ADDIS ABABA UNIVERSITY

INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

STUDY ON POWER LOSS MINIMIZATION FOR DISTRIBUTION NETWORK RECONFIGURATION USING GENETIC ALGORITHM

CASE STUDY: ADDIS NORTH 132/15 KV SUBSTATION

By

HABTAMU ESHETE

ADVISOR

Mr. KIROS TESFAY

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Electrical Engineering

June, 2018

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

STUDY ON POWER LOSS MINIMIZATION FOR DISTRIBUTION NETWORK RECONFIGURATION USING GENETIC ALGORITHM

CASE STUDY: ADDIS NORTH 132/15 KV SUBSTATION

Habtamu Eshete

Approval by Board of Examiners

_____	_____
Chairman, Dept. Graduate Committee	Signature
_____	_____
Advisor	Signature
_____	_____
Internal Examiner	Signature
_____	_____
External Examiner	Signature

Declaration

I, the undersigned, declare that this thesis is my original work, has not been Presented for a degree in this or any other university, and all sources of materials used for the thesis have been fully acknowledged.

Habtamu Eshete _____

Name Signature

Place: Addis Ababa Institute of Technology

Date of Submission: _____

This thesis has been submitted for examination with my approval as a university advisor.

_____ _____

Advisor's Name Signature

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor Mr. Kiros Tesfay for his help throughout the thesis including suggesting this interesting idea and guiding me to the end. I thank also, Addis Ababa University for giving me this opportunity to continue the MSc program. I would like to acknowledge Mr. Kassay and Abiy, who are Addis North 132/15 KV Substation worker for their cooperation and support.

ABSTRACT

Distribution system is a largest portion of network of electrical power system. Different actions have been taken to improve the efficiency of the distribution network. One of the most basic and the commonest way to improve the performance of the distribution network is the network reconfiguration. Electric distribution systems reconfiguration comprises tie and sectionalizing switches. Tie switches are normally open, and sectionalizing switches are normally closed. By opening and closing these switches, the distribution network can be reconfigured. This reconfiguration can be done for the objective of loss minimization. In order to get feasible results (loss minimization), the reconfiguration must meet some constraints, like Kirchhoff's voltage and current laws, other equality and inequality constraints.

Distribution network reconfiguration is an optimization problem and needs a suitable algorithm (method).The method used for this optimization problem is a genetic algorithm optimization method. The genetic algorithm has been described in detail and then applied specifically to the network reconfiguration problem. In the optimization process, load flow of the distribution system was computed. Then, computer simulation was performed by using DIgSILENT PowerFactory software for analyzing the distribution network reconfiguration. In addition, optimized genetic algorithm was used as a tool for network reconfiguration.

Addis North 132/15 kV substation feeders are used as test system for this particular study. Before the reconfiguration, the power loss is **3.983783 MW**, after reconfiguration the power loss has decreased from **3.983783 MW** to **1.594640 MW**, which is **2.389143 MW** (**59.91716%)** reduction. Besides, the maximum voltage drop before reconfiguration is **0.352639**, and the reconfiguration increased it to **5.050102**. The minimum voltage before reconfiguration is **0.992618 p.u**., and the minimum voltage is found to be **0.947288 p.u.** after reconfiguration. In addition, the maximum voltage before reconfiguration is **0.992069 p.u.,** and it is found to be **0.9953500 p.u.** after reconfiguration. Based on the findings of this research, it is concluded that reconfiguration of distribution networks can reduce power loss and operating cost as well as improves the voltage profile of distribution systems. Hence, it is recommended that all the distribution network feeders of Addis Ababa city to be reconfigured for the betterment of the Ethiopian Electric Utility (EEU) services.

Keywords: Network Reconfiguration, Genetic Algorithm Optimization, Power Loss Minimization

TABLE OF CONTENTS

ACKNOWLEDGEMENT III

ABSTRACT IV

LIST OF FIGURES VIII

LIST OF TABLES IX

LIST OF ACRONYMS X

Chapter 1 Introduction 1

1.1 Background 1

1.2 Motivation 3

1.3 Statement of the problem 3

1.4 Scope of the research 3

1.5 Objectives of the Study 3

1.5.1 General Objective 3

1.5.2 Specific Objectives 4

1.6 Methodology 4

1.7 Organization of the thesis 5

Chapter 2 6

Theoretical Background and Literature Review 6

2.1 Theoretical Background 6

2.2 Literature Review 6

2.2.1 Methods Based on Evolutionary Techniques for Network Reconfiguration 6

2.2.1.1 Genetic Algorithm (GA) 7

2.2.1.2 Simulated Annealing 8

2.2.1.3 Artificial Neural Network 8

2.2.1.4 Ant Colony Optimization 9

2.2.1.5 Particle Swarm Optimization 9

2.2.2 Knowledge Based Methods 10

2.2.2.1 Heuristic optimization methods For Network Reconfiguration ... 10
2.2.2.2 Linear programming Techniques ... 12
2.2.2.3 Fuzzy Logic Method ... 12
2.2.2.4 Tabu Search Method ... 13
2.2.3 Mixed Methods ... 13
Chapter 3 ... 15
Research Methodology ... 15
3.1 Problem formulation ... 15
3.1.1 Mathematical Model of the Network Reconfiguration Problem ... 15
3.2 Genetic Algorithm ... 18
3.2.1 Genetic Algorithm Optimization ... 18
3.2.2 How Genetic Algorithms Work ... 18
3.2.3 Genetic Algorithm Operators ... 21
3.2. 4 Encoding ... 30
3.3 DIgSILENT PowerFactory Overview ... 31
Chapter 4 ... 33
Simulation Studies and Analysis of Results ... 33
4.1 Addis North 132/15 KV Substation Test Systems ... 33
4.1.1 Addis North 132/15 KV Substation System Description ... 33
4.1.2 Addis North 132/15 KV Substation Simulation Result ... 33
4.2 Genetic Algorithm optimization result ... 40
4.3 Feasibility Analysis ... 43
Chapter 5 ... 44
Conclusions, Recommendations and Future Work ... 44
5.1 Conclusions ... 44
5.2 Recommendations ... 44
5.3 Future Work ... 45

References 46

Appendix A: Addis North 132/15 kV Substation feeders before reconfiguration 51

Appendix B: Addis North 132/15 kV Substation feeders network after reconfiguration..... 61

Appendix C: Addis North 132/15 kV Substation feeders data 71

Appendix D: Matlab program for reconfiguration of distribution network using Genetic algorithm 80

LIST OF FIGURES

Figure1.1 Addis North 132/15 kV Substation one line diagram 1
Figure 1.2 Distribution system structure..... 2
Figure 3.1 Optimization Process..... 18
Figure 3.2 Flow chart of Genetic Algorithm Optimization technique..... 20
Figure 3.3 Population, Chromosomes and Genes..... 21
Figure 3.4 Roulette wheel selections. [29], [30] 24
Figure 3.5 Rank selection effects. (a) Before ranking. (b) After ranking [31], [32], [33], [34] 25
Figure 3.6 Crossover point..... 26
Figure 3.7 Exchanging genes among parents 26
Figure 3.8 New offspring..... 26
Figure 3.9 Single Point Crossover 27
Figure 3.10 Multi point crossover..... 28
Figure 3.11 Uniform crossovers 28
Figure 3.12 Mutation: Before and After 29
Figure 3.13 Single point mutation 29
Figure 3.14 Multi point mutation..... 30
Figure 4.1 Feeders output calculation analysis chart before optimization..... 35
Figure 4.2 Feeders output calculation analyses chart after optimization..... 40
Figure A.1 Addis North 132/15 kV substation outgoing feeders online diagram 51
Figure A.2 Addis North 132/15 kV substation feeder 1 one line diagram 52
Figure A.3 Addis North 132/15 kV substation feeder 2 one line diagram 54
Figure A.4 Addis North 132/15 kV substation feeder 3 one line diagram 56
Figure A.5 Addis North 132/15 kV substation feeder 4 one line diagram 57
Figure A.6 Addis North 132/15 kV substation feeder 5 one line diagram 58
Figure A.7 Addis North 132/15 kV substation feeder 6 one line diagram 60
Figure A.8 Addis North 132/15 kV Substation feeders network after reconfiguration..... 70

LIST OF TABLES

Table 4-1 Feeder output calculation analysis before optimization 34
Table 4-2 Tie Open Point Optimization..... 35
Table 4-3 Optimal Tie Open Positions 36
Table 4-4 Necessary Switching Actions (From Initial to Optimized Configuration)..... 36
Table 4-5 Feeders Results..... 38
Table 4-6 Feeder output calculation analysis after optimization shown table below 39
Table 4-7 Genetic algorithm test results for feeder 1 and feeder 2..... 41
Table 4-8 Genetic algorithm test results for feeder 3 and feeder 4..... 41
Table 4-9 Genetic algorithm test results for feeder 5 and feeder 6..... 42
Table A-1 Network Data of Feeder 1 of Addis North 132/15 kV Substation 71
Table A-2 Network Data of Feeder 2 of Addis North 132/15 kV substation..... 74
Table A-3 Network Data of Feeder 3 of Addis North 132/15 kV substation..... 76
Table A- 4 Network Data of Feeder 4 of Addis North 132/15 kV substation..... 77
Table A-5 Network Data of Feeder 5 of Addis North 132/15 kV substation..... 77
Table A- 6 Network Data of Feeder 6 of Addis North 132/15 kV Substation 79

LIST OF ACRONYMS

F1	Feeder One
F2	Feeder Two
F3	Feeder Three
F4	Feeder Four
F5	Feeder Five
F6	Feeder Six
R	Resistance
X	Reactance
Z	Impedance
I	Current
P	Active Power
Q	Reactive Power
S	Apparent Power
S_{loss}	Loss Power
V_a	Voltage at node a
V_b	Voltage at node b
V_{min}	Minimum Voltage
V_{max}	Maximum Voltage

Chapter 1 Introduction

1.1 Background

Distribution system is a largest portion of network of electrical power system. Different actions have been taken to improve the efficiency of the distribution network. One of the most basic and the commonest way to improve the performance of the distribution network is the network reconfiguration. Feeder reconfiguration is performed by opening sectionalizing (normally closed) and closing tie (normally open) switches of the network.

Addis North 132/15 KV substation has six outgoing feeders. The one line diagram is shown in figure 1.1, which is obtained from Addis North 132/15 KV substation.

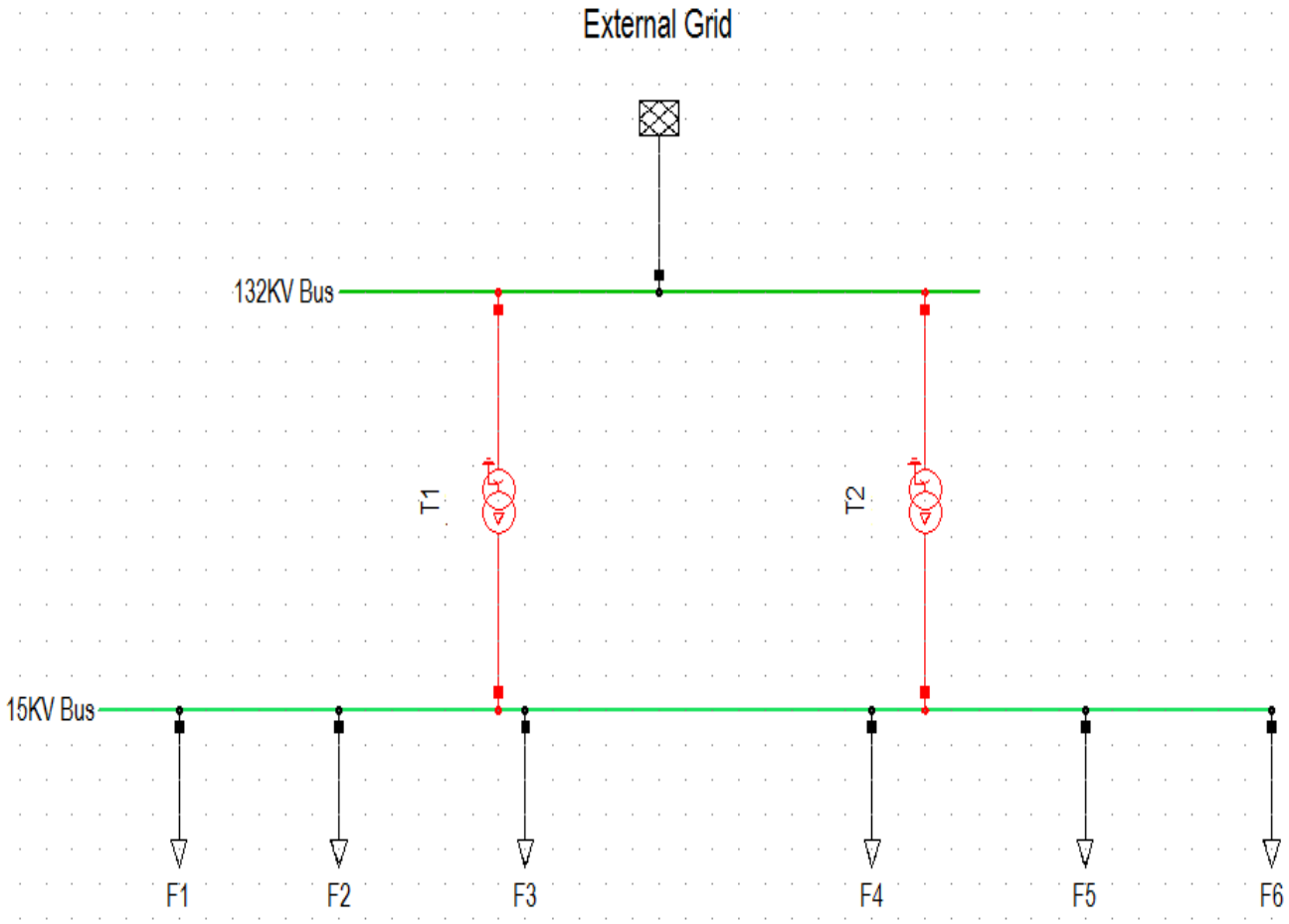


Figure1.1 Addis North 132/15 KV Substation one line diagram

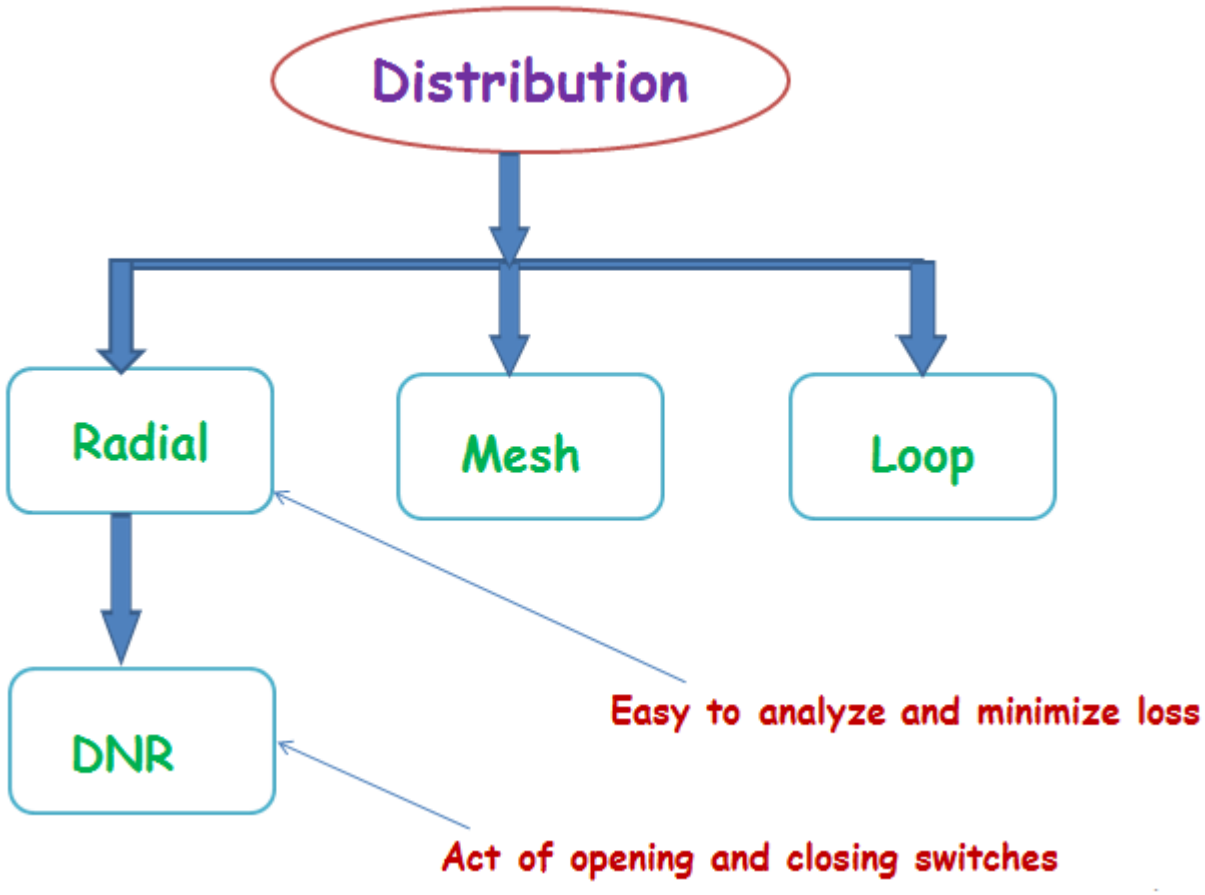


Figure 1.2 Distribution system structure

a. **Radial:** For radial, only one path between substation or service transformer and customer. The power flow is from substation to customer along single path. Furthermore, radial is cheap and predictable to use and simple to analyze. Besides, it is easy to analyze minimize power losses.
b. **Mesh:** For mesh, multiple paths between all point in the network. Power flow between any two points is split along several paths. It is most reliable method of distributing electricity. If failure occurs, power instantly reroutes itself. Other than that, it is usually used in high density urban area where maintenance and repairs are difficult and costly.
c. **Loop:** For loop, two paths between substation transformer and customer. Power flow is usually from both sides to the middle. The equipment is rated so that service can be maintained if an open point occurs in the system.

1.2 Motivation

In Addis Ababa, the growing industrialization and increasing standard of living considerably increase the usage of energy. The increasing demand of the electrical energy is quietly related to the power demand. The increasing electricity demand has made distribution system more complex and often caused power loss. To reduce the power loss, the network distribution system needs to be reconfigured.

1.3 Statement of the problem

In order to remain competitive, it is becoming more and more imperative for power distribution companies to be able to meet the demands of their customers efficiently. This means that one of their goals is finding an operating state for a large, balanced, three-phase, distribution network which minimizes the cost for the power company supplying the power, while satisfying the requirements of the customer. The demand for the electricity is rising due to the population growth. The distribution system has become more complex. The increasing of current draw during the distribution of electricity lead to the instability that often is the cause for power losses. Coordinated reconfiguration and upgrading of the distribution system can be a solution to this problem.

1.4 Scope of the research

This thesis is delimited to the study of power loss minimization in Addis North 132/15 KV Substation. The power loss and distribution system of the selected area has been first studied. This has been followed with reconfiguration of the networking system aiming to minimize the power loss.

1.5 Objectives of the Study

1.5.1 General Objective

The main objective of this thesis is to minimize the electrical power losses by considering reconfiguration method and applying a Genetic Algorithm for optimization in Addis North 132/15 KV distribution substation.

1.5.2 Specific Objectives

The specific objectives of the thesis are:

- To study the Addis North 132/15 kV substation and the distribution system.
- To formulate the objective function and solve the optimal reconfiguration problem using genetic algorithm.
- To study reconfiguration method of power loss reduction.
- To reconfigure the distribution system by the use of tie and sectionalizing switches.
- To minimize distribution losses by optimal reconfiguration of the existing network.
- To evaluate the distribution loss with and without reconfiguration of the distribution network.
- To investigate the voltage profile with and without reconfiguration of the distribution system.
- To draw conclusions based on the finding of this research and make recommendation to Ethiopia Electric Utility (EEU) for possible improvement in performance of Addis Ababa distribution system.

1.6 Methodology

The following methodology are adopted in carrying out this research and are itemized as follows;

i. Collection of data for radial distribution network of Addis North 132/15 KV distribution substation network.
ii. Establishment of the most suitable method for performing power flow analysis for the radial distribution network.
iii. Based on ii, a power flow analysis is carried out to estimate the initial configuration for distribution network.
iv. Development of genetic algorithm model used to determine the optimal location of tie and sectionalizing switches considering active power loss and total voltage profile.
v. Testing and validation of the developed model.
vi. A comparison of the reconfigure system with the original system is carried out, to determine the extent of loss reduction.

1.7 Organization of the thesis

The thesis is organized as follows. In Chapter 2, a theoretical background and literature about network reconfiguration and different optimization methods used earlier is reviewed. In chapter 3, a problem formulation and the new genetic algorithm methodology which is used in this thesis is discussed in detail. In Chapter 4, the Addis North 132/15 KV substation distribution network is reconfigured using DIgSILENT PowerFactory and the genetic algorithm optimization using Matlab softwares simulation results and discussions are discussed in detail. In chapter 5, Conclusion, recommendation and future work are made in the last chapter.

Chapter 2

Theoretical Background and Literature Review

2.1 Theoretical Background

More than 40 years, the French engineers Merlin and Back [8] introduced the opportunity to reduce technical losses by exploring a change in the status of normally closed and normally open switches. They proposed the "network reconfiguration problem", for which the solution should provide the best status for all the switches in a primary distribution network, best in the sense that they provide a radial configuration supplying loads with the minimum of power loss. Between 1988 and 1990, heuristic methods were used to solve the problem. The developments during this period focused on increasing the number of operating constraints. From 1990, new solution strategies appeared: linear programming simulated annealing, and genetic algorithms, whose objective function is power losses minimization and the operating constraints previously used. In addition, load models are improved with more precise models.

In 1993, solutions to the problem were presented through neural networks, which initially model few operating constraints and simple load models. In 1997, models with more constraints were used. Between 1995 and 1996 the heuristic method was proposed again in order to optimize energy losses using more precise load models. From 1997 until now, the techniques used are combinations of the previous techniques, aiming to complement each method's strengths. Also new methods like Ant Colony (ACO) and Particle Swarm Optimization (PSO) are used. According to the historic development, the computational searching methods are classified into three large groups:

- Methods based on evolutionary techniques for network reconfiguration.
- Knowledge-based methods for network reconfiguration.
- Mixed methods for network reconfiguration.

2.2 Literature Review

2.2.1 Methods Based on Evolutionary Techniques for Network Reconfiguration

Evolutionary algorithms (EAs) are developed to arrive at optimum or near-optimum solutions to a large scale optimization problem. The problem having very large number of decision variables and non-linear objective functions are often solved by EAs. EAs mimic the metaphor of natural biological evolution or social behavior like how ants find the shortest route to a source of food and

how birds find their destination during migration. The behavior of such species is guided by learning and adaptation. The evolutionary algorithms are based on population based search procedures that incorporate random variation and selection.

EAs have an important characteristic: the lack of a rigorous mathematical formulation that allows establishing their operation in each situation with certainty. These techniques start from a solution and improve it. Genetic algorithm, simulated annealing, neural networks, ant colony (ACO), particle swarm optimization (PSO), etc. belong to this category. Genetic algorithm is a search based on the mechanism of natural selection and natural genetics. It can be used to solve the multi objective optimization problem. It is used by [1], [2], [3]. Simulated annealing method can avoid local optima but requires an excessive computation time. It is used by [4]. Artificial neural network methods were also used by [5].

2.2.1.1 Genetic Algorithm (GA)

The first evolutionary-based optimization technique was the genetic algorithm (GA).Genetic algorithms have become very popular as a method of finding global optimums. As applied to reconfiguration [1], [2], the switch states are encoded in strings of 0/1 "chromosomes", and a population of, for example, 30 topologies is built at random. At each iteration, two parent topologies are selected at random for crossbreeding which is a process of combining the chromosomes according to some defined algorithm. Then mutation, a random alteration of some chromosomes, may occur with a certain probability. If the resulting child is better, it replaces an existing topology in the population of 30. This process of crossbreeding continues for a number of iterations. The population also has to be re-seeded periodically with random strings to avoid in breeding. As the population evolves, there will always be a best solution that should steadily improve.

Genetic algorithms are most attractive for parallel processing environments, and when each child can be evaluated quickly. The cross breeding and mutation algorithms must be custom designed and tested for each application. Parameters such as the number of crossbreeding per generation, mutation probability, number of generations, population size, and percentage of population reseeded must all be determined by testing.

Applications to reconfiguration have used simplified network analysis because many thousands of topologies are considered, so the resulting solution may not be optimal with a more detailed model.

Nara [2] used the genetic algorithm (GA) which is a search algorithm based on the mechanism of natural selection and natural genetics. It combines the adaptive nature of the natural genetics or the evolution procedure of organs with functional optimization. The simple feature of GA makes it suitable for different multi objectives optimization problem. The principle problem in using GA rests on an efficient coding and decoding mechanism of the chromosome representing the distribution network and the structure of fitness function.

Fudou, Fukuyama and Nakanishi [3] present a GA using three phases unbalanced load flow. A proper string representation for loads and power supplies is devised and a method to yield a good problem dependent initial string population is presented. A repair operator which modifies the string so as to improve the objective function of the problem and to satisfy the radial network constraints. A modification to the fitness function is made to reinforce the satisfaction of the power source limits and voltage as well as current constraints.

2.2.1.2 Simulated Annealing

The name and inspiration came from annealing process in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one. In the simulated annealing method each point of the search space is compared to a state of some physical system, and the function to be minimized is interpreted as the internal energy of the system in that state. Therefore the goal is to bring the system, from an arbitrary initial state, to a state with the minimum possible energy. Ray Daniel Zimmerman [4] used simulated annealing for loss reduction of three phase power distribution system. It is used on larger, more complex, unbalanced three-phase system. Computer program was developed and was able to find the optimal configuration of the 147 bus, 12 switch example systems, demonstrating the feasibility of such an approach for the solution of the network reconfiguration problem.

2.2.1.3 Artificial Neural Network

Ali Reza Fereidunian, Hamid Lesani and Caro Lucas [5] used an intelligent neural optimizer with two objective functions which is designed for electrical distribution systems. The presented method is faster than alternative optimization methods and is comparable with the most powerful and

precise ones. The optimizer is much smaller than similar neural systems. The proposed method has established a relation between two applications of neural networks: Optimization and Pattern Recognition.

H. Kim [6] presented the strategy of feeder reconfiguration to reduce the power loss by artificial neural (ANN) network. This approach developed is basically different in the aspects that the load transfer and the corresponding load flow solution during the search process are not required. The set of ANN is the optimal system topology corresponding to various load patterns which minimizes the load under given conditions.

2.2.1.4 Ant Colony Optimization

Ant colony (ACO) optimization is based on ant social behavior. In the real world, ants (initially) wander randomly, and upon finding food return to their colony while laying down pheromone trails. If other ants find such a path, they are likely not to keep traveling at random, but instead follow the trail laid by earlier ants, returning and reinforcing it, if they eventually find any food. Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate. A short path, by comparison, gets marched over faster, and thus the pheromone density remains high as it is laid on the path as fast as it can evaporate. Thus, when one ant finds a good (short) path from the colony to a food source, other ants are more likely to follow that path, and such positive feedback eventually leaves all the ants following a single path.

It is used by [9], [10], [11] used ACO for optimal reconfiguration of distribution network for loss minimization and load balancing. Matlab program which uses ACO was developed and tested on Baran and Wu 33 bus system and two large practical distribution systems. The results were also validated using Power World simulation software.

2.2.1.5 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) is inspired by the social behavior of a flock of migrating birds trying to reach an unknown destination. In PSO, each solution is a bird in the flock and is referred to as a particle. A particle is analogous to a chromosome in GAs. As opposed to GAs, the evolutionary process in the PSO does not create new birds from parent ones. Rather, the birds in the population only evolve their social behavior and accordingly their movement towards a destination. Each bird looks in a specific direction, and then when communicates together, they

identify the bird that is in the best location. Accordingly, each bird speeds towards the best bird using a velocity that depends on its current position. Each bird, then investigates the search space from its new local position, and the process repeats until the flock reaches a desired destination

Tamer M. Khalil, and Alexander V. Gorpinich [7] used selective particle swarm optimization (SPSO) for loss reduction. This algorithm is a simple modification to the binary particles warm optimization (BPSO). The search space of the algorithm is a set of branches (switches) which are normally closed or normally opened, this search space may be dissimilar for different dimensions. The process of solving reconfiguration problem is divided in to two steps. First, finding search spaces after closing all switches and second, using SPSO to find switches that would be opened. The presented technique is applied to a 33-node system and a 69-node system. The results obtained via SPSO are compared with some previous methods to demonstrate the effectiveness of the proposed algorithm.

2.2.2 Knowledge Based Methods

They are based on the operators' experience in the system operations. Based on this Knowledge, algorithms have been designed to facilitate searching for the new distribution network configuration, trying to find an option close to the optimal. Heuristic methods, linear programming, expert systems, fuzzy logic etc, belong to this category. Heuristic methods used many times by different researchers. The solution process leads to the optimum or near optimum in less computation time.

2.2.2.1 Heuristic optimization methods For Network Reconfiguration

Several methods have been proposed to solve the reconfiguration problem. In 1975, Merlin and Back [8] proposed a branch and bound type heuristic method to determine the network configuration for minimum line losses. Its solution scheme starts with a meshed network by initially closing all switches in the network. The switches are then opened one at a time until a new radial configuration is reached. In this process the switch to be opened at each stage is selected in order to minimize line losses of the resulting network.

Branch and bound method will work better if the initial solution is close to optimal, because more pruning will occur. Branch and bound also benefits from breaking the problem into sub problems, each of which can be optimized separately. Even so, branch and bound is a combinatorial method and hence too slow for practical use. Therefore, most of the recent work on reconfiguration has

used either a branch exchange method or sequential switch opening method. Heuristics are applied in most cases to reduce the number of switching options considered.

D. Shirmomohammadi and Hong [12] improved the method of Merlin and Back. As a result, it shares the two principle benefits of that methodology, convergence to the optimum or near optimum solution and the independence of the final solution from the initial status of the network switches. At the same time, this method avoids all the major drawbacks of Merlin and Back. Civanlar [13] developed a branch exchange method. In this method, loss reduction is achieved by exchange operation corresponds to the selection of a pair of switches, one for opening and the other for closing so that the resulting network has lower line losses while remaining connected and radial.

The major drawbacks of this method are:

- The final network reconfiguration is dependent on the initial state of the network switches.
- Optimum solution is not guaranteed.
- Selection of each switches exchange operation becomes very time consuming.

Baran and Wu [14] presented a heuristic reconfiguration methodology based on the branch exchange to reduce losses and balance the loads in the feeders. To assist in the search, two approximated load flows for radial networks with different degrees of accuracy are used. They are simple power flow method and back and forward update of power flow method. The method is very time consuming due to the complicated combinations in large scale system and converges to a local optimum solution, that is, convergence to the global optimum is not guaranteed.

It is used by [15], [16] used a combined method, the switch exchange (SEM) and sequential switch opening method (SSOM) for reconfiguration of the network for loss reduction.

Its main advantage is optimum or near optimum configuration is obtained using one of the two basic methods, SEM where closing switch and opening another in the loop formed. SSOM where all tie switches are initially closed and an optimal load flow is obtained. The system is returned to a radial configuration by successive opening of the switches having the least current flow until network radiality is obtained. SEM requires less computation time and SSOM is independent from the initial configuration and thus more likely leads to the actual optimum.

Broadwater and Khan [17] suggested a reconfiguration algorithm which calculates switch patterns as a function of time. Either seasonal or daily time studies may be performed. Both manual and automatic switches are used to reconfigure the system for seasonal studies whereas only automatic switches are considered for daily studies. Such a continuous reconfiguration is allowed by today's distribution automation, information technology and equipment. It is shown that switching at the system peak can reduce losses but cause a marginal increase in system peak. The practical aspects of such an optimization remain to be carefully analyzed through costs, transient effect and influence to system reliability.

R. Safri, M.M.A Salama and Y. Chickani [18] proposed an algorithm that is based on network portioning into groups of load buses such that the line section losses between the groups of nodes are minimized, the proposed method overcomes the size restriction imposed by reconfiguration techniques. By dividing the distribution network into groups of buses, the combinatorial nature of reconfiguration problem is reduced while simultaneously minimizing losses.

2.2.2.2 Linear programming Techniques

A. Borghetti, M. Paolone and C.A. Nucci [19] used mixed integer linear programming (MILP) model for the solution of the minimum loss reconfiguration problem of distribution networks, including embedded generation. The proposed MILP model takes into account the main operating constraints other than radiality, such as the lower and upper bounds of the bus voltages and the upper limits of the line currents. The solution of the MILP model does not require the knowledge of an initial feasible configuration of the network.

G. Celli, M. Loddo, F. Pilo and A. Abur [20] formulated and solved the network reconfiguration problem with DGs using a simple linear programming approach. Optimal configurations are determined by considering the effects of DG outputs, load variations, and various other contingences such as faults and maintenance outages. Demand Side Management actions have also been taken into account.

2.2.2.3 Fuzzy Logic Method

Ramadoni Syahputra [21] presented a fuzzy multi-objective approach for achieving the minimum active power loss and the maximum voltage magnitude in order to improve the efficiency of radial distribution networks with distributed generations. Multi-objective function are considered for load balancing among the feeders, minimization of the real power loss, deviation of nodes voltage, and

branch current constraint violation, while subject to a radial network structure in which all loads must be energized. The effectiveness of the method has been demonstrated by a 70-node distribution network test system.

Abhisek Ukil and Willy Siti [22] used Fuzy Logic for load balancing. They presented a fuzzy logic-based load balancing system along with a combinatorial optimization-based implementation system for implementing the load changes. The input to the fuzzy step is the total load (kW) per phase of the feeders. Output of the fuzzy step is the load change values, negative value for load releasing and positive value for load receiving. Sum of the positive and negative values is zero, i.e., the total load remains unchanged for the entire phase balancing.

2.2.2.4 Tabu Search Method

Tabu Search is a meta-heuristic that guides a local heuristic search procedure to explore the solution space beyond local optimality. One of the main components of Tabu search is its use of adaptive memory, which creates a more flexible search behavior. Memory-based strategies are therefore the hallmark of Tabu search approaches, founded on a quest for "integrating principles," by which alternative forms of memory are appropriately combined with effective strategies for exploiting them.

N. Rugthaicharoencheep and S. Sirisumrannukul [23] used Tabu Search (TS) for loss reduction of distribution networks with distributed generation. The developed methodology is tested with a 69-bus distribution system having 48 load points. The study results indicate that for a given set of distributed generators and their locations, the proposed method can identify optimal on/off patterns of the switches that yield the minimum loss while satisfying the constraints.

2.2.3 Mixed Methods

These methods are in use since 1996 and are combinations of previous methods to gather their combined strengths; hence better results are obtained. Mixed methods like linear programming with heuristics [24] and Fuzzy Logic and heuristics [25] were used.

King and Radha [26] used a fuzzy logic controller to adapt the cross over and mutation probabilities based on the fitness function. The main advantages of fuzzy control system over the conventional method are: ability of modeling the quantities aspects of human knowledge and reasoning process, model free estimator, robustness, and easy implementation. The fuzzy logic

controlled GA always finds the global optimum and has proved to have faster convergence than a GA using fixed cross over and adaptive mutation.

Mehdi Assadian [27] investigated the ability of particle swarm optimization (PSO) in cooperation with graph theory for network reconfiguration to reduce the power loss and voltage profile enhancement of distribution system.

Chapter 3

Research Methodology

3.1 Problem formulation

It is becoming more and more important to power distribution companies to be able to meet efficiently and reliably the demands of their customers. This means that one of their goals is to be able to find an operating state for a large, three-phase, distribution network which minimizes the loss for the power company supplying the power, while satisfying the requirements of the customer. In developing countries like Ethiopia, the utility companies should also reconfigure and redesign their distribution system in order to improve the reliability of their distribution system, which is unacceptably low by many standards. This chapter states introduces some mathematical model of network reconfiguration problem and presents a formulation of the network reconfiguration problem of loss reduction and besides, Genetic Algorithm Optimization.

3.1.1 Mathematical Model of the Network Reconfiguration Problem

Radial distribution system reconfiguration is done by opening/closing two types of switches, tie switches and sectionalizing switches. A feeder may be served for another feeder by closing a tie switch linking the two while a particular sectionalizing switch must be opened to maintain radial structures. In case of power loss reduction, the problem here to be addressed is to identify tie and sectionalizing switches that should be closed and opened, respectively, to achieve a maximum reduction in power losses. Theoretically, it is a straightforward matter to determine whether or not, the new system obtained through a feeder reconfiguration would incur lower power losses. The reduction in power losses can easily be computed from the results of two load flow studies of the system configurations before and after the feeder reconfiguration.

Let us consider a distribution network of n nodes. The optimization problem is then finding an optimal radial network u among all possible radial networks of search space S generated with the switch condition changes that minimize the objective function without violation of the constraints.

a. Objective Function

Assuming that $S = P + jQ$, and it turns out that $|S|$ is the total power that is transported through the circuit component in order to get a usable active power P. We call S the apparent power. It is clear from the definitions that a smaller phase difference results in a smaller reactive (useless) power, hence a smaller difference between the active power and the apparent power. In electrical

engineering it is important to take into account the reactive power as much as the active power. So we should work with the complex value S, and not just with the real value P. Now we can calculate the power loss through a line. If I is the complex current through a line from a to b with voltages Va, V_b, and Z = R + jX the impedance of this line, then the apparent power loss S_{loss} can be calculated simply as the difference between the powers at a and b.

$$S_{Loss} = S_a - S_b \quad (3.1)$$

$$= (V_a - V_b)I^* \quad (3.2)$$

$$= IZI^* \quad (3.3)$$

$$= Z\,|I|^2 \quad (3.4)$$

$$= R\,|I|^2 + j\,X\,|I|^2 \quad (3.5)$$

Using Ohm's law, now the reactance X only gives information about the phase difference between the current and the voltages due to the cable, where the resistance R induces the permeability of the cable. Only the latter plays a role for power loss, so the actual power loss is $P_{loss} = |I|^2R$. So that the objective of the optimal feeder reconfiguration problem to minimize the total power loss can be expressed as:

Minimize $P_L = \sum_{i=1}^{Nb} I_i^2 R_i$ (3.6)

Where, P_L = total power loss

N_b = number of branches

I_i = current flow in i^{th}branch

R_i = Resistance of i^{th}branch.

b. Constraints:

The objective function in equation 1.6 is subjected to the following constraints.

i. Bus voltage limits

Bus voltage limits is well known that a small change in nodal voltage affects the flow of reactive power whereas active power practically does not change. Further, the operating voltage at each node must be in safety range as given below.

$V_{imin} \le V_i \le V_{imax}$ $\quad i \in (1,2,3, \ldots, N_b)$

Where,

V_{imine} = Minimum voltage limits of i^{th} node respectively.

V_{imax} = Maximum voltage limits of i^{th} node respectively.

V_i = Voltage at i^{th} node

V_b = Number of buses.

ii. Feeder capacity limits

Power flow in each branch must be less than or equal to its maximum capacity as given below.

$$|I_i| \leq I_{imax} \qquad i \varepsilon \{1, 2, 3 \ldots, N_b\}$$

Where, I_{imax} = maximum current capacity of i branch

I_i = current in i^{th} branch

iii. Bus isolation

All nodes or buses have to be served after reconfiguration. The node must not be isolated without output supply from any feeder. It means only one switch should be opened in a loop.

3.2 Genetic Algorithm

Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and Natural Selection. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems, in research, and in machine learning.

One of the more challenging aspects of using genetic algorithms is to choose the configuration parameter settings. Discussion of Genetic Algorithm (GA) theory provides little guidance of proper selection of the settings. The population size, the mutation rate, and the type of recombination have the largest effect on search performance. They are used to control the run of a Genetic Algorithm (GA). They can influence the Population and the Reproduction part of the Genetic Algorithm GAs. In traditional GAs the parameters have fixed values [28].

3.2.1 Genetic Algorithm Optimization

Genetic algorithm optimization is the process of making something better. In any process, we have a set of inputs and a set of outputs as shown in figure 3.1.

Figure 3.1 Optimization Process

Besides, optimization refers to finding the values of inputs in such a way that we get the best output values. The definition of best varies from problem with problem, but in mathematical terms, it refers to maximizing or minimizing one or more objective functions, by varying the input parameters. The set of all possible solutions or values which the inputs can take make up the search space. In this search space, lies a point or a set of points which give the optimal solution. The aim at optimization is to find that points or set of points in the search space.

3.2.2 How Genetic Algorithms Work

Genetic algorithm maintains a population of individuals, say **P(t)**, for generation **t**. Each individual represents a potential solution to the problem at hand. Each individual is evaluated to give some

measure of its fitness. Some individuals undergo stochastic transformations by means of genetic operations to form new individuals. There are two type of transformation:

a. Mutation, which creates new individuals by making changes in a single individual.
b. Crossover, which creates new individuals by combining parts from two individuals.

The new individuals, called offspring **C (t)**, are then evaluated. A new population is formed by selecting the more fit individuals from the parent population and offspring population. After several generations, genetic algorithm converges to the best individual, which hopefully represents an optimal or suboptimal solution to the problem.

The general structure of the Genetic algorithms is as follow:

Begin

{

t=0;

Initialize **P(t)**;

Evaluate **P(t)**;

While (not termination condition) do Begin

{

Apply crossover and mutation to **P(t)** to yield **C(t)**;

Evaluate **C(t)**;

Select **P(t+1)** from **P(t)** and **C(t)**;

t=**t**+1;

}

End

}

End

The flowchart explains how genetic algorithms work is shown in figure 3.2.

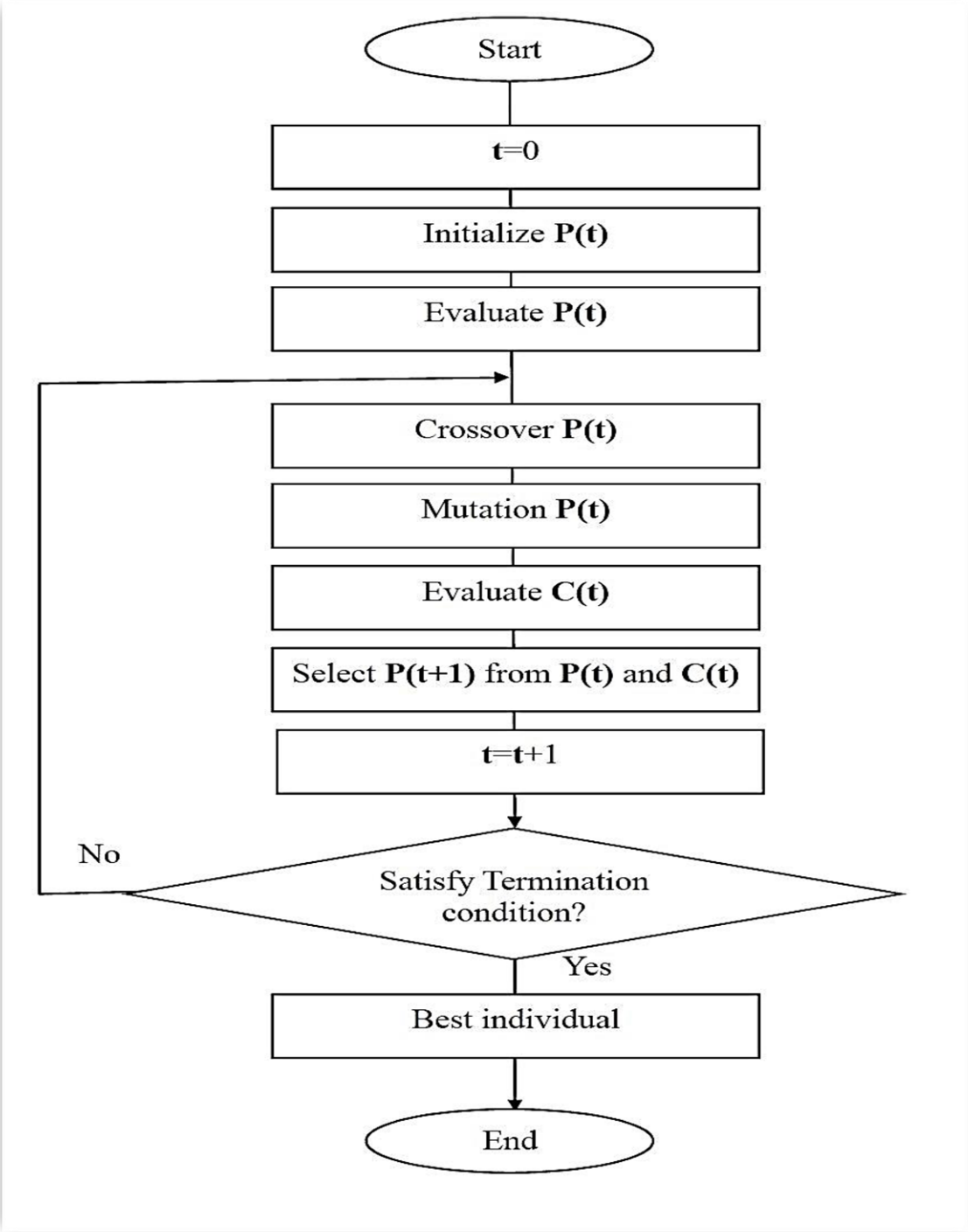


Figure 3.2 Flow chart of Genetic Algorithm Optimization technique

3.2.3 Genetic Algorithm Operators

There are five basic genetic algorithms operators which are Population, Fitness function, Selection, Crossover and mutation.

A. Population

The process begins with a set of individuals which is called a Population**.** Each individual is a solution to the problem you want to solve. An individual is characterized by a set of parameters (variables) known as Genes**.** Genes are joined into a string to form a Chromosome (solution).

In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. Usually, binary values are used (string of 1s and 0s). We say that we encode the genes in a chromosome, as shown in fig 3.3.

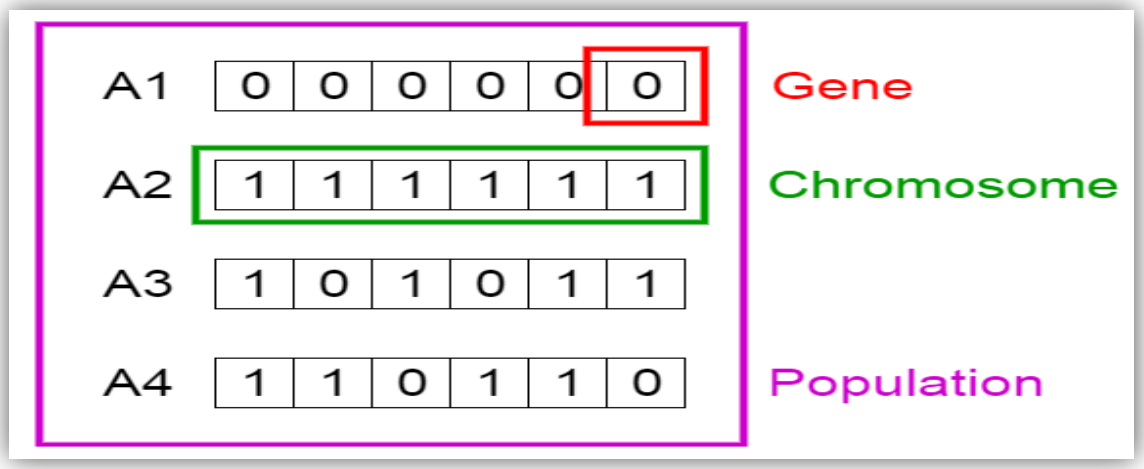


Figure 3.3 Population, Chromosomes and Genes

The population size is one of the most important parameters that play a significant role in the performance of the genetic algorithms. The population size dictates the number of individuals in the population. Larger population sizes increase the amount of variation present in the initial population at the expense of requiring more fitness evaluations. It is found that the best population size is both applications dependent and related to the individual size (number of chromosomes within). A good population of individuals contains a diverse selection of potential building blocks resulting in better exploration. If the population loses diversity the population is said to have "premature convergence" and little exploration is being done. For larger individuals and challenging optimization problems, larger population sizes are needed to maintain diversity (higher

diversity can also be achieved through higher mutation rates and uniform crossover) and hence better exploration. Many researchers suggest population sizes between 25 and 100 individual, while others suggest that it must be very much larger (1000 individual or more).

B. Fitness Function

The fitness function determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a fitness value to each individual. The probability that an individual will be selected for reproduction is based on its fitness value.

C. Selection

The idea of selection phase is to select the fittest individuals and let them pass their genes to the next generation. Two pairs of individuals **(**parents**)** are selected based on their fitness values [36]. Individuals with high fitness have more chance to be selected for reproduction.

Selection is the process of determining the number of times a particular individual is chosen for reproduction and, thus, the number of offspring that an individual will produce. The principle of genetic algorithms is essentially Darwinian natural selection. Selection provides the driving force in genetic algorithms. With too much force, genetic search will terminate prematurely. While with too little force, evolutionary progress will be slower than necessary.

Typically, a lower selection pressure is indicated at the start of genetic search in favor of a wide exploration of the search space, while a higher selection pressure is recommended to the end to narrow the search space. In this way, the selection directs the genetic search for promising regions in the search space and that will improve the performance of genetic algorithms. Many selection methods have been proposed, examined and compared. The most common types are:

i. Roulette wheels selection
ii. Rank selection
iii. Tournament selection
iv. Steady state selection
v. Elitism

i. Roulette Wheel Selection

Roulette wheel selection is the commonest selection method used in genetic algorithms for selecting potentially useful individuals (solutions) for crossover and mutation.

In roulette wheel selection, as in all selection methods, possible solutions are assigned fitness by the fitness function. This fitness level is used to associate a probability of selection of each

individual. While candidate solutions to a higher fitness will be less likely to be eliminated, there is still a chance that they may be. With roulette wheel selection there is a chance some weaker solutions may survive the selection process; this is an advantage, as though a solution may be weak, it may include some component which could prove useful following the recombination process. The analogy between a roulette wheel can be envisaged by imagining a roulette wheel in which each candidate solution represents a pocket on the wheel; the size of the pockets is proportionate to the probability of selection of the solution. Selecting N individual from the population is equivalent to playing N games on the roulette wheel, as each candidate is drawn independently, as shown in figure 3.4.

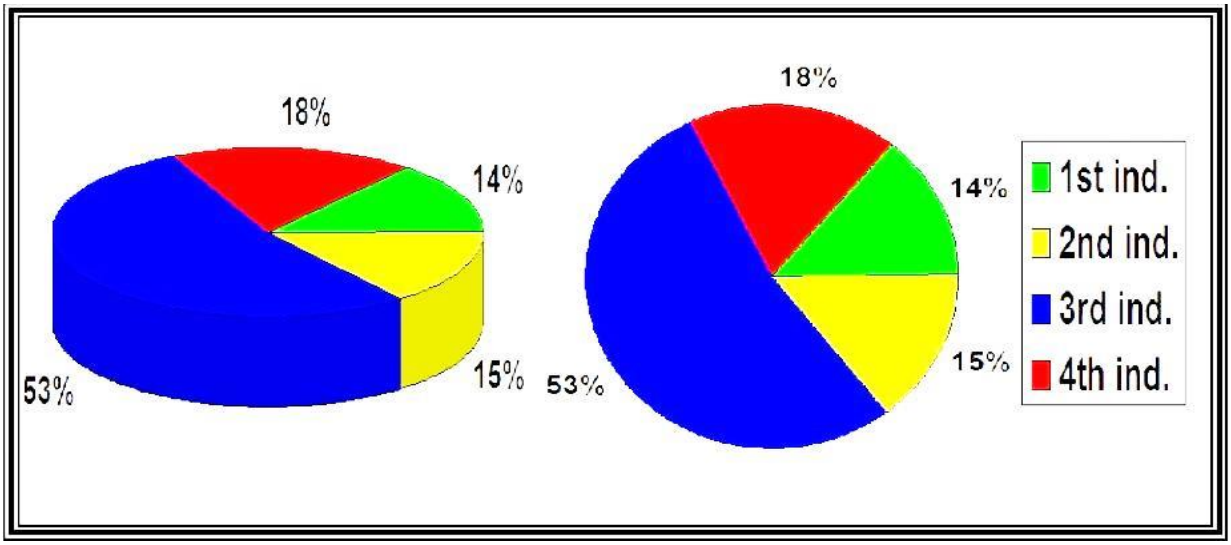


Figure 3.4 Roulette wheel selections. [29], [30]

ii. **Rank Selection**

In ranking selection, the individuals in the population are sorted through best to worst according to their fitness values. Each individual in the population is assigned a numerical rank based on fitness, and selection is based on this ranking rather than differences in fitness. The advantage of this method is that it can prevent very fit individuals from gaining dominance early at the expense of less fit ones, which would reduce the population's genetic diversity and might hinder attempts to find an acceptable solution. The disadvantage of this method is that it required sorting the entire population by rank which is a potentially time consuming procedure. Rank selection effect is shown in figure 3.5 (a and b).

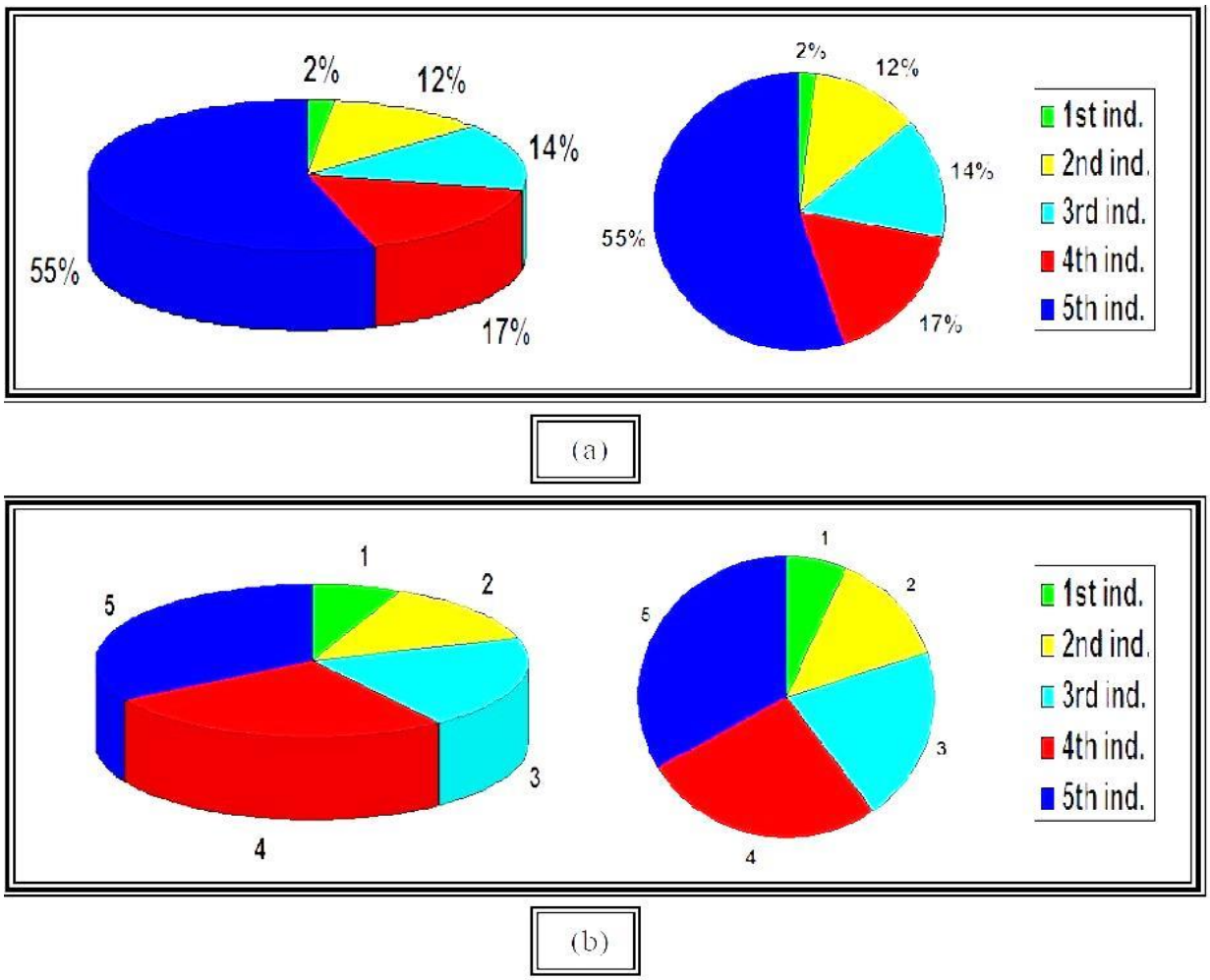


Figure 3.5 Rank selection effects. (a) Before ranking. (b) After ranking [31], [32], [33], [34]

iii. Tournament Selection

This method randomly chooses a set of individual and picks out the best individual for reproduction. The number of individual in the set is called the tournament size. A common tournament size is two, this is called binary tournament. By adjusting tournament size, the selection pressure can be made arbitrarily large or small. For example, using large Tournament size has the effect of increasing the selection pressure, since below average individuals are less likely to win a tournament while above average individuals are more likely to win it [35].

iv. Steady State Selection

The steady state selection will eliminate the worst of individuals in each generation [41]. It works; the offspring of the individuals selected from each generation go back into the pre-existing population, replacing some of the less fit members of the previous generation.

v. Elitism

Elitism is an addition to many selection methods that force genetic algorithms to retain some number of the best individual at each generation. It improves the selection process and save the best individuals. With elitist selection, the quality of the best solution in each generation monotonically increases from time. Without elitist selection, it is possible to lose the best individuals due to stochastic errors (due to crossover, mutation or selection pressure) [42].

D. Crossover

Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a crossover point is chosen at random from within the genes. For example, consider the crossover point to be three, as shown below in figure 3.6.

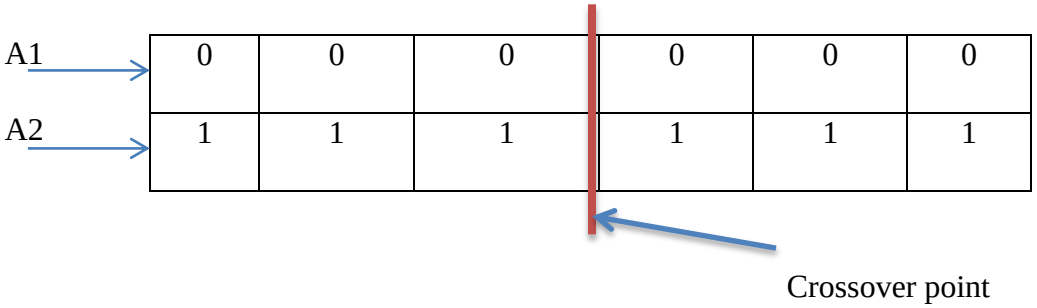


Figure 3.6 Crossover point

Offspring are created by exchanging the genes of parents among themselves until the crossover point is reached, as shown in figure 3.7. And the new offspring are added to the population, as shown in figure 3.8.

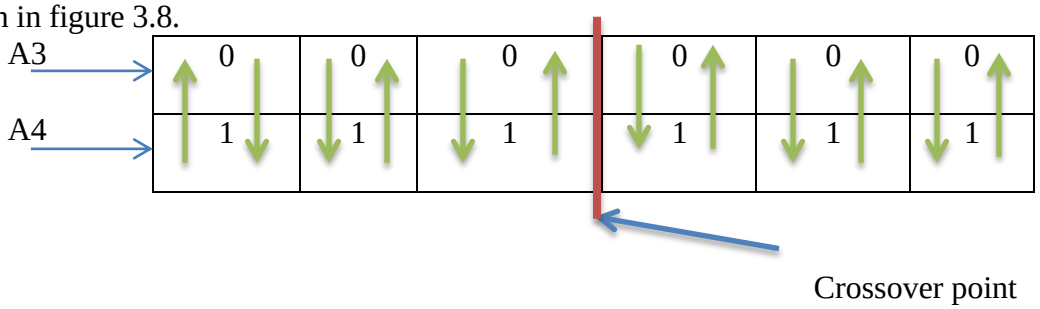


Figure 3.7 Exchanging genes among parents

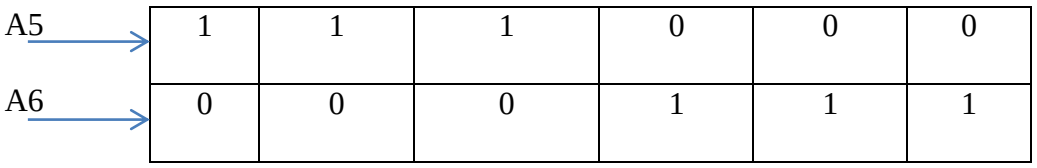


Figure 3.8 New offspring

One of the unique aspects of the work involving genetic algorithms (GAs) is the important role that Crossover (recombination) plays in the design and implementation of robust evolutionary systems. In most GAs, individuals are represented by fixed-length strings and crossover operates on pairs of individuals (parents) to produce new strings (offspring) by exchanging segments from the parents' strings. Crossover rate determines the probability that crossover will occur. The crossover will generate new individuals in the population by combining parts of existing individuals. The crossover rate is usually high and 'application dependent'. Many researchers suggest crossover rate to be between 0.6 and 0.95. [37], [38], [39], [40]

vi. **Single Point Crossover**

A commonly used method of crossover is called single point crossover. In this method, a single point crossover position (called cut point) is chosen at random (e.g., between the 10th and 5th point) and the parts of two parents after the crossover position are exchanged to form two offspring, as shown in figure 3.9.

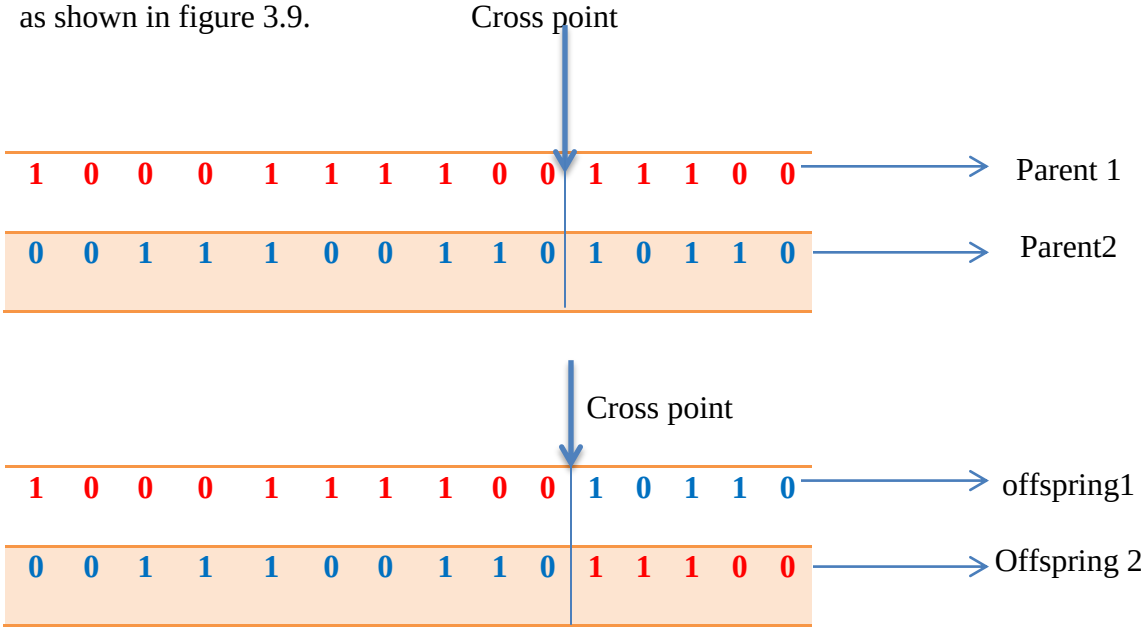


Figure 3.9 Single Point Crossover

vii. Multi Point Crossover

Multi-point crossover is a generalization of single point crossover, introducing a higher number of cut-points. In this case multi positions are chosen at random and the segments of them are exchanged, as shown in figure 3.10.

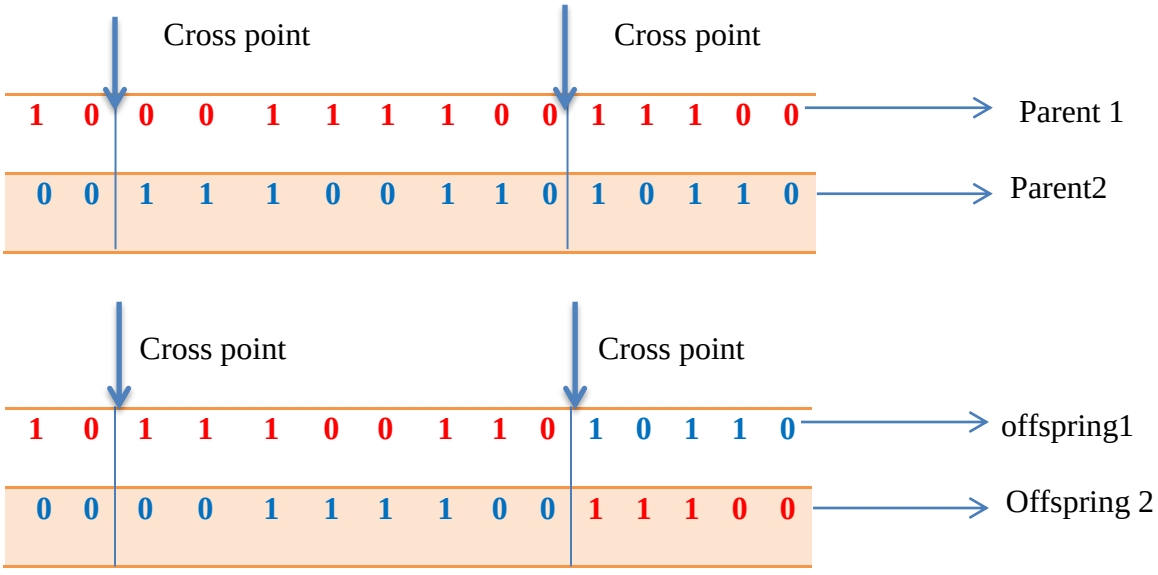


Figure 3.10 Multi point crossover

viii. Uniform Crossover

Uniform crossover does not use cut-points, but simply uses a global parameter to indicate the likelihood that each variable should be exchanged with two parents, as shown in figure 3.11.

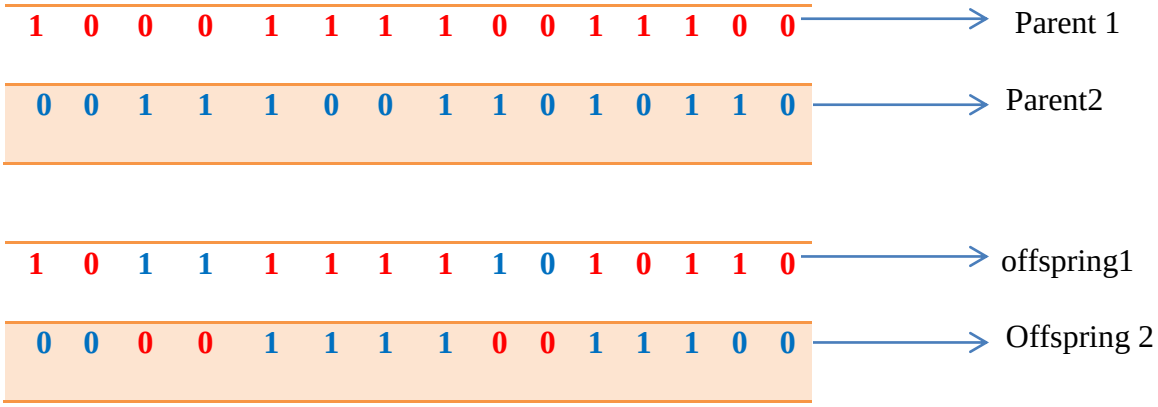
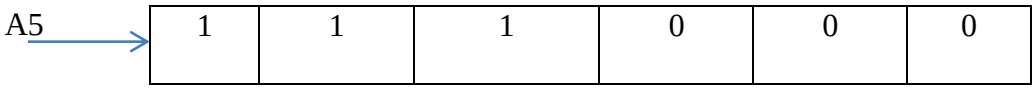


Figure 3.11 Uniform crossovers

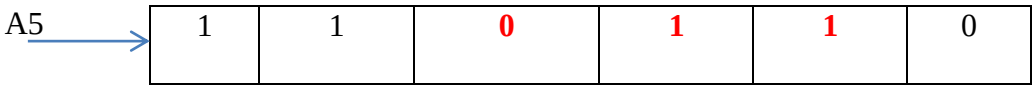
E. Mutation

Mutation rate determines the probability that a mutation will occur. Mutation is employed to give new information about the population (uncover new chromosomes) and also prevents the population of becoming saturated with similar chromosomes, simply said to avoid premature convergence. Large mutation rates increase the probability that good schemata will be destroyed, but increase population diversity. The best mutation rate is 'application dependent'. For most applications, mutation rate is between 0.001 and 0.1[43].

In certain new offspring formed, some of their genes can be subjected to a mutation with a low random probability. This implies that some of the bits in the bit string can be flipped, as shown in fig 3.12.



Before mutation



After Mutation

Figure 3.12 Mutation: Before and After

i. Single Point Mutation

Single gene (chromosome or even individual) is randomly selected to be mutated and its value is changed depending on the encoding type used, as shown in figure 3.13.

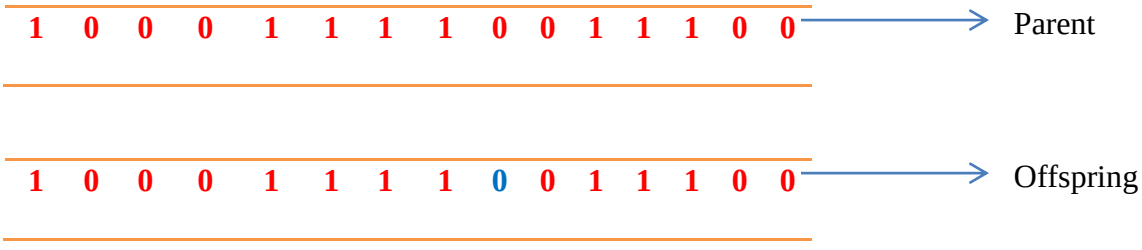


Figure 3.13 Single point mutation

ii. Multi Point Mutation

Multi genes (chromosomes or even individuals) are randomly selected to be mutated and there values are changed depending on the encoding type used, as shown in figure 3.14.

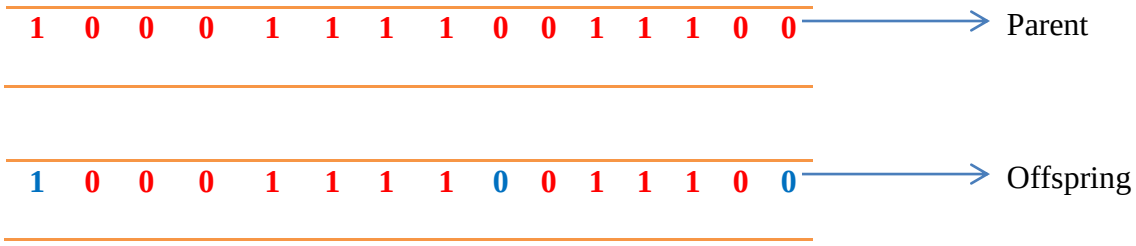


Figure 3.14 Multi point mutation

3.2. 4 Encoding

How to encode the solutions to the problem of chromosomes is a key issue when using genetic algorithms. One outstanding problem associated with encoding is that some individuals correspond to infeasible or illegal solutions to a given problem. This may become very severe for constrained optimization problems and combinatorial optimization problems.

Various encoding methods have been created for particular problems to provide effective implementation of genetic algorithms. According to what kind of symbol is used as the alleles of a gene, the encoding methods can be classified as follows:

i. Binary encoding
ii. Real-number encoding
iii. Integer or literal permutation encoding

i. Binary Encoding

Binary encoding (i.e., the bit strings) are the most common encoding used for several of reasons. One is historical: in their earlier work, Holland and his students concentrated on such encodings and genetic algorithms practices have tended to follow this lead. Another reason for that was because much of existing GAs theories are based on the assumption of using binary encoding. In spite of all that, binary encoding for function optimization problems is known to severe drawbacks due to the existence of Hamming cliffs, pairs of encoding having a large Hamming distance (The Hamming distance between two bit strings is defined as the number of corresponding positions in

these bit strings where the bits have a different value) while belonging to points of minimal distance in phenotype space.

The probability that crossover and mutation will occur can be very small. In this sense, the binary code doses not preserve the locality of points in the phenotype space. For many problems in the industrial engineering world, it is nearly impossible to represent their solution with binary encoding.

ii. Real Number Encoding

Real number encoding is best used for function optimization problems. It has been widely confirmed that real number encoding perform better than binary encoding for function optimization and constrained optimizations problems. In real number encoding, the structure of genotype space is identical to that of the phenotype. Therefore, it is easy to form effective genetic operators by borrowing useful techniques from conventional methods.

iii. Integer or Literal Permutation Encoding

Integer or literal permutation encoding is best used for combinational optimization Problems because the essence of this kind of problems is to search for the best permutation or combination of items subject to constrains.

3.3 DIgSILENT PowerFactory Overview

DIgSILENT power factory software is a powerful commercial program which converges on solving distribution system load flows in microseconds even though it uses full Newton-Raphson method.

The calculation program PowerFactory, as written by DIgSILENT, is a computer aided engineering tool for the analysis of transmission, distribution, and industrial electrical power systems. It has been designed as an advanced integrated and interactive software package dedicated to electrical power system and control analysis in order to achieve the main objectives of planning and operation optimization. “DIgSILENT" is an acronym for “**DIg**ital **SI**mu**L**ation of **E**lectrical **N**e**T**works". DIgSILENT Version 7 was the world’s first power system analysis software with an integrated graphical single-line interface. That interactive single-line diagram included drawing functions, editing capabilities and all relevant static and dynamic calculation features. PowerFactorywas designed and developed by qualified engineers and programmers with many years of experience in both electrical power system analysis and programming fields. The accuracy

and validity of results obtained with PowerFactoryhas been confirmed in a large number of implementations, by organizations involved in planning and operation of power systems throughout the world. To address users power system analysis requirements, PowerFactorywas designed as an integrated engineering tool to provide a comprehensive suite of power system analysis functions within a single executable program. Key features include:

1. PowerFactory core functions: definition, modification and organization of cases; core numerical routines; output and documentation functions.
2. Integrated interactive single line graphic and data case handling.
3. Power system element and base case database.
4. Integrated calculation functions (e.g. line and machine parameter calculation based on geometrical or nameplate information).
5. Power system network configuration with interactive or on-line SCADA access.
6. Generic interface for computer-based mapping systems.

Use of a single database, with the required data for all equipment within a power system (e.g. line data, generator data, protection data, harmonic data, controller data), means that PowerFactory can easily execute all power simulation functions within a single program environment - functions such as load-flow, short-circuit calculation, harmonic analysis, protection coordination, stability calculation, and modal analysis. Although PowerFactory includes some sophisticated power system analysis functions, the intuitive user interface makes it possible for new users to very quickly perform common activities such as load-flow and short-circuit calculations.

In this thesis DIgSILENT PowerFactory, is used for simulating the distribution network reconfiguration system.

Chapter 4

Simulation Studies and Analysis of Results

This chapter discusses the overall simulation results obtained from DIgSILENT PowerFactory on the distribution system of Addis North 132/15 KV substation and Matlab for Genetic Algorithm optimization. The overall Addis North 132/15 KV substation distribution system simulation modeling is represented in the DIgSILENT PowerFactory as shown in figures A-1 in the appendix.

4.1 Addis North 132/15 KV Substation Test Systems

4.1.1 Addis North 132/15 KV Substation System Description

The proposed network reconfiguration has been computed in Addis North 132/15 KV substation, Addis Ababa, which comprises six feeders as shown in Appendix-A before reconfiguration and Appendix-B after reconfiguration and For Genetic Algorithm optimization result the Matlab algorithm is shown in Appendix-D. The DIgSILENT PowerFactory simulation result is shown from Table 4-1 to Table 4- 6 and Genetic Algorithm optimization result is shown from Table 4-7 to Table 4-9. The system is a three phase system, 15 KV and the input data is shown in table Appendix-C, which is obtained from Ethiopia Electric Utility (EEU). The following conductors are used in the feeder.

1. Underground cables used for each feeder are listed below, which is obtained from Addis North 132/15 KV Substation.
   - Feeder 1: 240 X 2 sqmm AC
   - Feeder 2: 240 X 2 sqmm AC
   - Feeder 3: 240 X 2 sqmm AC
   - Feeder 4: 120/ 3φ/ sqmm Cu
   - Feeder 5: 300 X 1 sqmm Cu
   - Feeder 6: 240 X 2 sqmm AC
2. Overhead lines used for all feeders, which is obtained from Ethiopia Electric Utility (EEU).
   - AAC 50 sqmm, AAC 25 sqmm, 50 ACSR sqmm
3. The proposed overhead lines for all feeders
   - AAAC 240 sqmm , AAC 95 sqmm, AAC 50 sqmm

4.1.2 Addis North 132/15 KV Substation Simulation Result

a. Load flow Calculation

- "External Grid" is local reference in separated area of bus "B".

- Start Newton – Raphson algorithm . . .
    - Load flow iteration: 1
    - Load flow iteration: 2
- Newton – Raphson converged with 2 iterations.
- Load flow calculation successful.

b. **Feeder output calculation analysis before optimization**

In table 4 -1, the six feeders input current and total load have been obtained before optimization of the distribution network. As shown in the table 4-1, the total power loss is 3.983783 MW.

Name	Feeder	Input Current [kA]	Total Load [MW]	Generation [MW]	Losses [MW]	Max.Loading [%]	Mini.Voltage [p.u.]
BF_F1	F1	0.193	4.960	0.000	0.035	92.79	0.986
BF_F2	F2	0.589	13.660	0.000	1.550	107.07	0.845
BF_F3	F3	0.486	11.310	0.000	1.232	106.24	0.850
BF_F4	F4	0.115	2.980	0.000	0.007	95.40	0.993
BF_F5	F5	0.127	3.280	0.000	0.013	96.39	0.991
BF_F6	F6	0.503	11.850	0.000	1.146	105.87	00.858

Table 4-1 Feeder output calculation analysis before optimization

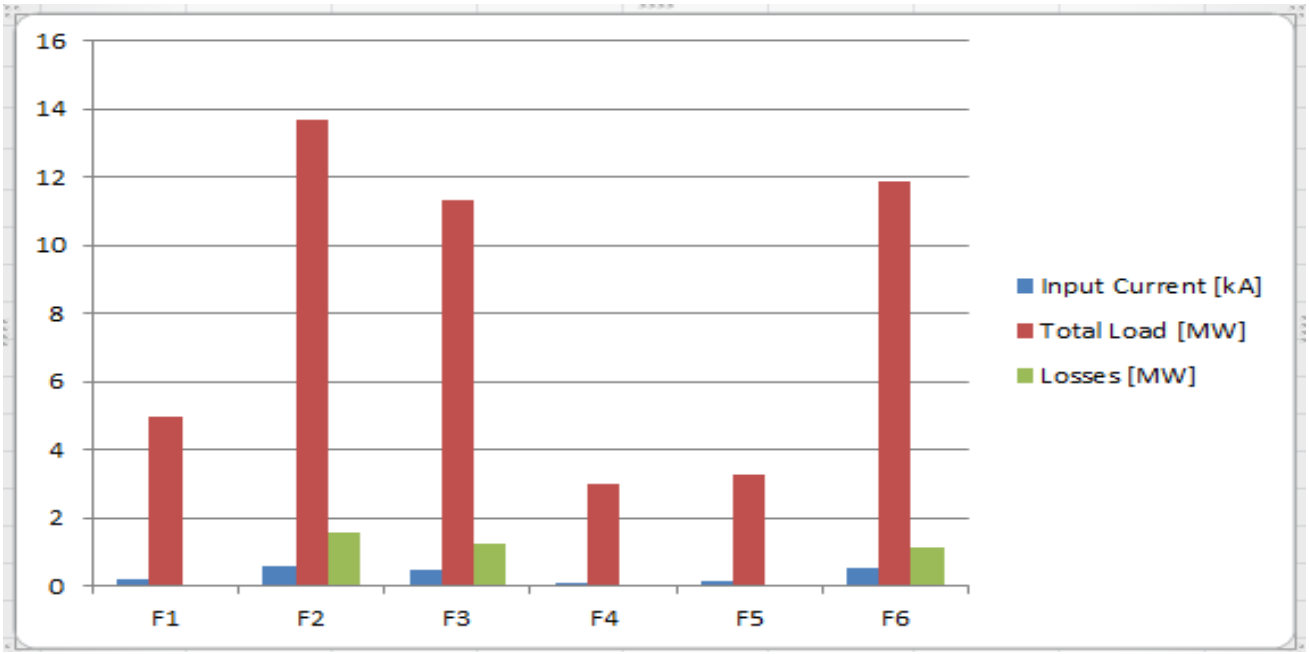


Figure 4.1 Feeders output calculation analysis chart before optimization

c. Tie Open Point Optimization

	Objective Function	Minimization of Losses
Constraints	Consider Thermal Constraints	YES
	Global constraint for all components	YES
	Maximum thermal loading of components	100.0 %
	Consider Voltage Constraints	YES
	Global constraint for all terminals	YES
	Lower voltage limit	0.95 p.u.
	Upper voltage limit	1.05 p.u.
	Consider Voltage Drop/Rise Constraints	NO
	Ignore all constraints for nominal voltage <=	1.00 kV

Table 4-2 Tie Open Point Optimization

d. Optimal Tie Open Positions

Name	Feeder 1	Feeder 2
Switch/Line033	BF_F3	BF_F4
Switch/Line028	BF_F2	BF_F1
Switch/Line37	BF_F6	BF_F5
Switch/Line(32)	BF_F4	
Switch/Line(27)	BF_F5	
Switch/Line033	BF_F4	
Switch/Line033	BF_F1	BF_F4
Switch/Line033	BF_F4	BF_F5

Table 4-3 Optimal Tie Open Positions

e. Necessary Switching Actions (From Initial to Optimized Configuration)

Name	Action
Switch/Line033	OPEN
Switch/Line028	OPEN
Switch/Line37	OPEN
Switch/Line	CLOSE
Switch/Line	CLOSE
Switch/Line	CLOSE

Table 4-4 Necessary Switching Actions (From Initial to Optimized Configuration)

f. Feeders Results Before and After Reconfiguration

The detail description of the feeders' result of loss, total load, number of customer, maximum voltage drop, maximum voltage rise, minimum and maximum voltages before optimization, after optimization and their difference is given in the table below.

Feeder		Losses [MW]	Total Load [MW]	Number of Customers	Maximum Voltage Drop [%]	Maximum Voltage Rise [%]	Minimum Voltage [p.u.]	Maximum Voltage [p.u.]
BF_F1	Before Optimization	0.035081	4.960000	10.000000	0.977528	0.000000	0.986369	0.994853
	After Optimization	0.333706	9.673195	28.000000	5.674761	0.000000	0.941041	0.995182
	Difference	0.298625	4.713195	18.000000	4.697234	0.000000	-0.045328	0.000328
BF_F2	Before Optimization	1.550367	13.660318	46.000000	15.117099	0.000000	0.844974	0.992069
	After Optimization	0.385894	8.946123	28.000000	5.872831	0.000000	0.939061	0.995350
	Difference	-1.164473	-4.714195	-18.000000	-9.244267	0.000000	0.094087	0.003281
BF_F3	Before Optimization	1.232391	11.309954	59.000000	14.645229	0.000000	0.849692	0.992789
	After Optimization	0.218436	6.733634	33.000000	4.821054	0.000000	0.949578	0.995979
	Difference	-1.013956	-4.576319	-26.000000	-9.824175	0.000000	0.099886	0.003190
BF_F4	Before Optimization	0.007463	2.980000	6.000000	0.352639	0.000000	0.992618	0.995374
	After Optimization	0.224793	7.555761	32.000000	5.050102	0.000000	0.947288	0.995765
	Difference	0.217330	4.575761	26.000000	4.697463	0.000000	-0.045330	0.000391

BF_F5	Before Optimization	0.012796	3.280000	8.000000	0.516859	0.000000	0.990976	0.995294
	After Optimization	0.220597	7.752629	29.000000	4.470545	0.000000	0.953083	0.995716
	Difference	0.207801	4.472629	21.000000	3.953686	0.000000	-0.037893	0.000421
BF_F6	Before Optimization	1.145686	11.849606	58.000000	13.767013	0.000000	0.858475	0.992681
	After Optimization	0.211215	7.376473	37.000000	4.480650	0.000000	0.952982	0.995816
	Difference	-0.934471	-4.473133	-21.000000	-9.286363	0.000000	0.094508	0.003135
TOTAL	Before Optimization	3.983783	48.039878	187.000000	0.352639	0.000000	0.992618	0.992069
	After Optimization	1.594640	48.037816	187.000000	5.050102	0.000000	0.947288	0.995350
	Difference	2.389143	-0.002062	0.000000	4.697463	0.000000	-0.045330	0.003281
	Difference[%]	59.97172	-0.004292	0.000000	1332.08644	100.0000	4.566754	0.330711

Table 4-5 Feeders Results

The distribution network is reconfigured for the real power loss minimization. As can be seen from Table 4-5, the loss after reconfiguration is 1.594640 MW and the difference is 2.389143 MW. The power loss before reconfiguration is 3.983783 MW. Therefore the reduction in power loss is Power loss reduction=3.983783 MW - 2.389143 MW = 1.594640 MW, which is approximately 59.97172 % of the loss after reconfiguration is reduced. High percentage of reduction in power loss obtained because the proposed tie open point switches are strategic ones and their overhead lines (cables) cross sections deliberately selected to be higher ones.

The total energy loss before and after optimization in a year

- Before reconfiguration total power loss = 3.983783 MW

- Before reconfiguration total energy loss for one year = 365 X 3.983783 MW X 24h
  = 34, 897, 939.08 KWh

- After optimization, can be saved = 2.389143 MW
  - After reconfiguration total energy, can be saved for one year = 365 X 2.389143 MW X 24h
    = **20,928,892.68 KWh**

g. Feeder output calculation analysis after optimization shown table below

In the table below, the 6 feeders input current and total load have been obtained after optimization of the distribution network. As shown in the table below, the total power loss is 1.594640.

Name	Feeder	Input Current [kA]	Total Load [MW]	Generation [MW]	Losses [MW]	Max.Loading [%]	Mini.Voltage [p.u.]
BF_F1	F1	0.386	9.673	0.000	0.334	185.59	0.941
BF_F2	F2	0.360	8.946	0.000	0.386	65.45	0.939
BF_F3	F3	0.268	6.734	0.000	0.218	58.68	0.950
BF_F4	F4	0.300	7.556	0.000	0.225	248.05	0.947
BF_F5	F5	0.308	7.753	0.000	0.221	233.01	0.953
BF_F6	F6	0.293	7.376	0.000	0.211	61.62	0.953

Table 4-6 Feeder output calculation analysis after optimization shown table below

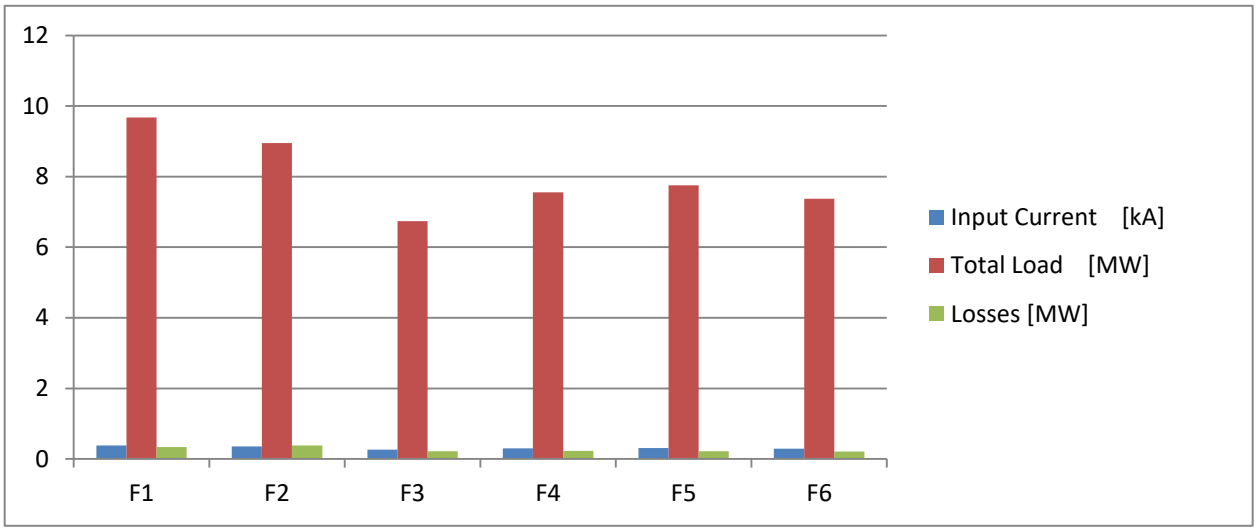


Figure 4.2 Feeders output calculation analyses chart after optimization

4.2 Genetic Algorithm optimization result

- Program to optimize the power for feeder 1 and feeder 2
  - Enter the size of initial population::30
  - Enter the bit size of each string::7
  - Enter the number of iterations to perform the process of genetic algorithm::1000
  - Enter the number of times you want to perform tournament selection::50

With the parameters chosen as above, the best results were achieved. A higher number of iterations always results in better output. The test results achieved when using the above parameter values can be found in Table 4-7, for 10 iterations and for 1000 iterations. For this test result the Matlab algorithm is shown in Appendix D.

	Power before optimization	Power optimization after 10 iterations	Power optimization after 1000 iterations
P_1 for feeder 1 (MW)	4.9600	8.8204	9.1174
P_2 for feeder 2 (MW)	13.6600	9.7996	9.5026
Total Power	18.6200	18.6200	18.6200
N^{o} of switches in feeder 1 (u)	10	23	24

Nº of switches in feeder 2 (V)	46	33	32
Total Nº of Switches (u + v)	56	56	56

Table 4-7 Genetic algorithm test results for feeder 1 and feeder 2

- Program to optimize the power for feeder 3 and feeder 4
    - Enter the size of initial population::30
    - Enter the bit size of each string::7
    - Enter the number of iterations to perform the process of genetic algorithm::100
    - Enter the number of times you want to perform tournament selection::1000

With the parameters chosen as above, the best results were achieved. A higher number of iterations always results in better output. The test results achieved when using the above parameter values can be found in Table 4-8, for 10 iterations and for 100 iterations. For this test result the Matlab algorithm is shown in Appendix D.

	Power before optimization	Power optimization after 10 iterations	Power optimization after 100 iterations
P_3 for feeder 3 (MW)	11.3100	6.9010	7.0927
P_4 for feeder 4 (MW)	2.9800	7.3890	7.1973
Total Power	14.29	14.29	14.29
Nº of switches in feeder 3 (u)	59	36	37
Nº of switches in feeder 4 (V)	6	29	38
Total Nº of Switches (u + v)	65	65	65

Table 4-8 Genetic algorithm test results for feeder 3 and feeder 4

- Program to optimize the power for feeder 5 and feeder 6
    - Enter the size of initial population::30
    - Enter the bit size of each string::7
    - Enter the number of iterations to perform the process of genetic algorithm::1000
    - Enter the number of times you want to perform tournament selection::1000

With the parameters chosen as above, the best results were achieved. A higher number of iterations always results in better output. The test results achieved when using the above parameter values can be found in Table 4-9, for 10 iterations and for 1000 iterations. For this test result the Matlab algorithm is shown in Appendix D.

	Power before optimization	Power optimization after 10 iterations	Optimization after 1000 iterations	
			Power	Loss
P_5 for feeder 5 (MW)	3.2800	6.9576	7.5705	0.214
P_6 for feeder 6 (MW)	11.8500	8.1724	7.5595	0.216
Total Power	15.1300	15.1300	15.1300	0.43
N^{o} of switches in feeder 5 (u)	8	26	29	
N^{o} of switches in feeder 6 (V)	58	40	37	
Total N^{o} of Switches (u + v)	66	66	66	

Table 4-9 Genetic algorithm test results for feeder 5 and feeder 6

4.3 Feasibility Analysis

- After reconfiguration total energy, can be saved for one year is 20,928.89268 MWh
- By using current average electric sale bill of EEU, it can be converting in ETB.
- Average electric sale bill = 0.655 cent per KWh
- Hence, 20,928,892.68 X 0.655 = 13,708,424.7054 ETB = $498,125.889435 per year
- **Tie and Sectionalizing switch specification**
    - Price: **$5.2/piece**
    - Minimum order: **10 piece**
    - Operating capacity: **Medium voltage**
    - Operation: **automatic type**
    - Total number of distribution network switches : 187
    - $5.2 X 187 = **$972.4**
- Assuming that the installation costs are 10% and annual maintenance costs are 2%.
    - 10% of $498,125.889435 = $49,812.6
    - 2% of $498,125.889435 = $9,962.52
    - Total cost = $972.4 + $49,812.6 + $9,962.52 = **$60,747.52**
    - difference $498,125.889435-$60,747.52 = $**437,378.369435**

Chapter 5

Conclusions, Recommendations and Future Work

5.1 Conclusions

Network reconfiguration means restructuring the power lines which connect various buses in a power system. Restructuring of specific lines leads to alternate system configurations. System reconfiguration can be accomplished by placing line interconnection switches into network. Opening and closing a switch connects or disconnect a line to the existing network.

The main objective of this thesis is to minimize the electrical power loss by considering network reconfiguration and applying a Genetic Algorithm in an electric distribution power system in the case of Addis North 132/15 KV substation, Addis Ababa. Obviously, Addis Ababa is experiencing frequent power interruptions in recent times as the distribution system is not designed and built to handle the current load demands. The power interruption can be minimized by using network reconfiguration.

As can be referred from the simulation and result discussion of chapter 4, the test system has been computed in Addis North 132/15 KV substation distribution network. The network reconfiguration has been simulated using DIgSILENT PowerFactory and Genetic Algorithm has been optimized using Matlab algorithm. An optimization problem is a maximization or minimization problem which involves finding the best solution out of a set of possible alternatives. It can be completely characterized by the search space and objective function. The search space is a finite or countable infinite set of possible solutions, and the objective function maps each point in the search space into the real line, to give a measure of how good a solution is relative to the others.

After network reconfiguration for loss minimization, the power loss has been reduced, which is 59.91716% reduction. Hence, reconfiguration of distribution networks can reduce power loss and operating cost as well as improves the voltage profile of distribution systems.

5.2 Recommendations

The following are the recommendations based on the study on power loss minimization for network reconfiguration.

1. From this thesis reconfiguration of distribution network has shown to decrease the power loss significantly. Therefore, it is highly recommended if all the distribution network feeders of Addis Ababa city to be reconfigured for the objectives wanted by the Ethiopian Eclectic Utility (EEU).

2. Most of the conductors used on the distribution network are 25 sq mm AAC, 50 sq mm AAC and 50 sq mm ACSR conductors' cross-section areas. In this case, some lines are loaded. Therefore, it is better to use AAAC 240 sqmm, AAC 95 sqmm, AAC 50 sqmm or above conductors cross-sectional areas, which is recommended for main lines of cities like Addis North 132/15 KV Substation, Addis Ababa, based on the loads connected. And the existing Addis Ababa distribution network is not designed and built according to international standards. Large cross-sections area conductors are used after small cross-section conductors on main lines and also on branch lines of the distribution network. In order to useful capacity of the large cross-section of conductors, it should be built before the small cross-sectional area conductors.
3. Most of the feeders network of the Addis Ababa city have no tie switches to a feeder, even though there are tie switches to feeders. Since these tie switches are very useful for network reconfiguration as shown by this thesis and has also other advantages, it is highly recommended if the distribution feeders have these tie switches.

5.3 Future Work

In the future researches can be done on the distribution networks of Addis Ababa city including the following suggestions

1. The simulation of the distribution network can also be used for optimization for capacitor placement. To reduce real power losses, improve the voltage profile and improve the load balancing of the network.
2. The simulation of the distribution network can be made to include strategically placed distribution generators (DG) and the network can be reconfigured. In doing so the voltage profiles, the real power losses and the system load balancing index can be improved.
3. Other optimization algorithms, some of which are described in literature review, can also be used to improve the reconfiguration of the distribution network and find global optimum solutions.

References

[1] K. Nara, A. Shiose, M. Kitagawa, T. Ishihara,"Implementation of genetic algorithm for distribution systems loss minimum re-configuration", IEEE Transactions on Power systems, Vol.7, No 3, August 1992, pp. 1044-1051.

[2] K. Nara, A. Shioss, M. Kitagwa and T. Ishihwara, "Implementation of GA for distribution system loss minimum reconfiguration", IEEE Transaction on Power Systems, Vol. 7, No. 3, 1992, pp. 1044-1051.

[3] H. Fudou, T. Genji, Y. Fukuyamam and Y. Nakanishi, "A genetic algorithm for network reconfiguration using three unbalanced load flow", Intelligent System Applications to Power System (ISAP'97), Seoul, Korea, 1997, pp. 1-5.

[4] Ray Daniel Zimmerman, Network Reconfiguration for Loss Minimization in Three Phase Power Distribution System, Cornel University, May 1992

[5] Ali Reza Fereidunian, Hamid Lesani and Caro Lucas "Distribution systems reconfiguration using pattern recognizer neural networks" IJE International: Applications Vol. 15, No. 2, July 2002 – 135.

[6] H. Kim, "ANN based feeder reconfiguration for loss reduction in distribution system", IEEE Transaction on Power Delivery, Vol. 8,No. 3, 1993, pp. 1356-1366.

[7] Tamer M. Khalil, and Alexander V. Gorpinich, "Reconfiguration for Loss Reduction of Distribution Systems Using Selective Particle Swarm Optimization" International Journal Of Multidisciplinary Sciences And Engineering, Vol. 3, No. 6, June 2012.

[8] A. Merlin, H. Back, "Search for a minimal-loss operating spanning tree configuration in an urban power distribution system" Proceedings of 5th Power System Computation Conference (PSCC), Cambridge, UK, (1975), pp.1-18.

[9] E.Dolatdar, S.Soleymani, B.Mozafari "A New Distribution Network Reconfiguration Approach Using a Tree Model", World Academy of Science, Engineering and Technology Vol: 3 2009-10-26.

[10] Hugh Rudnick , Ildefonso Harnisch , Raúl Sanhueza , "Reconfiguration of Electric Distribution Systems, Revista Facultad De Ingenieria", U.T.A. (Chile), Vol. 4, 1997.

[11] Abdullah M. Alshehri,"Optimal Reconfiguration of Distribution Networks Using Ant Colony Method", King Saud University, June, 2007.

[12] D. Shirmohammadi and H.W Hong,"Reconfiguration of electric distribution networks for resistive line losses reduction", IEEE Transactions on Power Delivery, Vol. 4, No. 2, 1989, pp. 1492-1498

[13] S. Civanlar, J.J. Grainger, H. Yin, S.S. Lee, "Distribution Feeder Reconfiguration for Loss Reduction", IEEE Transactions on Power Delivery, Vol. 3, No. 3, July 1988, pp. 1217-1223.

[14] M.E. Baran, F.F. Wu, "Network Reconfiguration in Distribution Systems for Loss Reduction and Load Balancing", IEEE Transactions on Power Delivery, Vol. 4, No. 2, April 1989, pp. 1401-1407.

[15] T. Taylor, D. Lubkeman, "Implementation of Heuristic Search Strategies for Distribution Feeder Reconfiguration", IEEE Transactions on Power Delivery, Vol. 5, No. 1, January 1990, pp. 239-246.

[16] G.J Peponis, M.P. Popadopoulos and N.D Hatziargyrion, "Distribution network reconfiguration for resistive line losses", IEEE Transactions on Power Delivery, Vol. 10, No. 3, 1995, pp. 1338-1342.

[17] R.P Broadwater, A.H. Khan, H.E Shalaan and R.E. Lee, "Time varying load analysis to reduce distribution losses through reconfiguration", IEEE Transactions on Power Delivery, Vol. 8, No. 1, 1993, pp. 294-300.

[18] R. Safri, M.M.A Salama and Y. Chickani, "Distribution system reconfiguration for loss reduction: an algorithm based on network portioning theory",a IEEE Transactions on Power System, Vol. 11, No. 1, 1996, pp. 504-510.

[19] A. Borghetti, M. Paolone and C.A. Nucci " A mixed integer linear programming approach to the Optimal configuration of electrical distribution Networks with embedded generators" 17th Power Systems Computation Conference, Stockholm Sweden - August 22-26, 2011.

[20] G. Celli, M. Loddo, F. Pilo and A. Abur " On-line network reconfiguration for loss reduction in distribution networks with distributed generation" 18th International Conference on Electricity Distribution, Turin, 6-9 June 2005.

[21] Ramadoni Syahputra " Fuzzy multi-objective approach for the Improvement of distribution network efficiency by considering DG" International Journal of Computer Science & Information Technology (IJCSIT) Vol 4, No 2, April 2012.

[22] Abhisek Ukil and Willy Siti "Feeder Load Balancing using Fuzzy Logic and Combinatorial Optimization-based Implementation"

[23] N. Rugthaicharoencheep and S. Sirisumrannukul "Feeder Reconfiguration for Loss Reduction in Distribution System with Distributed Generators by Tabu Search", GMSARN International Journal 3 (2009), pp 47 – 54.

[24] T.P. Wagner, A.Y. Chikhani, R. Hackam, "Feeder Reconfiguration for Loss Reduction: An Application of Distribution Automation", IEEE Transactions on Power Delivery, Vol. 6, No. 4, July 1991, pp. 1922-1931.

[25] Q. Zhou, D. Shirmohammadi, W. H. Liu, "Distribution Feeder Reconfiguration for Service Restoration and Load Balancing", IEEE Transactions on Power Systems, Vol. 12, No. 2, May 1997, pp. 724-729.

[26] R.T.F Ah King, B. Radha and H.C.S Rughooputh, "A fuzzy logic controlled GA for optimal electric distribution network reconfiguration", Proc.IEEE, International Confernce on Networking Sensing and Control, Taipei, Taiwan, 2004, pp. 577-582.

[27] Mehdi Assadian, Malihe M. Farsangi , Hossein Nezamabadi-pour " Optimal Reconfiguration of Distribution System by PSO and GA using graph theory" Proceedings of the 6th WSEAS International Conference on Applications of Electrical Engineering, Istanbul, Turkey, May 27-29, 2007.

[28] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley Publishing Company, January 1989.

[29] Bäck, Thomas, Evolutionary Algorithms in Theory and Practice (1996), p. 120, Oxford Univ. Press

[30] A. Lipowski, Roulette-wheel selection via stochastic acceptance (arXiv:1109.3627)

[31] Cedeno, W. and V. Vemuri (1994). Multi-niche crowding in genetic algorithms and its application to the assembly of DNA restriction fragments, Evolutionary Computation, Vol. 2, No. 4, pp 321-345. MIT Press.

[32] Cooper, M.G. (1994). Genetic Design of Rule-Based Fuzzy Controllers. PhD dissertation, Department of Computer Science, University of California, Los Angeles.

[33] DeJong, K.A. (1975). Analysis of the Behavior of a Class of Genetic Adaptive Systems. PhD dissertation, Department of Computer and Communication Sciences, University of Michigan.

[34] Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.

[35] D. Whitley, "The GENITOR algorithm and selection pressure: why rank based allocations of reproductive trials is best", Proc. ICGA 3, pp. 116-121, 1989.

[36] J. E. Baker, "Adaptive Selection Methods for Genetic Algorithms", Proc. ICGA 1, pp. 101-111, 1985.

[37] G. Syswerda, "Uniform crossover in genetic algorithms", Proc. ICGA 3, pp. 2-9, 1989.

[38] W. M. Spears and K. A. De Jong, "On the Virtues of Parameterised Uniform Crossover", Proc. ICGA 4, pp.230-236, 1991.

[39] G. Syswerda, "Uniform crossover in genetic algorithms", Proc. ICGA 3, pp. 2-9, 1989.

[40] W. M. Spears and K. A. De Jong, "An Analysis of Multi-Point Crossover", In Foundations of Genetic Algorithms, J. E. Rawlins (Ed.), pp. 301-315, 1991.

[41] J. E. Baker, “Reducing bias and inefficiency in the selection algorithm”, Proc. ICGA 2, pp. 14-21, Lawrence Erlbaum Associates, Publishers, 1987.

[42] R. A. Caruana, L. A. Eshelman, J. D. Schaffer, “Representation and hidden bias II: Eliminating defining length bias in genetic search via shuffle crossover”, In Eleventh International Joint Conference on Artificial Intelligence, N. S. Sridharan (Ed.), Vol. 1, pp. 750 755, Morgan Kaufmann Publishers, 1989.

[43] M. F. Bramlette, “Initialization, Mutation and Selection Methods in Genetic Algorithms for Function Optimization”, Proc ICGA 4, pp. 100-107, 1991.

Appendix A: Addis North 132/15 kV Substation feeders before reconfiguration

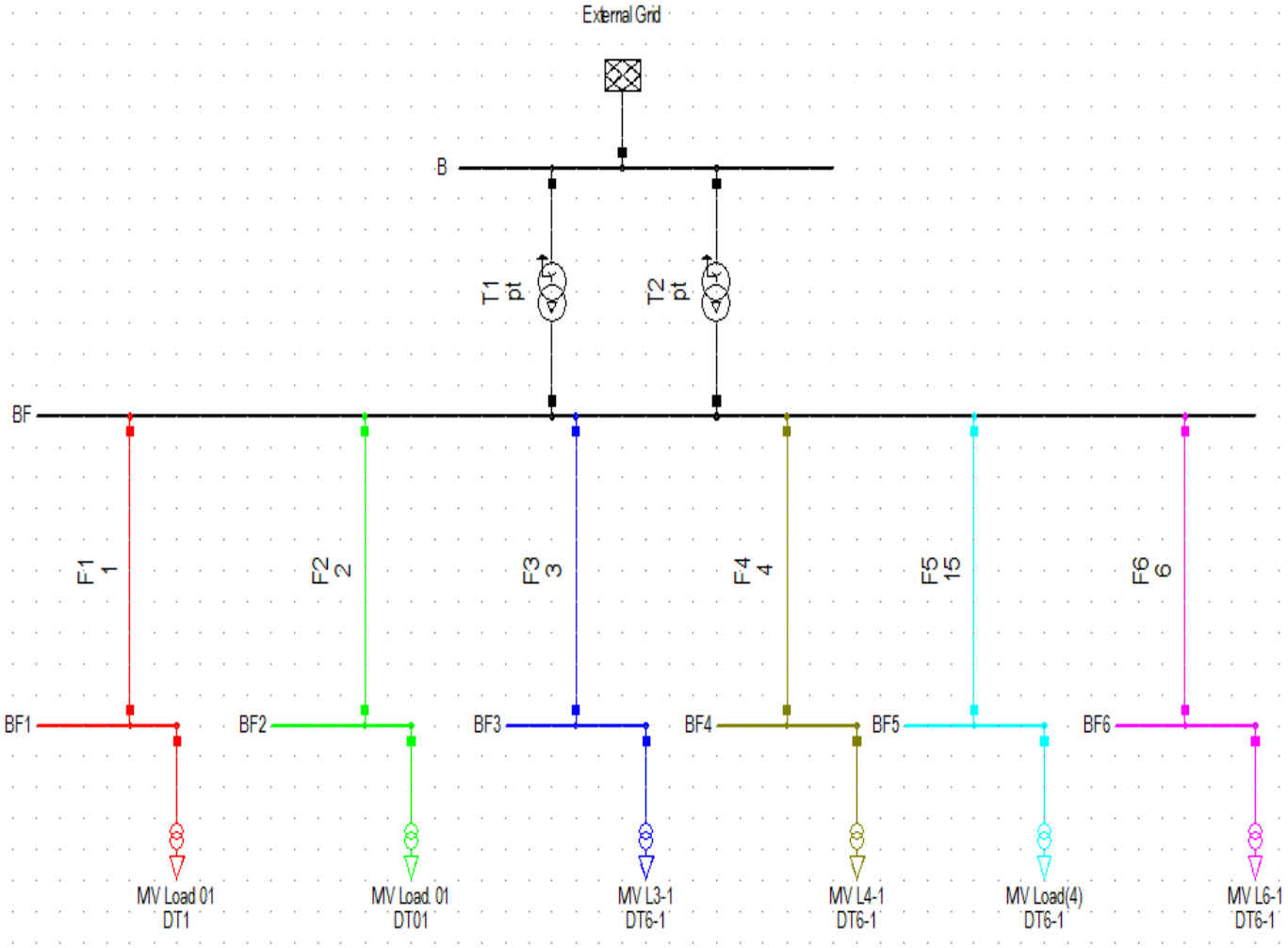


Figure A.1 Addis North 132/15 kV substation outgoing feeders online diagram

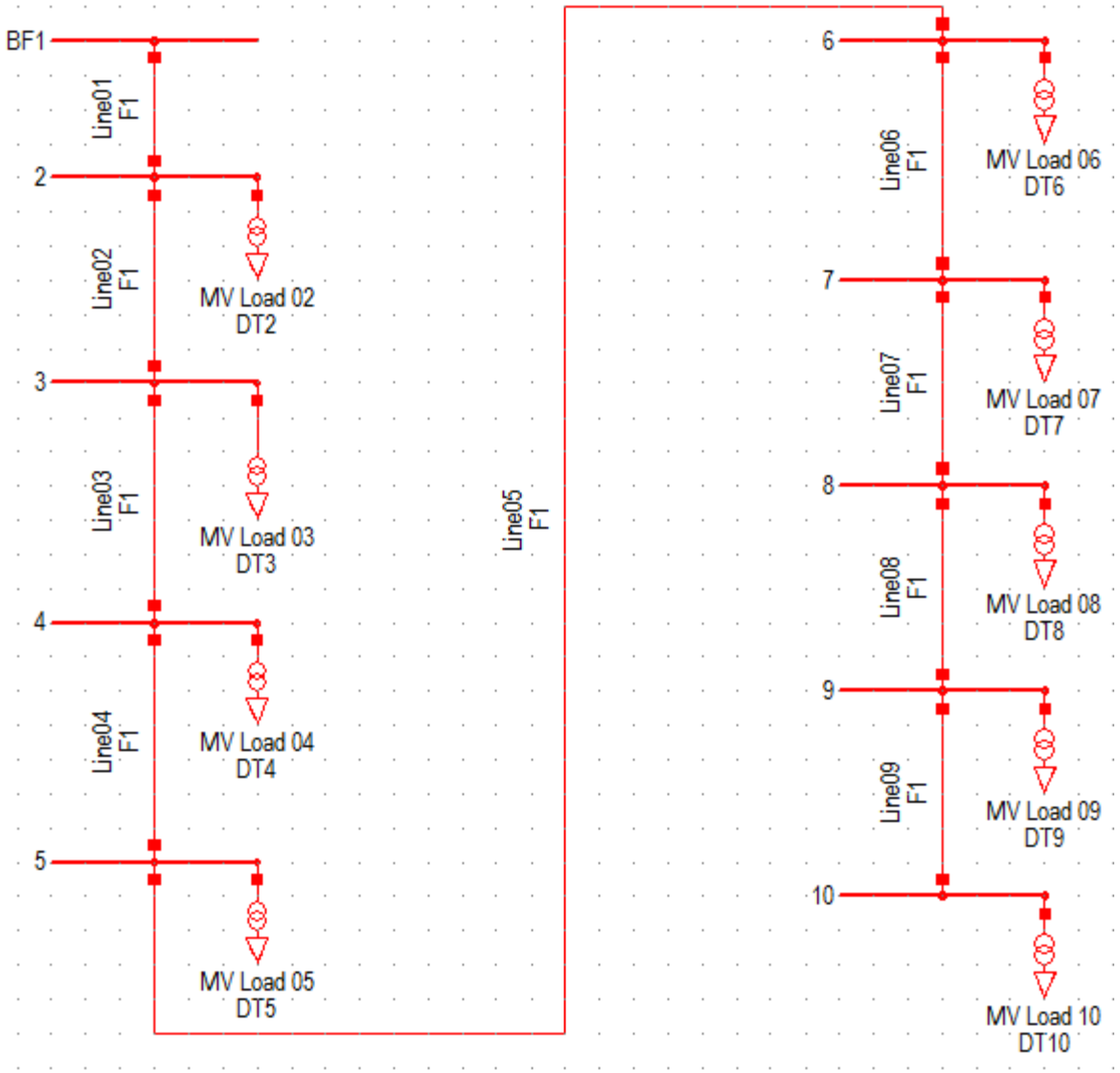
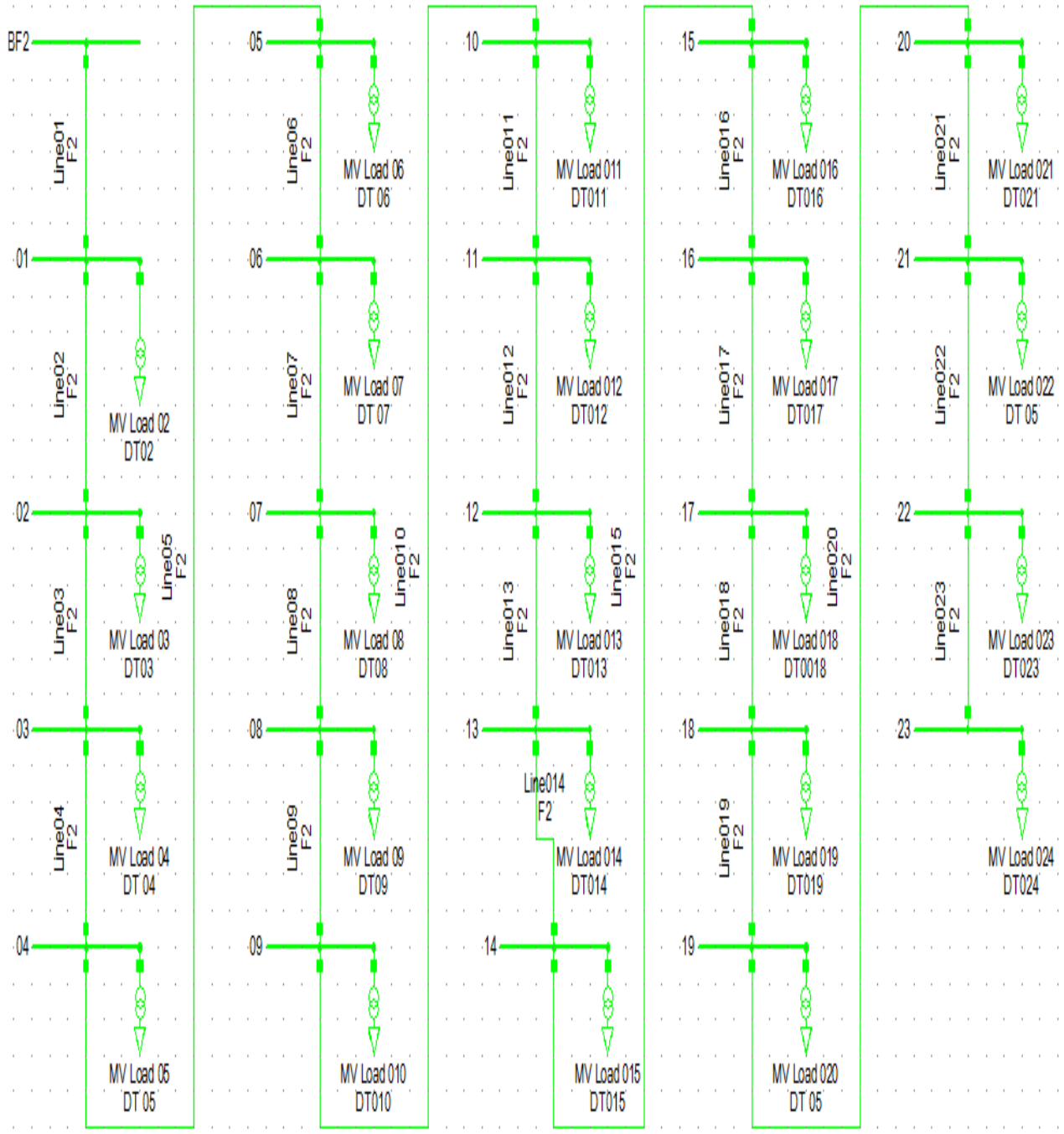


Figure A.2 Addis North 132/15 kV substation feeder 1 one line diagram


BF2
Line01 F2
MV Load 06 DT 06
Line06 F2
MV Load 011 DT011
Line011 F2
MV Load 016 DT016
Line016 F2
MV Load 021 DT021
Line021 F2
01
Line02 F2
MV Load 02 DT02
06
Line07 F2
MV Load 07 DT 07
11
Line012 F2
MV Load 012 DT012
16
Line017 F2
MV Load 017 DT017
21
Line022 F2
MV Load 022 DT 05
02
Line03 F2
MV Load 03 DT03
Line05 F2
07
Line08 F2
MV Load 08 DT08
Line010 F2
12
Line013 F2
MV Load 013 DT013
Line015 F2
17
Line018 F2
MV Load 018 DT0018
Line020 F2
22
Line023 F2
MV Load 023 DT023
03
Line04 F2
MV Load 04 DT 04
08
Line09 F2
MV Load 09 DT09
13
Line014 F2
MV Load 014 DT014
18
Line019 F2
MV Load 019 DT019
23
MV Load 024 DT024
04
MV Load 05 DT 05
09
MV Load 010 DT010
14
MV Load 015 DT015
19
MV Load 020 DT 05

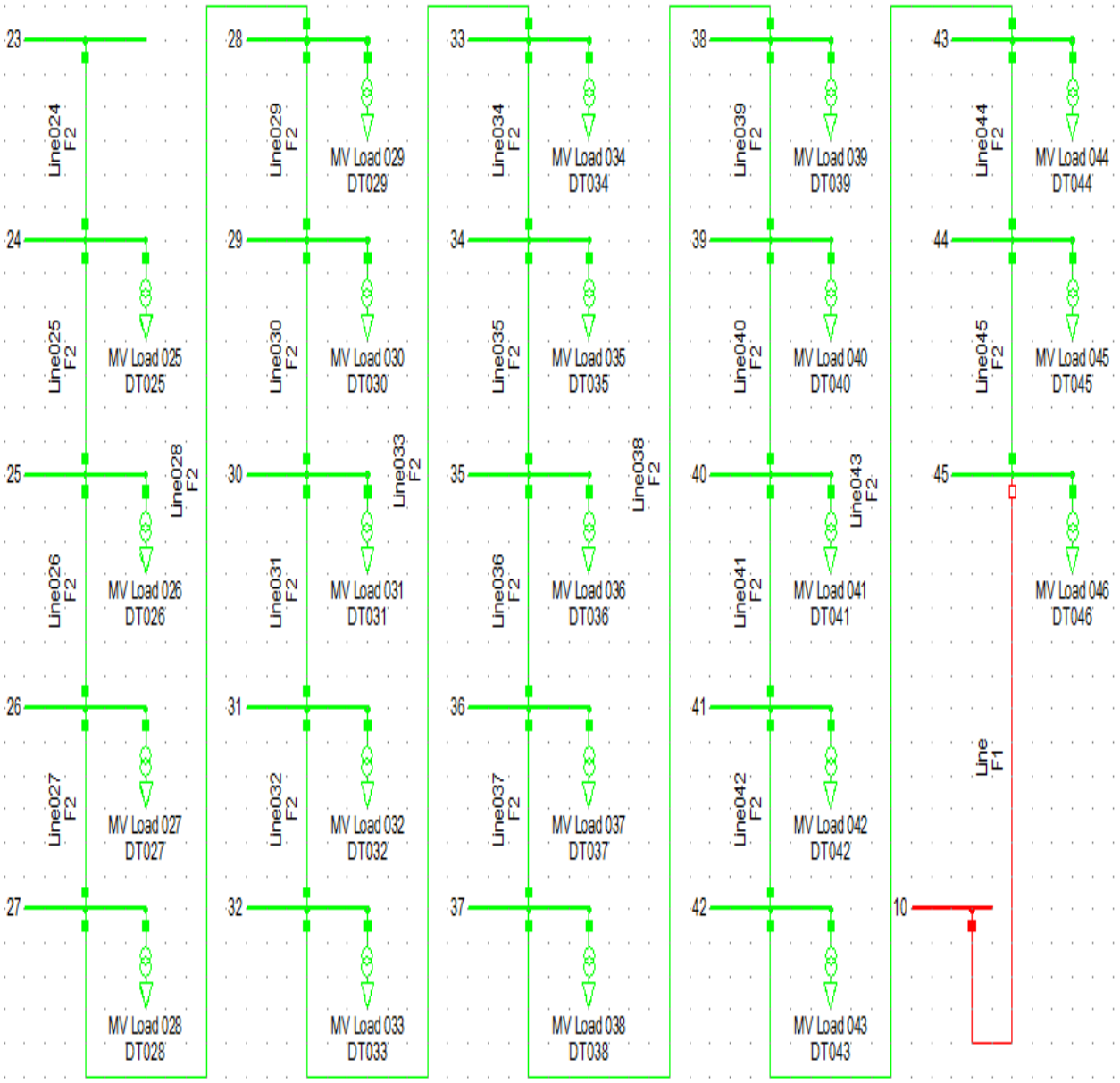
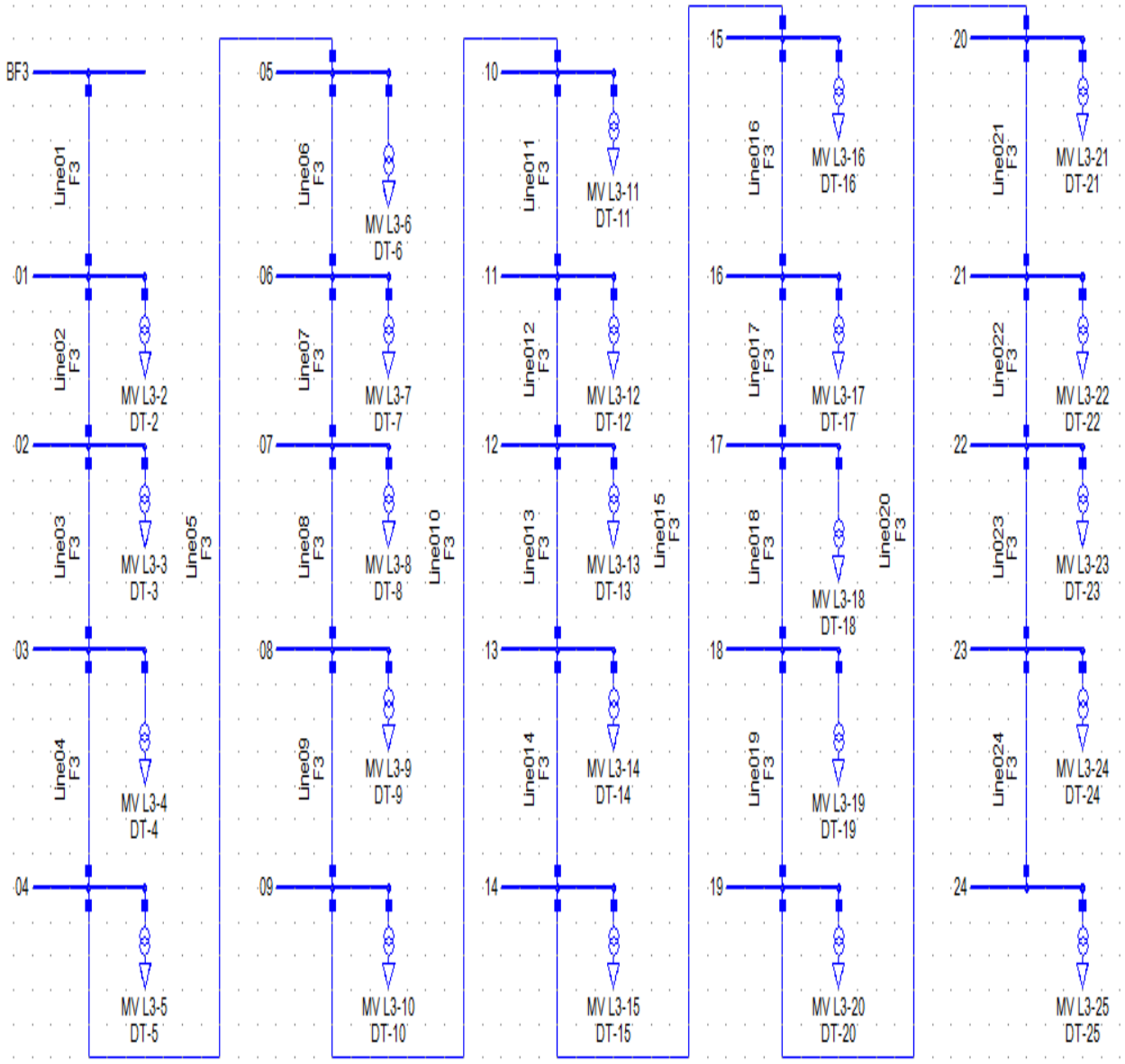


Figure A.3 Addis North 132/15 kV substation feeder 2 one line diagram


BF3
Line01 F3
01
Line02 F3
MV L3-2 DT-2
02
Line03 F3
MV L3-3 DT-3
Line05 F3
03
Line04 F3
MV L3-4 DT-4
04
MV L3-5 DT-5
05
Line06 F3
MV L3-6 DT-6
06
Line07 F3
MV L3-7 DT-7
07
Line08 F3
MV L3-8 DT-8
Line010 F3
08
Line09 F3
MV L3-9 DT-9
09
MV L3-10 DT-10
10
Line011 F3
MV L3-11 DT-11
11
Line012 F3
MV L3-12 DT-12
12
Line013 F3
MV L3-13 DT-13
Line015 F3
13
Line014 F3
MV L3-14 DT-14
14
MV L3-15 DT-15
15
Line016 F3
MV L3-16 DT-16
16
Line017 F3
MV L3-17 DT-17
17
Line018 F3
MV L3-18 DT-18
Line020 F3
18
Line019 F3
MV L3-19 DT-19
19
MV L3-20 DT-20
20
Line021 F3
MV L3-21 DT-21
21
Line022 F3
MV L3-22 DT-22
22
Lin023 F3
MV L3-23 DT-23
23
Line024 F3
MV L3-24 DT-24
24
MV L3-25 DT-25

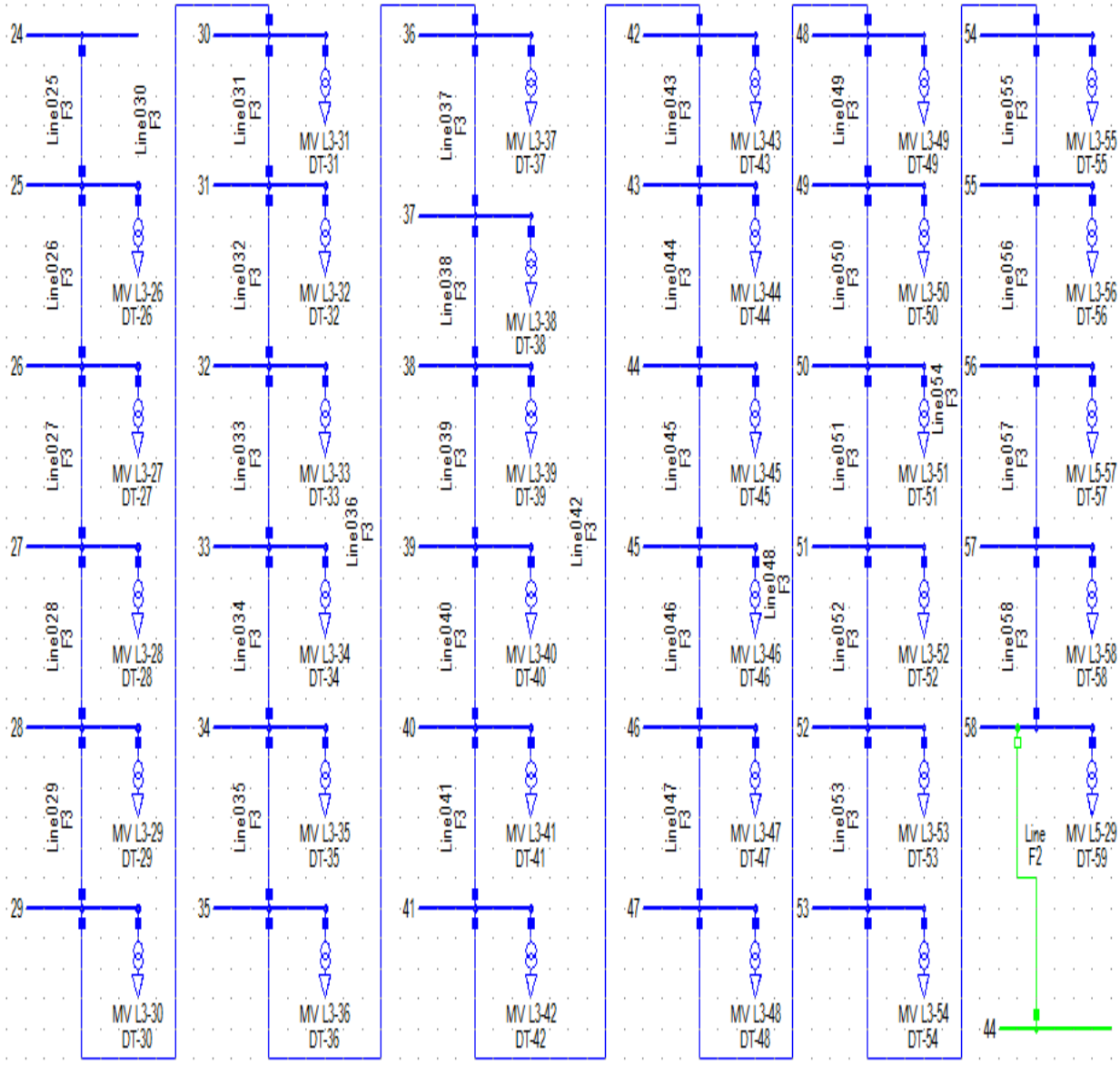


Figure A.4 Addis North 132/15 kV substation feeder 3 one line diagram

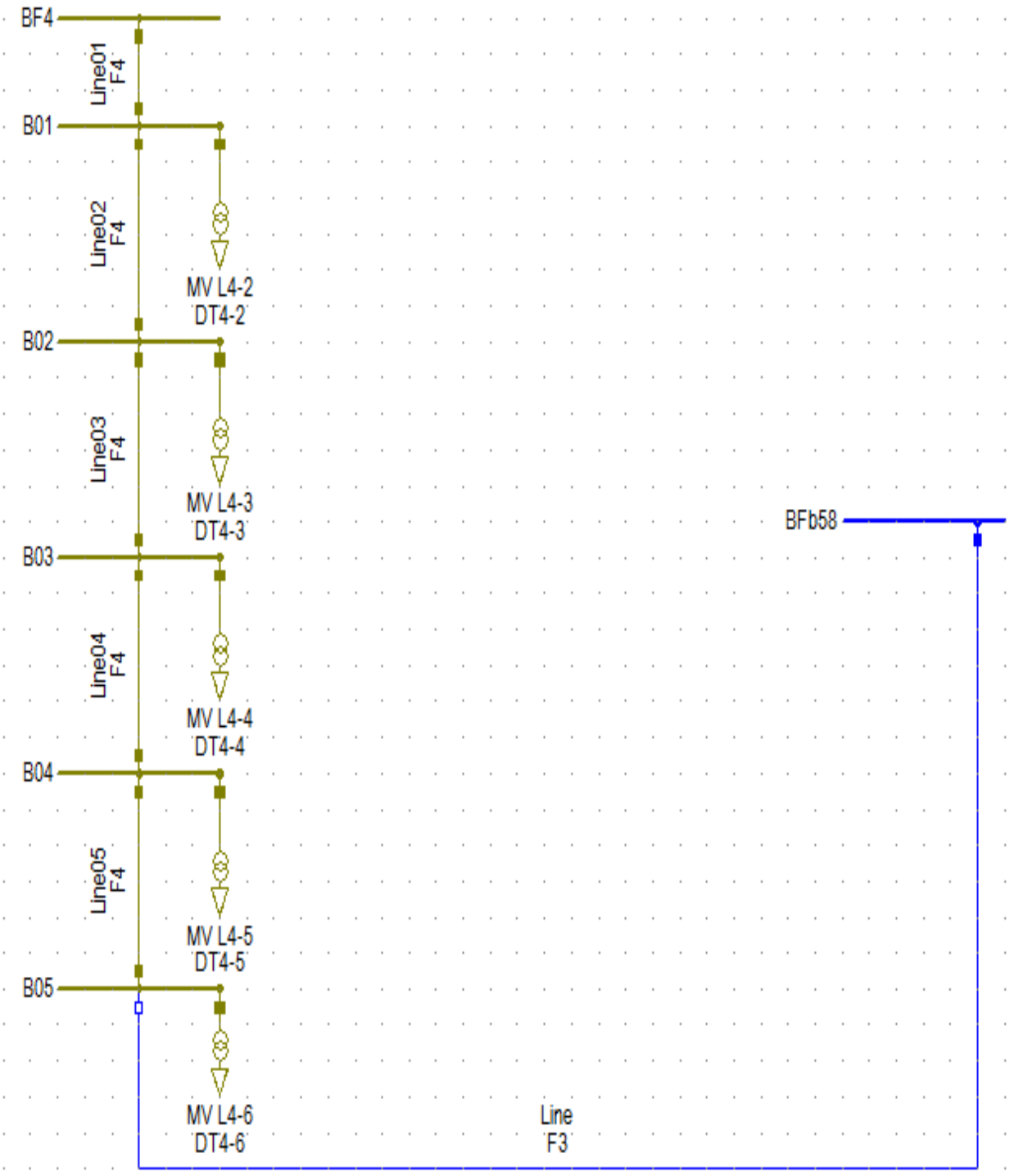


Figure A.5 Addis North 132/15 kV substation feeder 4 one line diagram

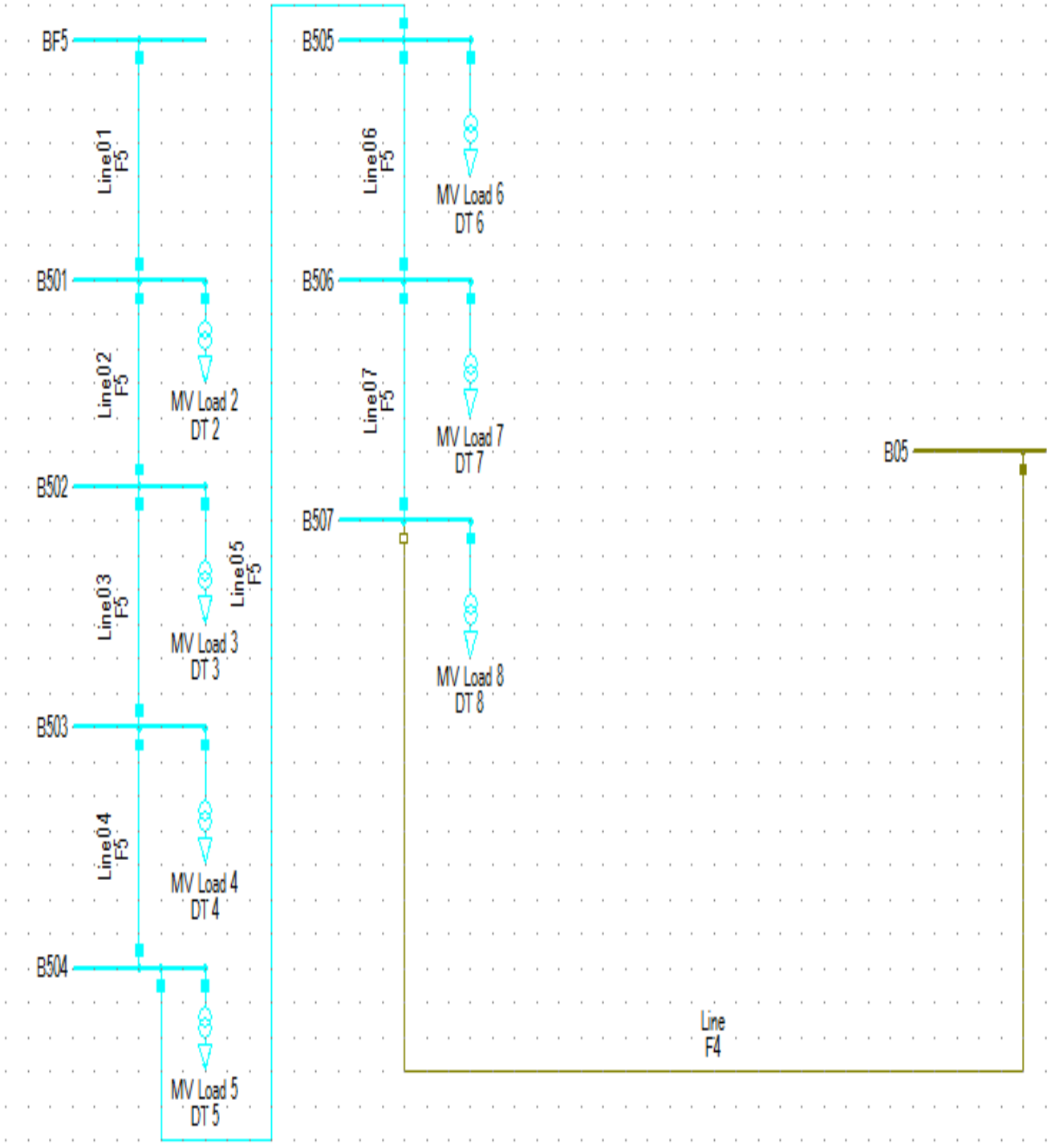
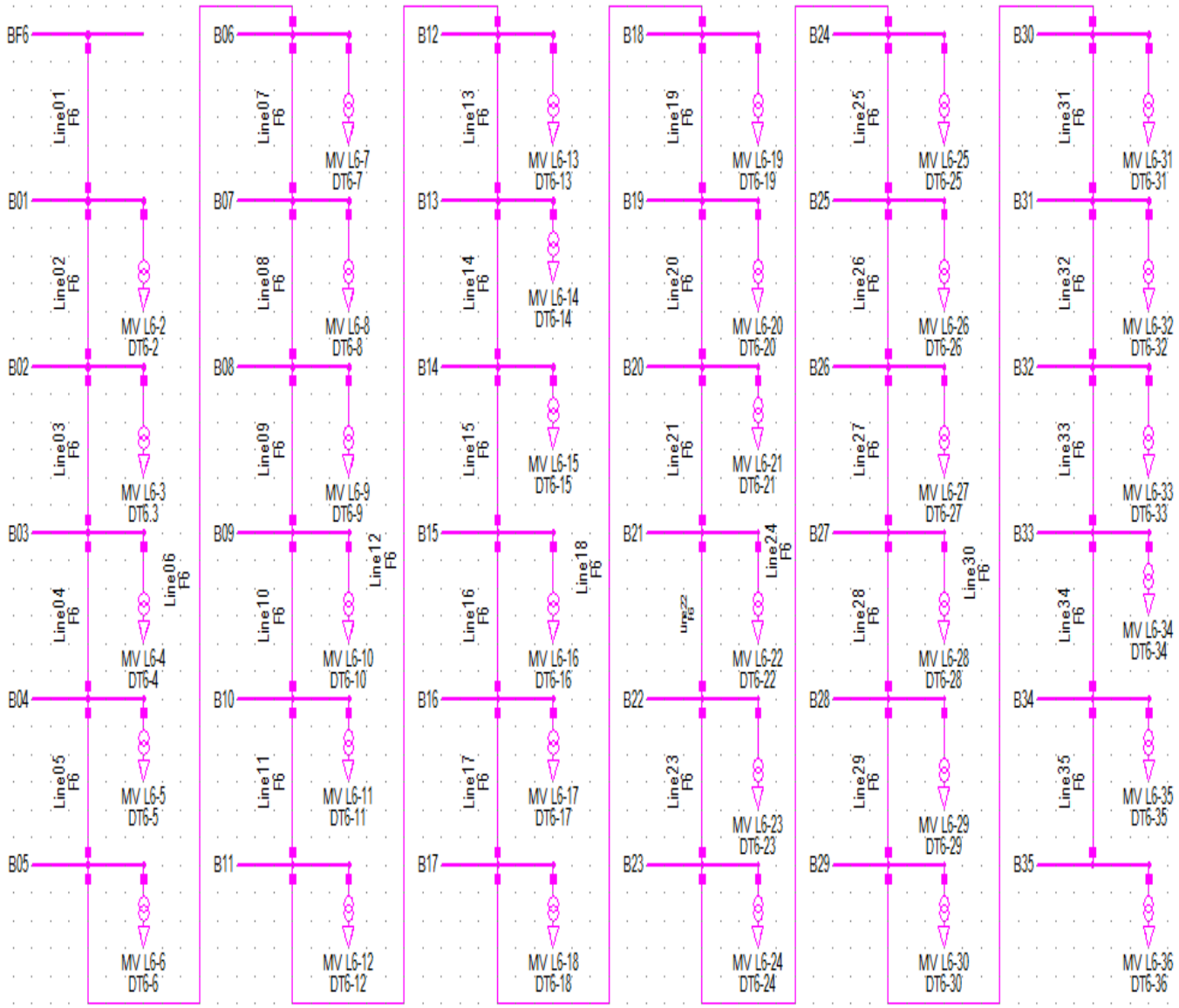


Figure A.6 Addis North 132/15 kV substation feeder 5 one line diagram


BF6
Line01 F6
B01
Line02 F6
MV L6-2 DT6-2
B02
Line03 F6
MV L6-3 DT6.3
B03
Line04 F6
Line06 F6
MV L6-4 DT6-4
B04
Line05 F6
MV L6-5 DT6-5
B05
MV L6-6 DT6-6
B06
Line07 F6
MV L6-7 DT6-7
B07
Line08 F6
MV L6-8 DT6-8
B08
Line09 F6
MV L6-9 DT6-9
B09
Line10 F6
Line12 F6
MV L6-10 DT6-10
B10
Line11 F6
MV L6-11 DT6-11
B11
MV L6-12 DT6-12
B12
Line13 F6
MV L6-13 DT6-13
B13
Line14 F6
MV L6-14 DT6-14
B14
Line15 F6
MV L6-15 DT6-15
B15
Line16 F6
Line18 F6
MV L6-16 DT6-16
B16
Line17 F6
MV L6-17 DT6-17
B17
MV L6-18 DT6-18
B18
Line19 F6
MV L6-19 DT6-19
B19
Line20 F6
MV L6-20 DT6-20
B20
Line21 F6
MV L6-21 DT6-21
B21
Line22 F6
Line24 F6
MV L6-22 DT6-22
B22
Line23 F6
MV L6-23 DT6-23
B23
MV L6-24 DT6-24
B24
Line25 F6
MV L6-25 DT6-25
B25
Line26 F6
MV L6-26 DT6-26
B26
Line27 F6
MV L6-27 DT6-27
B27
Line28 F6
Line30 F6
MV L6-28 DT6-28
B28
Line29 F6
MV L6-29 DT6-29
B29
MV L6-30 DT6-30
B30
Line31 F6
MV L6-31 DT6-31
B31
Line32 F6
MV L6-32 DT6-32
B32
Line33 F6
MV L6-33 DT6-33
B33
Line34 F6
MV L6-34 DT6-34
B34
Line35 F6
MV L6-35 DT6-35
B35
MV L6-36 DT6-36

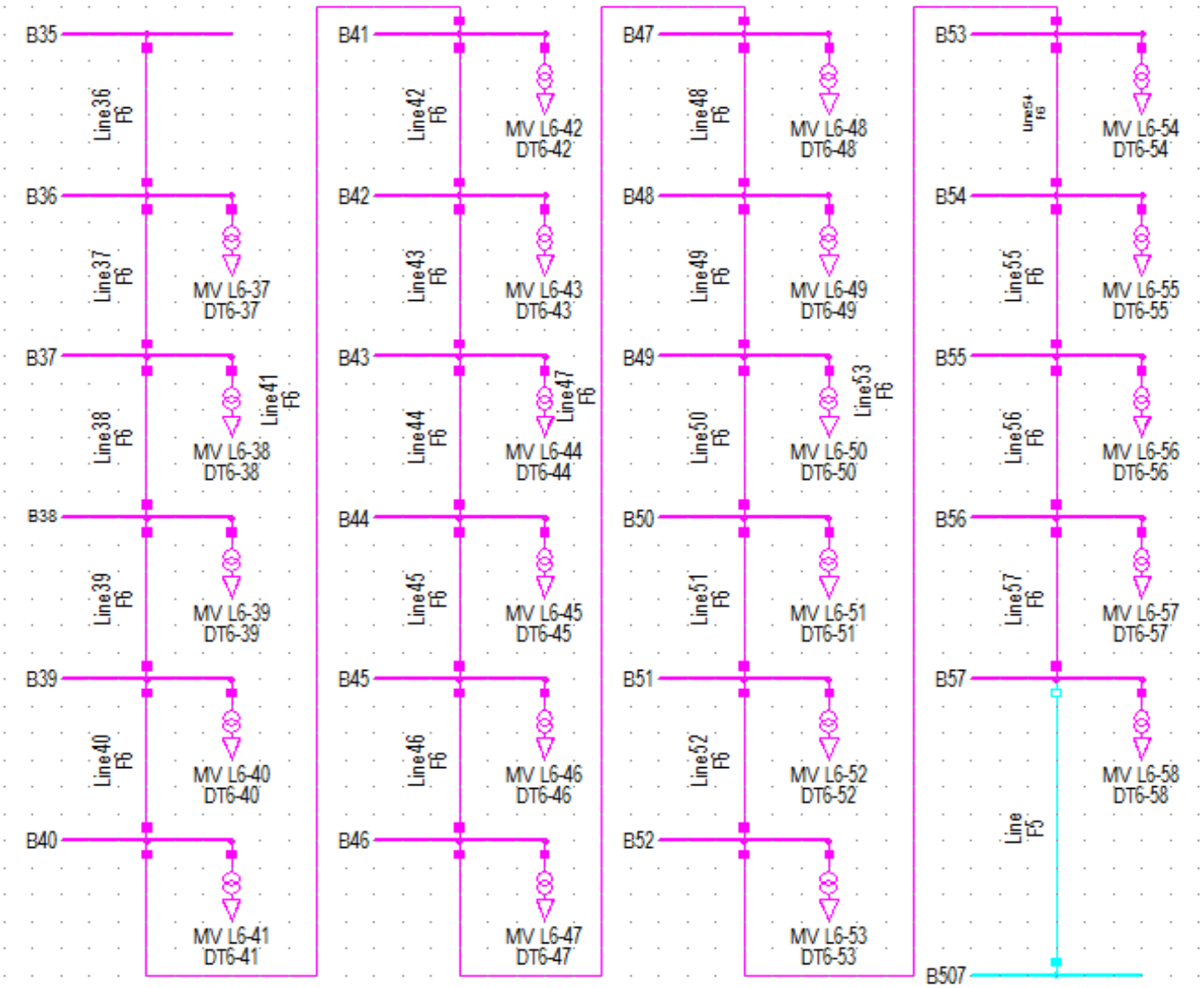
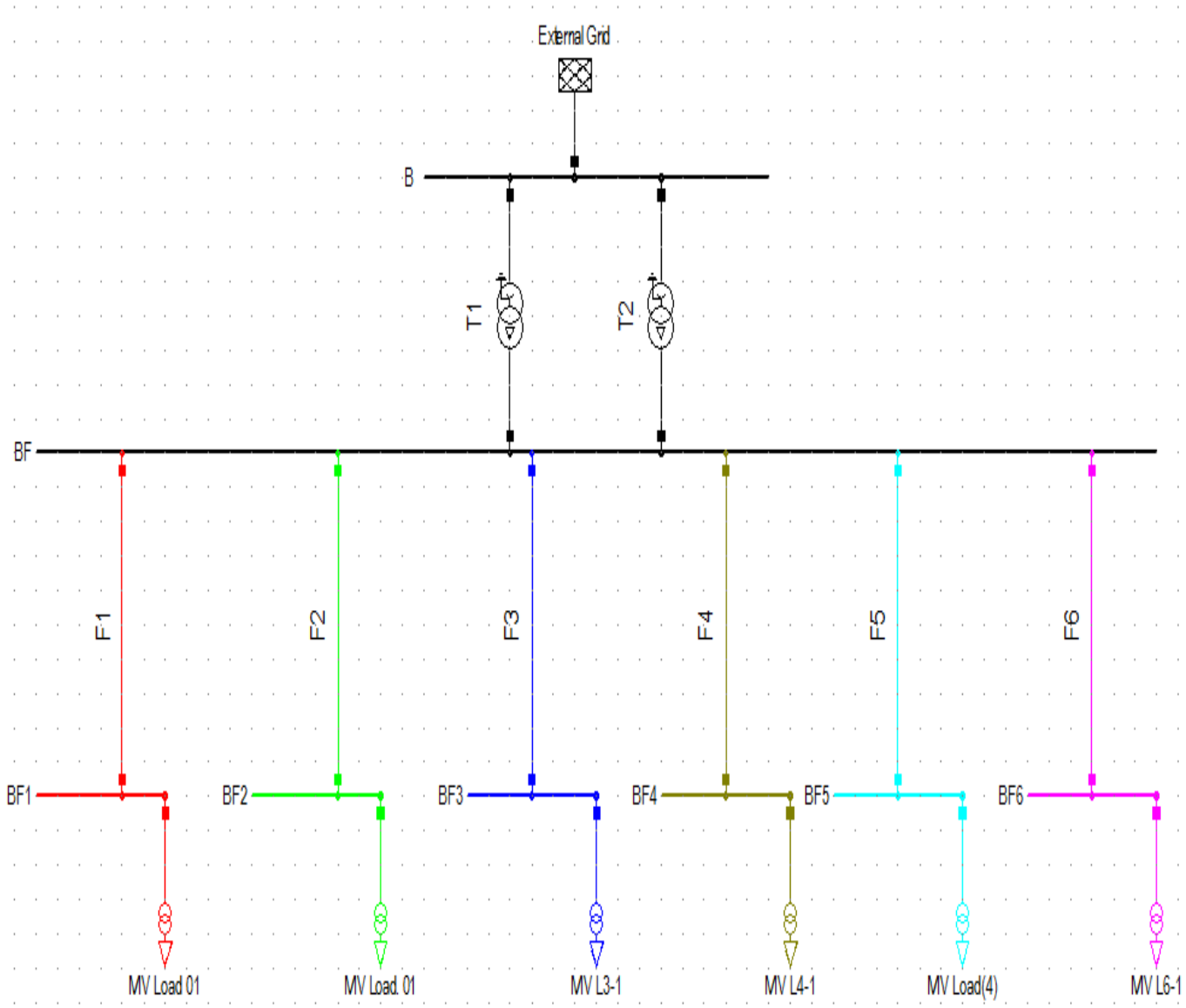
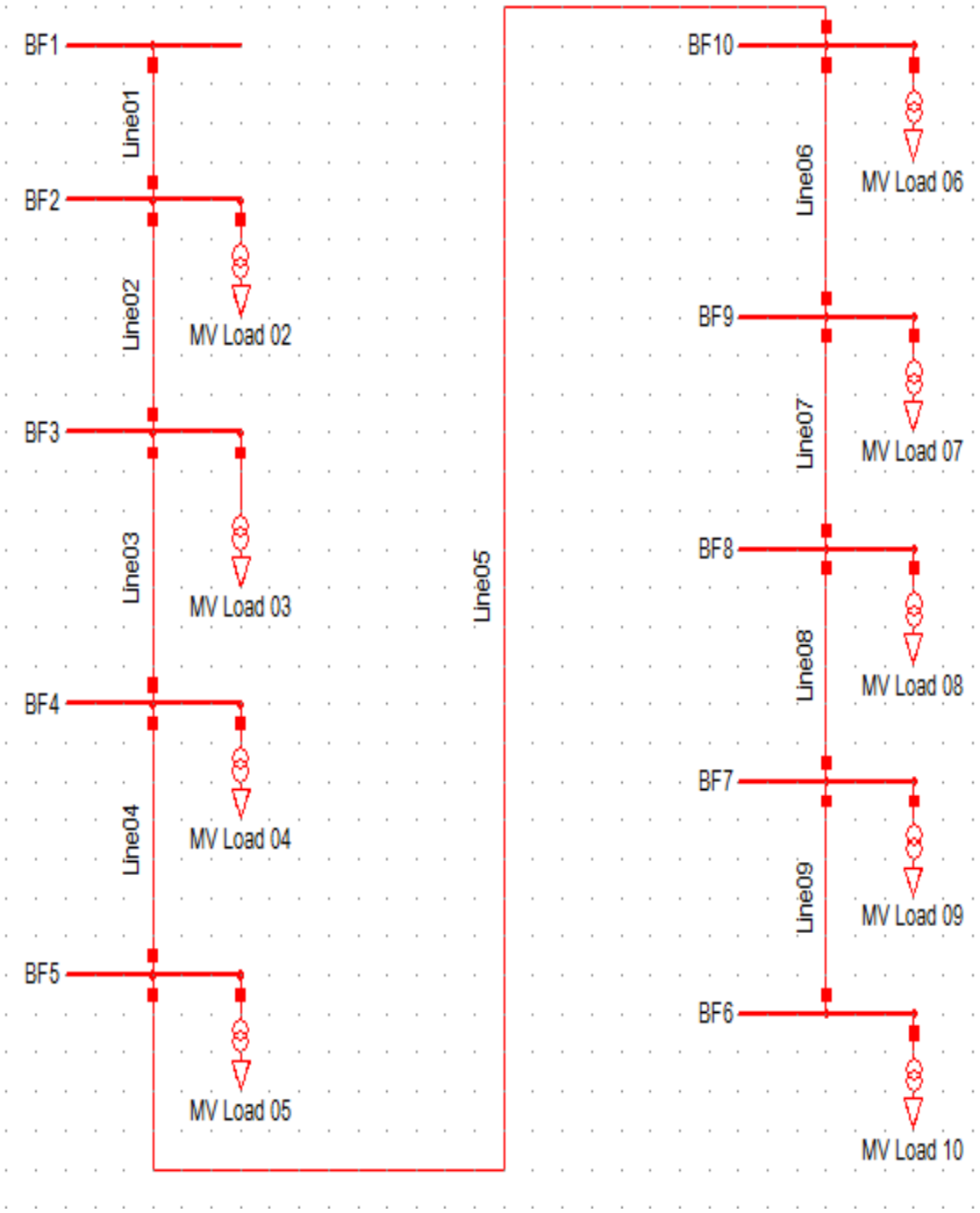
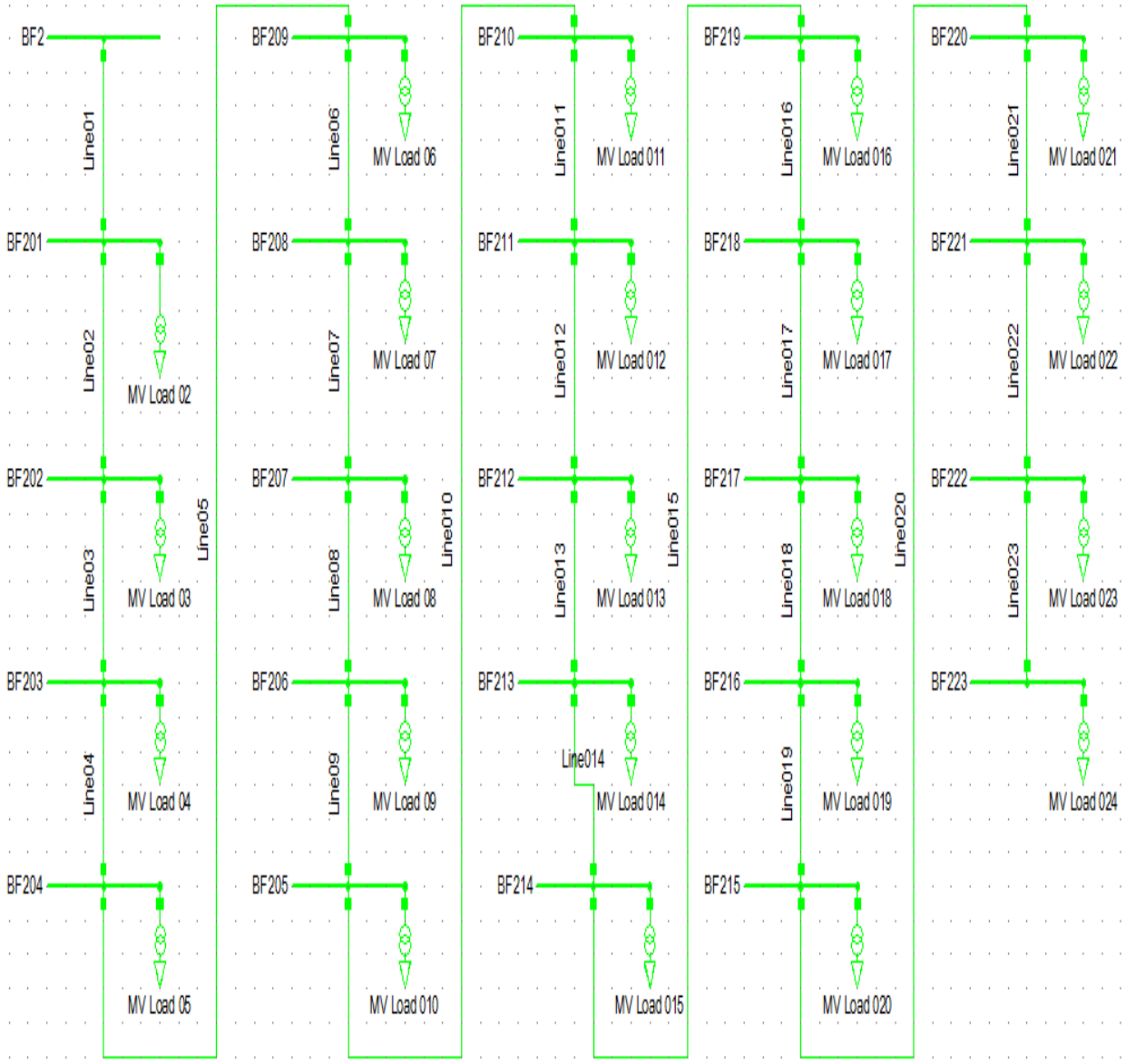


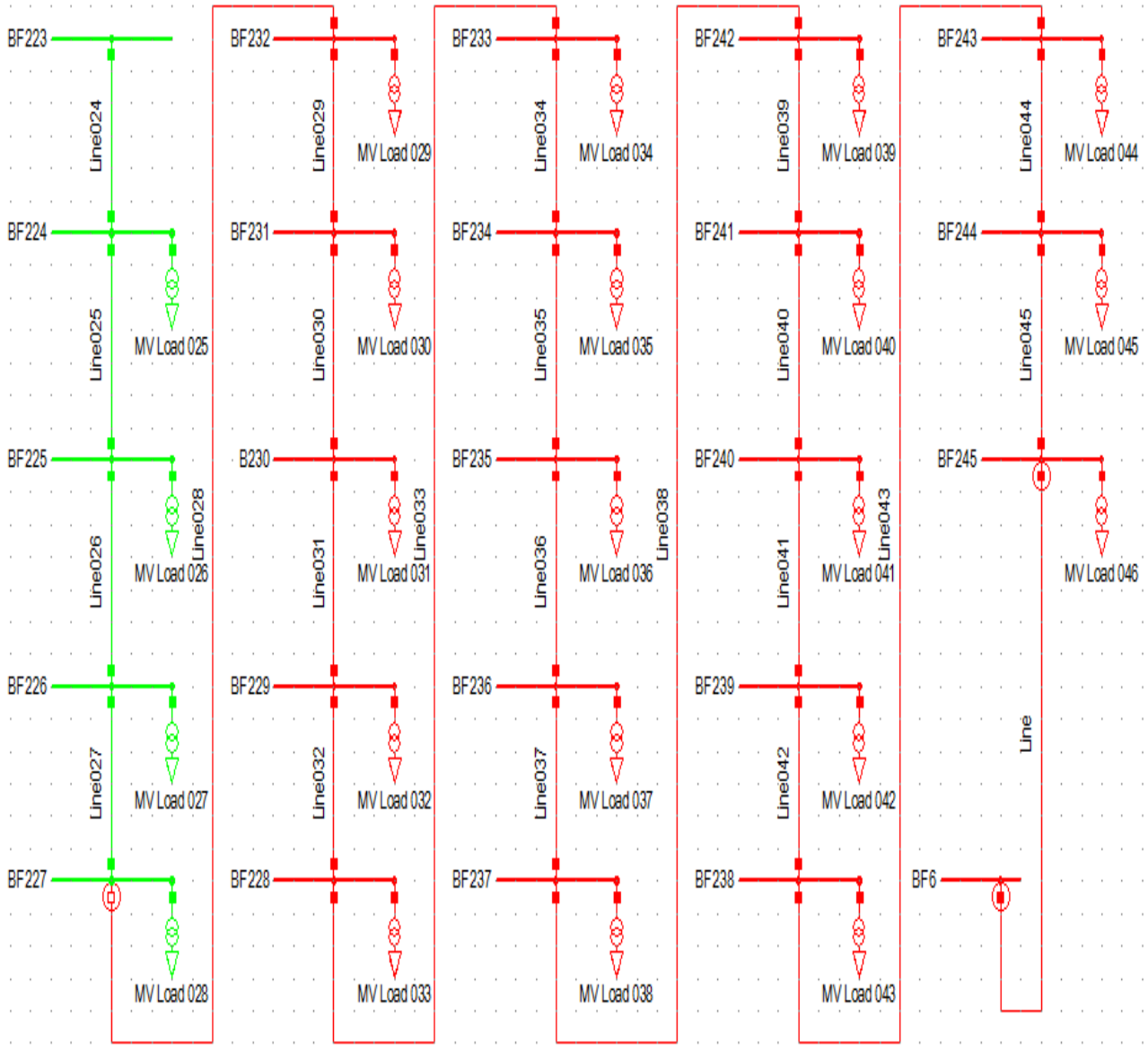
Figure A.7 Addis north 132/15 kV substation feeder 6 one line diagram

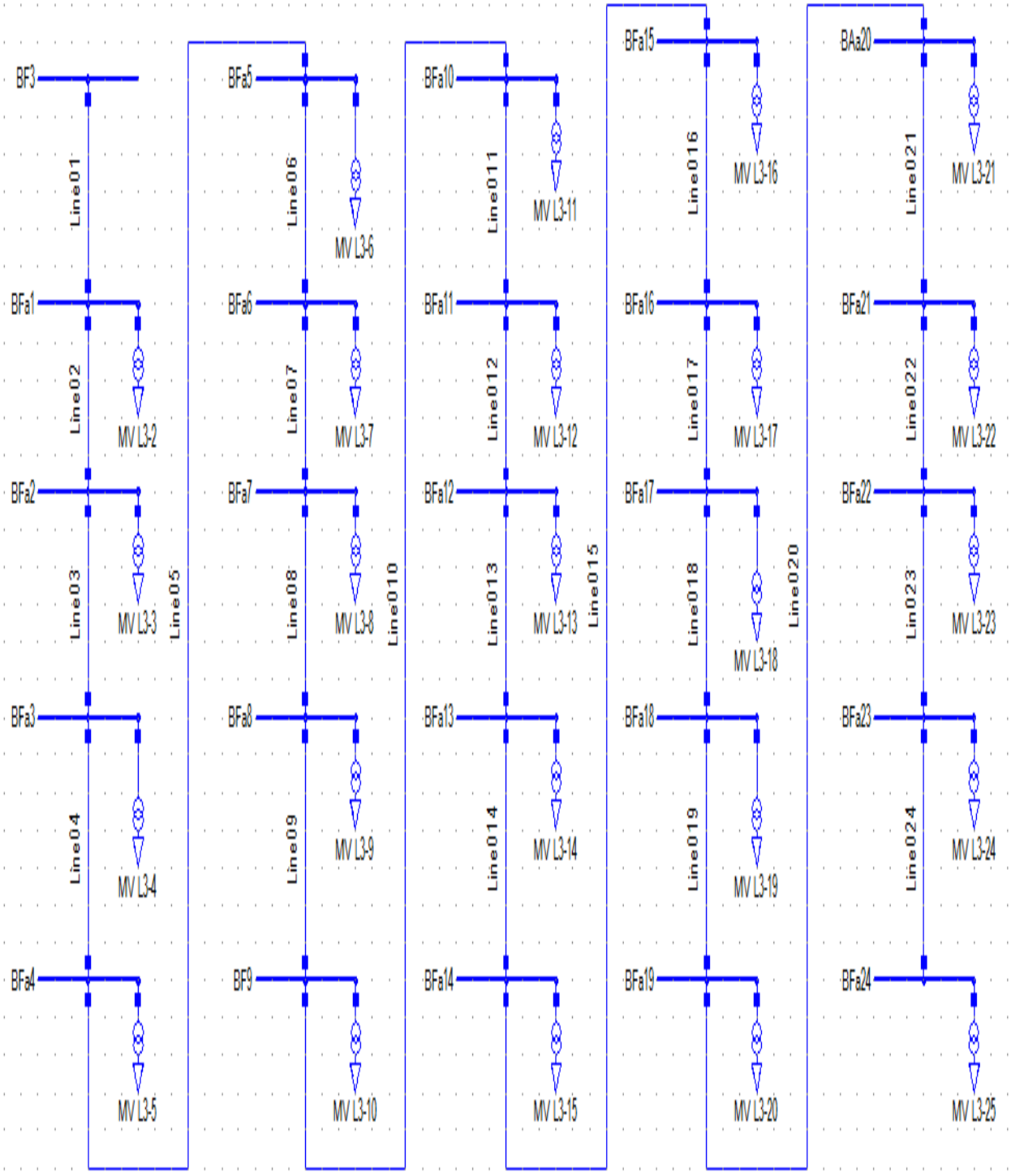
Appendix B: Addis North 132/15 kV Substation feeders network after reconfiguration

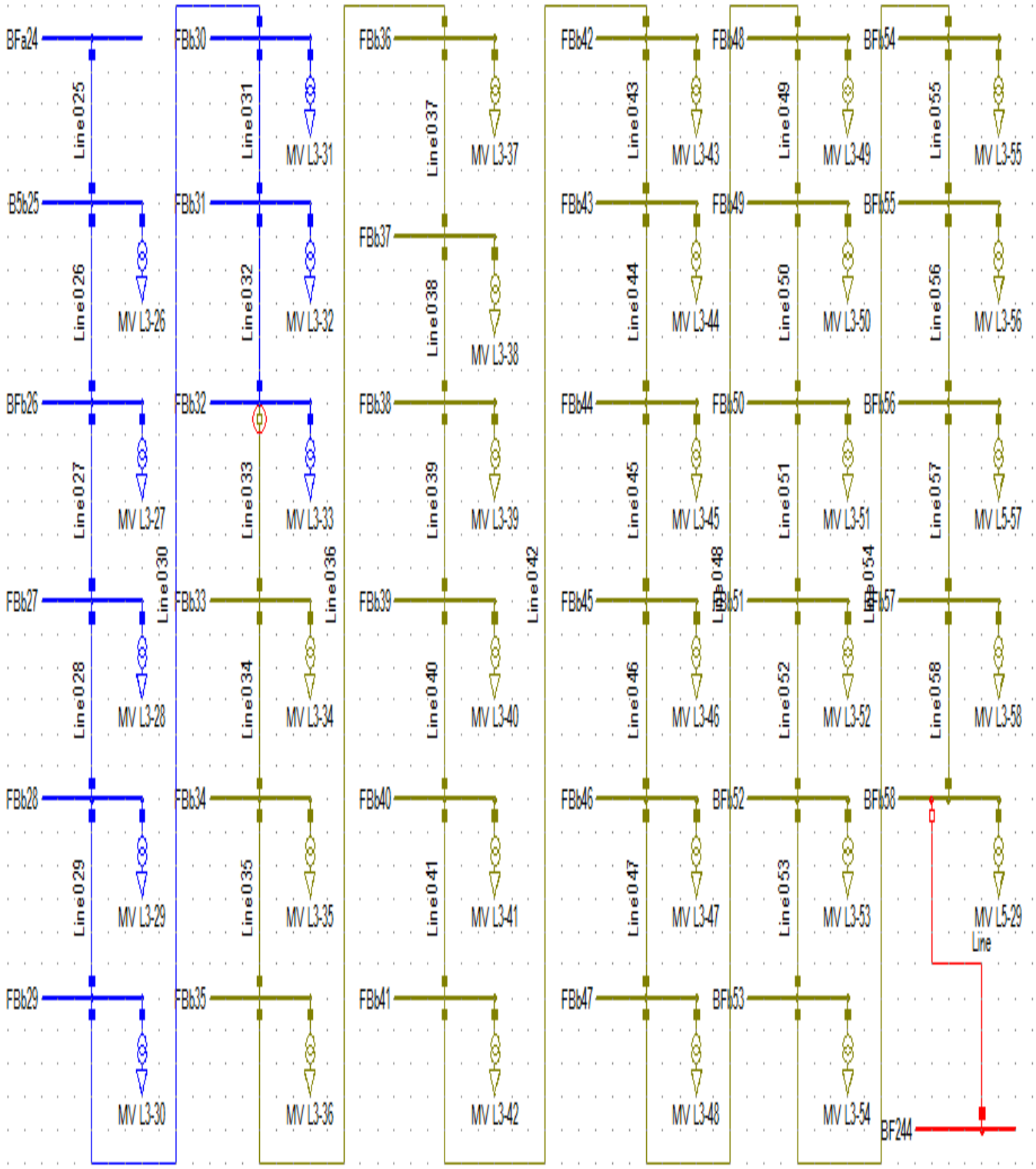


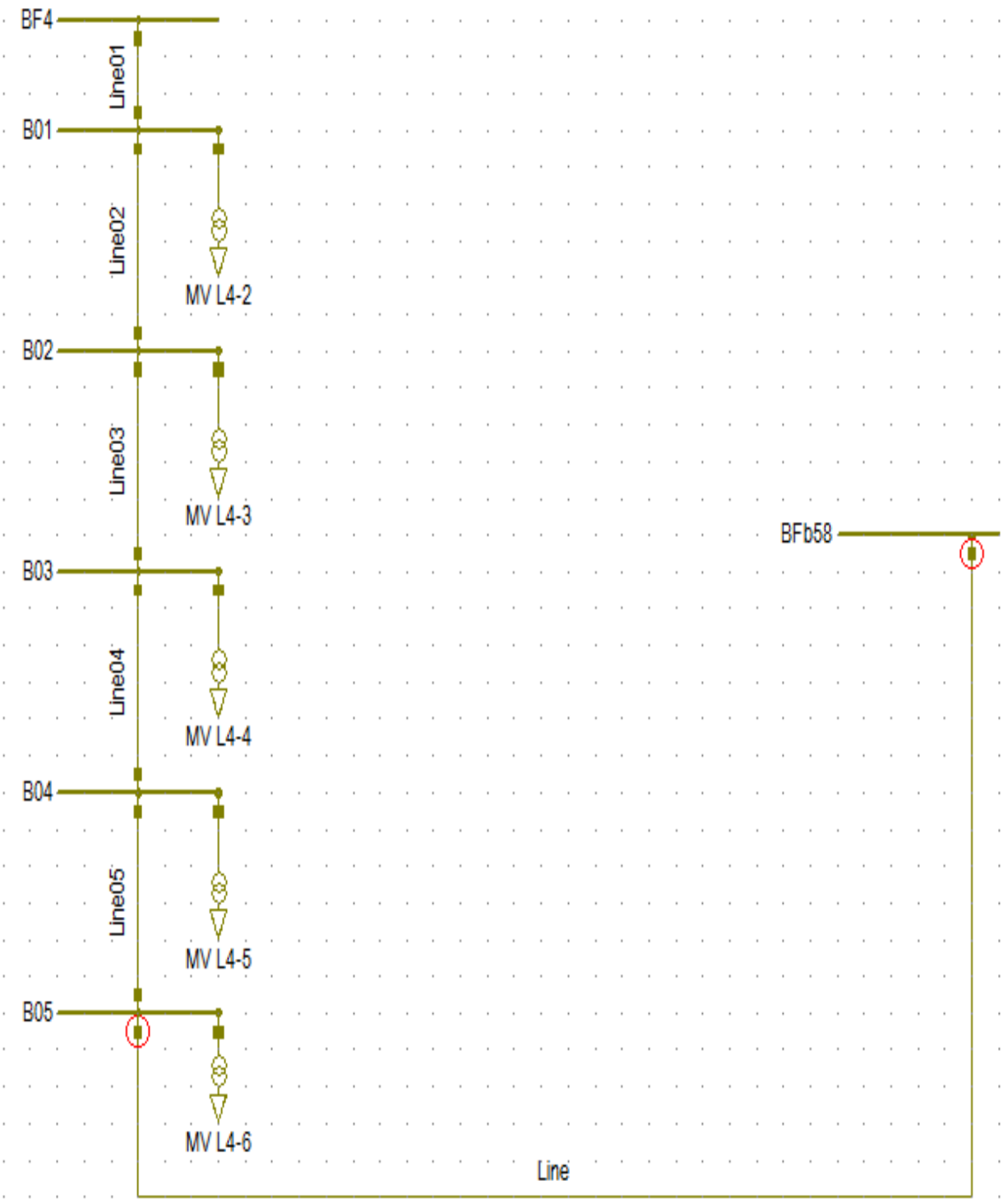

BF1
Line01
BF2
Line02
MV Load 02
BF3
Line03
MV Load 03
BF4
Line04
MV Load 04
BF5
MV Load 05
Line05
BF10
Line06
MV Load 06
BF9
Line07
MV Load 07
BF8
Line08
MV Load 08
BF7
Line09
MV Load 09
BF6
MV Load 10

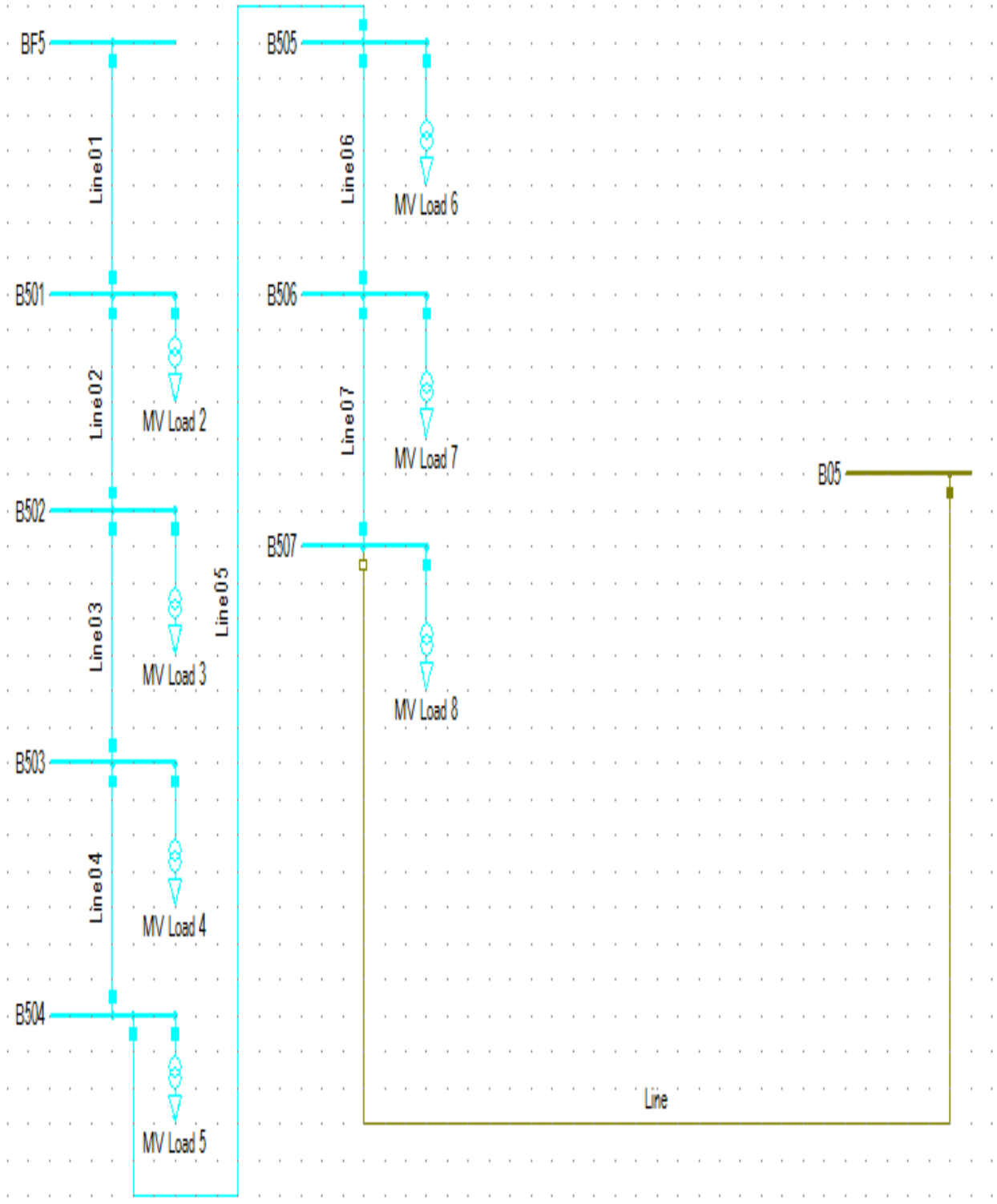

BF2
Line01
BF201
Line02
MV Load 02
BF202
Line03
MV Load 03
Line05
BF203
Line04
MV Load 04
BF204
MV Load 05
BF209
Line06
MV Load 06
BF208
Line07
MV Load 07
BF207
Line08
MV Load 08
Line010
BF206
Line09
MV Load 09
BF205
MV Load 010
BF210
Line011
MV Load 011
BF211
Line012
MV Load 012
BF212
Line013
MV Load 013
Line015
BF213
Line014
MV Load 014
BF214
MV Load 015
BF219
Line016
MV Load 016
BF218
Line017
MV Load 017
BF217
Line018
MV Load 018
Line020
BF216
Line019
MV Load 019
BF215
MV Load 020
BF220
Line021
MV Load 021
BF221
Line022
MV Load 022
BF222
Line023
MV Load 023
BF223
MV Load 024

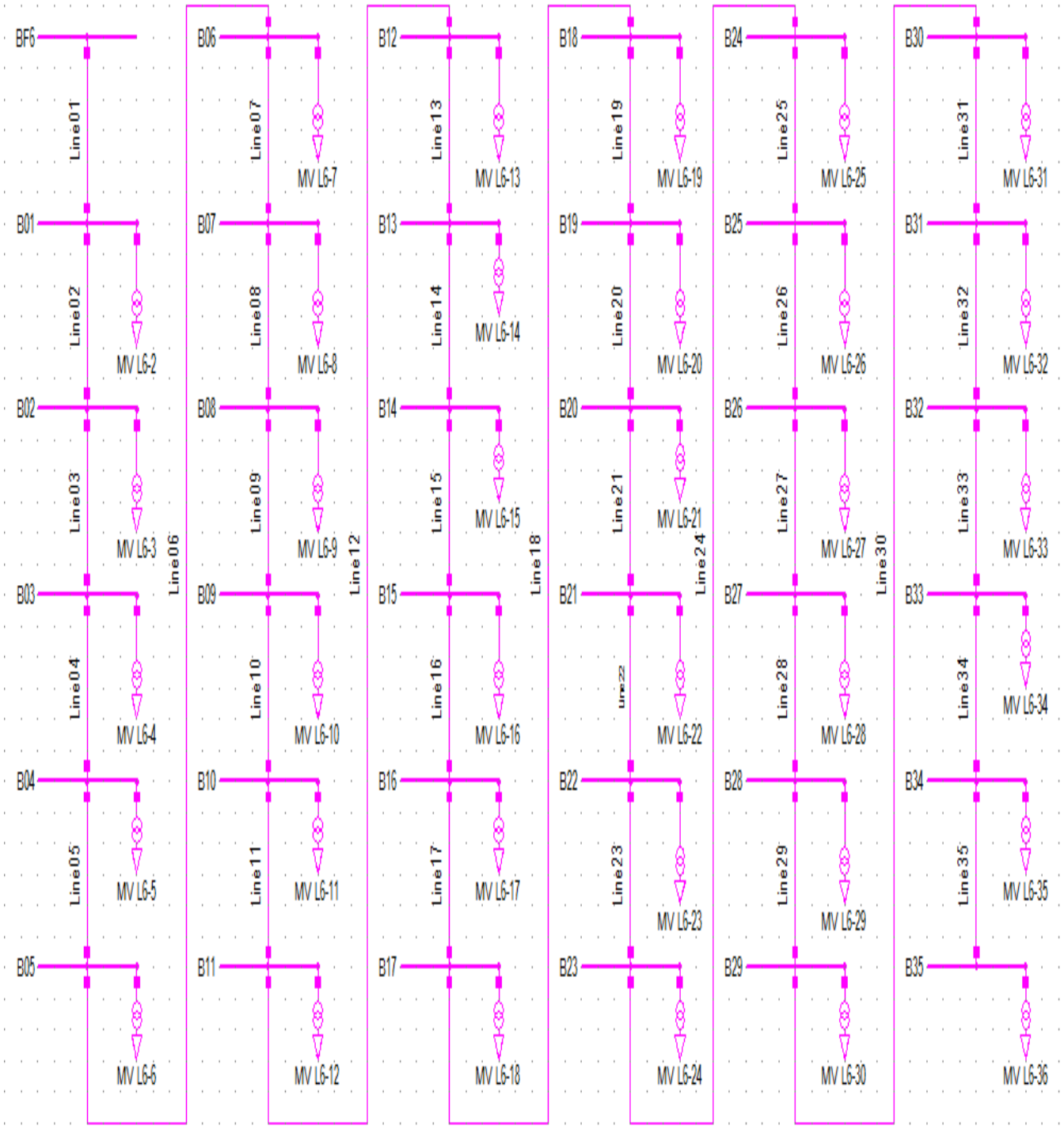

BF223
Line024
BF224
Line025
MV Load 025
BF225
Line026
Line028
MV Load 026
BF226
Line027
MV Load 027
BF227
MV Load 028
BF232
Line029
MV Load 029
BF231
Line030
MV Load 030
B230
Line031
Line033
MV Load 031
BF229
Line032
MV Load 032
BF228
MV Load 033
BF233
Line034
MV Load 034
BF234
Line035
MV Load 035
BF235
Line036
Line038
MV Load 036
BF236
Line037
MV Load 037
BF237
MV Load 038
BF242
Line039
MV Load 039
BF241
Line040
MV Load 040
BF240
Line041
Line043
MV Load 041
BF239
Line042
MV Load 042
BF238
MV Load 043
BF243
Line044
MV Load 044
BF244
Line045
MV Load 045
BF245
Line
MV Load 046
BF6


BF3
Line01
BFa1
Line02
MV L3-2
BFa2
Line03
MV L3-3
BFa3
Line04
MV L3-4
BFa4
MV L3-5
Line05
BFa5
Line06
MV L3-6
BFa6
Line07
MV L3-7
BFa7
Line08
MV L3-8
BFa8
Line09
MV L3-9
BF9
MV L3-10
Line010
BFa10
Line011
MV L3-11
BFa11
Line012
MV L3-12
BFa12
Line013
MV L3-13
BFa13
Line014
MV L3-14
BFa14
MV L3-15
Line015
BFa15
Line016
MV L3-16
BFa16
Line017
MV L3-17
BFa17
Line018
MV L3-18
BFa18
Line019
MV L3-19
BFa19
MV L3-20
Line020
BAa20
Line021
MV L3-21
BFa21
Line022
MV L3-22
BFa22
Lin023
MV L3-23
BFa23
Line024
MV L3-24
BFa24
MV L3-25


BFa24
Line025
FBb30
Line031
MV L3-31
FBb36
Line037
MV L3-37
FBb42
Line043
MV L3-43
FBb48
Line049
MV L3-49
BFb54
Line055
MV L3-55
B5b25
Line026
MV L3-26
FBb31
Line032
MV L3-32
FBb37
Line038
MV L3-38
FBb43
Line044
MV L3-44
FBb49
Line050
MV L3-50
BFb55
Line056
MV L3-56
BFb26
Line027
MV L3-27
FBb32
Line033
MV L3-33
FBb38
Line039
MV L3-39
FBb44
Line045
MV L3-45
FBb50
Line051
MV L3-51
BFb56
Line057
MV L5-57
FBb27
Line028
MV L3-28
Line030
FBb33
Line034
MV L3-34
Line036
FBb39
Line040
MV L3-40
Line042
FBb45
Line046
MV L3-46
Line048
FBb51
Line052
MV L3-52
Line054
BFb57
Line058
MV L3-58
FBb28
Line029
MV L3-29
FBb34
Line035
MV L3-35
FBb40
Line041
MV L3-41
FBb46
Line047
MV L3-47
BFb52
Line053
MV L3-53
BFb58
MV L5-29
Line
FBb29
MV L3-30
FBb35
MV L3-36
FBb41
MV L3-42
FBb47
MV L3-48
BFb53
MV L3-54
BF244


BF4
Line01
B01
Line02
MV L4-2
B02
Line03
MV L4-3
BFb58
B03
Line04
MV L4-4
B04
Line05
MV L4-5
B05
MV L4-6
Line


BF5
B505
Line01
Line06
MV Load 6
B501
B506
Line02
Line07
MV Load 2
MV Load 7
B05
B502
B507
Line03
Line05
MV Load 3
MV Load 8
B503
Line04
MV Load 4
B504
MV Load 5
Line


BF6
B01
B02
B03
B04
B05
B06
B07
B08
B09
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
B25
B26
B27
B28
B29
B30
B31
B32
B33
B34
B35
Line01
Line02
Line03
Line04
Line05
Line06
Line07
Line08
Line09
Line10
Line11
Line12
Line13
Line14
Line15
Line16
Line17
Line18
Line19
Line20
Line21
Line22
Line23
Line24
Line25
Line26
Line27
Line28
Line29
Line30
Line31
Line32
Line33
Line34
Line35
MV L6-2
MV L6-3
MV L6-4
MV L6-5
MV L6-6
MV L6-7
MV L6-8
MV L6-9
MV L6-10
MV L6-11
MV L6-12
MV L6-13
MV L6-14
MV L6-15
MV L6-16
MV L6-17
MV L6-18
MV L6-19
MV L6-20
MV L6-21
MV L6-22
MV L6-23
MV L6-24
MV L6-25
MV L6-26
MV L6-27
MV L6-28
MV L6-29
MV L6-30
MV L6-31
MV L6-32
MV L6-33
MV L6-34
MV L6-35
MV L6-36

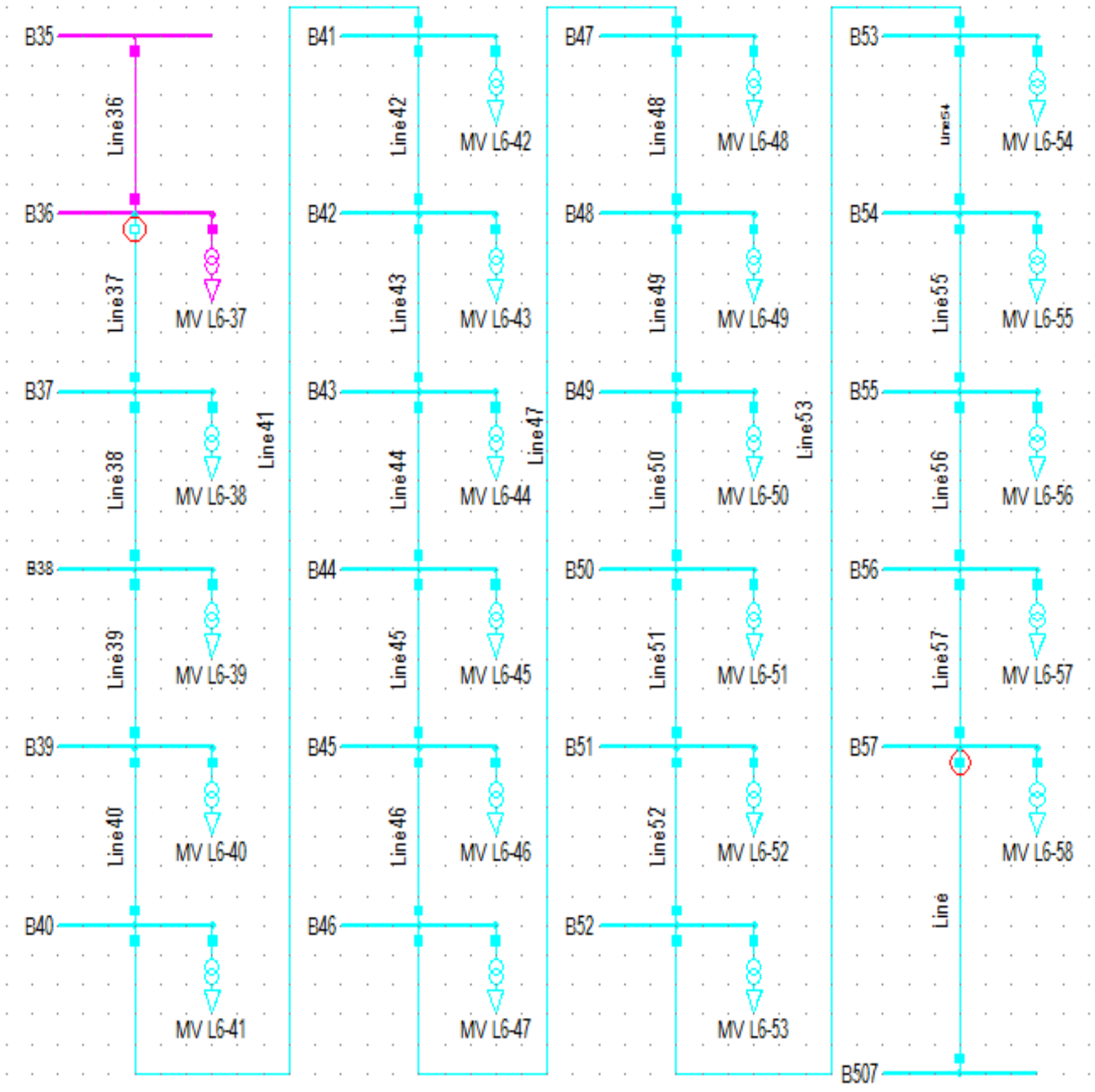


Figure A.1 Addis North 132/15 kV Substation feeders network after reconfiguration

Appendix C: Addis North 132/15 kV Substation feeders data

This appendix consists of tables for Addis North 132/15 KV Substation feeders data, which is obtained from Ethiopia Electric Utility (EEU).

Load No.	Load (MW)	Bus (From)	Bus (To)	R (ohm)	X (ohm)
01	0.25	BF	BF1	0.0578	0.036
02	0.25	BF1	2	0.085544	0.05328
03	1.00	2	3	0.085544	0.05328
04	0.49	3	4	0.083232	0.05184
05	0.49	4	5	0.087856	0.05472
06	1.00	5	6	0.087856	0.05472
07	0.25	6	7	0.087856	0.05472
08	0.49	7	8	0.083232	0.05184
09	0.25	8	9	0.083232	0.05184
10	0.49	9	10	0.085544	0.05328

Table A-1 Network Data of Feeder 1 of Addis North 132/15 KV Substation

Load No.	Load (MW)	Bus (From)	Bus (To)	R (ohm)	X (ohm)
01	0.2981436	BF	BF2	0.0578	0.036
02	0.2981436	BF2	01	0.10404	0.0648
03	0.2771436	01	02	0.07225	0.045
04	0.2771436	02	03	0.08381	0.0522
05	0.2771436	03	04	0.10115	0.063
06	0.2771436	04	05	0.10115	0.063
07	0.2771436	05	06	0.08381	0.0522
08	0.1495091	06	07	0.10693	0.0666
09	0.2771436	07	08	0.10693	0.10693
10	0.1495091	08	09	0.07225	0.045
11	0.1495091	09	10	0.10115	0.063
12	0.2771436	10	11	0.08381	0.0522
13	0.1495091	11	12	0.10404	0.0648
14	0.2771436	12	13	0.10693	0.0666
15	0.1495091	13	14	0.07225	0.045
16	0.934432	14	15	0.07225	0.045
17	0.934432	15	16	0.10693	0.0666
18	0.5747545	16	17	0.10115	0.063
19	0.2771436	17	18	0.08381	0.0522
20	0.1495091	18	19	0.10404	0.0648

21	0.2771436	19	20	0.08381	0.0522
22	0.1495091	20	21	0.10115	0.063
23	0.2771436	21	22	0.07225	0.045
24	0.2771436	22	23	0.07225	0.045
25	0.5747545	23	24	0.10693	0.0666
26	0.2354769	24	25	0.08381	0.0522
27	0.5747545	25	26	0.10115	0.063
28	0.1495091	26	27	0.08381	0.0522
29	0.1495091	27	28	0.10693	0.0666
30	0.1495091	28	29	0.07225	0.045
31	0.1495091	29	30	0.10115	0.063
32	0.5747545	30	31	0.10115	0.10115
33	0.2771436	31	32	0.08381	0.0522
34	0.1495091	32	33	0.10693	0.0666
35	0.1495091	33	34	0.10693	0.0666
36	0.2771436	34	35	0.07225	0.045
37	0.1495091	35	36	0.08381	0.0522
38	0.2771436	36	37	0.10404	0.0648
39	0.1495091	37	38	0.10693	0.0666
40	0.2771436	38	39	0.08381	0.0522
41	0.2771436	39	40	0.10115	0.063
42	0.2771436	40	41	0.07225	0.045

43	0.2771436	41	42	0.10693	0.0666
44	0.2771436	42	43	0.07225	0.045
45	0.2771436	43	44	0.10115	0.063
46	0.5980365	44	45	0.10404	0.0648

Table A-2 Network Data of Feeder 2 of Addis North 132/15 KV substation

Load No.	**Load (MW)**	**Bus (From)**	**Bus (To)**	**R (ohm)**	**X (ohm)**
1	0.231548	BF	BF3	0.0578	0.036
2	0.7867537	BF3	01	0.13294	0.0828
3	0.1157399	01	02	0.07514	0.0468
4	0.07350729	02	03	0.07514	0.0468
5	0.23154800	03	04	0.0578	0.036
6	0.07350729	04	05	0.0867	0.054
7	0.115774	05	06	0.0867	0.054
8	0.07350729	06	07	0.13294	0.0828
9	0.115774	07	08	0.07514	0.0468
10	0.1102609	08	09	0.0867	0.054
11	0.08269571	09	10	0.0578	0.036
12	0.231548	10	11	0.07514	0.0468
13	0.231548	11	12	0.0867	0.054
14	0.231548	12	13	0.13294	0.0828
15	0.07350729	13	14	0.0578	0.036
16	0.231548	14	15	0.0578	0.036

17	0.07350729	15	16	0.0867	0.054
18	0.07350729	16	17	0.0867	0.054
19	0.231548	17	18	0.07514	0.0468
20	0.115774	18	19	0.0867	0.054
21	0.07350729	19	20	0.07514	0.0468
22	0.1102609	20	21	0.0867	0.054
23	0.7867537	21	22	0.13294	0.0828
24	0.115774	22	23	0.07514	0.0468
25	0.2940292	23	24	0.0867	0.054
26	0.07350729	24	25	0.07514	0.0468
27	0.07350729	25	26	0.0867	0.054
28	0.231548	26	27	0.0578	0.036
29	0.1102609	27	28	0.07514	0.0468
30	0.1102609	28	29	0.0867	0.054
31	0.231548	29	30	0.07514	0.0468
32	0.231548	30	31	0.07514	0.0468
33	0.78675370	31	32	0.13294	0.0828
34	0.231548	32	33	0.0867	0.054
35	0.231548	33	34	0.0867	0.054
36	0.115774	34	35	0.0578	0.036
37	0.07350729	35	36	0.07514	0.0468
38	0.23154800	36	37	0.0867	0.054
39	0.7867537	37	38	0.13294	0.0828
40	0.231548	38	39	0.13294	0.0828
41	0.115774	39	40	0.0867	0.054
42	0.07350729	40	41	0.07514	0.0468
43	0.23154800	41	42	0.0578	0.036

44	0.07350729	42	43	0.0867	0.054
45	0.07350729	43	44	0.07514	0.0468
46	0.07350729	44	45	0.0867	0.054
47	0.23154800	45	46	0.0578	0.036
48	0.07350729	46	47	0.0867	0.054
49	0.1102609	47	48	0.07514	0.0468
50	0.115774	48	49	0.0867	0.054
51	0.231548	49	50	0.07514	0.0468
52	0.115774	50	51	0.13294	0.0828
53	0.1102609	51	52	0.13294	0.0828
54	0.07350729	52	53	0.07514	0.0468
55	0.231548	53	54	0.07514	0.0468
56	0.07350729	54	55	0.0867	0.054
57	0.231548	55	56	0.07514	0.0468
58	0.231548	56	57	0.13294	0.0828
59	0.2021451	57	58	0.07514	0.0468

Table A-3 Network Data of Feeder 3 of Addis North 132/15KV substation

Load No.	**Load (MW)**	**Bus (From)**	**Bus (To)**	**R (ohm)**	**X (ohm)**
1	0.25	BF	BF4	0.0578	0.036
2	0.99	BF4	01	0.0867	0.054
3	0.25	01	02	0.06936	0.0432
4	0.25	02	03	0.06936	0.0432
5	0.25	03	04	0.06358	0.0396

6	0.99	04	05	0.0867	0.054

Table A- 4 Network Data of Feeder 4 of Addis North 132/15 KV substation

Load No.	**Load (MW)**	**Bus (From)**	**Bus (To)**	**R (ohm)**	**X (ohm)**
1	0.25	BF	BF5	0.0578	0.036
2	0.25	BF5	01	0.0867	0.054
3	0.32	01	02	0.09826	0.0612
4	0.75	02	03	0.0867	0.054
5	0.75	03	04	0.06936	0.0432
6	0.32	04	05	0.0578	0.036
7	0.32	05	06	0.06936	0.0432
8	0.32	06	07	0.09826	0.0612

Table A-5 Network Data of Feeder 5 of Addis North 132/15 KV substation

Load No.	**Load (MW)**	**Bus (From)**	**Bus (To)**	**R (ohm)**	**X (ohm)**
1	0.2951094	BF	BF6	0.0578	0.036
2	0.09585703	BF6	01	0.09248	0.0576
3	0.09585703	01	02	0.07514	0.0468
4	0.4173342	02	03	0.0578	0.036
5	0.2951094	03	04	0.0578	0.036
6	0.4173342	04	05	0.07514	0.0468
7	0.2951094	05	06	0.07514	0.0468

8	0.09585703	06	07	0.0578	0.036
9	0.04792851	07	08	0.09248	0.0576
10	0.04792851	08	09	0.0578	0.036
11	0.09585703	09	10	0.0578	0.036
12	0.4173342	10	11	0.07514	0.0468
13	0.3019496	11	12	0.0578	0.036
14	0.09585703	12	13	0.09248	0.0576
15	0.2951094	13	14	0.09248	0.0576
16	0.04792851	14	15	0.0578	0.036
17	0.1437855	15	16	0.07514	0.0468
18	0.4173342	16	17	0.0578	0.036
19	0.4173342	17	18	0.07514	0.0468
20	0.2663594	18	19	0.0578	0.036
21	0.09585703	19	20	0.0578	0.036
22	0.4173342	20	21	0.09248	0.0576
23	0.09585303	21	22	0.07514	0.0468
24	0.4173342	22	23	0.0578	0.036
25	0.09585703	23	24	0.07514	0.0468
26	0.09585703	24	25	0.0578	0.036
27	0.02396426	25	26	0.07514	0.0468
28	0.04792851	26	27	0.0578	0.036
29	0.09585703	27	28	0.09248	0.0576
30	0.09585703	28	29	0.09248	0.0576
31	0.4173342	29	30	0.0578	0.036
32	0.02396426	30	31	0.0578	0.036
33	0.04792851	31	32	0.07514	0.0468
34	0.2663594	32	33	0.0578	0.036

35	0.09585703	33	34	0.07514	0.0468
36	0.4173342	34	35	0.0578	0.036
37	0.02396426	35	36	0.07514	0.468
38	0.09585703	36	37	0.0578	0.036
39	0.2663594	37	38	0.0578	0.036
40	0.09585703	38	39	0.09248	0.0576
41	0.2663594	39	40	0.0578	0.036
42	0.4173342	40	41	0.0578	0.036
43	0.4173342	41	42	0.07514	0.0468
44	0.2663594	42	43	0.0578	0.036
45	0.09585703	43	44	0.0578	0.036
46	0.09585703	44	45	0.09248	0.0576
47	0.09585703	45	46	0.0578	0.036
48	0.4173342	46	47	0.09248	0.0576
49	0.04792851	47	48	0.0578	0.036
50	0.09585703	48	49	0.07514	0.0468
51	0.2663594	49	50	0.0578	0.036
52	0.04792851	50	51	0.09248	0.0576
53	0.1437855	51	52	0.0578	0.036
54	0.4173342	52	53	0.0578	0.036
55	0.2663594	53	54	0.07514	0.0468
56	0.09585703	54	55	0.0578	0.036
57	0.1437855	55	56	0.0578	0.036
58	0.4173342	56	57	0.09248	0.0576

Table A- 6 Network Data of Feeder 6 of Addis North 132/15 Substation

Appendix D: Matlab program for reconfiguration of distribution network using Genetic algorithm

```
%%%%%GENETIC ALGORITHM FOR F1 AND F2 POWER OPTIMIZATION %%%%%%
clc
clear all
close all
display('Program to optimize the power');
p_no=input ('Enter the size of initial population ::');
bits=input('Enter the bit size of each strings::') ;
final=[];
iterations=input('Enter the number of iterations to perform the process of genetic algorithm::');
tournament_no=input ('Enter the number of times you want to perform tournament selection ::');
%% power before optimization
p_1=0;
p_2=0;
I_1=0.926354; %in ampere
I_2=0.716774444747; %in ampere
R=0.578; %in ohm
u=10; %number of switches in feeder 1
v=46; %number of switches in feeder 2
n=u+v;
for i=1:u
   p_1=p_1+(I_1^2)*R;
end;
for i=1:v
   p_2=p_2+(I_2^2)*R;
end;
display ('power before optimization')
p_1
p_2
%% generate the bits randomly
```

```
p=rand(p_no,bits);
for i=1:p_no
   for j=1:bits
      if (p(i,j)>=0.5)
         popu(i,j)=1;
      else
         popu(i,j)=0;
      end;
   end;
end;
for z=1:iterations
p_t=p_1+p_2;
val=zeros(p_no,1);
%%calculate the value of strings
for i=1:p_no
     for j=bits:-1:1
        k=bits-j;
        val(i,1)=val(i,1)+pow2(popu(i,j),k);
     end;
end;
val;
%% generate the random numbers to do the tournament selection
for i=1:tournament_no
   select(i,:)=[floor(1+rand(1)*(p_no-1))  ceil(1+rand(1)*(p_no-1))];%% have to make changes
here
end;
select;
%%calculate the fitness value and perform tournament selection
tour_val_1=zeros(tournament_no,2);
tour_val_2=zeros(tournament_no,2);
for i=1:tournament_no
```

```
tour_val_11=zeros(select(i,1),1);
tour_val_22=zeros(select(i,2),1);
for j=2:val(select(i))+1
   tour_val_11(j)=tour_val_11(j-1)+(I_2^2)*R;
   tour_val_1(i)=tour_val_11(j);
end;
for j=2:val(select(i,2))+1
   tour_val_22(j)=tour_val_22(j-1)+(I_2^2)*R;
   tour_val_2(i)=tour_val_22(j);
end;
tor_min1=min(abs(tour_val_1(i)-p_t/2),abs(tour_val_2(i)-p_t/2));
if tor_min1==abs(tour_val_1(i)-p_t/2)
      fitest(i,:)=popu(select(i,1),:);
else
      fitest(i,:)=popu(select(i,2),:);
end;
end;
tour_val_1;
tour_val_2;
fitest;
prob_crossover=0.8;
prob_mutation=0.01;
crossover_point=2;
for x=1:ceil(p_no/2)
%%selecting of parents
parents=[];
crossover=[];
crossover_val=[];
crossover_val_mapped=[];
crossover_fitness=[];
```

```
parents=[fitest(floor(1+rand(1)*(tournament_no-1)),:);      fitest(ceil(1+rand(1)*(tournament_no-1)),:)];
%selecting of childrens
val_gen=rand(1);
child=[];
if (val_gen<prob_crossover)
  child=[[parents(1,1:crossover_point)
parents(2,(crossover_point+1):bits)];[parents(2,1:crossover_point)
parents(1,(crossover_point+1):bits)]];
end;
%%calculate the value of parents and children
crossover=[parents;child];
crossover_val=zeros(size(crossover,1),1);
for i=1:size(crossover,1)
    for j=bits:-1:1
        k=bits-j;
        crossover_val(i,1)=crossover_val(i,1)+pow2(crossover(i,j),k);
    end;
end;
crossover_val;
%%calculate the fitness value to select the the most suitable solution
for j=1:size(crossover_val,1)
p_21=zeros(size(crossover_val,1),1);
% p_2=zeros(size(crossover_val,1),1);
p_2=zeros(size(crossover_val,1),1);
for i=2:crossover_val(j,1)+1
   p_21(i,1)=p_21(i-1,1)+(I_2^2)*R;
   p_2(j,1)=p_21(i,1);
end;
crossover_fitness=p_2;
end
```

```
select_min1=min(abs(crossover_fitness-p_t/2));
for i=1:size(crossover_val,1)
      if (select_min1==abs(crossover_fitness(i,1)-p_t/2))
         pos1=i;
         break;
      end;
end;
%%the position of the max fitness value after crossover
pos1;
% to select the 2nd bst fitness value.
select_min2=abs(crossover_fitness(1,1)-p_t/2);
pos2=1;
if select_min2==select_min1
   select_min2=abs(crossover_fitness(2,1)-p_t/2);
   pos2=2;
end;
for i=1:size(crossover_fitness,1)
      if(select_min1==abs(crossover_fitness(i,1)-p_t/2))
         continue;
      elseif (abs(crossover_fitness(i,1)-p_t/2)~=select_min2)
         select_min2=abs(crossover_fitness(i,1)-p_t/2);
         pos2=i;
      end;
end;
% the values of 2nd best fitness values and its position
select_min2;
pos2;
% taking the best 2 fitest solution and creating a different mating pool
final=[final;crossover(pos1,:);crossover(pos2,:)];
end;%% end of crossover
%performing mutation over the solutions generated after crossover.
```

```
for i=1:size(final,1)
   for j=1:size(final,2)
   val_gen=rand(1);
      if (val_gen<prob_mutation)
        final1(i,j)=xor(final(i,j),1);
      else
        final1(i,j)=final(i,j);
      end;
   end;
end;
final=[];
popu=final1;
p_no=size(popu,1);
val=zeros(p_no,1);
for i=1:p_no
     for j=bits:-1:1
        k=bits-j;
        val(i,1)=val(i,1)+pow2(popu(i,j),k);
     end;
end;
u2=val(1,1);
v2=n-val(1,1);
p_1=0;
p_2=0;
for i=1:u2
   p_2=p_2+(I_2^2)*R;
end;
p_1=p_t-p_2;
end;%%end to the number of iterations for the entire process;
display('The final best solution is given as....')
v2
```

```
u2
display('power after optimization')
p_1
p_2

%%%%%%GENETIC A LGORITHM FOR F3 AND F4 POWER OPTIMIZATION %%%%
clc
clear all
close all
display('Program to optimize the power ');
p_no=input ('Enter the size of initial population ::');
bits=input('Enter the bit size of each strings::') ;
final=[];
iterations=input('Enter the number of iterations to perform the process of genetic algorithm::');
tournament_no=input('Enter the number of times you want to perform tournament selection::');
%% power before optimization
p_3=0;
p_4=0;
I_3=0.57589244; %in ampere
I_4=0.9269762; %in ampere
R=0.578; %in ohm
u=59; %number of switches in feeder 3
v=6; %number of switches in feeder 4
n=u+v;
for i=1:u
   p_3=p_3+(I_3^2)*R;
end;
for i=1:v
   p_4=p_4+(I_4^2)*R;
end;
display('power before optimization')
```

```
p_3
p_4

%% generate the bits randomly
p=rand(p_no,bits);
for i=1:p_no
   for j=1:bits
      if (p(i,j)>=0.5)
         popu(i,j)=1;
      else
         popu(i,j)=0;
      end;
   end;
end;
for z=1:iterations
p_t=p_3+p_4;
val=zeros(p_no,1);
%%calculate the value of strings
for i=1:p_no
     for j=bits:-1:1
        k=bits-j;
        val(i,1)=val(i,1)+pow2(popu(i,j),k);
     end;
end;
val;
%% generate the random numbers to do the tournament selection
for i=1:tournament_no
   select(i,:)=[floor(1+rand(1)*(p_no-1))  ceil(1+rand(1)*(p_no-1))];%% have to make changes here
end;
select;
```

```
%%calculate the fitness value and perform tournament selection
tour_val_1=zeros(tournament_no,1);
tour_val_2=zeros(tournament_no,1);
for i=1:tournament_no
tour_val_11=zeros(select(i,1),1);
tour_val_22=zeros(select(i,2),1);
for j=2:val(select(i))+1
   tour_val_11(j)=tour_val_11(j-1)+(I_3^2)*R;
   tour_val_1(i)=tour_val_11(j);
end;
for j=2:val(select(i,2))+1
   tour_val_22(j)=tour_val_22(j-1)+(I_3^2)*R;
   tour_val_2(i)=tour_val_22(j);
end;
tor_min1=min(abs(tour_val_1(i)-p_t/2),abs(tour_val_2(i)-p_t/2));
if tor_min1==abs(tour_val_1(i)-p_t/2)
      fitest(i,:)=popu(select(i,1),:);
else
      fitest(i,:)=popu(select(i,2),:);
end;
end;
tour_val_1;
tour_val_2;
fitest;
prob_crossover=0.8;
prob_mutation=0.01;
crossover_point=2;
for x=1:ceil(p_no/2)
%%selecting of parents
parents=[];
crossover=[];
```

```
crossover_val=[];
crossover_val_mapped=[];
crossover_fitness=[];
parents=[fitest(floor(1+rand(1)*(tournament_no-1)),:);      fitest(ceil(1+rand(1)*(tournament_no-1)),:)];
%selecting of childrens
val_gen=rand(1);
child=[];
if (val_gen<prob_crossover)
  child=[[parents(1,1:crossover_point)
parents(2,(crossover_point+1):bits)];[parents(2,1:crossover_point)
parents(1,(crossover_point+1):bits)]];
end;
%%calculate the value of parents and children
crossover=[parents;child];
crossover_val=zeros(size(crossover,1),1);
for i=1:size(crossover,1)
    for j=bits:-1:1
        k=bits-j;
        crossover_val(i,1)=crossover_val(i,1)+pow2(crossover(i,j),k);
    end;
end;
crossover_val;
%%calculate the fitness value to select the the most suitable solution
for j=1:size(crossover_val,1)
p_31=zeros(size(crossover_val,1),1);
% p_4=zeros(size(crossover_val,1),1);
p_3=zeros(size(crossover_val,1),1);
for i=2:crossover_val(j,1)+1
   p_31(i,1)=p_31(i-1,1)+(I_3^2)*R;
   p_3(j,1)=p_31(i,1);
```

```
end;

crossover_fitness=p_3;
end
select_min1=min(abs(crossover_fitness-p_t/2));
for i=1:size(crossover_val,1)
      if (select_min1==abs(crossover_fitness(i,1)-p_t/2))
         pos1=i;
         break;
      end;
end;
%%the position of the max fitness value after crossover
pos1;
% to select the 2nd bst fitness value.
select_min2=abs(crossover_fitness(1,1)-p_t/2);
pos2=1;
if select_min2==select_min1
   select_min2=abs(crossover_fitness(2,1)-p_t/2);
   pos2=2;
end;
for i=1:size(crossover_fitness,1)
      if(select_min1==abs(crossover_fitness(i,1)-p_t/2))
         continue;
      elseif (abs(crossover_fitness(i,1)-p_t/2)~=select_min2)
         select_min2=abs(crossover_fitness(i,1)-p_t/2);
         pos2=i;
      end;
end;
% the values of 2nd best fitness values and its position
select_min2;
pos2;
```

```
% crossover1=dec2bin(crossover_fitness,7);
% taking the best 2 fitest solution and creating a different mating pool
final=[final;crossover(pos1,:);crossover(pos2,:)];
end;%% end of crossover
%performing mutation over the solutions generated after crossover.
for i=1:size(final,1)
   for j=1:size(final,2)
   val_gen=rand(1);
      if (val_gen<prob_mutation)
        final1(i,j)=xor(final(i,j),1);
      else
        final1(i,j)=final(i,j);
      end;
   end;
end;
final=[];
popu=final1;
p_no=size(popu,1);
val=zeros(p_no,1);
for i=1:p_no
     for j=bits:-1:1
        k=bits-j;
        val(i,1)=val(i,1)+pow2(popu(i,j),k);
     end;
end;
u2=val(1,1);
v2=n-val(1,1);
p_3=0;
p_4=0;
for i=1:u2
   p_3=p_3+(I_3^2)*R;
```

```
end;
p_4=p_t-p_3;
end;%%end to the number of iterations for the entire process;
display('The final best solution is given as....')
u2
v2
display('power after optimization')
p_3
p_4

%%%%%% GENETIC A LGORITHM FOR F5 AND F6 POWER OPTIMIZATION %%%%%
clc
clear all
close all
display('Program to optimize the power ');
p_no=input('Enter the size of initial population::');
bits=input('Enter the bit size of each strings::') ;
final=[];
iterations=input('Enter the number of iterations to perform the process of genetic algorithm::');
tournament_no=input('Enter the number of times you want to perform tournament selection ::');
%% power before optimization
p_5=0;
p_6=0;
I_5=0.84222477; %in ampere
I_6=0.59454; %in ampere
R=0.578; %in ohm
u=8; %number of switches in feeder 1
v=58; %number of switches in feeder 2
n=u+v;
for i=1:u
   p_5=p_5+(I_5^2)*R;
```

```
end;
for i=1:v
   p_6=p_6+(I_6^2)*R;
end;
display('power before optimization')
p_5
p_6
%% generate the bits randomly
p=rand(p_no,bits);
for i=1:p_no
   for j=1:bits
      if (p(i,j)>=0.5)
         popu(i,j)=1;
      else
         popu(i,j)=0;
      end;
   end;
end;
for z=1:iterations
p_t=p_5+p_6;
val=zeros(p_no,1);
%%calculate the value of strings
for i=1:p_no
     for j=bits:-1:1
        k=bits-j;
        val(i,1)=val(i,1)+pow2(popu(i,j),k);
     end;
end;
val;
%% generate the random numbers to do the tournament selection
for i=1:tournament_no
```

```
   select(i,:)=[floor(1+rand(1)*(p_no-1)) ceil(1+rand(1)*(p_no-1))];%% have to make changes
here
end;
select;
%%calculate the fitness value and perform tournament selection
tour_val_1=zeros(tournament_no,2);
tour_val_2=zeros(tournament_no,2);
for i=1:tournament_no
tour_val_11=zeros(select(i,1),1);
tour_val_22=zeros(select(i,2),1);
for j=2:val(select(i))+1
   tour_val_11(j)=tour_val_11(j-1)+(I_6^2)*R;
   tour_val_1(i)=tour_val_11(j);
end;
for j=2:val(select(i,2))+1
   tour_val_22(j)=tour_val_22(j-1)+(I_6^2)*R;
   tour_val_2(i)=tour_val_22(j);
end;
tor_min1=min(abs(tour_val_1(i)-p_t/2),abs(tour_val_2(i)-p_t/2));
if tor_min1==abs(tour_val_1(i)-p_t/2)
      fitest(i,:)=popu(select(i,1),:);
else
      fitest(i,:)=popu(select(i,2),:);
end;
end;
tour_val_1;
tour_val_2;
fitest;
prob_crossover=0.8;
prob_mutation=0.01;
crossover_point=2;
```

```
for x=1:ceil(p_no/2)
%%selecting of parents
parents=[];
crossover=[];
crossover_val=[];
crossover_val_mapped=[];
crossover_fitness=[];
parents=[fitest(floor(1+rand(1)*(tournament_no-1)),:); fitest(ceil(1+rand(1)*(tournament_no-
1)),:)];
%selecting of childrens
val_gen=rand(1);
child=[];
if (val_gen<prob_crossover)
  child=[[parents(1,1:crossover_point)
parents(2,(crossover_point+1):bits)];[parents(2,1:crossover_point)
parents(1,(crossover_point+1):bits)]];
end;
%%calculate the value of parents and children
crossover=[parents;child];
crossover_val=zeros(size(crossover,1),1);
for i=1:size(crossover,1)
    for j=bits:-1:1
        k=bits-j;
        crossover_val(i,1)=crossover_val(i,1)+pow2(crossover(i,j),k);
    end;
end;
crossover_val;
%%calculate the fitness value to select the the most suitable solution
for j=1:size(crossover_val,1)
p_61=zeros(size(crossover_val,1),1);
p_6=zeros(size(crossover_val,1),1);
```

```
for i=2:crossover_val(j,1)+1
   p_61(i,1)=p_61(i-1,1)+(I_6^2)*R;
   p_6(j,1)=p_61(i,1);
end;
crossover_fitness=p_6;
end
select_min1=min(abs(crossover_fitness-p_t/2));
for i=1:size(crossover_val,1)
      if (select_min1==abs(crossover_fitness(i,1)-p_t/2))
         pos1=i;
         break;
      end;
end;
%%the position of the max fitness value after crossover
pos1;
% to select the 2nd bst fitness value.
select_min2=abs(crossover_fitness(1,1)-p_t/2);
pos2=1;
if select_min2==select_min1
   select_min2=abs(crossover_fitness(2,1)-p_t/2);
   pos2=2;
end;
for i=1:size(crossover_fitness,1)
      if(select_min1==abs(crossover_fitness(i,1)-p_t/2))
         continue;
      elseif (abs(crossover_fitness(i,1)-p_t/2)~=select_min2)
         select_min2=abs(crossover_fitness(i,1)-p_t/2);
         pos2=i;
      end;
end;
% the values of 2nd best fitness values and its position
```

```
select_min2;
pos2;
% crossover1=dec2bin(crossover_fitness,7);
% taking the best 2 fittest solution and creating a different mating pool
final=[final;crossover(pos1,:);crossover(pos2,:)];
end;%% end of crossover
%performing mutation over the solutions generated after crossover.
for i=1:size(final,1)
   for j=1:size(final,2)
   val_gen=rand(1);
      if (val_gen<prob_mutation)
        final1(i,j)=xor(final(i,j),1);
      else
        final1(i,j)=final(i,j);
      end;
   end;
end;
final=[];
popu=final1;
p_no=size(popu,1);
val=zeros(p_no,1);
for i=1:p_no
     for j=bits:-1:1
        k=bits-j;
        val(i,1)=val(i,1)+pow2(popu(i,j),k);
     end;
end;
u2=val(1,1);
v2=n-val(1,1);
p_5=0;
p_6=0;
```

```
for i=1:u2
   p_6=p_6+(I_6^2)*R;
end;
p_5=p_t-p_6;
end;%%end to the number of iterations for the entire process;
display('The final best solution is given as....')
v2
u2
display('power after optimization')
p_5
p_6
```