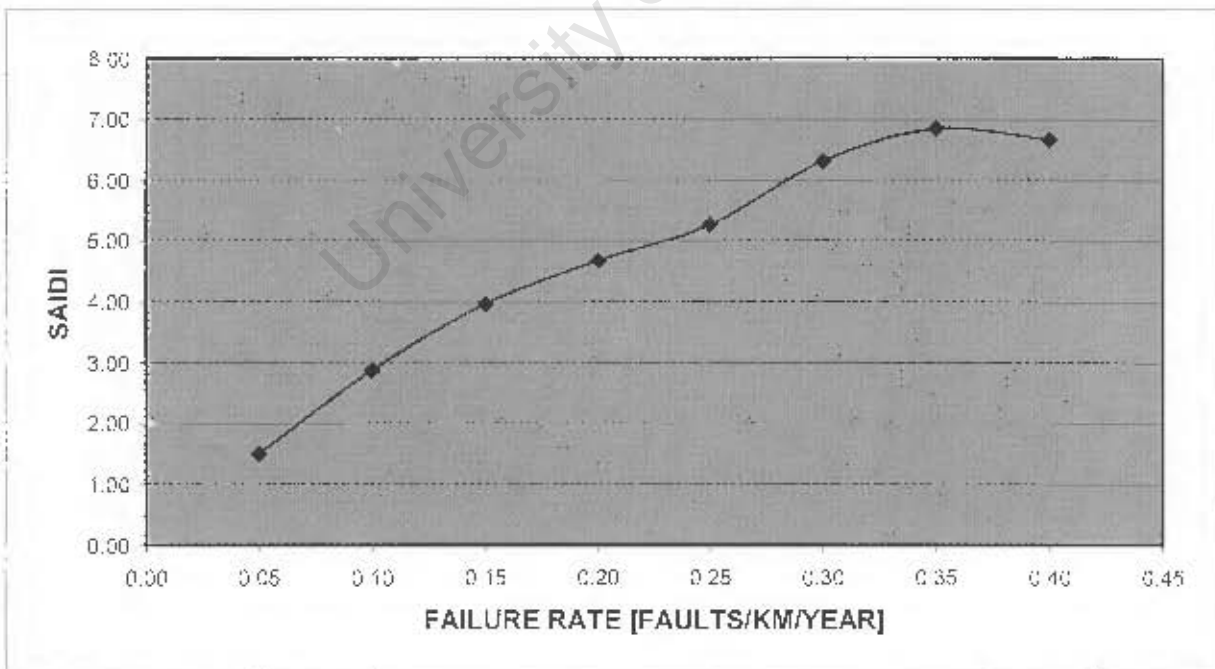
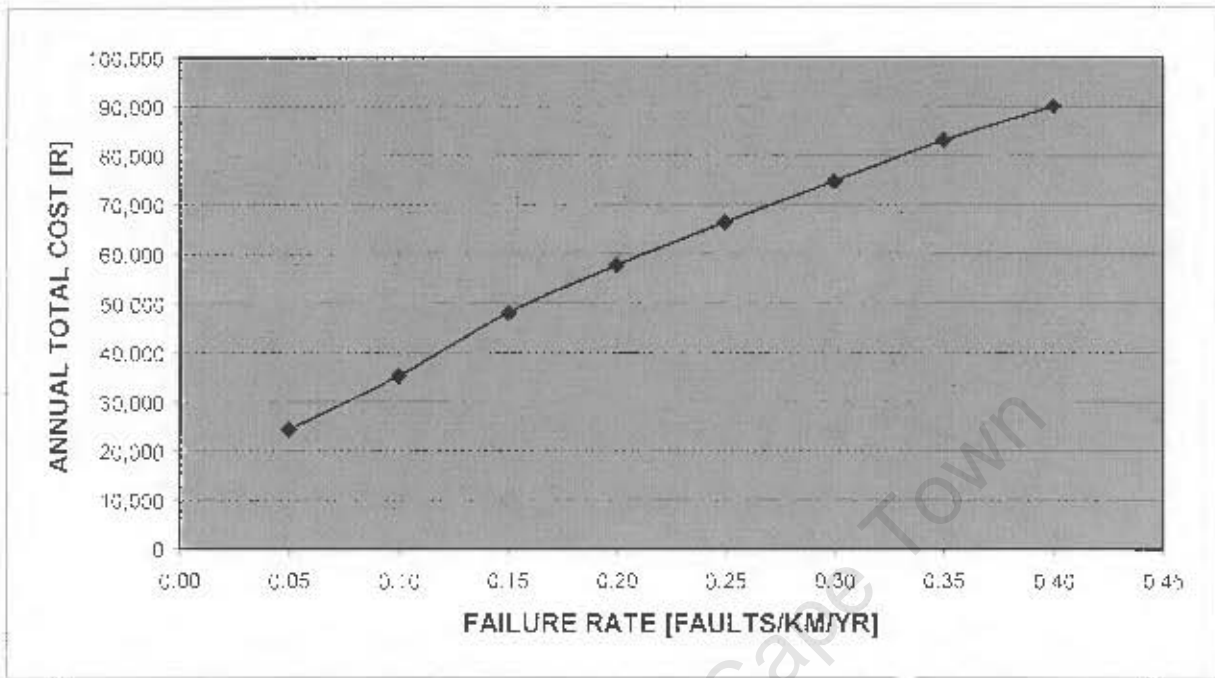
MINIMISATION OF THE ANNUAL COST OF ENS BY VARYING C_e FOR THE LEVITIN NETWORK

COST OF ENS	NUMBER OF SECTIONALISERS PLACED	SECTIONS SECTIONALISERS PLACED ON	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST ENS	TOTAL COST
[R/kWh]			[int/yr/cust]	[R]	[R]	[R]
0.25	4	72,36,9,3	5.26	16232	21060	37292
0.50	5	72,39,36,4,2	4.67	20290	37365	57655
1.00	8	74,57,39,36,21,9,4,2	3.66	32464	59227	91691
1.50	10	96,75,72,57,41,35,21,9,4,2	3.10	40580	76156	116736
2.00	12	96,78,72,70,57,54,41,35,21,9,4,2	2.78	48696	92228	140924
3.00	16	96,79,75,72,70,57,55,54,41,37,33,21,16,9,4,2	2.34	64928	118748	183676
4.00	16	96,78,75,72,70,57,55,54,41,35,33,21,16,9,4,2	2.32	64928	156571	221499
5.00	19	96,88,78,75,72,70,57,55,54,43,41,35,33,21,16,9,4,2	2.12	77102	181282	258384

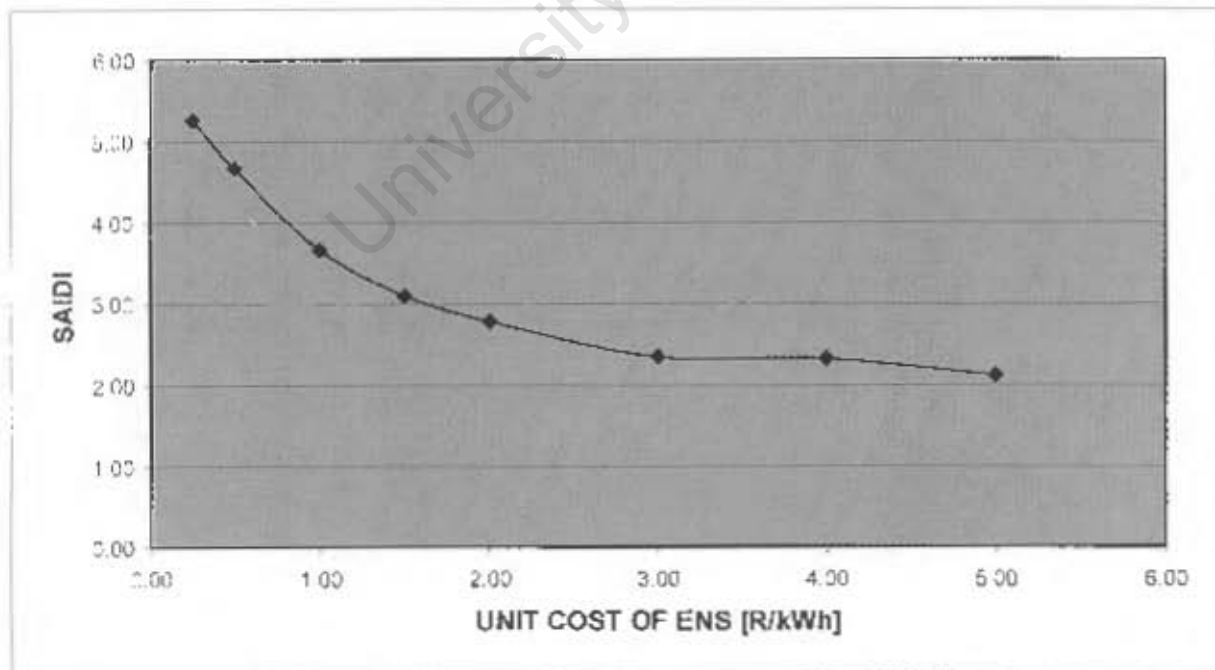
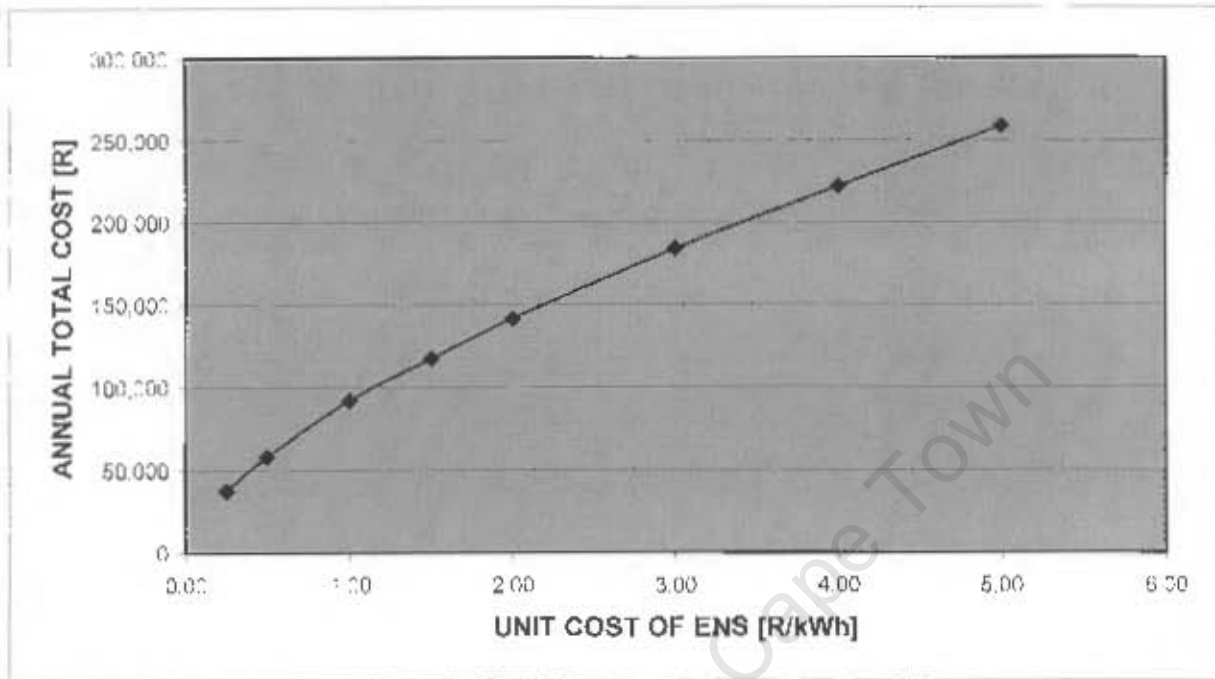
MINIMISATION OF THE ANNUAL COST OF ENS BY VARYING THE FAILURE RATE FOR THE LEVITIN NETWORK

FAILURE RATE	NUMBER OF SECTIONALISERS PLACED	SECTIONS SECTIONALISERS PLACED ON	SAIDI	ANNUAL COST OF SECTIONALISERS	ANNUAL COST ENS	TOTAL COST
[f/km/yr]			[int/yr/cust]	[R]	[R]	[R]
0.05	3	72,37,2	1.48	12174	12140	24314
0.10	3	72,36,2	2.87	12174	23080	35254
0.15	4	72,36,9,3	3.95	16232	31751	47983
0.20	5	72,39,36,4,2	4.67	20290	37365	57655
0.25	6	72,39,21,35,4,2	5.25	24348	42062	66410
0.30	6	72,39,35,,21,4,2	6.31	24348	50341	74689
0.35	7	72,57,39,36,21,4,2	6.84	28406	54613	83019
0.40	9	96,72,57,39,35,21,9,4,2	6.66	36522	53410	89932

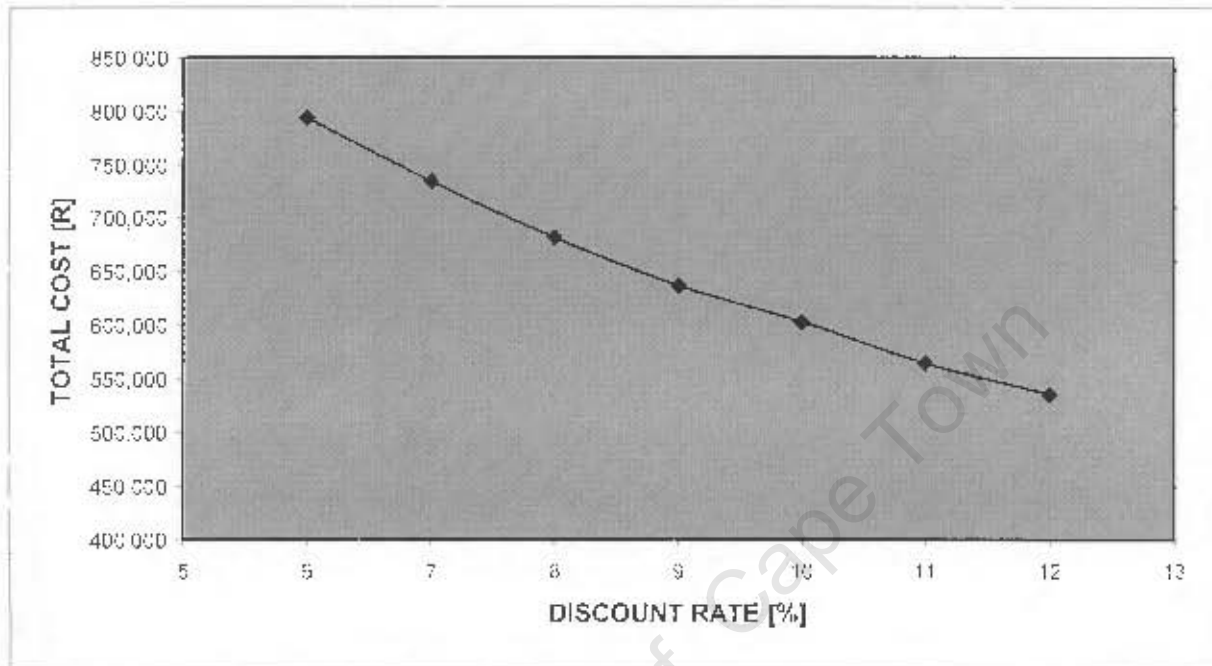
EFFECT OF FAILURE RATE ON ANNUAL TOTAL COST AND SAIDI FOR THE LEVITIN NETWORK



EFFECT OF C_e ON ANNUAL TOTAL COST AND SAIDI FOR THE LEVITIN NETWORK



EFFECT OF INCREASING THE DISCOUNT RATE ON THE TOTAL COST FOR THE LEVITIN NETWORK



University of Cape Town