



An-Najah National University
Faculty of Graduate Studies

**OPTIMUM SIZING AND ALLOCATION OF PV SYSTEMS
FOR IMPROVING THE PERFORMANCE OF
ELECTRICAL NETWORK – CASE STUDY NEDCO**

By
Omar Najam El dein Salah Breik

Supervisor
Dr. Imad Ibrik

**This Thesis is Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Electrical Power Engineering, Faculty of Graduate Studies, An-Najah National
University, Nablus, Palestine.**

2024

OPTIMUM SIZING AND ALLOCATION OF PV SYSTEMS FOR IMPROVING THE PERFORMANCE OF ELECTRICAL NETWORK – CASE STUDY NEDCO

By

Omar Najam El dein Salah Breik


This Thesis was Defended Successfully on 17/08/2024 and it was approved.

Dr. Imad Ibrik
Supervisor



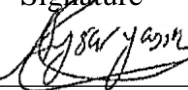
Signature

Dr. Samer Alsady
External Examiner



Signature

Dr. Aysar Yasin
Internal Examiner



Signature

Dedication

إلى والديّ اللى من أكرمني الله بأن جعلني ابنا لهم، إلى اللذان أهدقا وما زالا يغدقان علي الحب والعطف والرعاية،

إلى من حبانى الله بهم وجعلهم سندا وعونا لي إختي وأختي.

To my parents whom God honored by making me their son, and to my brothers and sister.

Acknowledgments

بسم الله الرحمن الرحيم والحمد لله رب العالمين والصلاة والسلام على سيد الخلق محمد رسول الله، أتقدم بجزيل الشكر والعرفان إلى الأستاذ الدكتور عماد بريك وإلى جميع أساتذتي الأفاضل وإلى المهندس احمد عصفور .

I would like to thank my advisor Dr. Imad Ibrik for his support and extend my thanks to all my instructors and engineer Ahmad Asfour.

Declaration

I, the undersigned, declare that I have submitted the thesis entitled:

OPTIMUM SIZING AND ALLOCATION OF PV SYSTEMS FOR IMPROVING THE PERFORMANCE OF ELECTRICAL NETWORK – CASE STUDY NEDCO

I further declare that the work presented in this thesis, unless otherwise referenced, is the researcher's own work and has not been submitted elsewhere for any other degree or qualification.

Student's Name: Omar Najam El dein Salah Breik

Signature: *Omar Breik*

Date: 17/08/2024

**OPTIMUM SIZING AND ALLOCATION OF PV SYSTEMS FOR IMPROVING
THE PERFORMANCE OF ELECTRICAL NETWORK – CASE STUDY
NEDCO**

**By
Omar Najam El dein Salah Breik
Supervisor
Dr. Imad Ibrik**

Abstract

Distribution networks may be affected if distribution generation is integrated into an existing system without a comprehensive network analysis since electrical systems are not built to support power generation sources at the distribution level. There may be an impact on voltage levels, current flow and protection coordination. Hence, the importance of studying the optimal size and location of solar cells in electrical networks.

In this research, the Chaotic Bees Algorithm is used to determine the optimal size and location of solar photovoltaic (PV) systems for integration into electrical distribution networks. Considering that solar energy integration has become an essential energy source, PV deployment needs to be supported by strong standards and research to optimize its benefits.

The traditional Bee algorithm in most studies solves the problem of optimal PV placement and sizing by focusing on minimizing losses while improving voltage stability only on small systems and networks such as the IEEE test system of 33 or 69 busses. We modified the algorithm so that the modified form of the algorithm using chaotic mapping increases the search space and returns better results faster and handles larger networks.

Three scenarios were compared in order to determine how well the distribution network performed: without solar systems, with distributed solar systems on remote and weak buses, and with optimally placed solar systems through both the original and improved Bee algorithm. The improved algorithm proved its efficacy, as it was able to improve the network performance significantly and thus provide optimal solutions.

the algorithm was applied to a more significant part of the Nablus network, which included 247 connection points and 213 transformers, based on information from

operational engineer we considering a load factor 0.4 and a power factor 0.95 of the transformers because we don't have measured data in all the 213 transformers.

This study was done with the help of the Python programming language and Jupyter Notebook for the implementation of the algorithm and analysis of the network. The results confirm that the developed algorithm finds the optimal locations and sizes of PV systems to improve the efficiency and stability of the electrical network with higher quality and speed compared to the traditional algorithm.

Keyword: optimization, PV systems, electrical network

List of Contents

Dedication.....	III
Acknowledgments.....	IV
Declaration.....	V
Abstract.....	VI
List of Contents.....	VIII
List of Tables	X
List of Figures	XI
List of Appendices	XI
Chapter One: Overview of The Energy Sector and Renewable Energy in Palestine and Literature review for Optimum sizing and allocation of PV system	1
1.1 Overview of the Palestinian Energy Sector	1
1.2 Palestinian Energy Institution Framework	2
1.3 Electricity Current Situation.....	4
1.3.1 West Bank Situation	4
1.3.2 Gaza Strip situation.....	5
1.4 Renewable Energy in Palestine	5
1.5 Electrical Energy Problems in Palestine.....	8
1.6 Theoretical Background of the Optimum Sizing and Allocation of PV System	9
1.7 Optimum Allocation and Sizing of PV Systems Techniques	11
1.7.1 Analytical approach	13
1.7.2 Classical (non-heuristic) approach.....	14
1.7.3 Meta-heuristic Approach.....	17
1.7.4 Hybrid approach	21
1.7.5 Assorted Approach	23
1.8 Conclusion	26
Chapter two: Mathematical modeling of Optimum sizing and allocation of PV system using chaotic artificial bee colony algorithm.....	27
2.1 Mathematical Modeling & problem formulation.....	27
2.1.1 Sizing and Allocation Problem Formulation.....	27
2.1.2 Sizing and Location of the PV Formula constraints	31
2.2 Power Flow Analysis	32
2.2.1 Newton-Raphson method.....	33
2.2.2 Gauss- Seidel method	35

2.2.3 Fast decoupled	39
2.2.4 Forward / Backward Sweep Method	41
2.3 Artificial Bee Colony Algorithm (Meta-Heuristic Approach)	44
2.4 Improved Artificial Bee Colony Algorithm (Assorted Axxprouch).....	47
Chapter Three: Results and Discussion	51
3.1 Load Flow Analysis of IEEE 33 Bus test System Without PV	52
3.2 Allocate and Size PV Random at the weak buses on IEEE 33 Bus Test System	53
3.3 Best PV Allocation and Sizing Using the Conventional ABC Algorithm IEEE 33 Bus Test System	53
3.4 Finding for the Best PV allocation and Sizing Using the improved ABC Method on 33 bus test system	55
3.5 Comparison of Adapted ABC Algorithms in Literature and the Improved ABC Algorithm.....	56
3.6 implementation of the improved ABC Method on Part of the Nablus network	57
Chapter Four: Conclusion and Future Work.....	59
4.1 Conclusion	59
4.2 Future work.....	59
List of Abbreviations.....	61
References.....	63
Appendices.....	71
الملخص	ب

List of Tables

Table 1.1: Total energy supply by year and type of energy	1
Table 1.2: Coupling points in the West Bank.....	4
Table 1.3: west bank Demand and Availability and Trend.....	5
Table 1.4: Gaza Strip Demand and Availability and Trend.....	5
Table 1.5: Installed Capacity of Photovoltaic in Palestine.....	8
Table 1.6: Distribution generation types based on the power	10
Table 2.1: Load Types and Exponent Values	28
Table 3.1: IEEE 33 Simulation Results	53
Table 3.2: Conventional ABC on IEEE 33 Results Summary	55
Table 3.3: Implementing Improved ABC on IEEE 33 Results Summary	56

List of Figures

Figure 1.1: Palestinian Electricity Sources	2
Figure 1.2: Palestinian Institution Energy Framework	2
Figure 1.3: The Solar Potential of Palestine	6
Figure 1.4: Wind Speed Map in Palestine and Jordan	7
Figure 1.5: Optimum allocation and Sizing of PV systems Techniques	12
Figure 1.6: Analytical Optimization Methods	13
Figure 1.7: Classical (non-heuristic) Optimization Methods	15
Figure 1.8: Meta-heuristic Optimization Methods.....	17
Figure 1.9: Hybrid Optimization Methods	22
Figure 1.10: assorted methods.....	24

List of Appendices

Appendix A: Comparison of the Optimization Techniques	71
Appendix B: IEEE 33 Network.....	75
Appendix C: IEEE 33 network parameters before installing PV	76
Appendix D: IEEE 33 network parameters after installing PV randomly.....	77
Appendix E: IEEE 33 Bus System Parameter Curves Before Installing PV System	80
Appendix F: IEEE 33 Bus System Parameter Curves After Installing PV System random at weak buses.....	82
Appendix G: IEEE 33 Conventional ABC python code	85
Appendix H: IEEE 33 Network Parameters After Installing PV Using Conventional ABC	96
Appendix I: IEEE 33 Results Summary of distribution of PV Using Conventional ABC	98
Appendix J: IEEE 33 Improved ABC Python Code.....	100
Appendix K: IEEE 33 Results Summary of random distribution of PV Using Improved ABC	118
Appendix L: IEEE 33 Network Parameters After Installing PV Using improved ABC Algorithm	121
Appendix M: Part of the Nablus network.....	124
Appendix N: Substations single line diagrams	130
Appendix O: Part of Nablus transformers names	132
Appendix P: Part of Nablus Network Parameters Before Installing PV	138
Appendix Q: Nablus Network Results Summary of random distribution of PV Using Improved ABC	143
Appendix R: Part of Nablus Network Parameters After Installing PV Using improved ABC Algorithm	146
Appendix S: Cables and over headlines specifications.....	159
Appendix T: Tables.....	162
Table 3.4: Literature and Improved ABC Results Comparison of the IEEE 33 Buss System.....	162
Table 3.5: Improved ABC on Nablus Network Results Summary	163
Appendix U: Figures	164

Figure 2.1: Typical Buses of Power System.....	164
Figure 2.2: Flowchart of ABC.....	164
Figure 2.3: Flowchart of Improved ABC	165
Figure 3.1: IEEE 33 Buss System Diagram.....	166
Figure 3.2: Flowchart of the ABC Algorithm.....	166
Figure 3.3: Fitness and Convergence Analysis for Different Configurations	167
Figure 3.4: Flowchart of the Improved ABC.....	168
Figure 3.5: Fitness and Convergence Analysis for Different Configurations - IEEE 33 Chaotic Bees Algorithm	169
Figure 3.6: The Convergence Characteristic of the Optimum Iteration Using Improved ABC on the Nablus network.....	169

Chapter One

Overview of The Energy Sector and Renewable Energy in Palestine and Literature review for Optimum sizing and allocation of PV system

1.1 Overview of the Palestinian Energy Sector

The following are the primary characteristics of the Palestinian energy sector (in Gaza and the West Bank):

- The overall amount of energy used is quite tiny when compared to regional consumption and even lower when compared to global usage.
- Electrical energy represents (28.7 %) of total energy consumed in the year 2020 [1] as shown in Table 1.1.

Table 1.1

Total energy supply by year and type of energy

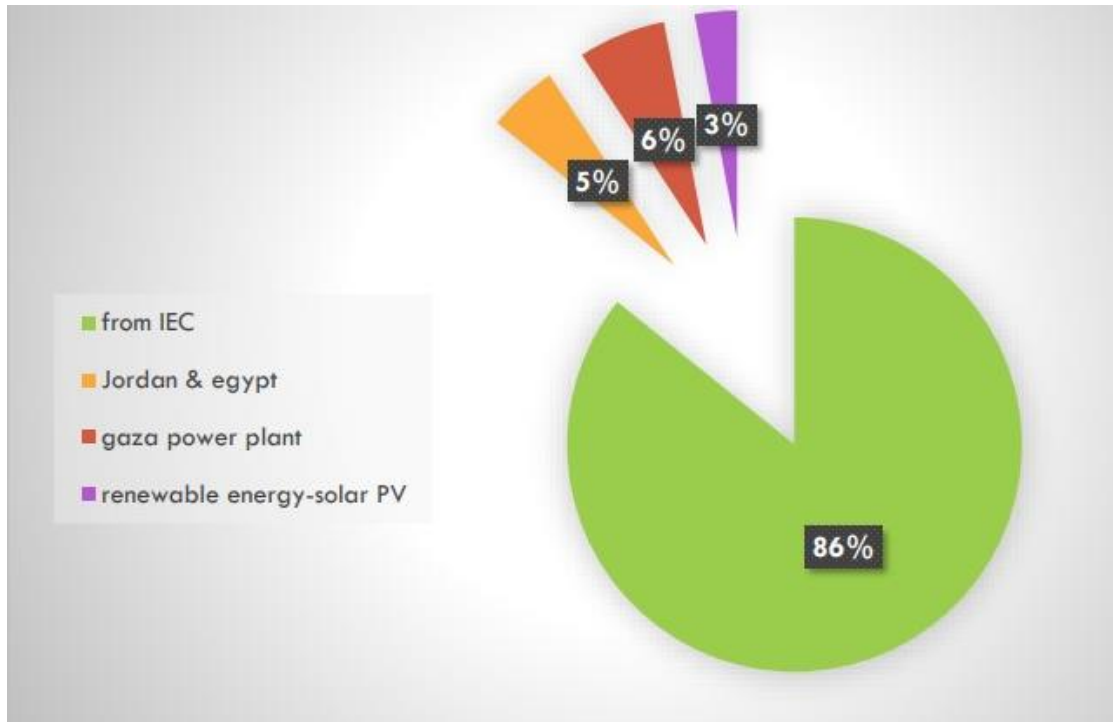
Year	Renewable energy (TJ)	Oil products (TJ)	Electricity (GWh)
2009	8438	29865	3983
2010	10503	29703	4159
2011	11285	33088	4622
2012	10734	31787	4909
2013	10350	35733	4736
2014	11827	41443	4938
2015	11807	43562	5422
2016	12098	47569	5516
2017	9660	47883	5622
2018	9725	43941	5976
2019	11187	48219	6369
2020	10904	48920	6698

Due to minor manufacturing activity, the residential sector has the highest energy requirement.

- The Israeli Electric Cooperation (IEC) provides 86% of the electrical energy that we used. [2]

Figure 1.1

Palestinian Electricity Sources

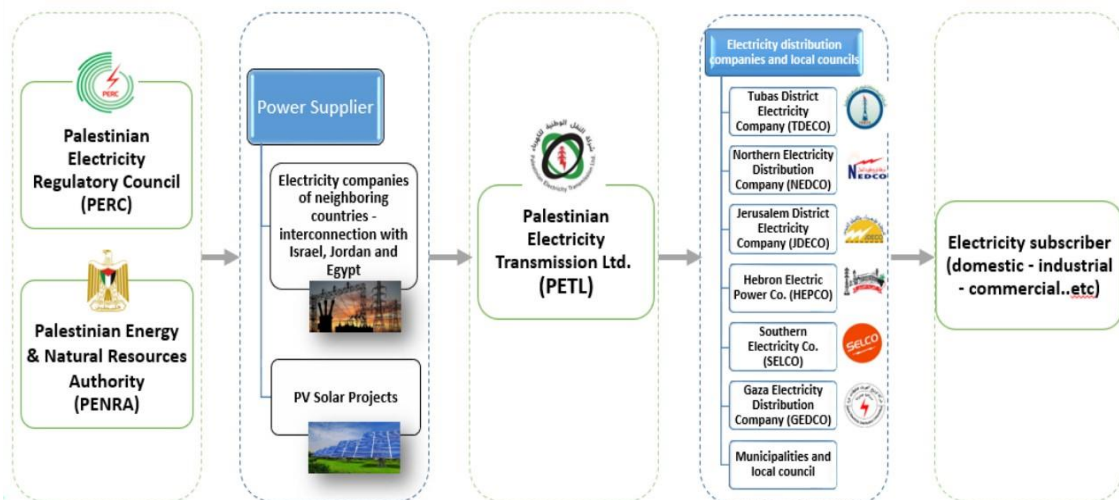


1.2 Palestinian Energy Institution Framework

In this section, the electricity sector in Palestine will be illustrated along with its sectors. The following figure demonstrates the Palestinian institution of energy framework.

Figure 1.2

Palestinian Institution Energy Framework



The following sections pertain to Palestinian institutions' energy sectors:

1. Regulatory sector: consists of two institutes:
 - a. Palestinian Electricity Regulator Council (PERC): it is responsible for:
 - Making pricing and tariff methodology recommendations.
 - Make recommendations for licenses, rules, and regulations.
 - Establishing service benchmarks.
 - Checking for license adherence.
 - Performance evaluation.
 - Customer satisfied with service.
 - Resolve conflicts.
 - Respond to inquiries and provide clear data.
 - b. Palestinian Energy & Natural Resources Authority (PENRA): it is responsible for:
 - Electrical projects in rural areas.
 - Information and preparation.
 - Energy and electricity.
 - Gas and petroleum derivatives.
 - Electric transport.
2. Generation sector: it is a private sector that allows to everyone invest in it, Investments are made in renewable energy without other energy sources because of the complex political situation.
3. Transmission sector: this sector is regulated by the Palestinian Electricity Transmission Line Company (PETL) which controls the low-voltage power substations:
 - 33KV/0.4KV
 - 22KV/0.4KV
 - 33KV/6.6KV
4. Distribution sector: distribution is done through five companies:
 - Jerusalem district electricity company (JDECO) serving the central area around Jerusalem.
 - Hebron Electric Power Company (HEPCO) serves the southern area around Hebron.
 - Southern Electric Company (SELCO) serves the rest of the southern area on the west bank.

- Northern Electricity Distribution Company (NEDCO) serves the northern area on the west bank.
- Gaza electricity distribution company (GEDCO) serving governorates of the Gaza Strip.

5. Consumers (end-user)

1.3 Electricity Current Situation

The power which is generated on the Israeli side is transmitted via 161 KV main transmission lines these lines then get transformed into 33 KV or 22 KV transmission lines.

1.3.1 West Bank Situation

- The west bank is mainly supplied by four 161KV/33KV substations from IEC:
 - The first one is the south close to Hebron.
 - The second is in the north in the Ariel settlement close to Nablus.
 - The third is in Atarot industrial area near Jerusalem.
 - The Fourth one in the north of Jenin.
- The electrical system in the west bank consists of many isolated distribution systems that are not integrated together into a unified distribution network.
- Coupling points control the amount of electricity that follows the

The Palestinian side, they are controlled by the Israeli side any increase in the ampacity is done based on political coordination, and the West Bank contains about 200 coupling points distributed as shown in Table 1.2.

Table 1.2

Coupling points in the West Bank

Distribution of connection points among regions	Number of connection points
Gaza	10
South west Bank	60
North west Bank	159
Central west Bank (outside JDECO)	6
JDECO	23
Total	258

- The Palestinian Authority has agreed with Jordan to connect the Palestinian power grid to that of Jericho through a 33KV line, JDECO submitted a new request to upgrade the line to 132KV.

- West Bank peak demand, availability, and trend (2030) [2].

Table 1.3

west bank Demand and Availability and Trend

Peak demand (MW)	Availability (MW)	Demand in 2030 (MW)
1200	1035	1686

1.3.2 Gaza Strip situation

- There is a power station that generates power in some regions of Gaza, high-cost electricity generated from the Gaza power plant leads to an inability to exploit the full power of the station.
- There is a 161KV connection with Israel as well as a 260KV connection with Egypt.
- Currently most of the electricity is being delivered by the 161 KV transmission line following the damage to the Gaza power plant damage in the 2014 war.
- Gaza Strip peak demand, availability, and trend (2030) [2].

Table 1.4

Gaza Strip Demand and Availability and Trend

Peak demand (MW)	Availability (MW)	Demand in 2030 (MW)
470	216	649

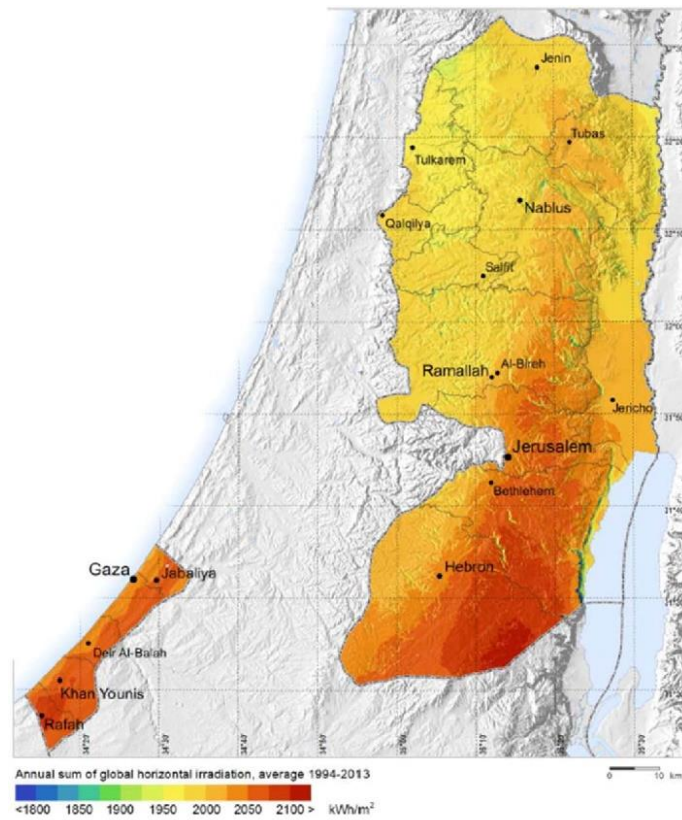
1.4 Renewable Energy in Palestine

The term "renewable energy" refers to energy sources including sun, wind, geothermal, hydropower, and bioenergy that regenerate themselves spontaneously over time comparable to that of humans. These energy sources are appealing because they frequently produce less pollution and have less of an adverse effect on the environment than fossil fuels, which are limited and emit dangerous greenhouse gases when burned. Additionally, switching to renewable energy can lessen reliance on fossil fuels, which have a limited supply and emit dangerous greenhouse gases when consumed.

Due to Palestine's favorable geographic location, where there are 300 sunny days per year and annual global horizontal irradiation (GHI) of more than 2,000 kWh/m². The renewable energy is employed in Palestine are mainly solar and photovoltaic energy, biomass energy, and currently the wind resources are being studied as shown in Figure 1.3 [3].

Figure 1.3

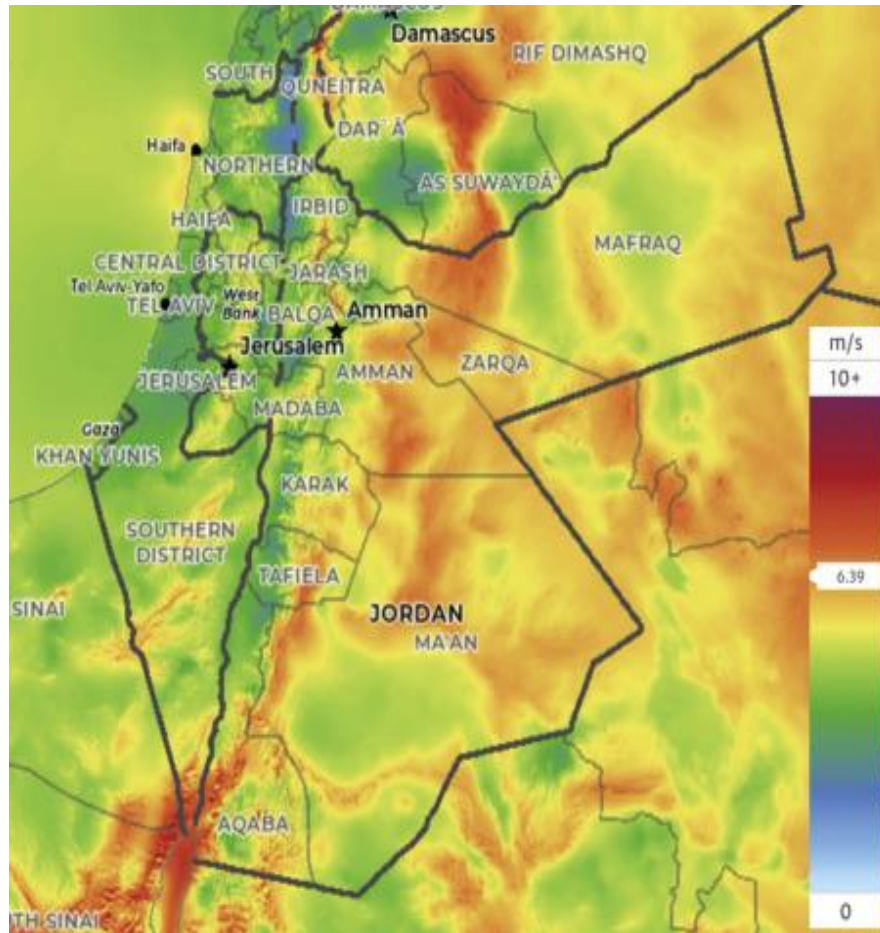
The Solar Potential of Palestine



The scenario for wind potential is not as favorable as it is for solar radiation because investing in it is unfeasible, but it is still considered an energy source that can integrate with PV systems to make up for the energy deficit. Figure 1.4 [4] shows the wind speed in Palestine.

Figure 1.4

Wind Speed Map in Palestine and Jordan



The Palestinian energy authority seeks to exploit the available renewable energies and invest in them to meet the growing energy demand and reduce dependence on the Israeli side. As a result, the Palestinian energy authority made electrical energy and renewable energy roadmap for 2030 [5] which states:

- Renewable target roadmap 2021- 2030 additional 500MW, 80% of the 2030 target will be achieved with PV systems and CSP, 4% with wind energy, and 16% with biogas – biomass.
- The electrical energy target for a roadmap 2020-2030 is 5,000 GWh (or 500 GWh per year). The implementation of the proposed plan should save 3.5 million tons of CO₂ each year and is expected to achieve around 10,000 GWh savings by 2030.

The Energy Authority has begun to implement the strategy somewhat effectively, as it published a report showing the installed capacity of photovoltaic cells in Palestine in 2021 according to the Palestinian Central Bureau of Statistics in Table 1.5.[6]:

Table 1.5*Installed Capacity of Photovoltaic in Palestine*

Governorate	Capacity (Kw)
Palestine	178495
West Bank	159512
Jenin	22853
Tubas & Northen valley	11000
Tulkarm	22788
Nablus	18373
Qalqilya	13825
Salfit	10406
Ramallah & Al-Bireh	10393
Ariha & Al- Aghwar	18311
Jerusalem	1582
Bethlehem	7867
Hebron	22114
Gaza Strip	18983

About 30 megawatts were added in the West Bank between 2021 and 2024, however the Energy Authority has not yet recorded these additions.

1.5 Electrical Energy Problems in Palestine

From the previous sections, we can see that the Palestinian energy sector faces several difficulties. Represented by adopting the Palestinian side primarily on imports to supply energy, as about 86% of the electrical energy is provided by IEC and 95% of the oil derivatives produced for energy are imported from Israel [7], as well as the high prices of this energy and the resulting high financial cost to the Palestinian consumer and producer in general. Power interruption continues to be a barrier due to technical difficulties as happened in Tukaram and the Gaza Strip because of the overload and shortage of supply [8].

Numerous factors adversely impacted the electricity network despite the numerous rehabilitation programs that were implemented in various parts of the West Bank, with the most significant issue being a lack of funding and a low collection rate for electricity bills, which resulted in a lack of network maintenance. Due to Palestinian municipalities and utilities' lack of control over connection points, feeders were extended at low voltage rather than medium voltage. This resulted in an increase in technical and non-technical losses, which are estimated to account for 20% to 30% of the total energy supply [9], as well as outages, a decline in supply quality, and overloading of feeders. To reduce the technical losses in the Palestinian power network, there has to be an investment in this

network by switching the low-voltage feeders to medium voltage feeders that are connected to a new substation.

In the realm of renewable energy, there are many obstacles related to the government management of the energy sector in general, specifically when it comes to solar investment, and the persistence of these obstacles until now that government agencies do not give seriousness to investing in the renewable energy sector, contrary to statements and objectives in this regard. Rather, it seems that the government does not see renewable energy as a solution to the electricity crisis in Palestine, which has led to other crises such as the crisis of debt accumulation, insisting that the solution lies in obtaining electricity from traditional sources, despite the rush of around the world to adopt renewable energy as the primary source of energy supply, we witness the focus of the efforts of some parties to secure the continuous growth in demand for electricity by importing more of the occupation, which is a wrong impression, so decisive and radical solution are required to change these obstacles.

1.6 Theoretical Background of the Optimum Sizing and Allocation of PV System

Most countries' current systems for electrical distribution rely heavily on centralized generating facilities. The load centers are located far from where these plants are installed. In some instances, during peak hours, power output from central plants may not be enough to fulfill the full demand or it may occasionally be seen as a costly investment when the load factor is weak, even though central plants are capable of meeting. As a result, several countries occasionally adopt distributed generating sources.

Generally, distribution generation is defined as "A generating plant connected directly to the grid at distribution level voltage or on the customer side of the met" [10]. Fuel cells, combustion turbines, thermal solar power, photovoltaic systems, microturbines, wind turbines, engines, and generator sets, as well as storage technologies, are examples of distributed generation systems. Distributed generation sources are divided into categories based on the power that they produce as shown in Table 1.6 [11].

Table 1.6*Distribution generation types based on the power*

Distribution generation type	Power factor	Technology
Produce reactive power (+Q)	Zero	Capacitor bank
Produce active(+P) and reactive(+Q) power	(0.8 – 0.99) Leading	Fuel cell, synchronous machine, solar PV system, microturbine
Consume active(-P) and produce reactive(+Q) power	(0.8 – 0.99) Lagging	Synchronous condenser, doubly fed induction generator (DFIG)

By distributed generation, consumers have access to a consistent and reasonably priced supply of power. Generally, integrating distributed generation systems into an existing system is crucial due to its many advantages. Increasing system dependability, voltage stability, and overall efficiency while minimizing greenhouse gas emissions also reduces power losses is the major objective of integrating distributed generation. Many Studies show that about 13% of the power losses of total power demand in the power system are on the distribution side [12].

Since Electrical systems are not designed to accommodate power production sources at the distribution level, therefore integrating distribution generation into an existing system without proper network study may have negative effects on distribution networks. They may be divided into five different issues:

1. Capacity: The addition of distributed generators at the distribution level can greatly affect the amount of power that equipment (cables, lines, transformers) must handle [13]. A reverse flow of power may occur, and the capacity of the transformer and cables must be able to handle this reverse flow.
2. Losses: The goal of constructing distributed generation is to lower network losses. Since the energy is produced close to the load, however occasionally energy losses rise because of the network's erratic current flow.
3. Power quality: Voltage and frequency are the two key factors that determine power quality, and distribution generation may either improve or worsen it. Installing and increasing the degree of distributed generation penetration on the system will undoubtedly affect the frequency and can make voltage flicker. the term "flicker" describes sudden changes in voltage that can disrupt electrical operations and produce significant fluctuations in illumination. When clouds pass across solar cells, for instance, flicker can happen, rapidly altering the power output and having an impact

on voltage stability [14]. Another significant problem with power quality distributed generation can be a source of harmonics.

4. Protection: Power can only flow in one direction, but when dispersed generation is connected in parallel to the current grid, the possibility of bidirectional power flow increases, making it more difficult to spot a defect and complicating coordination amongst preventive devices. Distribution networks are built using a "fit and forget" methodology in which protective equipment settings are left unchanged. DG units might increase current at a fault [14].
5. Voltage Regulation: Forecasted daily and seasonal fluctuations in loads are used to guide the design of the distribution system voltage regulation. The direction of the flow of power is changed using distribution generation power injections. The voltage level at the load centers may rise over the permitted levels in the event of minimal demand and maximum generation at the distribution generation system. This might cause the wrong operation of voltage regulation components in the distribution system, such as step voltage regulators, and load tap changers [15].

From the previous problems, we have a new challenge, which is where the optimum distributed generation sources should be located and what size they should be. Scientists have developed many methods to solve this problem.

1.7 Optimum Allocation and Sizing of PV Systems Techniques

The location and scale of renewable energy sources are essential tools in the effective management and planning of electric networks. Besides wind, solar energy is the second most popular RES, and during the past 20 years, it has significantly improved the security of the energy supply. The Optimal Allocation and Sizing of PV systems challenge is further complicated by the need to satisfy numerous operational and planning requirements unequal constraints as well as energy balancing equal constants.

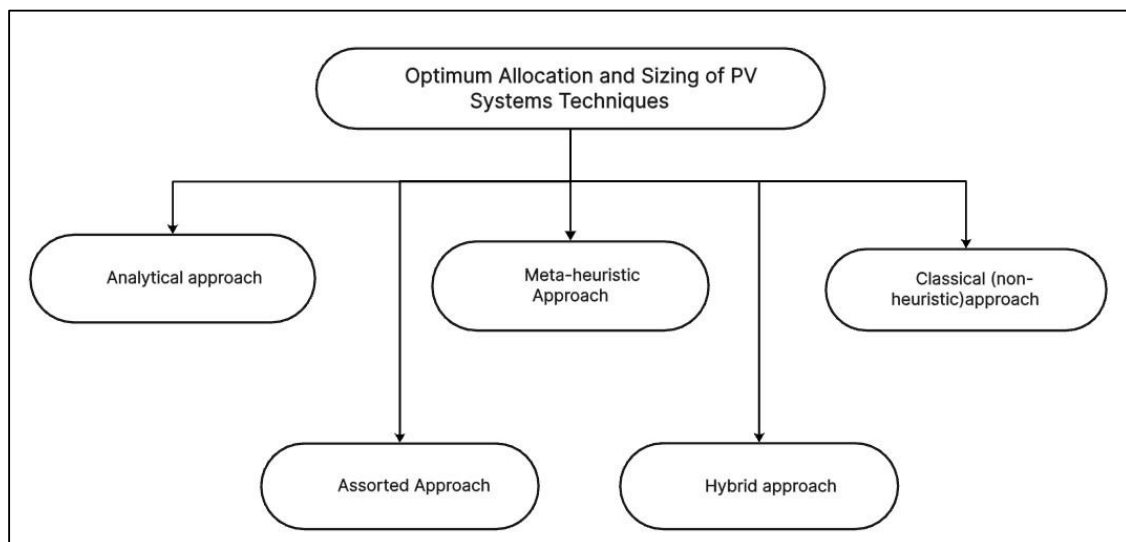
We will discuss in this chapter, the objective function, test systems, advantages, and limitations of a wide range of existing research varying techniques for optimum PV allocation and size.

The optimum allocation and the sizing of the PV systems can be achieved through various techniques that depend on exploration and exploitation. Generally, Analytical techniques

are effective for small, simple systems but fall short for systems with large, complex networks. Several meta-heuristic algorithms are performing better in terms of accuracy and convergence for very large and complex networks. Combining two or more optimization methods results in hybrid optimization. Assorted and future approaches have efficiency, accuracy, and flexibility in the optimization process and are being further improved by a variety of current and upcoming methods for PV allocation and size. This method provides accurate, efficient global optimum solutions for difficult multi-objective problems. A thorough literature examination of the principles of distributed generation is the foundation for looking at the various technological approaches for integrating it into the distribution system. The objective function, test system, advantages, and disadvantages of analytical, classical (non-heuristic), meta-heuristic, hybrid, and assorted optimization procedures have also been taken into consideration. These methods make use of recent technological developments, data accessibility, and processing power. The sizing and allocation of optimum PV in a distribution system will be significantly contributed by all the technical methods outlined above. The various methods used for optimum PV allocation and sizing are divided into five main methods shown in fig1.5 [16], [17], [18], [19].

Figure 1.5

Optimum allocation and Sizing of PV systems Techniques

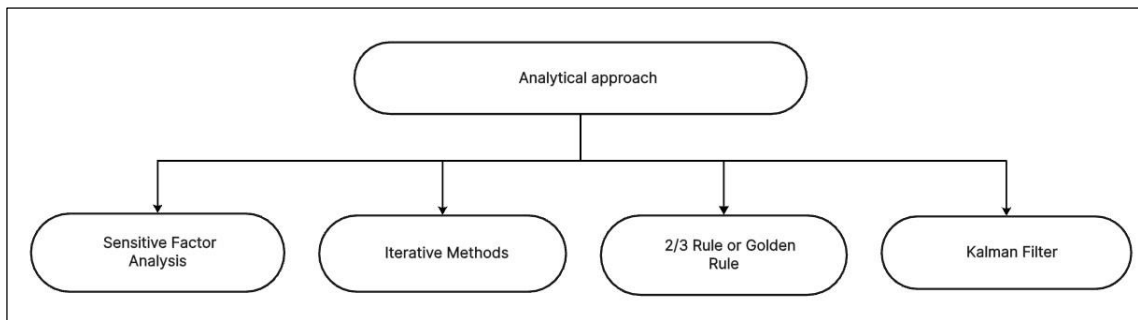


1.7.1 Analytical approach

The analytical optimization strategy relies on mathematical methods rather than stochastic or iterative approaches to discover the best solution to a problem. An explicit mathematical formula for the best PV system size and allocation based on objective functions and constraints is what the analytical optimization technique seeks to achieve in the context of PV allocation and sizing. The analytical optimization technique provides a straightforward route to establishing the best PV allocation and size options, making it advantageous when it can be used [20]. To get optimal outcomes, however, it may be essential to use additional optimization techniques in more complicated situations. Some of the analytical methods are reviewed in Figure 1.6.

Figure 1.6

Analytical Optimization Methods



1. 2/3 Rule or Golden Rule

A method for locating an extremum (minimum or maximum) of a function within a predetermined range is the golden-section search. It will converge to one of the extrema for an interval containing many extrema, which may include the interval borders, as opposed to finding the extremum of a strictly unimodal function with an extremum inside the interval. It will converge to that boundary point if the interval's single extremum is located there. The approach is very slow but extremely reliable since it works by gradually decreasing the range of values on the set interval [21].

2. Sensitive Factor Analysis

Sensitivity analysis evaluates, under a certain set of assumptions, how alternative values of an independent variable impact a specific dependent variable [22]. Numerous approaches are utilized in optimization, such as loss sensitivity factor, voltage sensitivity

index, exhaustive search with sensitivity factor, and existing sensitivity method, and this strategy is employed within specified bounds that rely on one or more input variables.

3. Kalman Filter

When the measuring equipment is inaccurate, a statistical procedure called the Kalman filter can be used to estimate values accurately in dynamic systems. There are only two phases in this basic algorithm: prediction and update. We anticipate an estimate of the starting value, and then when the actual value is received, we compare it to our estimate and generate a new estimate for the subsequent step. The number of samples determines the accuracy; the bigger the number of samples, the greater the accuracy [23].

4. Iterative Methods

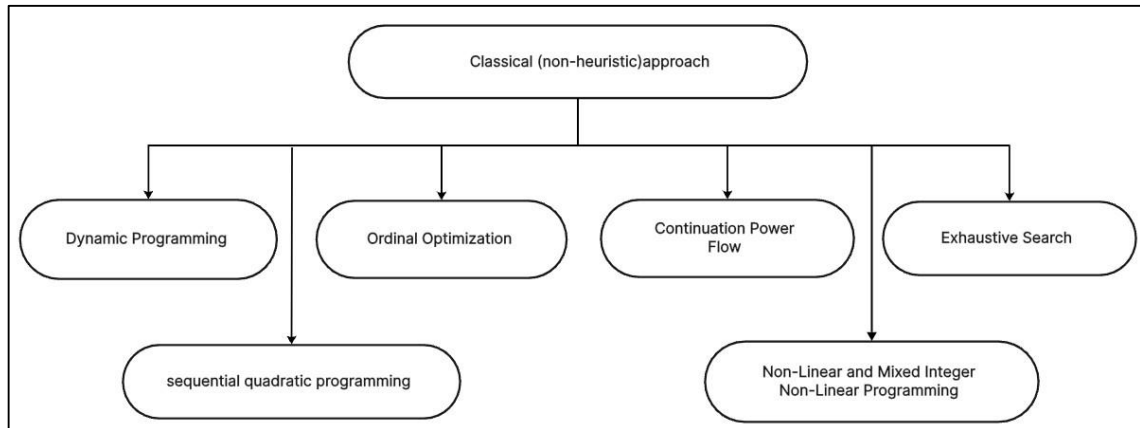
Iterative methods provide a tentative solution to the linear system after a limited number of iterations. These methods work well for complicated equation systems where speed is more important than accuracy. Iterative methods use the coefficient matrix only indirectly, via an abstract linear operator or a matrix-vector product. Although iterative techniques may be used with any matrix, they are frequently utilized with big sparse matrices since direct solutions are slow for these types of matrices. In contrast to a direct technique, an indirect method does not rely as much on the coefficient matrix's fill pattern to solve a linear problem quickly. However, employing an iterative approach often necessitates fine-tuning settings for each unique issue [17].

1.7.2 Classical (non-heuristic) approach

With traditional procedures, a solution would either be discovered, or its nonexistence would be demonstrated. Such approaches' primary drawbacks are their computational complexity and inability to handle uncertainty. Their utilization becomes fragile due to these drawbacks. This is a result of the unexpected and uncertain nature of such applications in their very nature. For more accurate near-optimal solution generation, many classical optimization techniques outperform analytical techniques. Some of the methods reviewed are as follows in Figure 1.7:

Figure 1.7

Classical (non-heuristic) Optimization Methods



1. Dynamic Programming (DP)

By continually splitting a complex problem into smaller, more manageable ones, dynamic programming takes a complex problem and simplifies it. Recursive analysis of decisions that span many periods is common, although certain choice problems cannot be divided in this fashion [24]. It utilizes DP to improve dependability and voltage profile and reduce the distribution system's power loss.

2. Ordinal Optimization (OO)

Finding the best option from a discrete, ordered list of options is the goal of an optimization problem known as ordinal optimization. For reducing the search space and the processing load, it offers a probabilistic framework. Depending on the type of issue and the constraints involved, several ordinal optimization approaches may be used. It outlines trade-offs between maximizing DG capacity and minimizing losses [25].

3. Non-Linear and Mixed Integer Non-Linear Programming (NLP and MINLP)

Because integer variables are discrete in addition to other NLP variables, MINLP problem solving is more difficult than NLP issue solving. The global optimum for issues is frequently found using global optimization methods, because non-linear functions can introduce several local optimal points, making it tough to determine the global optimum, NLP issues are more complex to solve than linear programming problems. In both radial and meshed networks, NLP is employed for minimal DG unit allocation with increased voltage stability [18]. Multiple DG unit allocation is favored for lowering total power loss

and generating cost. MINLP transforms discrete probabilistic generating load models into deterministic load models to solve time-varying load models [26].

4. Exhaustive Search (ES)

Finding the best answer to an optimization problem may be done simply and thoroughly via comprehensive search optimization. The option that best maximizes the goal function is chosen after thoroughly examining every viable option. Exhaustive search optimization is frequently applied to small-scale issues or as a standard to verify the outcomes. It is an appropriate solution for the creation and demand processes that fluctuate over time [18].

5. sequential quadratic programming (SQP)

The iterative technique of sequential quadratic programming (SQP) is used for constrained nonlinear optimization. In mathematical problems where the objective function and constraints may be discriminated twice continuously, SQP techniques are applied. SQP is renowned for its effectiveness and capacity to solve complex, equality- and inequality-constrained nonlinear optimization problems. When the size of the DG is restricted so that it doesn't go above a specific percentage of the power supplied by the substation and the flow restriction in the feeders, to make sure they don't become too hot and to prevent the danger of running into reverse energy flows [27].

6. Continuation Power Flow (CPF)

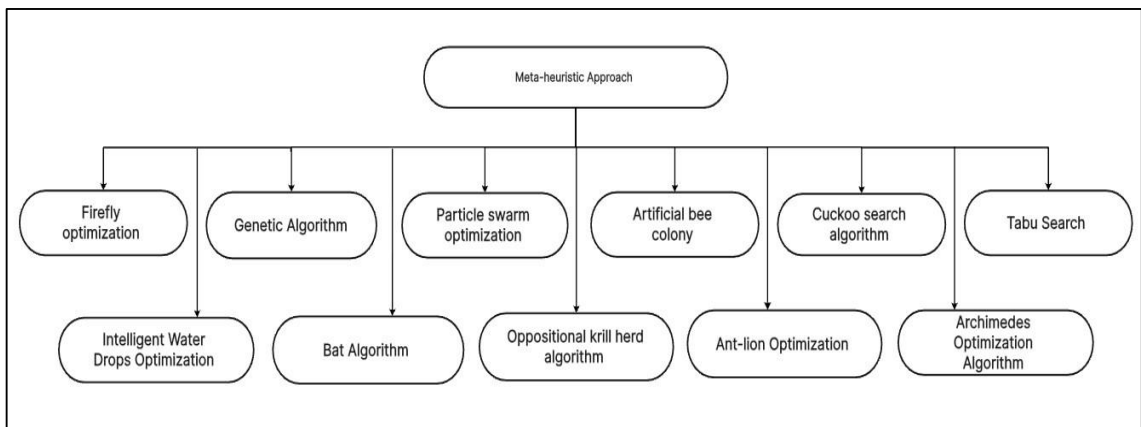
Finding the ideal or nearly ideal operating conditions for a power system that fulfills certain goals and limitations is the goal of continuation power flow (CPF) optimization, which blends continuation power flow with optimization techniques. Continuation power flow finds load flow solutions. It consists of prediction and correction steps. CPF optimization is an effective tool for dealing with complex power systems issues like optimal power flow, economical dispatch, unit commitment, and various other planning and operational studies. It is especially helpful when dealing with nonlinear and highly constrained power system problems [28].

1.7.3 Meta-heuristic Approach

Meta-heuristics is a problem-independent, high-level algorithmic framework that provides a set of recommendations or strategies for creating algorithms that improve heuristics. These algorithms are always heuristic and are generated according to the methods described in the meta-work framework. Metascience has proven to be a realistic and often better alternative to more traditional methods. Metaheuristics is often able to provide a better trade-off between solution quality and processing time, especially for complex problems or large problem cases. Meta-inference is also more adaptable than micro-techniques [29]. There are many meta-heuristics algorithms, some of which are inspired by physics and others by animal behavior. We will represent some of the algorithms as follows in Figure 1.8:

Figure 1.8

Meta-heuristic Optimization Methods



1. Genetic Algorithm (GA)

A genetic algorithm (GA), which is a subset of the wider class of evolutionary algorithms (EA), is a metaheuristic used in computer science and operations research that draws inspiration from the process of natural selection. Genetic algorithms frequently utilize biologically inspired operators like mutation, crossover, and selection to produce high-quality solutions to optimization and search issues are employed to provide optimum voltage stability and minimal loss reduction. However, GAs do not ensure that the world's best solution will be found [30].

2. Particle swarm optimization (PSO)

Is a technique inspired by the social behavior of birds flocking. In PSO, a population of particles (representing potential solutions) cooperatively moves through the search space to find the optimal or near-optimal solution to an optimization problem. The key idea behind PSO is that particles adjust their positions based on their own previous experiences and the experiences of their neighbors, to converge to the best solution in the search space. Each particle's position and velocity are updated iteratively using rules that are influenced by its own historical best position and the best position found by any particle in the swarm. It can be applied to a wide range of optimization problems across different domains to provide optimum minimal loss. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found [31].

3. Bat Algorithm (BA)

Bat technology (BA) is an optimization technology inspired using echolocation by bats. The bat algorithm simulates the hunting behavior of bats considering territorial exploitation and global exploration. The Bat algorithm has shown high performance in reducing network losses and is particularly useful in situations involving continuous optimization [32]. Compared to some other optimization techniques, it is less dependent on parameter modification.

4. Artificial bee colony (ABC)

is a new population-based random algorithm that shows effectiveness in searching through a variety of optimization problems. However, the original ABC shows a poor convergence velocity during the search. It is a newly developed optimization method that simulates the intelligent foraging activity of honeybees. There are three different types of bees in the ABC algorithm: worker bees, observer bees, and scout bees. Worker bees forage near a fresh food source in their minds while also informing the foraging bees of these food sources. From the food sources explored by the renter bees, the observer bees often choose the best one. The probability of observer bees choosing a high-quality food source over a low-quality one is much higher. A few worker bees that leave their food sources and look for other ones become scout bees. The number of solutions in the swarm is equal to the number of worker or observer bees. The method seeks to actively explore the search space and find optimal or near-optimal solutions to difficult optimization

problems through balancing exploration, which searches for new answers and is used to reduce loss, improve the voltage profile, increase reliability [33].

5. Cuckoo search algorithm (CSA)

Cuckoo Search imitates the behavior of cuckoos, who deposit their eggs in the nests of other birds, creating a form of competition and variety in the population. It seeks to identify optimum or nearly optimal solutions to optimization issues. It was motivated by some cuckoo species' compulsion to parasitize host birds of other species' species by depositing their eggs there. Some host birds and the intrusive cuckoo may directly clash. As an illustration, if the host bird learns that the eggs are not its own, it may either discard the unusual eggs or depart the nest and start a new one elsewhere. This breeding behavior has been optimized by Cuckoo's study, which makes it applicable to many optimization challenges, such as minimizing energy loss in the system [34].

6. Oppositional krill herd algorithm (OKHA)

The krill herd (KH) algorithm is a novel meta-heuristic algorithm for solving global optimization problems. The two major objectives of the herding of the krill individuals are to increase krill density and get nourishment. An individual krill goes toward the optimal outcome in this process when it seeks the maximum density and food. Each krill modifies its location by changing its movement owing to induction, foraging motions and random diffusion as it travels in the multidimensional space. By leveraging opposition-based learning to consider solutions from various directions in the search space, the Oppositional Krill Herd Algorithm seeks to improve the exploration and exploitation abilities of the original Krill Herd Algorithm. This could enable the algorithm to escape local optima and arrive at more accurate conclusions, this method is used to minimize network power losses and to improve voltage stability [35].

7. Ant-lion Optimization (ALO)

The ALO imitates the natural hunting strategy of ants. The ant lion larva excavates a pit-shaped hole in the sand, walks in a circle, and then uses its enormous jaws to empty the hole of sand. After constructing the trap, the larva hides beneath the cone's base and waits for the insects to stay imprisoned there. Insects may readily fall to the bottom of the trap because of the cone's sharp edge. The antelope starts to catch its victim as soon as it knows

it is in a trap. After that, it is buried and eaten. The ants discard the remnants from the hole after devouring the meal and set up the well for the subsequent hunt [36]. Ants and blacks are two of the groups in the ALO algorithm. Ants are search agents in the decision space, and ants hiding there can track them down and capture their position to obtain the shape. It has had positive results when used to lower power losses and enhance the voltage profile.

8. Firefly optimization (FO)

It is a naturally inspired optimization technique that addresses optimization issues by modeling the flashing behavior of fireflies. The Firefly Algorithm mimics the attraction behavior of fireflies to locate optimum or nearly optimal solutions. Fireflies gravitate toward solutions with higher fitness values, which correlate to brighter fireflies. The algorithm can travel the search space and converge toward better results with the aid of this movement and attraction mechanism. The Firefly Algorithm's performance might vary based on the nature of the issue and the parameter choices, much like with other optimization techniques [37].

9. Tabu Search (TS)

It is an optimization method made to discover roughly correct answers to combinatorial optimization issues. The notion of avoiding local optima by permitting "tabu" motions that momentarily prevent going back to previously visited states is the inspiration behind it. The fundamental idea underlying Tabu Search is to use a memory mechanism to prevent being stuck in local optima while exploring the solution space by performing a variety of motions. Here is a brief explanation of the Tabu Search algorithm. Many different optimization loss minimizations through ODG allocation issues are solved using Tabu Search. It is renowned for its capacity to efficiently explore complicated solution domains and locate excellent solutions even when there are restrictions and noisy goal functions. It is not guaranteed to discover the global optimum like other metaheuristic algorithms, but it can offer effective solutions for a variety of issues [38].

10. Archimedes Optimization Algorithm (AOA)

According to Archimedes' principles, the upward buoyant force acting on an object immersed, either wholly or partially, in a fluid is equal to the weight of the fluid displaced

by the object moving upwards at the center of mass of the displaced fluid. This concept of physics is mimicked by a population-based metaheuristic technique called AOA. AOA followed the same steps as in all other population-based algorithms, including the initialization step to generate a random population, population evaluation, parameter modification in the exploration and exploitation phases as well as the termination phase, and this algorithm was used for reducing emissions of pollution gases [39].

11. Intelligent Water Drops Optimization (IWDO)

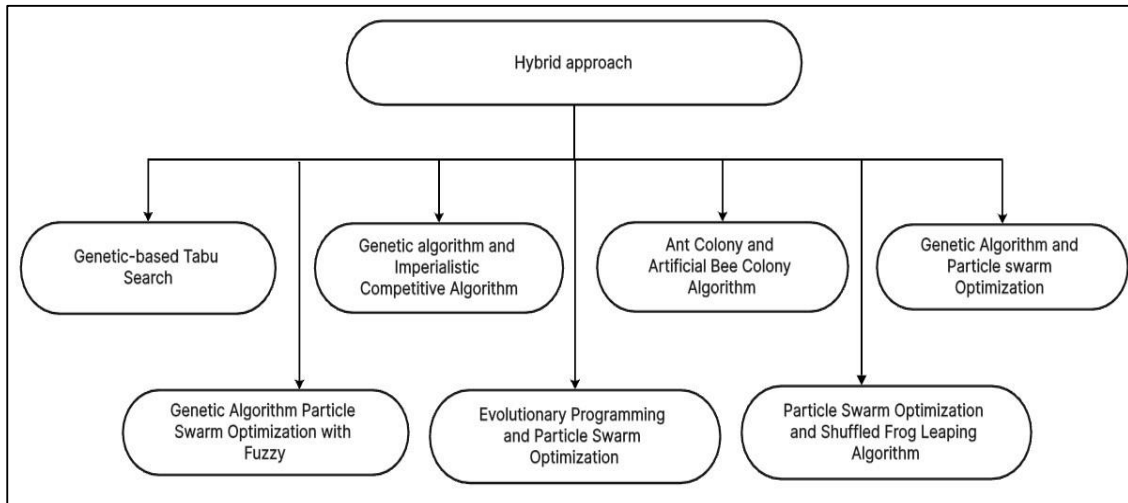
It is based on nature and is motivated by the way that water droplets naturally interact and move to lower places. Different search and optimization issues are solved with this approach. The method mimics the behavior of water droplets as they flow down slopes and converge at the lowest point, which, in the context of optimization issues, represents an ideal solution. Given that this approach is employed to reduce losses in power lines, it is particularly helpful for resolving problems involving continual improvement. It is advised to experiment with various parameters and issue states in the smart water drop optimization method that was utilized to find the settings that work best for your application [40].

1.7.4 Hybrid approach

In a mixed optimization strategy, two or more different optimization algorithms or approaches are combined to produce a single, more reliable algorithm that benefits from the advantages of each component separately. By integrating different metaheuristic techniques, or metaheuristic techniques with mathematical programming, machine learning, or artificial intelligence (AI), hybridization is accomplished. Hybridization tries to solve the shortcomings of individual algorithms and enhance their overall performance in tackling challenging optimization issues with better outcomes and more durable solutions while requiring less search time. We will represent some algorithms as follows in Figure 1.9:

Figure 1.9

Hybrid Optimization Methods



1. Genetic-based Tabu Search:

The two approaches have already been described, and when they are merged into a single algorithm, GA-based TS is used to allocate DG in the demand side of the power system in the most efficient way possible [41].

2. Genetic algorithm and Imperialistic Competitive Algorithm

The idea of imperialism and rivalry between imperialist nations served as the inspiration for the Imperial Competitive Algorithm (ICA). For actual power allocation and interaction, it performs well when combined with the fairy algorithm [42].

3. Particle Swarm Optimization and Shuffled Frog Leaping Algorithm

A population-based optimization approach is the Shuffled Frog Leaping approach (SFLA). The cooperative search metaphor, which was influenced by natural memetics, carries out optimization. It is utilized to optimize voltage profiles with the greatest possible advantages. Combining PSO and SFLA can be used to overcome the problem of transient stability [43].

4. Genetic Algorithm and Particle swarm Optimization

To handle the voltage, and increase issues brought on by DG integration, PSO and Gravitational Search Algorithm (GSA) are combined [44].

5. Genetic Algorithm Particle Swarm Optimization with Fuzzy

To simplify the multi-objective problem and minimize the number of iterations, a combination of GA and PSO with fuzzy is utilized. This allows us to quickly obtain the global optimum point [45].

6. Ant Colony and Artificial Bee Colony Algorithm

A combination of ACO and ABC algorithms is used for ODG allocation to reduce total power losses, emissions, and electricity costs [46].

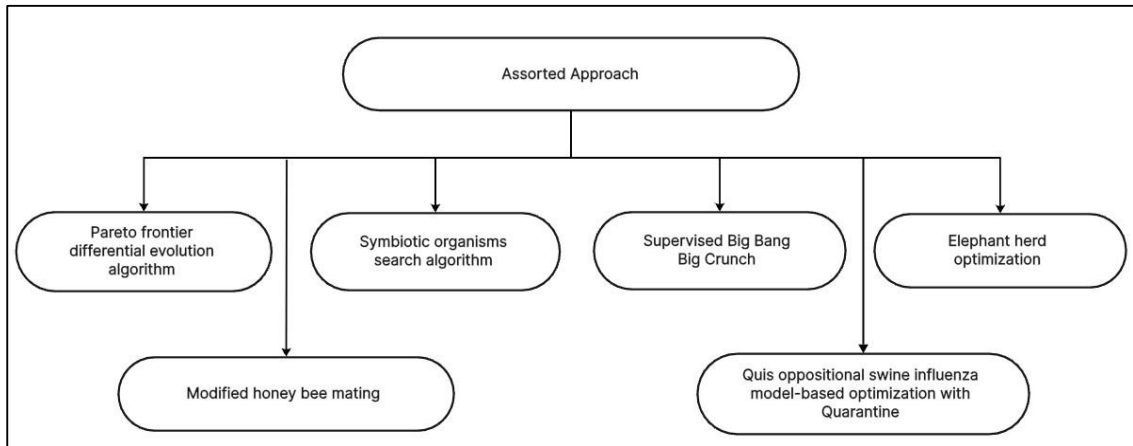
7. Evolutionary Programming and Particle Swarm Optimization

Hybrid Evolutionary Programming and Particle Swarm Optimization Minimize power for radial networks with less iterations [47].

1.7.5 Assorted Approach

Analytical and traditional (non-heuristic) methodologies perform well for small, simple systems but fall short for systems with complicated networks. However, several meta-heuristic strategies now do better. Due to its excellent accuracy and rapid convergence, it is appropriate for big and complicated systems. Combining two or more distinct optimization techniques is known as hybrid optimization. For complex multi-objective problems, it offers optimal global solutions that are more dependable and effective. Then, to make some different algorithms more successful, deal with the most complicated and vast networks, and produce results more quickly than with other technologies, some of their best qualities were combined with certain alterations by using machine learning and artificial intelligence. The assorted methods are the name of this approach, we will represent some of these algorithms in the following:

Figure 1.10
assorted methods



1. Pareto frontier differential evolution algorithm (PFDEA)

A different version of the Differential Evolution (DE) algorithm that specializes in resolving multi-objective optimization issues is the Pareto Frontier Differential Evolution (PFDE) method. Multi-objective optimization entails simultaneously maximizing a number of competing goals, when achieving one goal may result in the detriment of another. The purpose is to identify a collection of solutions that tradeoff between these goals, establishing the Pareto frontier or Pareto front. According to the PFDE method, solutions for mutation, crossover, and selection are often chosen in a way that favors variety preservation and convergence towards the Pareto frontier. To do this, it could employ specialized mutation and crossover operators [48].

2. Modified honey bee mating (MHBM)

to address optimization problems, the software simulates the foraging, mating, and communication processes of honeybees. The main concept is to build a group of artificial "bees" that iteratively search the space for possible solutions. The HBMO algorithm must be modified to handle multi-objective problems. Selection of the 'best solutions' (queens) should be done after the production of the initial population and corresponding evaluation of the objective functions, but not solely based on a comparison of the values of a single objective function, like the idea of Pareto dominance, is required in a multi-objective approach to dealing with multiple solutions and categorizing them as dominant or non-dominant. The non-dominant solutions are the "best solutions" (queens) selected from the indigenous population. The iterative process begins in the same way as in the individual

objective case after the identification of non-dominant solutions (queens) (mating flights, generation of new queens, improvement of existing queens and new generation, selection of new queens). Thus, we get the optimal global solution [49].

3. Supervised Big Bang Big Crunch (SBB-BC)

The Supervised Big Bang-Big Crunch algorithm is a potent and effective technique with high accuracy quick convergence and simple implementation. This approach enhances the Big Bang-Big Crunch algorithm's capacity for improved exploration through the use of particle swarm optimization. The hybrid Big Bang-Big Crunch algorithm also employs a mutation operator following position updates in order. this algorithm is used to solve the multi-objective allocation of Distributed Generators in a Distribution Network. The reduction of bus voltage variation and overall actual power losses are the goals that are taken into account in balanced and unbalanced distribution systems [50].

4. Elephant herd optimization (EHO)

The elephant herding behavior serves as the basis for the Elephant Herding Optimization approach. In the wild, elephants from various clans coexist under the direction of a matriarch, and when male elephants mature, they separate from their family units. These two actions can be represented by the two operator's clan updating and clan separating. Through the clan updating operator in EHO, the elephants in each clan are updated with information about their matriarch and present location. The use of the separation operator, which can increase population diversity at a later stage of the search process, comes next. By adopting EHO, distributed generation is intelligently and precisely allocated in electric distribution networks, reducing equipment overload, voltage swell and sag, active power, reactive power, and the cost of producing energy [51].

5. Quasi oppositional swine influenza model-based optimization with Quarantine (QOSIMBO-Q)

Quasi-opposition-based Swine Influenza Model-Based Optimization with Quarantine (QOSIMBO-Q) performs the optimization through quarantine and treatment loops. In this optimization technique initially, all suspected individuals are identified according to the current health of an individual and then they undergo the swine flu test. After the swine flu test, the confirmed case is quarantined or isolated from the population. Then treatment

is given to all individuals depending upon their current health. The basic steps of the (SIMBO-Q) algorithm is: Evaluate health, Swine flu test, and Quarantine. this algorithm is used for solving optimal placement and sizing problem DG in the radial distribution system [52].

6. Symbiotic organisms search algorithm (SOSA)

The Symbiotic Organisms Search algorithm is a nature-inspired optimization technique that draws inspiration from the linkages and interactions between numerous species in ecosystems. This approach is used to address optimization and search-related problems, and it typically takes its cues from the mutualistic relationships seen in nature. In biological environments, symbiosis refers to the mutually beneficial cooperation of two different species. The SOS algorithm mimics these interactions during optimization. The search is run through three cycles that imitate the mutualism phase, commensalism phase, and parasitism phase of three symbiotic partnerships. Through the execution of these three phases, SOS seeks to relocate a population, referred to as an ecosystem of potential solutions, to promising regions of the search space throughout the pursuit of the ideal answer. The SOS algorithm attempts to strike a balance between exploring and exploiting the search region by duplicating these many symbiotic connections. Like many algorithms derived from nature, the effectiveness of SOS depends on how well its parameters are adjusted and how well it suits the specific circumstances at hand [53].

1.8 Conclusion

The transition to a future powered by sustainable and renewable energy depends heavily on the effective deployment of distributed generation. Maximizing power output, enhancing economic viability, reducing environmental effects, minimizing losses in the electrical grid, and optimizing the voltage profile all depend on the scale and distribution of distributed generation. As there are several methodologies in this subject, many of which we have already studied, and numerous solution approaches, each of which is appropriate for a particular situation, research in this area is still in its early stages.

Advanced techniques address complex networks while setting several goals for improvement, while also having limitations. As a result, the best answer may be found in the shortest amount of time and with the fewest iterations. A summary table in Appendix A, illustrates the objectives and the most important advantages and disadvantages of the techniques previously mentioned in this chapter (see Appendix A).

Chapter Two

Mathematical modeling of Optimum sizing and allocation of PV system using chaotic artificial bee colony algorithm

2.1 Mathematical Modeling & problem formulation

mathematical modeling is a strong and adaptable tool used in many domains to describe and evaluate real-world occurrences. It entails the development of mathematical models that replicate or characterize the behavior of intricate systems or procedures, giving rise to predictions, insights, and rational decision-making. Understanding complicated systems, forecasting future results, streamlining procedures, and testing hypotheses are the goals of mathematical modeling. It assists in bridging the knowledge gap between theory and practical applications. In conclusion, mathematical modeling is a useful tool for deciphering, evaluating, and forecasting complicated systems and processes in a variety of fields.

2.1.1 Sizing and Allocation Problem Formulation

It is a challenging optimization issue to determine the best size and distribution of PV systems, as it involves dealing with numerous linear and nonlinear inequality constraints. The target function must be mathematically defined with its parameters, and the restrictions relating to cables and overhead lines thermal limits, and voltage constraints must be defined as well.

As for the formulation of objective function parameters, in many optimization situations, creating an objective function is a critical first step. The task you wish to minimize or optimize is sometimes referred to as the *objective function*. The specific problem you are trying to solve determines the form and structure of the objective function. Formulating function parameters is one of the most important steps in constructing an objective function. in our case, the most important parameters are:

1. Load models and Impact Indices

The distribution system's load model illustrates the mathematical connection between a bus voltage and the electricity or current entering the load bus. Residential, commercial, and industrial load models are among the several that are considered in the distribution systems.

The following equations use voltage-dependent load models, which may be stated mathematically as follows:

$$Load = P + jQ \quad 2.1$$

$$P_i = P_{oi} * V_i^\alpha \quad 2.2$$

$$Q_i = Q_{oi} * V_i^\beta \quad 2.3$$

Where v_i is the voltage at bus i , P_{oi} and Q_{oi} are real and reactive operating points at the bus i . P_i and Q_i are real and reactive power exponents, α and β are the real and reactive power exponents. According to Table 2.1 [54], Commercial, residential, and industrial loads are all constant when α and β are equal. Voltage stability index, voltage profile, power loss index, and line flow limit index are among the technical issues taken into consideration in order to construct the objective function in this technique. The definitions of these relevant affect indexes are as follows:.

Table 2.1

Load Types and Exponent Values

Load type	α	β
Constant	0	0
Industrial load	0.18	6.00
Residential	0.92	4.04
Commercial	1.51	3.40

2. Real and reactive Power Loss Index (PLR) and (QLR)

In power system analysis, the Real and Reactive Power Loss Index (QLR and PLR) is a metric used to evaluate the effectiveness of power distribution and transmission networks. It measures the quantity of power lost during the transmission of electricity via the grid infrastructure, including both reactive and actual power losses. These two parameters serve as an efficient indicator of how much the installation of solar cells or distribution generators has reduced both active and reactive power losses in the electrical grid [55].

$$PLR = \frac{P \text{ losses with PV}}{P \text{ losses without pv}} \quad 2.4$$

$$QLR = \frac{Q \text{ losses with PV}}{Q \text{ losses without pv}} \quad 2.5$$

3. Voltage Profile Index (VPI)

In electrical power systems, a metric called the Voltage Profile Index (VPI) is used to evaluate the stability and quality of voltage levels throughout a network. It is essentially a metric that aids in assessing the degree to which the voltage is kept within allowable bounds at different grid locations.

The difference between required or nominal voltage levels and actual voltage levels is the basis for calculating VPI. It helps identify potential contact points of excessively high or low voltage, which could lead to problems such as equipment damage, inefficiency, or service interruptions. A high VPI value indicates significant deviations from nominal voltage levels and indicates a poor voltage profile. The voltage profile index is calculated by the following formula

$$VPI = \frac{nn}{i = 2} \left(\frac{|V_{nominal}| - |v_i|}{|V_{nominal}|} \right) \quad 2.6$$

Where v_i is the voltage at bus i , nn Number of buses, $V_{nominal}$ is ranged from 1.05 to 1.1 P. U according to IEC60038

4. voltage stability Index (VSI)

The Voltage Stability Index (VSI) is a metric used in power systems to evaluate the system's capacity to sustain constant voltages within reasonable bounds throughout a range of operational circumstances, particularly in high-stress or high-demand scenarios. It is useful in determining how close the system is to voltage collapse, which is the point at which a minor disruption could result in a large reduction in voltage. A high VSI value suggests a larger danger of voltage breakdown under fluctuating conditions since it shows that the system is getting closer to its stability limits. To improve voltage stability and avoid widespread power disruption, utilities, and grid operators employ VSI analysis to foresee possible problems with voltage instability, plan system upgrades or reinforcements, and make appropriate modifications in real-time, for example, modifying

generation or load shedding. There are many formulas used to find the voltage stability index, including complex ones that require many data, and simple ones, and for each of them there is accuracy in giving results, one of these formulas is the Simplified Fast Voltage Stability Index (SFVSI) which is shown in the equation below [56].

$$SFVSI = \frac{nn}{\max_{i=1}} \sqrt{\frac{V_{ri}}{V_{si}}} \left(1 - \frac{V_{ri}}{V_{si}}\right) / \quad 2.7$$

Where V_{ri} is the voltage of the receiving node and V_{si} is the voltage at the sending node at bus i , nn Number of buses.

5. Line Flow Limit Index (LFLI)

In power system analysis, the Line Flow Limit Index (LFLI) is a statistic that is used to evaluate the capacity of transmission lines and ascertain the maximum power flow that a transmission line can manage without going above specific operational boundaries, including heat or voltage limitations. To avoid overheating, which can cause equipment damage or failure, electrical lines must be operated within their temperature limitations. An elevated LFLI score implies that the line is functioning near its maximum capacity, implying a greater possibility of overheating or beyond its thermal limitations. LFLI is commonly computed using variables such as line ratings, and current power flow as follows .

$$LFLI = \frac{nn}{\max_{i=1}} \left(\frac{|I_{branch}|}{|CI_{branch}|} \right) \quad 2.8$$

After defining the key parameters that affect the sizing and the location of the PV, the multi-objective performance index (MOPI) is calculated.

MOPI indicates the extent of network improvement based on the network's trend towards minimum values, where coefficient values range from zero to one. The closer the coefficients are to zero, the greater the improvement. Where there are fewer losses, better stability, and improvement of voltage while ensuring an acceptable limit of power flow. In the following formula, each factor can be given a certain amount of importance according to the weight factor by which it will be multiplied as following:

$$MOPI = w1 * PLR + w2 * QLR + w3 * VPI + w4 * SFVSI + +w5 * LFLI \quad 2.9$$

$$\sum_{p=1}^5 wp = 1 \quad \wedge \quad wp \in [0,1] \quad 2.10$$

Where w_p is the weight factors that choose in the study based on index importance.

2.1.2 Sizing and Location of the PV Formula constraints

Constraints are the limitations, restrictions, or criteria imposed on parameters or variables in an optimization problem. These limitations define the possible solution space and direct the search for the best solution that fulfills the objective function and the given constraints, making them crucial components of optimization models.

These limitations let optimization algorithms focus their search for the best potential answer. Finding the values for variables that maximize the objective function while abiding by all the given constraints is a necessary step in solving optimization issues. These limitations are essential in guaranteeing that the solution obtained is realistically possible within the given issue context and optimal analytically. The most important constraint in this problem is illustrated as follows along with their equations [57].

1. Distributed Generators Real and Reactive Power Generation Limits

To ensure that stations are not too large or too small, their electrical capacity should be restricted.

$$P_{DG}^{min} \leq P_{DG} \leq P_{DG}^{max} \quad 2.11$$

$$Q_{DG}^{min} \leq Q_{DG} \leq Q_{DG}^{max} \quad 2.12$$

2. Voltage Profile Limits

Because voltages outside of this range will negatively impact the equipment, the voltage at each distribution system node must be kept within certain bounds.

$$V_i^{min} \leq V_i \leq V_i^{max} \quad 2.13$$

3. Line Thermal Limits

the feeder should not exceed the thermal capacity of the line

$$S_{(i,j)} \leq S_{(i,j)}^{max} \quad 2.14$$

2.2 Power Flow Analysis

The electrical power network can be described as a circuit that consists of a source, load, and transmission line T.L. Although T.L. is characterized using characteristics like resistance, inductance, and capacitance, the source is typically represented as a rotating machine. Lastly, loads might be thought of as an apparent power consumer or an impedance. The network's constituent parts are arranged in clusters known as buses or nodes. They can be either swing or slack buses, load buses, producing buses, or voltage-controlled buses, depending on the function for which they are designed. Slack buses are used to balance the network's active power (P) and reactive power (Q). Meanwhile, generating buses provide the network with constant voltage and active power. Finally, load buses are used to represent the loads that are linked to the network. [58].

The electrical network's performance under steady-state circumstances is often examined using load flow analysis. In order to plan for future expansion and figure out how to run the system as effectively as possible, this research is required. The most crucial information that load flow analysis can offer is the actual and reactive power of each line, together with the voltage and phase angle of each bus. The following formulas are used to determine the load flow:

$$P_{ij} = \sum_{j=1}^{j \in N_i} |v_j| |v_i| (G_{ij} \cos(\delta_i - \delta_j) + B_{ij} \sin(\delta_i - \delta_j)) \quad 2.15$$

$$Q_{ij} = \sum_{j=1}^{j \in N_i} |v_j| |v_i| (G_{ij} \sin(\delta_i - \delta_j) - B_{ij} \cos(\delta_i - \delta_j)) \quad 2.16$$

Where:

- P_{ij} and Q_{ij} are the real and reactive power flow in the line.
- V_i and δ_i are the voltage and power angle at the sending end.
- V_j and δ_j are the voltage and power angle at the receiving end.
- G_{ij} and B_{ij} are the real and imaginary parts of the admittance matrix element Y_i .

Figure 2.1 -Appendix U- shows typical buses of power system.

Usually, two of the four variables are known:

- For the load bus we know real and reactive power (P and Q), and we find voltage and power angle (V and δ).

- For the generator bus we know real power and voltage (P and V), we find reactive power and power angle (Q and δ)
- For the slack bus we know voltage and power angle (V and δ), and we find real and reactive power (P and Q)

Real and reactive in load flow analysis, the voltage and phase angle of the slack bus are known. Meanwhile, the voltage and phase of the load bus are calculated, and its active and reactive capacities are known. Eventually, the voltage and active power of the generating bus are known, but the reactive power and phase angle still need to be checked [59].

Nowadays, a variety of software programs, like MATLAB, PS cad, ETAP, Power Factory, and POWER WORLD simulator, may be utilized to do a load flow analysis. This includes load flow and fault calculations, voltage stability, etc. Load flow analysis is usually done based on an iterative numerical solution; many methods are used in this thesis to find load flow solutions based on the case as follows:

2.2.1 Newton-Raphson method

The first estimations for all unknown variables (voltage magnitude and angles at load buses and voltage angles at generator buses) are the starting point of the iterative Newton-Raphson approach [58]. Next, for each of the power balance equations in the system of equations, a Taylor Series is created, ignoring the higher-order elements. A linear system of equations is the outcome, and it may be written as shown in the equation below:

$$\begin{bmatrix} \Delta\delta \\ \Delta|v| \end{bmatrix} = J^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad 2.17$$

Where ΔP and ΔQ are called the mismatch equations.

$$\Delta P_i = -P_i + \sum_{\substack{j \in Ni \\ j=1}}^j |v_j| |v_i| (G_{ij} \cos(\delta_i - \delta_j) + B_{ij} \sin(\delta_i - \delta_j)) \quad 2.18$$

$$\Delta Q_i = -Q_i + \sum_{\substack{j \in Ni \\ j=1}}^j |v_j| |v_i| (G_{ij} \sin(\delta_i - \delta_j) - B_{ij} \cos(\delta_i - \delta_j)) \quad 2.19$$

J is a matrix of partial derivatives known as a Jacobian:

$$J = \begin{bmatrix} \frac{\partial \Delta P}{\partial \delta} & \frac{\partial \Delta P}{\partial |V|} \\ \frac{\partial \Delta Q}{\partial \delta} & \frac{\partial \Delta Q}{\partial |V|} \end{bmatrix}$$

Based on the following equations, the linearized system of equations is solved to provide the next estimate ($m + 1$) of voltage magnitude and angles.

$$\begin{aligned} \delta_{m+1} &= \delta_m + \Delta\delta & 0.1 \\ |v|_{m+1} &= |v|_m + \Delta|v| & 0.2 \end{aligned}$$

The process keeps going until a halting condition is satisfied. Terminating if the norm of the mismatch equations is less than a certain tolerance is a typical stopping condition.

Newton Raphson's algorithm steps for the solution of the power-flow problem are as follows:

1. Initialization: For load buses, the voltage and angle need to be approximated, but the active and reactive power are known. Set the voltage magnitude to 1 and the angle to the known values for the slack bus.
2. Calculate power mismatch: Utilizing the provided values as well as the expected values of phase angles and voltage magnitudes, compute P and Q injection for load buses; P injection is calculated for generator buses. after that Power mismatch can be identified.
3. Using the partial derivative equations in terms of voltage magnitude and angle, the Jacobian matrix is created.
4. A matrix form of equations is solved using one of these two methods: Inverting the Jacobian matrix and multiplying it by the power mismatch. using Gaussian elimination for the Jacobian matrix to find the voltage magnitude matrix.
5. The voltage magnitude and angle have new estimated values.
6. As indicated by the accompanying equations, these procedures are continued until the power mismatch is smaller than the required precision:

$$\left| \Delta P_i^{[k]} \right| \leq \epsilon \quad 2.3$$

$$\left| \Delta Q_i^{[k]} \right| \leq \epsilon \quad 2.4$$

Where, ϵ is the accuracy is used to determine the final solution.

2.2.2 Gauss- Seidel method

for the purposes of using the Gauss-seidel method to solve the load flow solution All buses except the slack bus are regarded as PQ buses. This approach is also easily modified to accommodate PV buses. Furthermore, as is well known, all the other (n-1) bus voltages start at a value whose magnitudes and angles are assumed, except the slack bus voltage. An iterative procedure is then used to update those values [58].

$$V^{(1+k)} = \frac{\frac{P_i^{sch} - j Q_i^{sch}}{V_i^{*(k)}} + \sum y_{ij} V_j^{(k)}}{\sum y_{ij}} \quad j \neq i \quad 2.24$$

Where:

y_{ij} denoted by lowercase: the actual admittance per unit.

P_i^{sch} and Q_i^{sch} : the net real and reactive powers per unit.

For buses where real and reactive powers are injected into the bus, such as generator buses, P_i^{sch} and Q_i^{sch} have positive values. and for load buses where real and reactive powers are flowing away from the bus, P_i^{sch} and Q_i^{sch} have negative values.

$$P_i^{[k+1]} = R \left\{ V_i^{*[k]} \left[V_i^{[k]} \sum_{j=0}^n y_{ij} - \sum_{j=1}^n y_{ij} V_j^{[k]} \right] \right\} \quad j \neq i \quad 2.25$$

$$Q_i^{[k+1]} = -I \left\{ V_i^{*[k]} \left[V_i^{[k]} \sum_{j=0}^n y_{ij} - \sum_{j=1}^n y_{ij} V_j^{[k]} \right] \right\} \quad j \neq i \quad 2.26$$

The power flow equation is expressed in terms of the element bus admittance matrix (Y-bus). Since the off-diagonal elements of the bus admittance matrix Y_{bus} shown by uppercase letters, $Y_{ij} = -y_{ij}$ and the diagonal elements are $Y_{ii} = \sum y_{ij}$. The equations may then be written as follows:

$$V^{[1+k]} = \frac{\frac{P_i^{sch} - j Q_i^{sch}}{V_i^{*(k)}} + \sum_{j=1}^n Y_{ij} V_j^{[k]}}{Y_{ii}} \quad j \neq i \quad 2.27$$

$$P_i^{[k+1]} = R \left\{ V_i^{*[k]} \left[V_i^{[k]} Y_{ii} - \sum_{j=1}^n Y_{ij} V_j^{[k]} \right] \right\} \quad j \neq i \quad 2.28$$

$$Q_i^{[k+1]} = -I \left\{ V_i^{*[k]} \left[V_i^{[k]} Y_{ii} - \sum_{j=1}^n Y_{ij} V_j^{[k]} \right] \right\} \quad j \neq i \quad 2.29$$

y_{ii} consists of the ground's susceptibility to life charging as well as any other set ground admittance. It is necessary to solve two (n-1) equations repeatedly because both components (V and δ) are provided for the slack bus. Under typical operating circumstances, the $|V|$ of buses is near the $|V|$ of the slack buses or around 1.0 per unit. While the scheduled voltage at the generator buses is higher, the $|V|$ of the load buses is lower than the slack bus value, contingent on reactive power demand. While the δ of generator buses may be higher than the reference value based on the quantity of real power entering the bus, the δ of load buses is lower due to power demand. Thus, for the Gauss-Seidel method, an initial voltage estimates of $1.0+j0.0$ for unknown voltage is satisfactory and the converged solution correlates with actual operating states.

For load buses, P_i^{sch} and Q_i^{sch} are known. Starting with an initial estimate, $V_i^{[k+1]}$ is solved for the real and imaginary components of voltage. For regulated buses, P_i^{sch} and $|v_i|$ re-specified, $Q_i^{[k+1]}$ is solved, then used in $V_i^{[k+1]}$. Since it is specified, only imaginary part of $V_i^{[k+1]}$ is retained. And its real part is selected to satisfy [48].

$$e_i^{[k+1]} = \sqrt{|V_i^{[sch]}|^2 - (f_i^{[k+1]})^2} \quad 2.30$$

where $e_i^{[k+1]}$ and $f_i^{[k+1]}$ are the real and imaginary components of the voltage $V_i^{[k+1]}$ in the iterative sequence.

The rate of convergence is increased by applying acceleration factor (α) to the approximate solution obtained from each iteration as shown in the equation below:

$$V_i^{(k+1)} = V_i^{(k)} + \alpha \left(V_i^{(k)} - V_i^{(k)} \right) \quad 2.31$$

α is the factor of acceleration, and it falls between 1.3 and 1.7. Until the real and imaginary components of bus voltages vary between iterations within a predetermined range of precision, the operation is repeated. as indicated by the following equations:

$$|e_i^{(k+1)} - e_i^{(k)}| \leq \epsilon \quad 2.32$$

$$|f_i^{(k+1)} - f_i^{(k)}| \leq \epsilon \quad 2.33$$

Where, ϵ is the accuracy is used to determine the final solution.

Gauss- seidel algorithm steps for solution of the power-flow problem is:

1. Assume a flat voltage profile $1+j0b$ for all buses except the slack bus.
2. Assume a suitable value of ϵ called convergence certain.
3. Set iteration count, $k=0$ and assumed voltage profile of the buses are denoted as follows:

$$V_p^0, V_2^0, \dots, V_n^0$$

4. Set the bus count $i = 1$
5. Check for slack bus, if it is a slack bus go to step 12 otherwise continue
6. Check for generator bus, if it is generator bus go to next step, else go to step 9.
7. Temporarily set spec $|v_i^K| = |v_i|_{sch}$ where $|v_i^*|_{sch}$ is specified reactive power of generator bus. As shown in the following equation:

$$Q_{i,cal}^{[k+1]} = -\Im \{ V_i^{*[k]} [V_i^{[k]} \sum_{j=0}^n y_{ij} - \sum_{j=1}^n y_{ij} V_j^{[k]}] \} \quad j \neq i \quad 2.34$$

It is possible for the computed reactive power to fall within or above the given limitations. Consider the bus to be a generator bus if the computed reactive power falls within the designated range and set, $Q_i = Q_{i,cal}^{K+1}$ for this iteration and go to step 8.

If the calculated reactive power violates the specified limit, then treat the bus as load bus, as in the following equations:

$$Q_{i,cal}^{k+1} \leq Q_{i \min} \text{ then } Q_i = Q_{i \min} \quad 2.35$$

$$Q_{i,cal}^{k+1} \geq Q_{i \max} \text{ then } Q_i = Q_{i \max} \quad 2.36$$

Since the bus is treated as load bus U_i^k need not be replaced by spec $|V_i|_{sch}$ go to step 9

8. For generator bus the phase voltage of the bus can be calculated as shown in the below equations:

$$V_{i,temp}^{[1+k]} = \frac{\frac{P_i^{sch} - j Q_i^{sch}}{V_i^{*(k)}} - \sum_{j=1}^{i-1} Y_{ij} V_j^{[k+1]} - \sum_{j=i+1}^n Y_{ij} V_j^{[k]}}{Y_{ii}} \quad 2.37$$

$$\delta_i^{[k+1]} = \tan^{-1} \left[\frac{\Im V_{i,temp}^{[1+k]}}{\Re V_{i,temp}^{[1+k]}} \right] \quad 2.38$$

Now the (k+1) th iteration voltage of the generator bus is given by:

$$v_i^{k+1} = |v_i|_{sch} < \delta_i^{k+1} \quad 2.39$$

9. For load bus, v_i^{k+1} is calculated as follow:

$$V_i^{[1+k]} = \frac{\frac{P_i^{sch} - j Q_i^{sch}}{V_i^{*(k)}} - \sum_{j=1}^{i-1} Y_{ij} V_j^{[k+1]} - \sum_{j=i+1}^n Y_{ij} V_j^{[k]}}{Y_{ii}} \quad 2.40$$

10. An acceleration factor, α , can be used for faster convergence.

$$V_{i,acc}^{(k+1)} = V_i^{(k)} + \alpha (V_{i,cal}^{(k)} - V_i^{(k)}) \quad 2.41$$

$$\text{Then, } V_i^{(k+1)} = V_{i,acc}^{(k+1)} \quad 2.42$$

11. Calculate the change in bus-P voltage as shown in the following equation:

$$\Delta V_i^{k+1} = V_i^{k+1} - V_i^{(k)} \quad 2.43$$

12. Repeat steps 5 to 11 until all the bus voltages are calculated. Increment the bus count by 1 until the bus count is n .

13. Find the largest among $\Delta V_1^{k+1}, \Delta V_2^{k+1}, \dots, \dots, \Delta V_n^{k+1}$. If $|\Delta v_{max}|$ is less than ϵ , then move to the next step, else increment the iteration count and go to step 4.

14. Calculate the line flow and slack bus power using the bus voltages.

2.2.3 Fast decoupled

One Jacobian matrix for the voltage angle and one for the voltage magnitude are used in the quick decoupled load flow approach. As a result, convergence can occur more quickly since each set of equations can be solved separately. The voltage angle equations do not affect the linearization of the voltage magnitude equations. It is simpler to solve a set of nonlinear equations numerically when they are linearized. This guarantees that answers converge precisely and swiftly.

ΔP are less sensitive to changes in the voltage magnitude and are most sensitive to changes in power angle $\Delta\delta$, ΔQ are less sensitive to changes in the power angle and are most sensitive to changes in voltage magnitude $\Delta|v|$, we express this in the following equations:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \delta} & \frac{\partial P}{\partial |V|} \\ \frac{\partial Q}{\partial \delta} & \frac{\partial Q}{\partial |V|} \end{bmatrix} = \begin{bmatrix} \Delta\delta \\ \Delta|v| \end{bmatrix} \quad 2.44$$

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} = \begin{bmatrix} \Delta\delta \\ \Delta|v| \end{bmatrix} \quad 2.45$$

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & 0 \\ 0 & J_4 \end{bmatrix} = \begin{bmatrix} \Delta\delta \\ \Delta|v| \end{bmatrix} \quad 2.46$$

Then,

$$\Delta P = J_1 * \Delta\delta = \begin{bmatrix} \Delta P \\ \Delta\delta \end{bmatrix} * \Delta\delta \quad 2.47$$

$$\Delta Q = J_4 * \Delta|v| = \begin{bmatrix} \Delta Q \\ \Delta|v| \end{bmatrix} * \Delta|v| \quad 2.48$$

Matrix equation is separated into the following decoupled equations as follows:

$$\frac{dP_i}{d\delta_i} = \sum_{j \neq i}^n |V_i| |V_j| |y_{ij}| \sin(\theta_{ij} + \delta_i - \delta_j) \quad 2.49$$

$$\frac{dP_i}{d\delta_i} = \sum_{j=1}^n |V_i| |V_j| |y_{ij}| \sin(\theta_{ij} + \delta_i - \delta_j) - |v_i|^2 |y_{ii}| \sin(\theta_{ii}) \quad 2.50$$

$$\frac{dP_i}{d\delta_i} = -Q_i - |v_i|^2 |y_{ii}| \sin(\theta_{ii}) \quad 2.51$$

$$\frac{dP_i}{d\delta_i} = -Q_i - |v_i|^2 B_{ii} \quad 2.52$$

And,

$$\frac{dP_i}{d\delta_j} = -|V_i| |V_j| |y_{ij}| \sin(\theta_{ij} + \delta_j - \delta_i) \quad , \quad \delta_j - \delta_i \approx 0 \quad 2.53$$

$$\frac{dP_i}{d\delta_j} = -|V_i| |V_j| |y_{ij}| \sin(\theta_{ij}) \quad 2.54$$

$$\frac{dP_i}{d\delta_j} = -|V_i| |V_j| B_{ij} \quad 2.55$$

$$\frac{dP_i}{d\delta_j} = -|V_i|B_{ij} \quad 2.56$$

Similarly, for $J_4 = \frac{dQ}{d|V|}$:

$$\frac{dQ_i}{d|V_i|} = -|V_i|B_{ii} \quad 2.57$$

$$\frac{dQ_i}{d|V_j|} = -|V_i|B_{ij} \quad 2.58$$

As a result, four main equations will appear:

$$\frac{\Delta P}{|V|} = -B^{\backslash} \Delta\delta \quad \Rightarrow \quad \Delta\delta = -[B^{\backslash}]^{-1} \frac{\Delta P}{|V|} \quad 2.59$$

$$\frac{\Delta Q}{|V|} = -B^{\wedge} \Delta|v| \quad \Rightarrow \quad \Delta|v| = -[B^{\wedge}]^{-1} \frac{\Delta Q}{|V|} \quad 2.60$$

Where B^{\backslash} & B^{\wedge} are real and imaginary parts of admittance matrix.

Fast decoupled algorithm steps for solution of the power-flow problem can be achieved by following the steps below:

1. Define the initial estimates for the voltage magnitudes and angles at each bus.
2. Linearize the power balance equations for each bus using the initial estimates of the voltage magnitudes and angles. This gives equations for the voltage magnitude and angle at each bus.
3. Form the Jacobian matrices J_1 and J_4 by taking partial derivatives of the power balance equations concerning the voltage magnitudes and angles, respectively.
4. Solve the voltage magnitude equations using the Jacobian matrix J_1 and J_4 . This gives an updated estimate of the voltage magnitudes and voltage angles respectively at each bus.
5. Check the convergence criteria.
6. Update the voltage magnitude and angle estimates using the new estimates from step 4.
7. Repeat steps 2 to 6 until the convergence criteria are met.

2.2.4 Forward / Backward Sweep Method

The majority of the electric power distribution system is radial, and some buses are subject to heavy loads, an imbalanced load, and a high R/X ratio. It has been demonstrated that a number of power flow strategies are effective for transmission networks. Power flow analysis in radial distribution networks, however, cannot employ the design assumptions taken into account for power flow methodologies in transmission networks due to their convergence.

The backward/forward sweep (BFS) is one of the best power flow strategies for radial networks. The requirements of a radial distribution network are met by a proposed load flow solution. Two computation operations are needed for every iteration of the fundamental operating principle of BFS. Every node's voltage is calculated by the forward sweep. The backward process transfers the power and current from the end nodes' branch to the branch connected to the reference node by using power or current flow solutions. Both the voltage and the current or power value are kept constant throughout the forward and backward sweeps, respectively, from the reference node to the end nodes [60].

The first step in BFS is to calculate the carrier injection current which can be calculated as shown in the following equation:

$$I_i^{(k)} = \text{conj } S_i / v_i^{(k-1)} \quad 2.61$$

where $I_i^{(k)}$ is the injection current at node i at iteration k , S_i is the injection power at node i and $v_i^{(k-1)}$ is the voltage of node i at iteration $k-1$.

The second step is back sweep which is performed using Equation 62 Starting from the last ordered branch of the root node.

$$J_{i-1,i}^{(k)} = I_i^{(k)} + \sum (J_{i,i+1}^{(k)}) \quad 2.62$$

Where $J_{i-1,i}^{(k)}$ is the current of the branch connecting node i to its upstream node $i-1$ and $\sum (J_{i,i+1}^{(k)})$ is the sum of all currents of branches emanating from the node.

The next step is forward sweep which calculates voltages for each node by the following equation:

$$v_i^{(k)} = v_{i-1}^{(k)} - J_{i-1}^{(k)} * z_{i-1,i} \quad 2.63$$

Where $v_i^{(k)}$ is the voltage of node i at iteration k and $v_{i-1}^{(k)}$ is the direct source node voltage of node i, $J_{i-1}^{(k)}$ is the current of branch connecting node i to its immediate upstream node and $z_{i-1,i}$ It is the impedance of the branch connecting node i to the immediate upstream node.

The stopping criteria for BFS are the convergence of voltage magnitudes obtained by tracking voltage magnitude differences between two successive iterations as a following equation:

$$\left(v_i^{(k-1)} - v_i^{(k)} \right) < \alpha \quad 2.64$$

Where α is the tolerance limit $\alpha = 0.0001$

Forward / Backward sweep algorithm steps for solution of the power-flow problem are as shown below:

1. Initialization:

- i. Initial voltage is 1p.u ($v_i=1-0j$, for $i=1$ to n)
- ii. Initial power losses both real and reactive are zero

$$(P_{lj} = 0 \text{ and } Q_{lj} = 0 \text{ for } j = 1 \text{ to } m)$$

Where:

n: total number of nodes.

m: Total number of branches.

2. find out different network matrices

- i. Convert the resistance, reactance, voltages and power in per unit form.
- ii. Calculate matrix [A]: (Branch-Node incidence matrix).

$$A_{i,j} = \begin{cases} -1 & \text{if } j = \text{sending node} \\ + & \text{if } j = \text{receiving node} \end{cases}$$

- iii. Calculate number of possible paths by computing number of end nodes.

iv. Find out the number of nodes on each possible path. Let the lateral have maximum 'm' number of branches, so the bus matrix [B] will be having dimensions (l*m).

v. find out linked branches beyond a branch, form next- linked node matrix [C]

3. Set iteration count t= 1, where t= 1 to t-max

4. Calculate current from each branch as shown in the following equation:

$$I_j = \left(\frac{S_{i+1}}{v_{i+1}} \right) \text{ for } I = 1 \text{ to } m \quad 2.65$$

The above equation i

$$\text{Here, } s_{j+1} = (P_{i+1} + JQ_{i+1})$$

5. Backward sweep: update current starting from the end nodes

$$I_k = \sum_j I_j \text{ for } k=1 \text{ to } m$$

where , $j \in c_j$

Here c_j , is the set of next linked node beyond k branch.

6. Forward sweep: from branch currents update the nodal voltages starting from source node.

$$v_{k+1} = V_i - (I_k * z_k) , \text{for } k=1 \text{ to } n$$

7. t=t+1 until t=tMAX

8. These steps are repeated until the power mismatch is less than the specified accuracy (α) or t=tMAX

Four power flow analysis methods (Newton Raphson, Guas-seidel, Fast-decupled, and forward\backward sweep methods) have been discussed. These previously discussed methods will be implemented in the code of this research, where the most suitable method based on the characteristics of the system being analyzed, computational resources available, the desired level of accuracy, and speed of convergence will be chosen.

2.3 Artificial Bee Colony Algorithm (Meta-Heuristic Approach)

"It is a swarm based meta-heuristic algorithm that was introduced by Karaboga in 2005 (Karaboga, 2005) for optimizing numerical problems". It has shown its effectiveness in searching through a variety of optimization problems [61]. However, the original ABC shows a poor convergence velocity during the search. It is a newly developed optimization method that simulates the intelligent foraging activity of honeybees.

The ABC algorithm uses three different kinds of bees: scout, observer, and worker bees. After gathering food from nearby sources, worker bees advise the other bees about these feeding opportunities. Many a time, from among the food sources under the scrutiny of the worker bees, the worker bees select the best food source. Observer bees simply observe the dances made by worker bees to locate better food sources with a higher probability of selecting a better food source. Few worker bees, who give up on their present food sources in a bid to find better food sources, are called scout bees. The number of worker or observer bees is equal to the number of solutions in the swarm.

In ABC, A colony, of (n) agents, or bees, searches the solution space, with each potential solution viewed as a food source. An artificial bee assesses a solution's profitability (fitness) each time it falls on a blossom. The primary search cycle of the bee's method is iterated for a predetermined number (MCN) of times until a solution of acceptable fitness is satisfied. The algorithm also includes an initialization process. Five operations make up each search cycle: site desertion, neighborhood limitation, local search, recruiting, and global search. Scout bees are thrown into the search space at random during the initialization phase, and they assess the suitability of the solutions based on where they fall.

The scouts that visited the fittest solutions enlist foragers to scout out the areas around the most likely solutions. While the remaining scouts recruit the scouts that found the best solutions recruit foragers each. Thus, the profitability of the food supply determines how many foragers are hired. A forager takes over the role of the new scout if they find a solution in a location that is more fit than the one the scout visited. The search patch's size is reduced if no forager discovers a solution that is more fitness. Consequently, the local exploration's scope gradually narrows to the vicinity of the local fitness peak. A new scout is created at random locations, and the patch is abandoned (site abandoned) if after

a certain number of searches cycles the fitness value does not get better or is not likely to be enhanced in that location, the current search location (food source) is considered. This is known as the local maximum of fitness.

such as locations with a high level of fitness (a technique called global search). The global search approach is used to generate solutions at random for the final search patches. The scout population is again composed of elite site foragers, scouts generated by the global search procedure, and the best site foragers remaining from the local search procedure (some may have been re-initialized by the site abandonment procedure) after a single search cycle. The size of the entire artificial bee colony is (scouts + elite sites foragers + best sites foragers). real bee colonies, a limited number of scouts continuously search the solution space for novel [62].

The algorithm shown in figure 3.2 [63] illustrates ABC method and can be summarized by the following steps:

1. Set parameters of ABC: N = swarm population

MCN = maximum cycle number

D= diminution of problem (number of locations)

$$\text{Limit} = \left(\frac{N}{2}\right) * D$$

$$\text{Number of food sources} = \left(\frac{N}{D}\right)$$

2. Read system data which contains buses and lines data of the system.
($P_{load}, Q_{load}, X_{line}, R_{line}$)
3. Initialize the food source positions all the vectors of the population of food sources, x_m are initialized ($m=1,2,\dots,N$, x_{mi} (matrix of locations and sizes), here $i = 1,2, \dots$ by the following equations

$$x_{mi} = l_i + rand(0,1) * (U_i - l_i) \tag{2.66}$$

where l_i and U_i are the lower and upper bound of the parameter x_{mi} , respectively.

4. Run power flow analysis, and evaluate fitness function.

5. Set the cycle iteration to 1.
6. Employed bee phase modifying on initial values locations and sizes by determine a neighbor food source V_m using the formula given by the following equation:

$$v_{mi} = x_{mi} + \Phi_{mi}(x_{mi} - x_{ki}) \quad 2.67$$

Where:

x_k : a randomly selected food source.

i : a randomly chosen parameter index.

Φ_{mi} : a random number within the range $[-a, a]$.

After producing the new food source (v_m) its fitness is calculated.

7. Run power flow with constrains checking, evaluate fitness function, and applying the greedy selection and maintain the best value by using the following equation:

$$fit_m(\vec{x}_m) = \begin{cases} \frac{1}{1+fn(\vec{x}_m)} & \text{if } fn(\vec{x}_m) \geq 0 \\ 1 + |fn(\vec{x}_m)| & \text{if } fn(\vec{x}_m) < 0 \end{cases} \quad 2.68$$

where $fn(x_m)$ is the multi objective function value of solution x_m . MOPI is calculated as follows shown previously in equation 3.9.

8. Onlooker bee phase evaluate the probability p_m and modifying on locations and sizes values to get richer sources and positive feedback behavior. P_m is calculated as shown in the following equation:

$$p_m = \frac{fit_m(\vec{x}_m)}{\sum_{m=1}^n fit_m(\vec{x}_m)} \quad 2.69$$

9. Run power flow with constrains checking, evaluate fitness function, and apply the greedy selection and maintain the best value.

10. Check if (trial > limit) \Rightarrow yes: go to step (11)

\Rightarrow no: go to step (13)

11. Scout bee phase to generate new locations and sizes values & Sources that are initially impoverished or have become impoverished due to exploitation are abandoned.

12. Run power flow with constrains checking, evaluate fitness function & apply the greedy selection and maintain the best value.
13. Keep track of the best value.
14. Check if (MCN = MAX) \Rightarrow yes: go to step (15)
 \Rightarrow no: go to step (6)
15. Print the results.

Figure 2.2 -Appendix U- illustrates the flowchart of the ABC algorithm.

2.4 Improved Artificial Bee Colony Algorithm (Assorted Approach)

As we mentioned in the previous part, where we explained the bee algorithm in detail, in this part we will build on the previous explanation, as the development will be by adding Chaotic Local Search. The ABC algorithm is employed in the exploration phase to update the bees' food source, and when combined with chaos theory, it is used in the exploitation phase to alter the best food source that is produced by ABC. Additionally, the application of chaos theory enhances search behavior and helps avoid solutions that become stuck in a local optimum.

Chaotic maps are mathematical functions that produce a very random pattern, given an initial value. A continuous or discrete time parameter can be used to specify map parameters. Repeated functions are typically used as a format for discrete maps. Random values are produced iteratively using chaotic maps and recursive functions to get away from the initial value. This behavior helps to get out from the local optimum value of the function searching away from the previous values. There are many types of chaotic maps used in optimization, in this thesis, we use a hybrid map that combines tent and logistic maps and implementation into ABC algorithm can be summarized as the following equations:

$$cx_i^{k+1} = 4cx_i^k(1 - cx_i^k) \quad i = 1, 2, \dots, n \quad 2.70$$

where k is the maximum number of iterations. The chaotic local search has the following procedure the decision variables x_i^k is converted into chaotic variables cx_i^k using the equation below:

$$cx_i^k = \frac{x_i^k - lb_i}{ub_i - lb_i} \quad 2.71$$

where, lb_i and ub_i are the lower and upper bound of the x (food source).

Determine the chaotic variables cx_i^{k+1} using the equation (3.70) for the next iteration

The chaotic variables cx_i^{k+1} is converted into decision variables x_i^{k+1} using the following equation

$$x_i^k = lb_i + cx_i^k (ub_i - lb_i) \quad i = 1, 2, \dots, n \quad 2.72$$

Calculate the new solution with decision variables x_i^{k+1} . Update the new solution if it has better fitness when compared to old one.

The following steps are the modified steps for the chaotic artificial bee algorithm, along with modified flowchart to match the algorithm's steps:

1. Set parameters of ABC: N = swarm population

MCN = maximum cycle number

D = diminution of the problem (number of locations)

$$\text{Limit} = \left(\frac{N}{2}\right) * D$$

$$\text{Number of food sources} = \left(\frac{N}{D}\right)$$

2. Read system data which contains buses and lines data of the system.
($P_{load}, Q_{load}, X_{line}, R_{line}$)
3. Initialize the food source positions all the vectors of the population of food sources, x_m are initialized ($m=1,2,\dots,N$, x_{mi} (matrix of locations and sizes), here $i = 1, 2, \dots$ by the following equ:

$$x_{mi} = l_i + rand(0,1) * (U_i - l_i)$$

where l_i and U_i are the lower and upper bound of the parameter x_{mi} , respectively.

4. Run power flow & evaluate fitness function.
5. Set the cycle =1
6. Employed bee phase modifying on initial values locations and sizes by determining a neighbor food source V_m using the formula given by equ:

$$v_{mi} = x_{mi} + \phi_{mi}(x_{mi} - x_{ki})$$

where x_k is a randomly selected food source, i is a randomly chosen parameter index and ϕ_{mi} is a random number within the range $[-a, a]$. After producing the new food source v_m its fitness is calculated.

7. Run power flow with constraints checking, evaluate fitness function & apply the greedy selection, and maintain the best value by using the following formula:

$$fit_m(\vec{x}_m) = \begin{cases} \frac{1}{1+f_m(\vec{x}_m)} & \text{if } f_m(\vec{x}_m) \geq 0 \\ 1 + |f_m(\vec{x}_m)| & \text{if } f_m(\vec{x}_m) < 0 \end{cases}$$

where $f_m(x_m)$ is the multi-objective function value of solution $x_m \Rightarrow$ (MOPI) equ (9)

8. The Onlooker bee phase evaluates the probability p_m and modifying locations and size values to get richer sources and positive feedback behavior appears. As shown in the following equation:

$$p_m = \frac{fit_m(\vec{x}_m)}{\sum_{m=1}^n fit_m(\vec{x}_m)} \quad 2.73$$

9. Run power flow with constraints checking, evaluate fitness function & apply the greedy selection, and maintain the best value.

10. Check if (trial > limit) \Rightarrow yes: go to step 11

\Rightarrow no: go to step 13

11. Scout bee phase to generate new locations and sizes values & Sources that are initially impoverished or have become impoverished due to exploitation are abandoned.
12. Run power flow with constraints checking, evaluate fitness function & apply the greedy selection, and maintain the best value.

13. apply the chaotic local search to modify the locations and sizes of PV and apply the greedy selection to maintain the best values by applying the equation 3.70 – 3.72.
14. Memorize the best solution.
15. Check if (MCN = MAX) ⇒ yes: go to step 15
⇒ no: go to step 6
16. Print results -Appendix U-.

Chapter Three

Results and Discussion

This chapter presents an examination of the optimal distribution and sizing of photovoltaic (PV) systems using an improved Artificial Bee Colony (ABC) algorithm within the IEEE 33 bus test system. The study involves analyzing load flow results with and without PV systems, comparing randomly distributed PV systems, and evaluating PV systems distributed using both the ABC and Improved ABC (IABC) algorithms. Additionally, the algorithm is applied to a portion of the Nablus network, and the outcomes are scrutinized.

The code for this analysis was developed in Python using Jupyter Notebook. The load flow analysis was conducted with the help of the Panda power library, along with other libraries such as NumPy and Pandas. Various scenarios were considered during the code development process.

3.1 Methodology

The methodology follows a series of structured steps:

1. **Data Extraction:** Network data is initially extracted from an Excel sheet.
2. **Network Modeling:** This data is then used to construct a network model tailored to the specific requirements of the case, whether for transmission or distribution systems, as these systems require different optimization methodologies.
3. **Load Flow Analysis:** A suitable load flow analysis method is applied, considering the unique characteristics of the network.
4. **Algorithmic Procedures:** Finally, algorithmic procedures are executed to process the data, ultimately yielding the results.

The comprehensive approach ensures that the analysis captures the nuances of each scenario, providing a detailed understanding of the optimal distribution and sizing of PV systems within the test network.

3.2 Load Flow Analysis of IEEE 33 Bus test System Without PV

The IEEE 33 bus system, a small electrical network with one generation bus and 32 loading buses, is commonly used for testing and studying various parameters such as active and reactive power losses, voltage profiles, magnitudes, and phase angles. Detailed information about this system can be found in Appendix B. The network features a radial configuration, as illustrated in Figure 3.1 -Appendix U-.

Load Flow Analysis Results:

The load flow analysis of the IEEE 33 bus network, conducted without the installation of photovoltaic (PV) systems (see Appendix C), provides the following results:

- Total Generation:
- Active Power (P_g): 3.8178 MW
- Reactive Power (Q_g): 2.3589 MVAR
- Total Loads:
- Active Power (P_{load}): 3.615 MW
- Reactive Power (Q_{load}): 2.22 MVAR

The loads are distributed across 32 load buses.

Power Losses:

- Active Power Losses (P_{losses}): 0.2028 MW, representing 5.31% of the total active power generation.
- Reactive Power Losses (Q_{losses}): 0.1389 MVAR, representing 6.25% of the total reactive power generation.

The distribution of active and reactive power losses across each line, along with the line flow limit index representing the load factor on each line, is illustrated in Figure 4.2.

Voltage Profile and Stability

The analysis of the voltage profile and voltage stability indices for each bus is also shown in Appendix D. The voltage magnitudes of buses within the range of bus 5 to 17 and bus 25 to 32 are below 0.95 p.u., indicating the lowest voltage magnitudes in the network.

These findings provide essential insights into the performance and stability of the IEEE 33 bus network under current load conditions without PV installation

3.3 Allocate and Size PV Random at the weak buses on IEEE 33 Bus Test System

For choosing the size and location of PV we look at the load flow without PV and noticed we have many buses have low voltage profile. Based on this, these buses will have the most priority to put PV in the location. In this section we will install 3 MWP of PV distributed on one, two, and three locations, where these locations will be chosen randomly. See Appendix E, that shows the parameters curves. Refer to Appendix F for the network parameters after installing PV at the weak buses.

The following table shows the simulation results in table 3.1

Table 3.1

IEEE 33 Simulation Results

	Original case	PV in one location	PV in two locations	PV in three locations
Bus number	-	17	17, 33	17, 22, 33
PV size (KWP)	-	3000	1500, 1500	1000, 1000, 1000
Total PV capacity (KWP)	-	3000	3000	3000
Power factor of PV	1	1	1	1
Min bus voltage	0.9037	0.9559	0.981	0.9725
Max bus voltage	1.00	1.0958	1.0323	1.0106
total active power losses (KW)	211	399.25	162.84	111.63
Percentage of losses active reduction	0%	-89.21%	22.8%	47.09%
Total reactive power losses (KVAR)	143.03	338.60	133.04	87.79
Percentage of reactive losses reduction	0%	-136.7%	7%	38.6%
Max Line limit index	42.17%	25.78%	23.3%	22.76%
Max voltage stability index (magnitude)	0.0749	0.075562	0.038661	0.025134

The findings in the above table show that it may not be beneficial to choose the locations and size of PV systems randomly, depending on the voltage drop on bus or distance from the source because this would increase network losses and negatively impact stability coefficients.

3.4 Best PV Allocation and Sizing Using the Conventional ABC Algorithm IEEE 33 Bus Test System

This thesis employs Jupyter Notebook to implement the Python code for the conventional Artificial Bee Colony (ABC) algorithm to determine the optimal allocation and sizing of a photovoltaic (PV) system on the IEEE 33 bus test network.

The implementation is contained in a file titled "conventional ABC" (refer to Appendix G). The code is organized into several sections:

- **Library Imports:** This section imports the necessary libraries for power flow analysis and allows for the selection of methods detailed in Chapter 3, with the Newton-Raphson method being utilized in this instance.
- **Network Data Import:** Network data is imported from an Excel sheet named "IEEE 33 data" to construct the network model.
- **Load Flow Initialization:** This section runs the initial load flow calculations.
- **Fitness Function Calculation:** Here, the fitness function is calculated for initialization, following the methodologies described in Chapter 3.
- **Algorithm Parameters and Constraints:** This section sets the parameters and constraints for the algorithm, which include:
 - PV size range: 100–2000 KWP
 - Number of PV locations: 1 (first case), 2 (second case), and 3 (third case)
 - Number of bees: 50
 - Maximum number of iterations: 100
 - Scout bee limit: 10
 - Penetration factor: 1 (as referenced in other research papers), ensuring the load is not exceeded and allowing for comparative analysis with other studies.
- **Algorithm Execution:** The final section contains the algorithm functions, which utilize the data from the previous sections to perform calculations. The parameters are weighted in a multi-objective function to yield the final results, as depicted in Figure 3.2 -Appendix U-.

This structured approach ensures a comprehensive and methodical implementation of the conventional ABC algorithm for optimal PV system allocation and sizing on the IEEE 33 bus test system.

In this scenario we implement the conventional ABC algorithm on the IEEE 33 bus network and run the code to find the optimum location and size of PV if we install in one location, two locations, and three locations to represent the algorithm's behavior in each case as shown in figure 3.3 -Appendix U- and compare the results of each case with original case (see Appendix H and Appendix I).

According to previous results curves, we noticed that every parameter has been improved so, the summary of improvement shown in Table 3.2

Table 3.2

Conventional ABC on IEEE 33 Results Summary

	Original case	PV in one location	PV in two locations	PV in three locations
Bus number	-	6	6, 14	13, 23, 28
PV size (KWP)	-	2598.6	1764.2, 624.5	753.6, 1100, 1070.3
Total PV capacity	-	2598.6	2388.7	2923.9
Min bus voltage	0.9037	0.944	0.952	0.95989
Max bus voltage	1.00	1.00	1.00	1.00
total active power losses (KW)	202.8	105.39	86.92	76.49
Percentage of losses active reduction	0%	48.03%	57.14%	62.28%
Total reactive power losses (KVAR)	138.9	78.85	61.63	54.28
Percentage of reactive loss reduction	0%	43.23%	55.62%	60.92%
Max Line limit index	40.93%	23.33%	23.96%	21.89%
Max voltage stability index	0.0747	0.0479	0.0444	0.0326
Best Fitness function value	0.5646	0.3141	0.2620	0.2334

The previous table and figures present the outcomes of applying the conventional ABC algorithm to determine the optimal locations and size of PV systems. The results show that this approach greatly enhanced network performance, with losses declining, voltages rising, and network stability also increasing. The figures additionally display the algorithm's performance in terms of results obtaining speed and improvement values.

3.5 Finding for the Best PV allocation and Sizing Using the improved ABC Method on 33 bus test system

An improved ABC used for extending the space of exploration of algorithm by using a chaotic map that produce a random number every iteration which help to get out of local optimum and looking for global optimum solution in huge search space.

A file used for implementation named "improved ABC" (see Appendix J). This file has the same parts and content as the previous file "conventional ABC", but it contains an addition to the algorithm's section that adds a chaotic map to make the algorithm better more robust and generally better in searching for the minimum fitness function value. Figure 3.4 -Appendix U- shows the flowchart of the chaotic ABC (Improved ABC).

As for this modification of the Chaotic ABC algorithm, the IEEE 33 bus system will be used to demonstrate the behavior of this algorithm by making the same comparisons of the results in section 3.3. Figure 3.5 -Appendix U- illustrates the convergence graphs of

the algorithm along with its fitness values (see Appendix K) that shows the comparison results of each parameter as in the previous section (see Appendix L).

According to previous results curves we noticed that every parameter has been improved and the summary of improving shows in table 3.3

Table 3.3

Implementing Improved ABC on IEEE 33 Results Summary

	Original PV in one case	PV in two location	PV in two locations	PV in three locations
Bus number	-	6	14, 30	13, 24, 30
PV size (KWP)	-	2547.6	785.4, 1146.7	837.4, 763.7, 1062.8
Total PV capacity	-	2547.6	1932.1	2663.9
Min bus voltage	0.9037	0.9439	0.962	0.977
Max bus voltage	1.00	1.00	1.00	1.00
total active power losses (KW)	202.8	103.82	82.42	71.08
Percentage of losses active reduction	0%	48.8%	59.35%	64.95%
Total reactive power losses (KVAR)	138.9	78.76	56.9	49.58
Percentage of reactive losses reduction	0%	43.3%	59.03%	64.3%
Max Line limit index	40.93%	23.58%	26.27%	22.7%
Max voltage stability index	0.0747	0.0480	0.0305	0.0309
Best Fitness function value	0.5646	0.3114	0.2469	0.2174

From the previous results, the performance of the improved ABC algorithm is better than the conventional ABC algorithm, as it is faster in obtaining results it takes around 1.5 minute to give as result while the conventional takes around 4 minutes, and the results are better for IEEE 33 bus network as we will show in section 3.5.

3.6 Comparison of Adapted ABC Algorithms in Literature and the Improved ABC Algorithm

This section compares various algorithms in research publications with different fitness functions for all scenarios of installing PV in one, two, and three places with our improved ABC approach.

The paper titled “ABC algorithm based optimal sizing and placement of DGs in distribution networks considering multiple objectives” [46], this research designed for distributed energy resources (DERs) on distribution systems to determine the best placement and size. The suggested approach combines the strengths of the ABC and ACO algorithms' local and global search capabilities, respectively, by utilizing discrete (location optimization) and continuous (size optimization) structures. A set of non-dominated solutions that are saved in the external archive are also produced by the

suggested algorithm using a multi-objective ABC. The goals include reducing power losses, lowering the overall cost of electrical energy, lowering the emissions from the substation and resources, and enhancing voltage stability. To tackle the optimization issue in a stochastic setting, this study uses an efficient point estimate technique (PEM). On the IEEE 33 bus distribution systems, the suggested method is tested. The outcomes show that, when compared to the existing evolutionary optimization techniques, the suggested algorithm has efficacy. the results of first case (IEEE 33) comparing with result of our improved ABC is shown in the following table3.4 -Appendix T-.

After investigating the results in the previous table, it's evident to us that the improved beehive algorithm performs better than other algorithms because it improves network performance more than other algorithms while requiring a smaller capacity of solar systems, which will also result in lower capital and operating costs.

3.7 implementation of the improved ABC Method on Part of the Nablus network

The Israeli Electricity Company delivers electricity to the Northern Electricity distribution Company, which distributes electricity across the city of Nablus. The Israeli Electricity Company is in control of the generating and connection points. The Northern Electricity Company is responsible of all electrical equipment in the city, and the most significant connecting points between the Israeli network and the city of Nablus are: Salem, Hawarah, Innab, and Sarra from connection point No. 1 to connection No. 6. These points supply power to a number of the power substations in the city of Nablus, namely: the southern substation (Tel), Karkon, Cnentral, Askar, Wadi al-Tuffah, and An-Najah University substations.

In this thesis, four substations were taken as a case study, which are Wadi al-Tuffah substation, the An-Najah University substation, the Karkon substation, and the Central substation. These substations feed 247 buses which includes 213 distribution transformers (See Appendix M).

The attached figures in Appendix N show a line diagram for each substation, the number of buses and transformers, their capacities, and the cables and overhead lines parameters that are used to transmit power (see Appendix O and Appendix S).

According to available information, we suggested the transformers were placed in constant load through simulations. based on information from operational engineer we considering a load factor 0.4 and a power factor 0.95 of the transformers because we don't have measured data in all the 213 transformers were suggested for the simulation process. Therefore, the focus of this study is on analyzing and assessing the behavior of the algorithm that was built, which has a rather big network form and many branches, such as part of the Nablus electrical network. Total energy in this part of the network is 41.5 MWh we concede the penetration of PV is 20% of the total energy. We made a load flow in original case (see Appendix P), then To speed up the simulation, the connecting buses were deleted by reducing the number of cables and overhead lines and placing equivalent lines with a maximum current load equal to the maximum current of the line with the least cross-section. The following figures show the results of the simulation results. Figure 4.6 -Appendix U- shows the convergence of the fitness function through the iterations. shows the comparison results of each parameter (see Appendix Q and Appendix R) -Appendix T-.

Based on the aforementioned findings and the available data of network, tell and Nablus al-jadedah area bus numbers 106 and 112 in south of Nablus, the Water Street area to the east buses (165 and 169), and the Titi Bridge area to the north (214) are the best places to connect PV systems to the network. These connection points may not be suitable on the real due to the presence of residential areas, so stations can be established in the closest available area of Theas buses. By connecting the PV systems, the network performs better and the burden on the Israeli side's connection points is lessened, which lowers energy imports and losses within the network and enhances voltages and stability.

Chapter Four

Conclusion and Future Work

4.1 Conclusion

This thesis presents an improved IABC algorithm for sizing and allocation optimization of the PV systems in IEEE 33-bus networks and a part of the Nablus network concerning loss reduction, voltage profile, and stability improvement. The enhancement to be done includes incorporating a chaotic map to help the algorithm avoid trapping in local optima due to the generation of chaotic sequences, hence exploring a larger space of the solution. This improvement to the conventional ABC considerably enhances the performance of this technique.

The implementation was done using Python, where the data input is read from Excel sheets to simulate the network, load flow analysis, and apply both conventional ABC and improved IABC algorithms. The IEEE 33-bus network was taken for the simulations under different scenarios: without PV, randomly distributed PV, with conventional ABC distribution, and with improved IABC distribution.

As the results in Chapter 4 show, the IABC algorithm performed the best overall by giving minimum active and reactive losses, reduction of loading factor, maintenance of voltages within limits above 0.95 p.u., and improvement of the stability coefficient with minimal PV system sizes. The IABC algorithm, when applied on part of the Nablus network, showed good potential and satisfactory results in handling a large and complex network, as elaborated on in Chapter 4. The research concludes with statements that depict optimal placements and sizing of PV systems as key to maximizing benefits, which include loss minimization and the enhancement of voltage stability. "Great size and cost reductions result from strategic placement of PV systems rather than sitting at the furthest buses or weakest voltage; hence, there is a need to develop advanced algorithms, like the IABC, to derive the best outcomes in integrating a PV system".

4.2 Future work

This research is an excellent stake for electrical power system sizing and allocation based on PV systems. However, the following facts should be taken into account for future studies to contribute more to this science:

1. Exploring More Complex Power Networks: Future studies will focus on analyzing more complex power networks, taking into account different network topologies to understand the broader applicability of the proposed methods.
2. Multi-objective optimization: is the application of multi-objective optimization problems taking into account the economic factors and other primary essential considerations of the power networks, such as the total harmonic distortion and the placement of PV to improve the power factor.
3. Integration of Diverse Distributed Generation (DG) Units: Research the integration with other DG units, such as wind turbines and micro-turbines, and compare the results to when PV systems are installed. Improving Chaotic Map: Investigating new methods to enhance the chaotic map further for better performance of the ABC algorithm.
4. Algorithm Variable Optimization: An algorithm for determining the best values of variables and random numbers in the algorithm to optimize performance for different cases.
5. Comparison of optimization techniques: The use of several optimization techniques to compare their output with that of the proposed Improved ABC algorithm, seeking the most effective methodologies that will be used for PV system integration.

These areas would be the foundation for future research to build up the findings of this study further and continue to fine-tune the efficiency and effectiveness of PV system sizing and allocation in the electrical power network.

List of Abbreviations

Abbreviation	Meaning
IEC	Israeli Electrical Corporation
PERC	Palestinian Electricity Regulator Council
PENRA	Palestinian Energy & Natural Resources Authority
PETL	Palestinian Electricity Transmission Line Company
JDECO	Jerusalem District Electricity Company
HEPCO	Hebron Electric Power Company
SELCO	Southern Electric Company
NEDCO	Northern Electricity Distribution Company
GEDCO	Gaza Electricity Distribution Company
PV	Photovoltaic
CSP	Concentrated Solar Power
KV	Kilo Volt
KWh	Kilowatt Hour
GWh	Gigawatt Hour
MW	Megawatt
P	Active Power
Q	Reactive Power
DFIG	Doubly-Fed Induction Generator
DG	Distribution Generation
DP	Dynamic Programming
OO	Ordinal Optimization
NLP	Non-Linear Programming
MINLP	Mixed Integer Non-Linear Programming
ES	Exhaustive Search
SQP	Sequential Quadratic Programming
CPF	Continuation Power Flow
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
BA	Bat Algorithm
ABC	Artificial Bee Colony
CSA	Cuckoo Search Algorithm
OKHA	Oppositional Krill Herd Algorithm
ALO	Ant Lion Optimization

FO	Firefly Optimization
TS	Tuba Search
AOA	Archimedes Optimization Algorithm
IWDO	Intelligent Water Drop Optimization
SFLA	Shuffled Frog Leaping Algorithm
PFDEA	Pareto Frontier Differential Evolution Algorithm
DE	Differential Evolution
SOSA	Symbiotic Organisms Search Algorithm
SBB-BC	Supervised Big Bang – Big Crunch
EHO	Elephant Herd Optimization
QOSIMBO-Q	Quis Oppositional Swine Influenza Model-Based Optimization with Quarantine
MHBM	Modified Honey Bee Mating
PLR	Real Power Losses Index
QLR	Reactive Power Losses Index
VPI	Voltage Profile Index
VSI	Voltage Stability Index
SFVSI	Simplified Fast Voltage Stability Index
LFLI	Line Flow Limit Index
T.L	Transmission Line
MCN	Maximum Cycle Number

References

- [1] P. C. B. of Statistics, “Total Energy Supply by Year and Type of energy, 2009-2018,” 2020. [Online]. Available: https://www.pcbs.gov.ps/Portals/_Rainbow/Documents/Energy-Supply-2009-2020_E.html
- [2] “PCBS | Annual Energy Tables and Energy Balance,” 2020. [Online]. Available: https://www.pcbs.gov.ps/site/lang__en/1030/Default.aspx
- [3] E. Ajlouni and H. Alsamamra, “A Review of Solar Energy Prospects in Palestine,” *American Journal of Modern Energy*, vol. 5, no. 3, 2019, doi: 10.11648/j.ajme.20190503.11.
- [4] A. Juaidi, R. Abdallah, O. Ayadi, T. Salameh, A. A. Hasan, and A. Ramahi, “Wind energy in Jordan and Palestine: current status and future perspectives,” in *Renewable Energy Production and Distribution: Recent Developments*, 2022. doi: 10.1016/B978-0-323-91892-3.00006-6.
- [5] M. Al Deen Al Arda, N. Amin, W. Abu Al Halaweh, O. Sharqieh, G. Director, and A. Papadopoulou, “Manal Taha-(MDLF) of the Projects Department at the Ministry of Local Government in Palestine (National Focal Point CES-MED),” Nader Al Betar.
- [6] “PCBS | Annual Energy Tables and Energy Balance,” 2021. [Online]. Available: https://www.pcbs.gov.ps/site/lang__en/886/Default.aspx
- [7] S. Henderson, “Natural gas in the Palestinian Authority: the potential of the Gaza Marine offshore field (Policy Brief),” 2014.
- [8] B. S. Udasin, “Power cut again in Palestinian cities despite collective punishment; accusations,” Nov. 2015, [Online]. Available: https://www-jpost-com.translate.googleusercontent.com/translate/g/translate/arab-israeli-conflict/israel-electric-corporation-cuts-off-power-in-palestinian-cities-for-second-time-392163?_x_tr_sl=en&_x_tr_tl=ar&_x_tr_hl=ar&_x_tr_pto=rq

- [9] A. El-Ghussain, "Infrastructure and development in the occupied Palestinian territories." [Online]. Available: <https://poica.org/2010/10/%D8%A7%D9%84%D8%A8%D9%86%D9%8A%D8%A9-%D8%A7%D9%84%D8%AA%D8%AD%D8%AA%D9%8A%D8%A9-%D9%88%D8%A7%D9%84%D8%AA%D9%86%D9%85%D9%8A%D8%A9-%D9%81%D9%8A-%D8%A7%D9%84%D8%A3%D8%B1%D8%A7%D8%B6%D9%8A-%D8%A7%D9%84/>
- [10] P. Fraser, "Distributed Generation in Liberalised Electricity Markets Distributed Generation in Liberalised Electricity Markets," *International Energy Agency*, 2002.
- [11] A. El-Fergany, "Optimal allocation of multi-type distributed generators using backtracking search optimization algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 64, 2015, doi: 10.1016/j.ijepes.2014.09.020.
- [12] T. Yuvaraj, K. Ravi, and K. R. Devabalaji, "Optimal Allocation of DG and DSTATCOM in Radial Distribution System Using Cuckoo Search Optimization Algorithm," *Modelling and Simulation in Engineering*, vol. 2017, 2017, doi: 10.1155/2017/2857926.
- [13] L. Schneider and M. Peht, "Embedding Micro Cogeneration in the Energy Supply System," in *Micro Cogeneration*, 2006. doi: 10.1007/3-540-30821-0_9.
- [14] P. P. Barker and R. W. De Mello, "Determining the impact of distributed generation on power systems: Part 1 - Radial distribution systems," in *Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference*, 2000. doi: 10.1109/pess.2000.868775.
- [15] G. Corporate Research and N. York, "DG Power Quality, Protection, and Reliability Case Studies Report," 2003. [Online]. Available: <http://www.osti.gov/bridge>
- [16] M. H. Moradi and M. Abedini, "A combination of genetic algorithm and particle swarm optimization for optimal DG location and sizing in distribution systems,"

- International Journal of Electrical Power and Energy Systems*, vol. 34, no. 1, 2012, doi: 10.1016/j.ijepes.2011.08.023.
- [17] R. Viral and D. K. Khatod, "An analytical approach for sizing and siting of DGs in balanced radial distribution networks for loss minimization," *International Journal of Electrical Power and Energy Systems*, vol. 67, 2015, doi: 10.1016/j.ijepes.2014.11.017.
- [18] M. Pesaran, A. A. Mohd Zin, A. Khairuddin, and O. Shariati, "Optimal sizing and siting of distributed generators by a weighted exhaustive search," *Electric Power Components and Systems*, vol. 42, no. 11, 2014, doi: 10.1080/15325008.2014.921947.
- [19] Y. Cao, X. Wang, Y. Li, Y. Tan, J. Xing, and R. Fan, "A comprehensive study on low-carbon impact of distributed generations on regional power grids: A case of Jiangxi provincial power grid in China," 2016. doi: 10.1016/j.rser.2015.09.008.
- [20] M. Louzazni, A. Khouya, S. Al-Dahidi, M. Mussetta, and K. Amechnoue, "Analytical optimization of photovoltaic output with Lagrange Multiplier Method," *Optik (Stuttg)*, vol. 199, 2019, doi: 10.1016/j.ijleo.2019.163379.
- [21] L. R. Taylor, W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, "Numerical Recipes: The Art of Scientific Computing," *J Anim Ecol*, vol. 56, no. 1, 1987, doi: 10.2307/4830.
- [22] V. V. S. N. Murthy and A. Kumar, "Comparison of optimal DG allocation methods in radial distribution systems based on sensitivity approaches," *International Journal of Electrical Power and Energy Systems*, vol. 53, no. 1, 2013, doi: 10.1016/j.ijepes.2013.05.018.
- [23] S. H. Lee and J. W. Park, "Selection of optimal location and size of multiple distributed generations by using Kalman Filter algorithm," *IEEE Transactions on Power Systems*, vol. 24, no. 3, 2009, doi: 10.1109/TPWRS.2009.2016540.
- [24] X. S. Yang, *Introduction to algorithms for data mining and machine learning*. 2019. doi: 10.1016/C2018-0-02034-4.

- [25] R. A. Jabr and B. C. Pal, "Ordinal Optimisation approach for locating and sizing of distributed generation," *IET Generation, Transmission and Distribution*, vol. 3, no. 8, 2009, doi: 10.1049/iet-gtd.2009.0019.
- [26] M. Esmaili, "Placement of minimum distributed generation units observing power losses and voltage stability with network constraints," *IET Generation, Transmission and Distribution*, vol. 7, no. 8, 2013, doi: 10.1049/iet-gtd.2013.0140.
- [27] M. F. Alhajri, M. R. Alrashidi, and M. E. El-Hawary, "Improved sequential quadratic programming approach for optimal distribution generation deployments via stability and sensitivity analyses," *Electric Power Components and Systems*, vol. 38, no. 14, 2010, doi: 10.1080/15325008.2010.492451.
- [28] V. Ajjarapu and C. Christy, "The continuation power flow: A tool for steady state voltage stability analysis," *IEEE Transactions on Power Systems*, vol. 7, no. 1, 1992, doi: 10.1109/59.141737.
- [29] C. Blum and A. Roli, "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison," 2003. doi: 10.1145/937503.937505.
- [30] F. Gerges, G. Zouein, and D. Azar, "Genetic algorithms with local optima handling to solve sudoku puzzles," in *ACM International Conference Proceeding Series*, 2018. doi: 10.1145/3194452.3194463.
- [31] M. R. Bonyadi and Z. Michalewicz, "Particle swarm optimization for single objective continuous space problems: A review," 2017. doi: 10.1162/EVCO_r_00180.
- [32] S. K. Sudabattula and K. M, "Optimal allocation of solar based distributed generators in distribution system using Bat algorithm," *Perspect Sci (Neth)*, vol. 8, 2016, doi: 10.1016/j.pisc.2016.04.048.
- [33] A. A. Seker and M. H. Hocaoglu, "Artificial Bee Colony algorithm for optimal placement and sizing of distributed generation," in *ELECO 2013 - 8th International Conference on Electrical and Electronics Engineering*, 2013. doi: 10.1109/eleco.2013.6713817.

- [34] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings*, 2009. doi: 10.1109/NABIC.2009.5393690.
- [35] S. Sultana and P. K. Roy, "Krill herd algorithm for optimal location of distributed generator in radial distribution system," *Applied Soft Computing Journal*, vol. 40, 2016, doi: 10.1016/j.asoc.2015.11.036.
- [36] S. Mouassa, T. Bouktir, and A. Salhi, "Ant lion optimizer for solving optimal reactive power dispatch problem in power systems," *Engineering Science and Technology, an International Journal*, vol. 20, no. 3, 2017, doi: 10.1016/j.jestch.2017.03.006.
- [37] M. H. Sulaiman, M. W. Mustafa, A. Azmi, O. Aliman, and S. R. Abdul Rahim, "Optimal allocation and sizing of distributed generation in distribution system via firefly algorithm," in *2012 IEEE International Power Engineering and Optimization Conference, PEOCO 2012 - Conference Proceedings*, 2012. doi: 10.1109/PEOCO.2012.6230840.
- [38] K. Nara, Y. Hayashi, K. Ikeda, and T. Ashizawa, "Application of tabu search to optimal placement of distributed generators," in *Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference*, 2001. doi: 10.1109/pesw.2001.916995.
- [39] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, no. 3, 2021, doi: 10.1007/s10489-020-01893-z.
- [40] K. Lenin, "REDUCTION OF ACTIVE POWER LOSS BY IMPROVED INTELLIGENT WATER DROP ALGORITHM," *International Journal of Research -GRANTHAALAYAH*, vol. 5, no. 11, 2017, doi: 10.29121/granthaalayah.v5.i11.2017.2335.
- [41] M. Gandomkar, M. Vakilian, and M. Ehsan, "A genetic-based tabu search algorithm for optimal DG allocation in distribution networks," *Electric Power*

Components and Systems, vol. 33, no. 12, 2005, doi: 10.1080/15325000590964254.

- [42] M. H. Moradi, A. Zeinalzadeh, Y. Mohammadi, and M. Abedini, "An efficient hybrid method for solving the optimal siting and sizing problem of DG and shunt capacitor banks simultaneously based on imperialist competitive algorithm and genetic algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 54, 2014, doi: 10.1016/j.ijepes.2013.06.023.
- [43] M. Nayeripour, E. Mahboubi-Moghaddam, J. Aghaei, and A. Azizi-Vahed, "Multi-objective placement and sizing of DGs in distribution networks ensuring transient stability using hybrid evolutionary algorithm," 2013. doi: 10.1016/j.rser.2013.05.016.
- [44] M. H. Moradi and M. Abedini, "A combination of genetic algorithm and particle swarm optimization for optimal distributed generation location and sizing in distribution systems with fuzzy optimal theory," *Int J Green Energy*, vol. 9, no. 7, 2012, doi: 10.1080/15435075.2011.625590.
- [45] F. Valdez, P. Melin, and O. Castillo, "An improved evolutionary method with fuzzy logic for combining Particle Swarm Optimization and Genetic Algorithms," in *Applied Soft Computing Journal*, 2011. doi: 10.1016/j.asoc.2010.10.010.
- [46] M. Kefayat, A. Lashkar Ara, and S. A. Nabavi Niaki, "A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources," *Energy Convers Manag*, vol. 92, 2015, doi: 10.1016/j.enconman.2014.12.037.
- [47] J. J. Jamian, M. W. Mustafa, and H. Mokhlis, "Optimal multiple distributed generation output through rank evolutionary particle swarm optimization," *Neurocomputing*, vol. 152, 2015, doi: 10.1016/j.neucom.2014.11.001.
- [48] M. H. Moradi, S. M. Reza Tousi, and M. Abedini, "Multi-objective PFDE algorithm for solving the optimal siting and sizing problem of multiple DG sources," *International Journal of Electrical Power and Energy Systems*, vol. 56, 2014, doi: 10.1016/j.ijepes.2013.11.014.

- [49] T. Niknam, S. I. Taheri, J. Aghaei, S. Tabatabaei, and M. Nayeripour, "A modified honey bee mating optimization algorithm for multiobjective placement of renewable energy resources," *Appl Energy*, vol. 88, no. 12, 2011, doi: 10.1016/j.apenergy.2011.06.023.
- [50] M. Kumawat, N. Gupta, N. Jain, and R. C. Bansal, "Optimally Allocation of Distributed Generators in Three-Phase Unbalanced Distribution Network," in *Energy Procedia*, 2017. doi: 10.1016/j.egypro.2017.12.122.
- [51] R. Vijay and M. Abhilash, "Elephant Herding Optimization for Optimum Allocation of Electrical Distributed Generation on Distributed Power Networks," *Asian Journal of Electrical Sciences*, vol. 7, no. 2, 2018, doi: 10.51983/ajes-2018.7.2.2108.
- [52] S. Sharma, S. Bhattacharjee, and A. Bhattacharya, "Quasi-Oppositional Swine Influenza Model Based Optimization with Quarantine for optimal allocation of DG in radial distribution network," *International Journal of Electrical Power and Energy Systems*, vol. 74, 2016, doi: 10.1016/j.ijepes.2015.07.034.
- [53] B. Das, V. Mukherjee, and D. Das, "DG placement in radial distribution network by symbiotic organisms search algorithm for real power loss minimization," *Applied Soft Computing Journal*, vol. 49, 2016, doi: 10.1016/j.asoc.2016.09.015.
- [54] D. Singh, R. K. Misra, and D. Singh, "Effect of load models in distributed generation planning," *IEEE Transactions on Power Systems*, vol. 22, no. 4, 2007, doi: 10.1109/TPWRS.2007.907582.
- [55] D. Singh, D. Singh, and K. S. Verma, "Multiobjective optimization for DG planning with load models," *IEEE Transactions on Power Systems*, vol. 24, no. 1, 2009, doi: 10.1109/TPWRS.2008.2009483.
- [56] I. Musirin and T. K. Abdul Rahman, "Novel fast voltage stability index (FVSI) for voltage stability analysis in power transmission system," in *2002 Student Conference on Research and Development: Globalizing Research and Development in Electrical and Electronics Engineering, SCORED 2002 - Proceedings*, 2002. doi: 10.1109/SCORED.2002.1033108.

- [57] A. Shaheen, A. Elsayed, A. Ginidi, R. El-Sehiemy, and E. Elattar, "Reconfiguration of electrical distribution network-based DG and capacitors allocations using artificial ecosystem optimizer: Practical case study," *Alexandria Engineering Journal*, vol. 61, no. 8, 2022, doi: 10.1016/j.aej.2021.11.035.
- [58] H. Saadat, *Power System Analysis: Third Edition*, vol. 3. 2011.
- [59] W. D. Stevenson and S. A. Sebo, "Elements of Power System Analysis, Third Edition," *IEEE Trans Syst Man Cybern*, vol. SMC-6, no. 7, 2008, doi: 10.1109/tsmc.1976.4309538.
- [60] B. Jangid and S. Ghosh, "Load flow Analysis using Backward Forward Sweep Method," 2017.
- [61] D. Karaboga, "Artificial Bee Colony Algorithm," *Scholarpedia*, vol. 5, no. 3, p. 6915, 2010, doi: 10.4249/scholarpedia.6915.
- [62] V. Tereshko and A. Loengarov, "Collective Decision-Making in Honey Bee Foraging Dynamics," *Computing and Information Systems Journal*, vol. 9, no. 3, 2005.
- [63] E. A. Al-Ammar *et al.*, "ABC algorithm based optimal sizing and placement of DGs in distribution networks considering multiple objectives," *Ain Shams Engineering Journal*, vol. 12, no. 1, 2021, doi: 10.1016/j.asej.2020.05.002.
- [64] M. Lalitha, N. Reddy, and V. Reddy, "Optimal DG placement for maximum loss reduction in radial distribution system using ABC algorithm," *International journal of reviews in computing*, vol. 3, 2010.

Appendices

Appendix A

Comparison of the Optimization Techniques

Approach	Algorithm	Objective	Advantage	Disadvantage	ref
Analytical	2/3 rule or Golden rule	Minimize energy losses	Very Simple	Just stable for uniformly distributed loads and very slow	[21]
Analytical	Sensitive factor analysis	Minimize the cost of losses at unity and lagging power factors	Reduces reactive power burden on the input side of the distribution system	Only considered unity and 0.9 lagging power factors	[22]
Analytical	Kalman filter	Minimize total power losses	Fast convergence compared with other analytical methods	Just Uses for simple network	[23]
Analytical	Iterative methods	Minimize active and reactive power losses	The size of DG units depends only on the base caseload flow	The proposed method is not applicable to unbalanced meshed networks	[24]
Classical (non – heuristic)	Dynamic Programming	Minimize the power loss with enhanced reliability and voltage profile	Financial and technical benefits	Effect of reactive power not considered	[25]
Classical (non – heuristic)	Ordinal Optimization	Minimize loss and maximize DG size	Guaranteed solution	High computational burden	[26]
Classical (non – heuristic)	Non-Linear and Mixed Integer Non-Linear Programming	Minimize energy losses by incorporating loss adjustment factors along with individual generation load factors	Maintaining voltage limits, feeder capacity maximum penetration limits, and discrete size of DG units	Suitable only for radial distribution system	[27]
Classical (non – heuristic)	Exhaustive Search	Minimize line losses for active and reactive parts together with improving bus voltage deviation	guarantee of optimality and simplicity	Thermal limits not considered	[28]
Classical (non – heuristic)	sequential quadratic programming	Minimize loss	deal with a wide class of distribution systems	quickly converge to local solutions	[29]
Classical (non – heuristic)	Continuation Power Flow	Improve voltage limit load ability	Reliability improvement	System losses not evaluated	[30]
Metaheuristic	Genetic Algorithm	minimize electrical network losses	evaluation of losses, voltage profile based on	for high-quality solutions accuracy is low	[32]

			power flow method, and reliability indices by analytical method	premature convergence	
Metaheuristic	Particle swarm optimization	Improve voltage stability margin and Minimize energy cost and losses	Effective analysis of voltage profile and stability	Complicated methodology	[33]
Metaheuristic	Bat Algorithm	Minimize loss	stochastic nature of solar irradiance is studied and losses reduced	effectiveness of reactive power not considered	[34]
Metaheuristic	Artificial bee colony	Minimize loss	Effective in handling complex problems	Converge towards local minima	[35]
Metaheuristic	Cuckoo search algorithm	Minimize real power loss and enhance voltage stability	considered alternate methods for the same objective	effectiveness of reactive power not considered	[36]
Metaheuristic	Oppositional krill herd algorithm	Minimize annual energy losses through RERs	RER-based DG are considered	Self-compared algorithm	[37]
Metaheuristic	Ant-lion optimization	minimize loss	perform better at lagging power factors	effectiveness of reactive power not considered	[38]
Metaheuristic	Firefly optimization	Minimize real and reactive power loss	Improved voltage profile	Only suitable for single-objective	[39]
Metaheuristic	Tabu Search	Minimize cost of power, energy losses, and total required reactive power	suitable for real and reactive power DG allocation	Thermal limits not considered	[40]
Metaheuristic	Archimedes optimization algorithm	reducing emissions and losses, improving the voltage profile	can easily incorporate constraints related to the problem	Complicated methodology	[41]
Metaheuristic	Intelligent water drop optimization	Minimize total line losses	Easy to implement and obtain feasible solution near optimum with less computation time	Problem with voltage deviation	[42]
Hybrid	Genetic-based tabu search	Minimize power loss	Harmonic power loss also reduced	Thermal limits not considered	[43]
Hybrid	genetic algorithm and imperialistic competitive algorithm	Minimize power loss, improve system voltage profile, and load balancing, and minimize cost	Transmission and distribution relief capacity for both utilities and customers	Applicable only for uniform and constant power systems	[44]

Hybrid	particle swarm optimization and shuffled frog leaping algorithm	Minimize power loss, improve system voltage profile, improve the transient stability	Fast convergence	Limited Handling of Constraints	[45]
Hybrid	genetic algorithm particle swarm optimization with fuzzy	minimize power losses, improve voltage regulation and stability, and system security	effective search and improved convergence	Integration and coordination among these algorithms are complex task	[46]
Hybrid	genetic algorithm and particle swarm optimization	minimize power losses, improve voltage regulation and stability, and system security	Less convergence time	DG penetration more than 30% will affects actual system performance	[47]
Hybrid	ant colony and artificial bee colony algorithm	Minimize power loss, total emissions and electrical cost	Stochastic nature of wind generation is enlightened	Complicated methodology	[48]
Hybrid	Evolutionary programming and particle swarm optimization	Minimize power loss	Less number of iterations and low computation time	Unsuitable for meshed networks	[49]
assorted	Pareto frontier differential evolution algorithm	Improve voltage stability, minimize power loss and network voltage variations	Non-dominated ranking methodology for finding the optimal solution	The effect of reactive power is ignored	[50]
assorted	Symbiotic organisms search algorithm	Minimize loss	handling complex problems	The high computational time required	[51]
assorted	Supervised Big Bang Big Crunch	Minimize power and energy losses	Applicable for balanced and unbalanced systems	Self-compared algorithm	[52]
assorted	Elephant herd optimization	reduces the overloading of the equipment, voltage swell & sag, active power, reactive power, and production cost of electricity	handling high-dimensional solution spaces, making it suitable for problems with a large number of decision variables	Lack of Benchmark Performance, low Convergence Speed	[53]
assorted	Quis oppositional swine influenza	Minimize network power losses, improve	Computationally efficient	Premature convergence	[54]

	model-based optimization with Quarantine	voltage stability and voltage regulation			
assorted	Modified honey bee mating	Minimize costs, emissions and losses	Performs better in case of single and multi-objective problem	Problem comparison with	[55]

Appendix B
IEEE 33 Network

from	to	R(ohm)	X(ohm)	P(KW)	Q(KW)	Imax(A)
1	2	0.0922	0.047	0	0	400
2	3	0.493	0.2511	100	60	400
3	4	0.366	0.1864	90	40	400
4	5	0.3811	0.1941	120	80	400
5	6	0.819	0.707	60	30	400
6	7	0.1872	0.6188	60	20	400
7	8	1.7114	1.2351	200	100	400
8	9	1.03	0.74	200	100	400
9	10	1.044	0.74	60	20	400
10	11	0.1966	0.065	60	20	400
11	12	0.3744	0.1238	45	30	400
12	13	1.468	1.155	60	35	400
13	14	0.5416	0.7129	60	35	400
14	15	0.591	0.526	120	80	400
15	16	0.7463	0.545	60	10	400
16	17	1.289	1.721	60	20	400
17	18	0.732	0.574	60	20	400
2	19	0.164	0.1565	90	40	400
19	20	1.5042	1.3554	90	40	400
20	21	0.4095	0.4784	90	40	400
21	22	0.7089	0.9373	90	40	400
3	23	0.4512	0.3083	90	40	400
23	24	0.898	0.7091	90	50	400
24	25	0.896	0.7011	420	200	400
6	26	0.203	0.1034	420	200	400
26	27	0.2842	0.1447	60	25	400
27	28	1.059	0.9337	60	25	400
28	29	0.8042	0.7006	60	20	400
29	30	0.5075	0.2585	120	70	400
30	31	0.9744	0.963	200	600	400
31	32	0.3105	0.3619	150	70	400
32	33	0.341	0.5302	210	100	400

Appendix C

IEEE 33 network parameters before installing PV

Bus number	Voltage magnitude (P.U)	Line from-to	Line load index	active power losses(KW)	Reactive power losses(KVAR)	Voltage stability index
1	1.000000	1-2	40.932508	11.585876	5.906032	0.011518
2	0.997112	2-3	36.284642	48.680369	24.794403	0.054079
3	0.983444	3-4	26.999017	20.009593	10.190678	0.030277
4	0.975943	4-5	25.651946	18.807924	9.579160	0.030171
5	0.968525	5-6	25.028541	38.478331	33.216337	0.074777
6	0.950068	6-7	11.752961	1.939375	6.410710	0.014767
7	0.946547	7-8	9.598614	10.443779	8.534545	0.051891
8	0.934104	8-9	7.415941	4.248455	3.052288	0.026880
9	0.927784	9-10	6.798442	3.618933	2.565144	0.025114
10	0.921922	10-11	6.177784	0.562743	0.186054	0.003758
11	0.921055	11-12	5.647373	0.895550	0.296125	0.006555
12	0.919543	12-13	4.961226	2.709975	2.132167	0.026630
13	0.913380	13-14	4.271500	0.741141	0.975553	0.009983
14	0.911095	14-15	2.861155	0.362854	0.322946	0.006242
15	0.909671	15-16	2.260888	0.286110	0.208937	0.006055
16	0.908292	16-17	1.626619	0.255791	0.341518	0.008981
17	0.906248	17-18	0.991900	0.054014	0.042355	0.002700
18	0.905636	2-19	3.617126	0.160928	0.153569	0.002118
19	0.996584	19-20	2.715747	0.832042	0.749734	0.014307
20	0.993006	20-21	1.811120	0.100742	0.117692	0.002836
21	0.992302	21-22	0.905851	0.043627	0.057684	0.002568
22	0.991665	3-23	8.510116	2.450763	1.674579	0.012694
23	0.980313	23-24	7.556206	3.845433	3.036522	0.023191
24	0.974596	24-25	3.783634	0.962028	0.752765	0.011659
25	0.971747	6-26	13.064249	2.598521	1.323581	0.008102
26	0.948139	26-27	12.491510	3.325946	1.693400	0.010782
27	0.945577	27-28	11.922786	11.290487	9.954606	0.047785
28	0.934142	28-29	11.390992	7.826154	6.817960	0.034864
29	0.925928	29-30	10.112128	3.892086	1.982471	0.015302
30	0.922372	30-31	4.667611	1.592164	1.573537	0.017956
31	0.918213	31-32	3.024309	0.212998	0.248257	0.003982
32	0.917298	32-33	0.717234	0.013156	0.020456	0.001236
33	0.917014					

Appendix D

IEEE 33 network parameters after installing PV randomly

	Voltage profile			
	original	One PV	two PV	three PV
1	1.000000	1.000000	1.000000	1.000000
2	0.997112	0.998681	0.998582	0.998892
3	0.983444	0.993416	0.992783	0.994758
4	0.975943	0.992144	0.991115	0.991790
5	0.968525	0.991223	0.989779	0.989096
6	0.950068	0.986899	0.984581	0.980960
7	0.946547	0.983519	0.982015	0.978535
8	0.934104	0.971574	0.976303	0.974036
9	0.927784	0.965508	0.974471	0.973040
10	0.921922	0.959883	0.973133	0.972553
11	0.921055	0.959051	0.973105	0.972686
12	0.919543	0.957600	0.973183	0.973071
13	0.913380	0.951688	0.973283	0.974373
14	0.911095	0.949496	0.973312	0.972233
15	0.909671	0.948130	0.971980	0.970899
16	0.908292	0.946807	0.970690	0.969608
17	0.906248	0.944847	0.968778	0.967694
18	0.905636	0.944260	0.968206	0.967121
19	0.996584	0.998154	0.998054	0.998365
20	0.993006	0.994582	0.994482	0.994794
21	0.992302	0.993879	0.993779	0.994091
22	0.991665	0.993242	0.993142	0.993454
23	0.980313	0.990317	0.989682	0.994785
24	0.974596	0.984659	0.984020	0.989153
25	0.971747	0.981839	0.981199	0.986346
26	0.948139	0.985047	0.982724	0.980492
27	0.945577	0.982586	0.980257	0.979971
28	0.934142	0.971604	0.969249	0.976258
29	0.925928	0.963716	0.961340	0.968408
30	0.922372	0.960301	0.957917	0.965010
31	0.918213	0.956308	0.953914	0.961037
32	0.917298	0.955430	0.953033	0.960162
33	0.917014	0.955157	0.952760	0.959892

	Active losses				Reactive losses			
	original	One PV	two PV	three PV	original	One PV	two PV	three PV
1-2	11.585876	4.596857	3.753421	3.583608	5.906032	2.343300	1.913349	1.826785
2-3	48.680369	18.919147	15.360926	19.141894	24.794403	9.636101	7.823790	9.749553
3-4	20.009593	9.010023	7.950883	6.398666	10.190678	4.588711	4.049302	3.258774
4-5	18.807924	8.991116	8.101259	5.958971	9.579160	4.579312	4.126094	3.034994

5-6	38.478331	18.894184	17.094579	12.306772	33.216337	16.310364	14.756859	10.623794
6-7	1.939375	3.974035	0.556026	0.350515	6.410710	13.136393	1.837975	1.158646
7-8	10.443779	41.624940	5.828279	2.185727	8.534545	30.040296	4.206209	1.577417
8-9	4.248455	29.352163	4.724762	1.359274	3.052288	21.087962	3.394489	0.976566
9-10	3.618933	31.248137	5.333214	1.567260	2.565144	22.149062	3.780247	1.110893
10-11	0.562743	6.174762	1.116789	0.341328	0.186054	2.041503	0.369233	0.112850
11-12	0.895550	12.155958	2.279345	0.709496	0.296125	4.019518	0.753694	0.234604
12-13	2.709975	49.844720	9.839709	3.201227	2.132167	39.217065	7.741733	2.518676
13-14	0.741141	19.222702	3.995940	1.370482	0.975553	25.302555	5.259796	1.803945
14-15	0.362854	22.879906	5.255477	2.012383	0.322946	20.363503	4.677463	1.791055
15-16	0.286110	30.196459	7.295888	2.964228	0.208937	22.051548	5.327963	2.164685
16-17	0.255791	54.421077	13.779757	5.897609	0.341518	72.659948	18.397953	7.874155
17-18	0.054014	0.036926	0.041613	0.044796	0.042355	0.028955	0.032631	0.035127
2-19	0.160928	0.160431	0.160368	0.436283	0.153569	0.153095	0.153034	0.416331
19-20	0.832042	0.829468	0.829139	5.029053	0.749734	0.747414	0.747118	4.531564
20-21	0.100742	0.100430	0.100390	1.697948	0.117692	0.117328	0.117281	1.983634
21-22	0.043627	0.043492	0.043475	3.593277	0.057684	0.057505	0.057482	4.750993
3-23	2.450763	3.114823	3.106866	3.124651	1.674579	2.128324	2.122888	2.135040
23-24	3.845433	5.035495	5.022612	5.051410	3.036522	3.976247	3.966073	3.988814
24-25	0.962028	1.260284	1.257049	1.264281	0.752765	0.986144	0.983613	0.989271
6-26	2.598521	2.295908	1.686015	1.251086	1.323581	1.169443	0.858788	0.637253
26-27	3.325946	2.934994	2.405020	1.691843	1.693400	1.494348	1.224512	0.861399
27-28	11.290487	9.951536	9.186857	6.141538	9.954606	8.774078	8.099876	5.414876
28-29	7.826154	6.893250	7.244658	4.636181	6.817960	6.005236	6.311375	4.038931
29-30	3.892086	3.418494	4.831582	2.808402	1.982471	1.741243	2.461013	1.430487
30-31	1.592164	1.465303	7.317685	2.423350	1.573537	1.448160	7.232071	2.394998
31-32	0.212998	0.196015	2.950931	1.119740	0.248257	0.228463	3.439426	1.305101
32-33	0.013156	0.012107	4.388271	1.969720	0.020456	0.018824	6.823054	3.062597

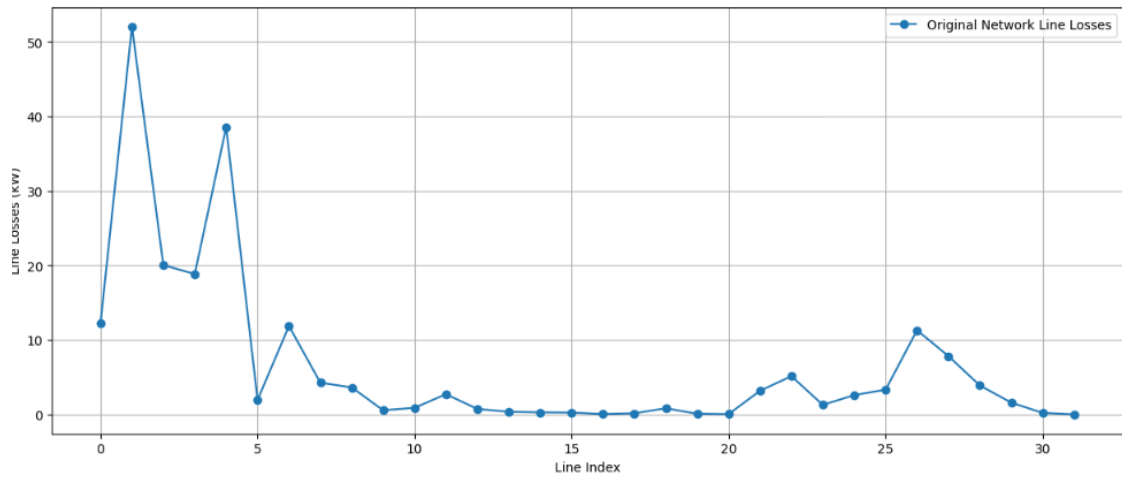
	line flow limit index				Voltage stability index			
	original	One PV	two PV	three PV	original	One PV	two PV	three PV
1-2	40.932508	25.783058	23.297940	22.764817	0.011518	0.005419	0.004636	0.004465
2-3	36.284642	22.620213	20.382366	22.752984	0.054079	0.021737	0.017583	0.028720
3-4	26.999017	18.117229	17.019097	15.267698	0.030277	0.004317	0.001216	0.009636
4-5	25.651946	17.736041	16.835505	14.438947	0.030171	0.002844	-0.00037	0.008448
5-6	25.028541	17.538461	16.682327	14.154662	0.074777	0.018057	0.009649	0.028472
6-7	11.752961	16.824117	6.293092	4.996543	0.014767	0.004400	0.007032	0.008897
7-8	9.598614	18.008204	6.738508	4.126592	0.051891	-0.06095	-0.01025	0.009166
8-9	7.415941	19.492647	7.820612	4.194733	0.026880	-0.04399	-0.01349	-0.00190
9-10	6.798442	19.977042	8.253033	4.473935	0.025114	-0.04685	-0.01583	-0.00410
10-11	6.177784	20.463868	8.702884	4.811315	0.003758	-0.00986	-0.00392	-0.00171
11-12	5.647373	20.806357	9.009619	5.026625	0.006555	-0.01929	-0.00798	-0.00380
12-13	4.961226	21.277271	9.453605	5.392182	0.026630	-0.07346	-0.03012	-0.01391
13-14	4.271500	21.753906	9.918352	5.808536	0.009983	-0.02587	-0.01083	-0.00502
14-15	2.861155	22.719708	10.888842	6.737998	0.006242	-0.03267	-0.01588	-0.00962
15-16	2.260888	23.226868	11.416993	7.277270	0.006055	-0.04297	-0.02164	-0.01378
16-17	1.626619	23.726106	11.938883	7.810537	0.008981	-0.07556	-0.03866	-0.02513
17-18	0.991900	0.820120	0.870615	0.903303	0.002700	0.001846	0.002080	0.002239
2-19	3.617126	3.611537	3.610822	5.955680	0.002118	0.002112	0.002111	-0.00191
19-20	2.715747	2.711542	2.711004	6.676666	0.014307	0.014263	0.014257	-0.02284
20-21	1.811120	1.808314	1.807955	7.435404	0.002836	0.002827	0.002826	-0.00724
21-22	0.905851	0.904446	0.904267	8.220954	0.002568	0.002560	0.002559	-0.01490
3-23	8.510116	9.594039	9.581778	9.609164	0.012694	0.014238	0.014202	0.014282
23-24	7.556206	8.646738	8.635669	8.660391	0.023191	0.026501	0.026434	0.026583
24-25	3.783634	4.330615	4.325053	4.337476	0.011659	0.013341	0.013307	0.013383

6-26	13.064249	12.280008	10.523304	9.064941	0.008102	0.007277	-0.00040	0.002082
26-27	12.491510	11.734402	10.622253	8.909172	0.010782	0.009665	-0.00111	0.002365
27-28	11.922786	11.193514	10.754863	8.793463	0.047785	0.043075	0.002994	0.016036
28-29	11.390992	10.690533	10.959640	8.767333	0.034864	0.031330	0.000202	0.010250
29-30	10.112128	9.476954	11.266682	8.589759	0.015302	0.013559	-0.00620	0.000044
30-31	4.667611	4.477798	10.006624	5.758492	0.017956	0.016537	-0.02096	-0.00933
31-32	3.024309	2.901235	11.256881	6.934212	0.003982	0.003665	-0.00818	-0.00456
32-33	0.717234	0.688033	13.099023	8.775956	0.001236	0.001137	-0.01167	-0.00782

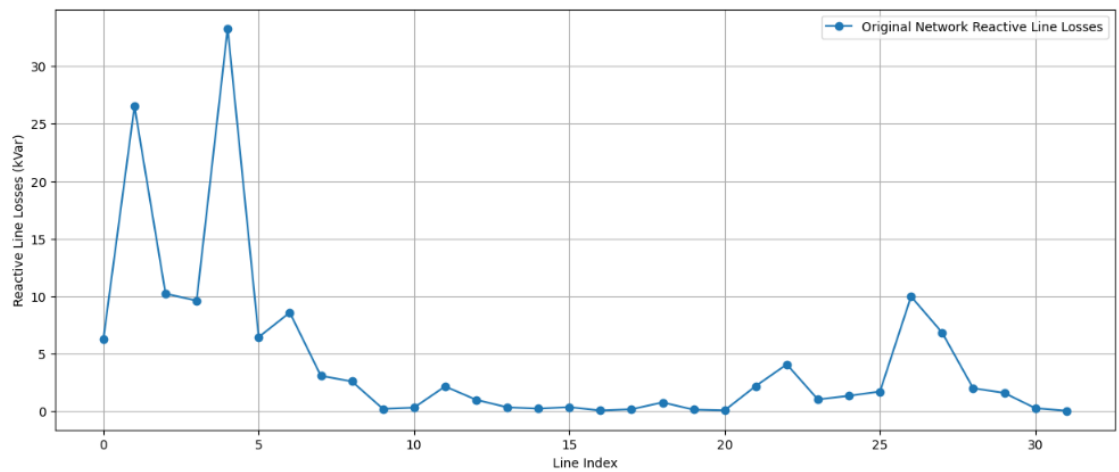
Appendix E

IEEE 33 Bus System Parameter Curves Before Installing PV System

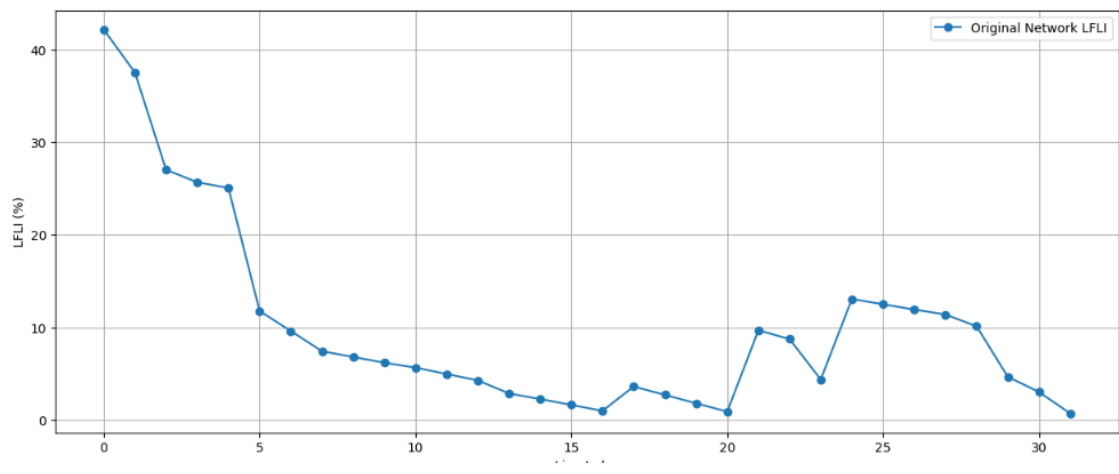
IEEE 33 Active Power Losses in Each Line



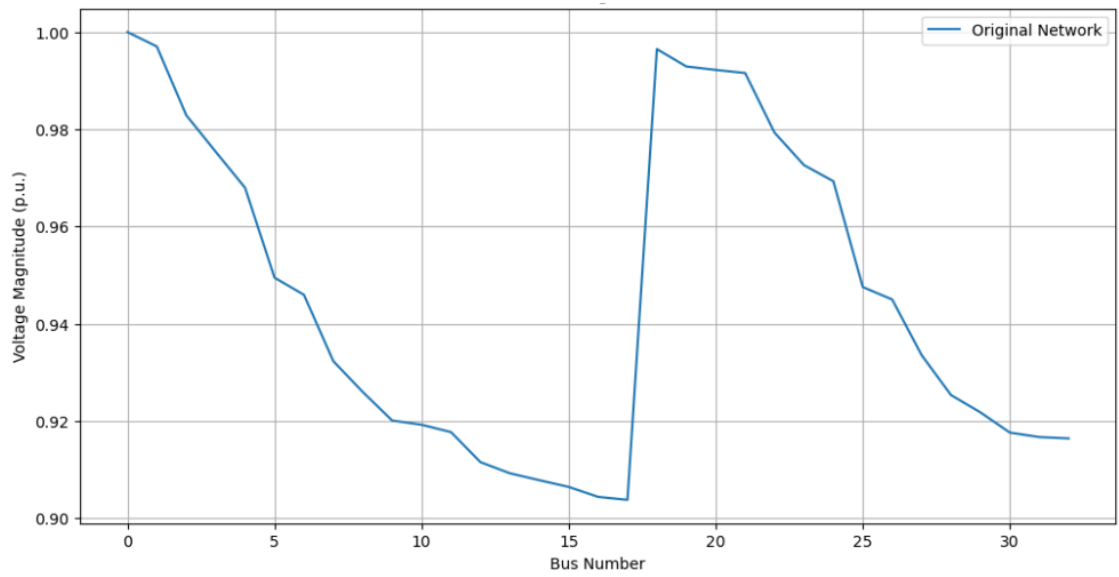
IEEE 33 Reactive Power Losses in Each Line



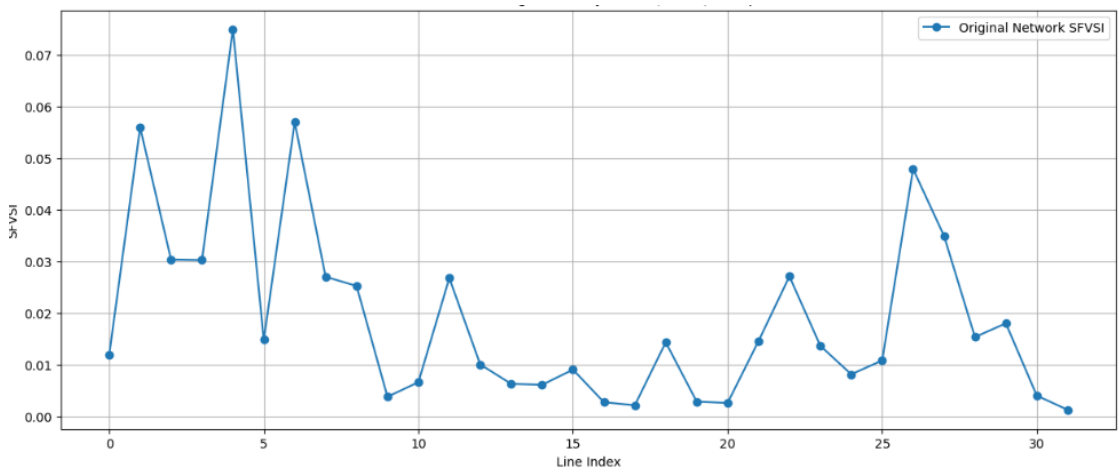
IEEE 33 Line Flow Limit Index in Each Line



IEEE 33 Voltage Profile in Each Bus



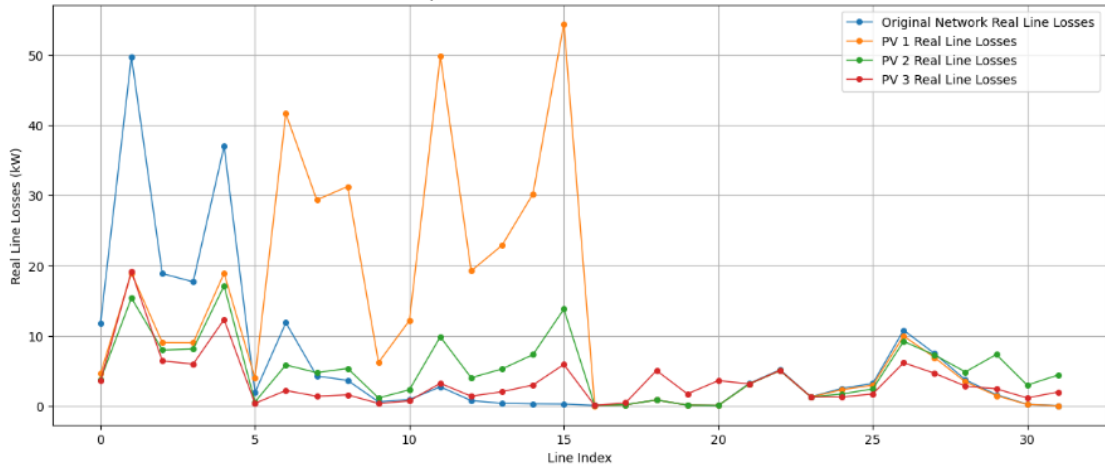
IEEE 33 Voltage Stability Index in Each Bus



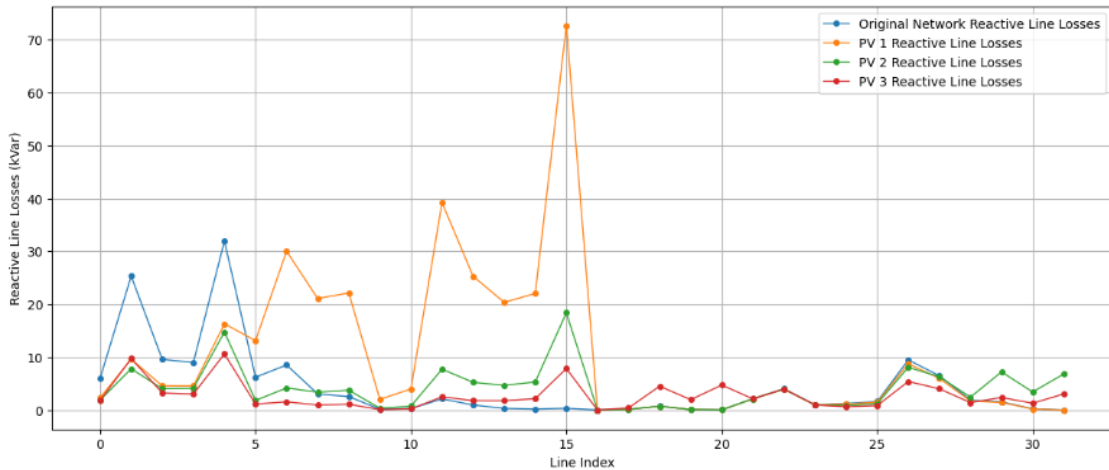
Appendix F

IEEE 33 Bus System Parameter Curves After Installing PV System random at weak buses

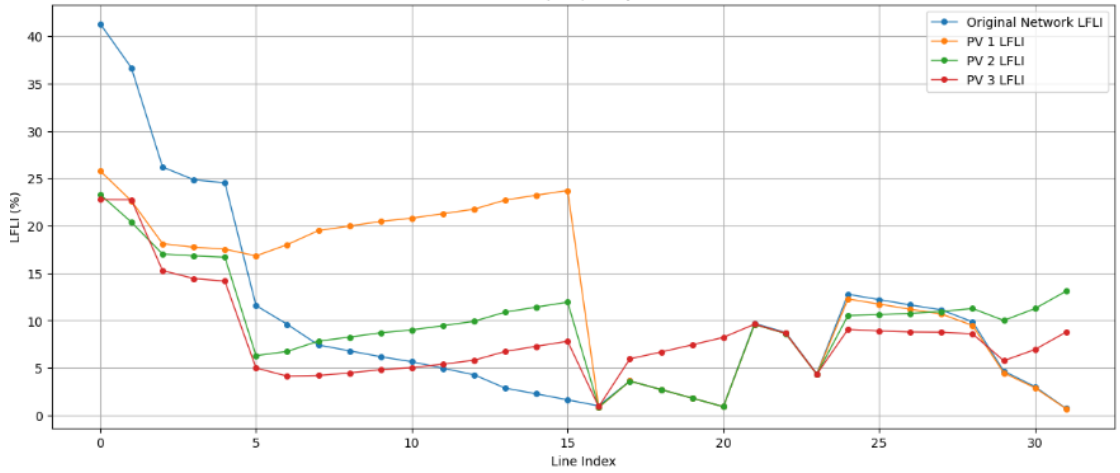
Comparison for IEEE 33 active power losses in each line for random PV-distributed at weak buses



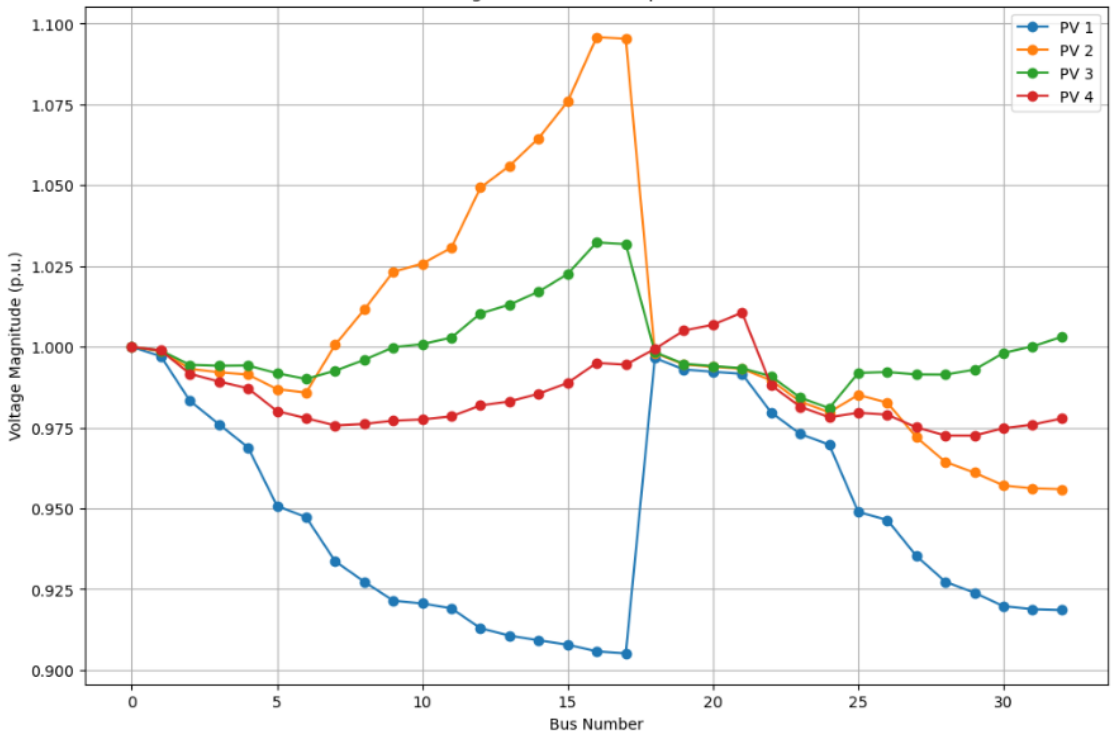
Comparison for IEEE 33 reactive power losses in each line for random PV-distributed at weak buses



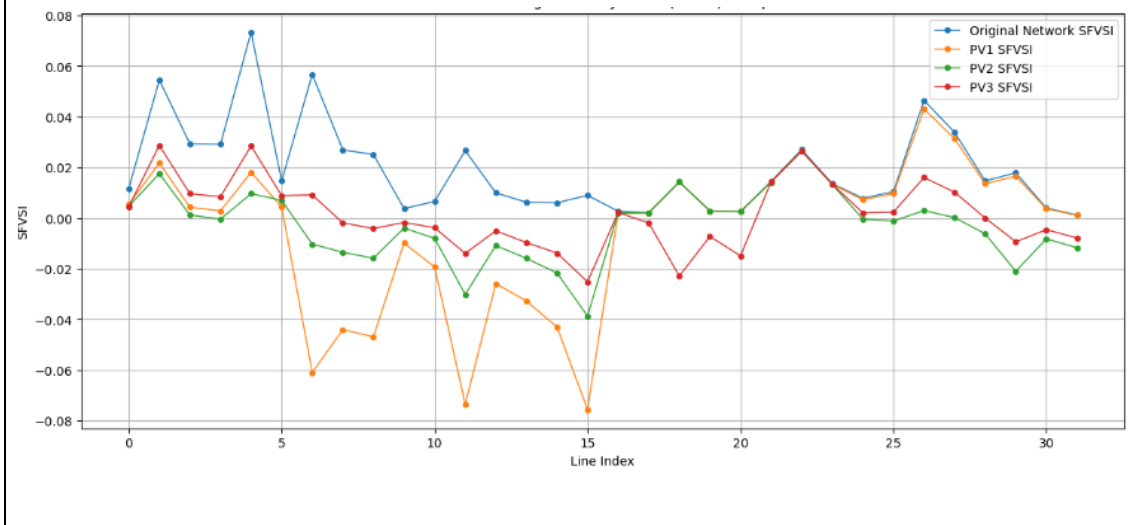
Comparison for IEEE 33-line flow limit index in each line for random PV-distributed at weak buses



Comparison for IEEE 33 Voltage Profile in Each bus for random PV-distributed at weak buses



Comparison for IEEE 33 voltage stability index in each bus for random PV-distributed at weak buses



Appendix G

IEEE 33 Conventional ABC python code

Import Libraries

```
import pandapower as pp
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
import numba
```

Network Data

```
# Read data from Excel for Buses
excel_file = 'IEEE 33 Data.xlsx'
buses = pd.read_excel(excel_file, sheet_name='Buses', engine='openpyxl')
# Display the DataFrame to ensure it's loaded correctly
#buses

# Read data from Excel for Lines
lines = pd.read_excel(excel_file, sheet_name='Lines', engine='openpyxl')
# Display the DataFrame to ensure it's loaded correctly
#lines
```

Building the Network

```
def build_network(buses_df, lines_df):
    # Create an empty pandapower network
    net = pp.create_empty_network()
    # Create buses and attach loads, create slack bus if necessary
    for _, row in buses_df.iterrows():
        # Create a bus
```

```

    bus = pp.create_bus(net, vn_kv=12.66, name=row['name'], zone=1.0,
max_vm_pu=1.1, min_vm_pu=0.9)

    # Create a load at the bus

    pp.create_load(net, bus=bus, p_mw=row['p_kw']/1000,
q_mvar=row['q_kvar']/1000)

    # Check if the bus should be a slack bus and create it if so

    if row.get('is_slack', False):

        pp.create_ext_grid(net, bus=bus, vm_pu=1.0)

        slack_bus_created = True

        print(f"Slack bus created at Bus {bus}, Name: {row['name']}")

    # Check if at least one slack bus was created, print a message if not
    if 'slack_bus_created' not in locals():

        print("No slack bus was created from the Excel data!")

    # Create lines between buses

    for _, row in lines_df.iterrows():

        # Create a line between two buses

        pp.create_line_from_parameters(net, from_bus=row['from_bus']-1,
to_bus=row['to_bus']-1,
length_km=1, r_ohm_per_km=row['r'],
x_ohm_per_km=row['x'],
c_nf_per_km=0, max_i_ka=1)

    return net

newTest = build_network(buses, lines)

def run_power_flow_analysis(net):

    try:

        pp.runpp(net, algorithm='bfs', max_iteration=100, numba= True)

        print("Power flow executed successfully!")

```

```

    real_power_loss = net.res_line.pl_mw.sum() * 1000 # Convert from MW to kW
    reactive_power_loss = net.res_line.ql_mvar.sum() * 1000 # Convert from MVar to
kVar

    return real_power_loss, reactive_power_loss, net.converged

except Exception as e:

    print("Error running power flow:", e)

    return None, None, False

run_power_flow_analysis(newTest)

fitness = newTest.res_line.pl_mw.sum() * 1000

starting_loss = fitness

# Before starting the ABC algorithm, calculate the base losses

net = build_network(buses, lines)

pp.runpp(net) # Run the initial power flow to determine the base losses

base_real_loss = net.res_line.pl_mw.sum() * 1000 # Convert from MW to kW

base_reactive_loss = net.res_line.ql_mvar.sum() * 1000 # Convert from MVar to kVar

Fitness Function

def run_load_flow_and_calculate_fitness(net, pv_config, base_real_loss,
base_reactive_loss):

    # Backup original Loads

    original_loads = net.load.p_mw.copy()

    # Subtract PV sizes from the loads at corresponding buses

    for bus_id, pv_size in pv_config.items():

        load_indices = net.load[net.load.bus == bus_id].index

        for load_index in load_indices:

            net.load.at[load_index, 'p_mw'] = max(net.load.at[load_index, 'p_mw'] - pv_size
/ 1000, 0)

```

```

# Run the power flow analysis

try:

    pp.runpp(net)

    # Calculate real and reactive power losses in kW and kVar

    real_power_loss = net.res_line.pl_mw.sum() * 1000 # Convert from MW to kW

    reactive_power_loss = net.res_line.ql_mvar.sum() * 1000 # Convert from MVar to
kVar

    # Calculate SFVSI for each line using the provided formula

    vs = net.res_bus.loc[net.line['from_bus'], 'vm_pu'].values

    vr = net.res_bus.loc[net.line['to_bus'], 'vm_pu'].values

    sfvsi = (4 * vr / vs) * (1 - (vr / vs))

    max_SFVSI = sfvsi.max()

    # Calculate IVR for each bus

    net.res_bus['IVR'] = (net.bus['max_vm_pu'] - net.res_bus['vm_pu']) /
net.bus['max_vm_pu']

    max_IVR = net.res_bus['IVR'].max()

    # Calculate LFLI for the network

    current_loading_percent = np.abs(net.res_line.loading_percent) / 100 # Convert
percentage to a decimal

    line_capacity_percent = np.ones_like(current_loading_percent) # Assume line
capacity is 100% for all lines

    lfli = np.max(current_loading_percent / line_capacity_percent) # Compute LFLI

    # Combine all metrics into a fitness value

    weights = {'w1': 0.35, 'w2': 0.15, 'w3': 0.15, 'w4': 0.25, 'w5': 0.1}

    fitness_value = (

        weights['w1'] * real_power_loss / base_real_loss +

        weights['w2'] * reactive_power_loss / base_reactive_loss +

```

```

        weights['w3'] * max_SFVSI +
        weights['w4'] * max_IVR +
        weights['w5'] * Ifli
    )
except Exception as e:
    print("Error running load flow:", e)
    fitness_value = float("inf") # Assign a high cost to an invalid configuration
# Restore original Loads
net.load.p_mw = original_loads
return fitness_value

```

artificial Bees Algorithm

Algorithm Parameters

```

# Constants and Parameters
PV_SIZE_RANGE = (100, 2000)
NUMBER_OF_PV_LOCATIONS = 3
NUMBER_OF_BEES = 50
NUMBER_OF_ITERATIONS = 100
PENETRATION_FACTOR = 1
SCOUT_BEE_LIMIT = 10

```

Algorithm Functions

```

def initialize_population(net, num_bees, num_pv_locations, pv_size_range,
penetration_factor):
    population = []
    total_pv_capacity_kw = calculate_total_pv_capacity(net, penetration_factor)
    for _ in range(num_bees):
        remaining_capacity_kw = total_pv_capacity_kw

```

```

bee_solution = {}

# Randomly choose bus locations and PV sizes within the total capacity
while remaining_capacity_kw > 0 and len(bee_solution) < num_pv_locations:

    bus_id = random.choice([bus for bus in net.bus.index if bus not in bee_solution])

    max_pv_size = min(remaining_capacity_kw, pv_size_range[1])

    pv_size = random.uniform(pv_size_range[0], max_pv_size)

    bee_solution[bus_id] = pv_size

    remaining_capacity_kw -= pv_size

population.append(bee_solution)

return population

def employed_bee_phase(net, population, pv_size_range, base_real_loss,
base_reactive_loss):

    new_population = []

    for bee_solution in population:

        new_solution = {}

        for bus_id, pv_size in bee_solution.items():

            # Generate a new PV size using a random neighbor's PV size

            partner_solution = random.choice(population)

            partner_pv_size = partner_solution.get(bus_id, pv_size_range[0])

            phi = random.uniform(-1, 1) # Random number between -1 and 1

            new_pv_size = pv_size + phi * (partner_pv_size - pv_size)

            # Ensure new PV size is within bounds

            new_pv_size = max(min(new_pv_size, pv_size_range[1]), pv_size_range[0])

            new_solution[bus_id] = new_pv_size

```

```

    # Evaluate the fitness of the new solution

    fitness = run_load_flow_and_calculate_fitness(net, new_solution, base_real_loss,
base_reactive_loss)

    current_fitness = run_load_flow_and_calculate_fitness(net, bee_solution,
base_real_loss, base_reactive_loss)

    # Choose the solution with better fitness

    if fitness < current_fitness:

        new_population.append(new_solution)

    else:

        new_population.append(bee_solution)

    return new_population

def onlooker_bee_phase(net, population, fitnesses, pv_size_range, base_real_loss,
base_reactive_loss):

    new_population = []

    # Calculate the total fitness to normalize probabilities

    total_fitness = sum(fitnesses)

    # Handle the case where the total fitness is 0 to avoid division by zero

    if total_fitness == 0:

        total_fitness = 1

    # Calculate the probability of selecting each solution

    selection_probabilities = [f / total_fitness for f in fitnesses]

    # Iterate over the full population size

    for _ in range(len(population)):

        # Select a solution based on fitness

        selected_index = np.random.choice(range(len(population)),
p=selection_probabilities)

        selected_solution = population[selected_index]

```

```

# Perturb the selected solution

partner_solution = random.choice(population)

new_solution = {}

for bus_id, pv_size in selected_solution.items():

    phi = random.uniform(-1, 1) # Random number between -1 and 1

    partner_pv_size = partner_solution.get(bus_id, pv_size)

    # Calculate new PV size and ensure it is within the specified size range

    new_pv_size = pv_size + phi * (partner_pv_size - pv_size)

    new_pv_size = max(min(new_pv_size, pv_size_range[1]), pv_size_range[0])

    new_solution[bus_id] = new_pv_size

# Evaluate the fitness of the new solution

fitness = run_load_flow_and_calculate_fitness(net, new_solution, base_real_loss,
base_reactive_loss)

# Choose the solution with better fitness

if fitness < fitnesses[selected_index]:

    new_population.append(new_solution)

    fitnesses[selected_index] = fitness # Update fitness value

else:

    new_population.append(selected_solution)

# Ensure new_population has the same size as the original population

while len(new_population) < len(population):

    new_population.append(random.choice(population))

return new_population

def scout_bee_phase(net, population, trials, limit, pv_size_range, num_pv_locations):

    for i, trial_count in enumerate(trials):

        if trial_count > limit: # If a solution hasn't improved for a certain number of trials

```

```

# Generate a new random solution
new_solution = {}

remaining_capacity_kw = calculate_total_pv_capacity(net,
PENETRATION_FACTOR)

while remaining_capacity_kw > 0 and len(new_solution) < num_pv_locations:
    bus_id = random.choice(net.bus.index)

    pv_size = random.uniform(pv_size_range[0], pv_size_range[1])

    new_solution[bus_id] = pv_size

    remaining_capacity_kw -= pv_size

    population[i] = new_solution

    trials[i] = 0 # Reset the trial count for this new solution

return population

def abc_algorithm_full(net, num_iterations, pv_size_range, num_pv_locations,
penetration_factor, num_bees, scout_bee_limit, base_real_loss, base_reactive_loss):

    # Initialize bee population

    population = initialize_population(net, num_bees, num_pv_locations, pv_size_range,
penetration_factor)

    # Initialize the best solution and its fitness

    best_solution = None

    best_fitness = float('inf')

    # Initialize a list to track the best fitness value in each iteration

    best_fitness_values = []

    # Initialize a list to track all fitness values in each iteration

    iteration_fitness_values = []

    # Initialize trial counters for scout bees

    trials = [0] * len(population)

```

```

for i in range(num_iterations):

    # Employed Bee Phase

    population = employed_bee_phase(net, population, pv_size_range, base_real_loss,
base_reactive_loss)

    # Onlooker Bee Phase

    fitnesses = [run_load_flow_and_calculate_fitness(net, sol, base_real_loss,
base_reactive_loss) for sol in population]

    population = onlooker_bee_phase(net, population, fitnesses, pv_size_range,
base_real_loss, base_reactive_loss)

    # Scout Bee Phase

    population = scout_bee_phase(net, population, trials, scout_bee_limit,
pv_size_range, num_pv_locations)

    # Update fitnesses after scout bee phase

    new_fitnesses = [run_load_flow_and_calculate_fitness(net, sol, base_real_loss,
base_reactive_loss) for sol in population]

    iteration_fitness_values.append(new_fitnesses)

    # Find the best solution and its fitness in the current iteration

    for idx, fitness in enumerate(new_fitnesses):

        if fitness < best_fitness:

            best_fitness = fitness

            best_solution = population[idx]

    # If there is no improvement, increment the trial counter

    if fitness > fitnesses[idx]:

        trials[idx] += 1

    else:

        trials[idx] = 0 # Reset the trial counter if there was an improvement

    # Store the best fitness of this iteration

```

```

    best_fitness_values.append(best_fitness)

return best_solution, best_fitness_values, iteration_fitness_values

best_solution, best_fitness_values, iteration_fitness_values = abc_algorithm_full(

    net=newTest,

    num_iterations=NUMBER_OF_ITERATIONS,

    pv_size_range=PV_SIZE_RANGE,

    num_pv_locations=NUMBER_OF_PV_LOCATIONS,

    penetration_factor=PENETRATION_FACTOR,

    num_bees=NUMBER_OF_BEES,

    scout_bee_limit=SCOUT_BEE_LIMIT,

    base_real_loss=base_real_loss, # Include the base real loss

    base_reactive_loss=base_reactive_loss # Include the base reactive loss

)

```

Appendix H

IEEE 33 Network Parameters After Installing PV Using Conventional ABC

	Voltage profile			
	original	One PV	two PV	three PV
1	1.000000	1.000000	1.000000	1.000000
2	0.997025	0.997819	0.998189	0.998550
3	0.982893	0.987933	0.990282	0.992576
4	0.975383	0.983562	0.984998	0.989543
5	0.967957	0.979402	0.979886	0.986779
6	0.949479	0.968612	0.966868	0.978852
7	0.945954	0.965162	0.965246	0.977604
8	0.932298	0.951796	0.962148	0.976567
9	0.925966	0.945599	0.962104	0.977737
10	0.920092	0.939851	0.962571	0.979423
11	0.919223	0.939001	0.962784	0.979844
12	0.917708	0.937519	0.963324	0.980776
13	0.911532	0.931477	0.957448	0.975007
14	0.909242	0.929236	0.955269	0.972867
15	0.907815	0.927840	0.953911	0.971535
16	0.906433	0.926488	0.952596	0.970244
17	0.904385	0.924485	0.950648	0.968331
18	0.903772	0.923885	0.950065	0.967759
19	0.996497	0.997291	0.997662	0.998023
20	0.992919	0.993717	0.994088	0.994450
21	0.992215	0.993013	0.993384	0.993747
22	0.991577	0.992376	0.992748	0.993111
23	0.979307	0.984366	0.989778	0.991028
24	0.972636	0.977729	0.989387	0.988506
25	0.969310	0.974422	0.986118	0.985235
26	0.947549	0.966722	0.964975	0.977980
27	0.944985	0.964212	0.962460	0.976895
28	0.933543	0.953011	0.951237	0.970938
29	0.925324	0.944964	0.943175	0.966896
30	0.921765	0.941481	0.939685	0.963493
31	0.917603	0.937407	0.935603	0.959513
32	0.916688	0.936510	0.934705	0.958637
33	0.916404	0.936233	0.934427	0.958366

	Active losses				Reactive losses			
	original	One PV	two PV	three PV	original	One PV	two PV	three PV
1-2	11.585876	3.764012	3.971740	3.314407	5.906032	1.918748	2.024640	1.689557
2-3	48.680369	14.620754	15.238450	13.257945	24.794403	7.446798	7.761409	6.752677
3-4	20.009593	6.510835	6.221882	6.634649	10.190678	3.315901	3.168740	3.378958
4-5	18.807924	6.360984	5.960933	6.063556	9.579160	3.239745	3.035993	3.088261
5-6	38.478331	13.465288	12.496906	12.362345	33.216337	11.623881	10.787928	10.671768
6-7	1.939375	1.789382	0.572853	0.455113	6.410710	5.914903	1.893596	1.504402
7-8	10.443779	9.627728	2.323074	1.860108	8.534545	7.867677	1.898391	1.520061
8-9	4.248455	3.913394	0.679010	0.704196	3.052288	2.811565	0.487833	0.505927
9-10	3.618933	3.332762	0.586207	0.716703	2.565144	2.362303	0.415511	0.508008
10-11	0.562743	0.518154	0.101599	0.146031	0.186054	0.171312	0.033591	0.048281
11-12	0.895550	0.824464	0.175451	0.288360	0.296125	0.272619	0.058015	0.095350

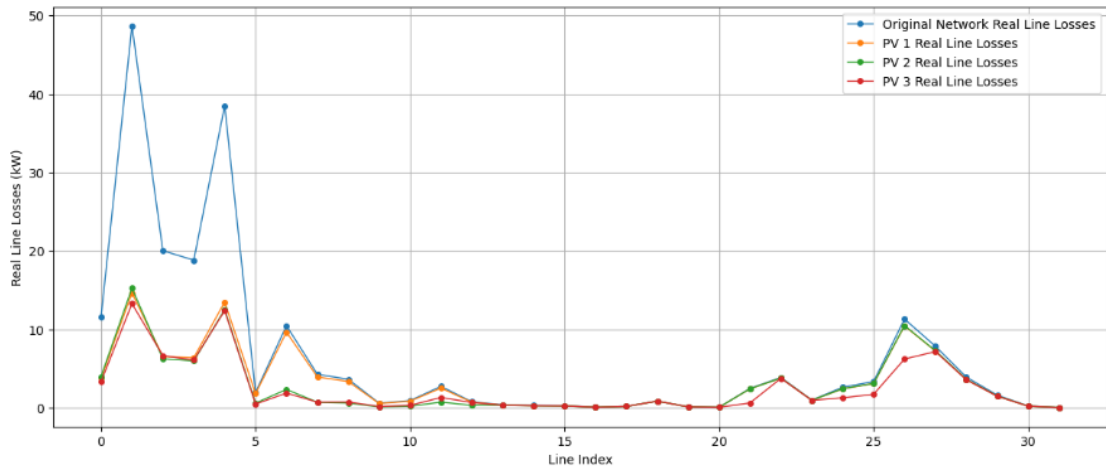
12-13	2.709975	2.494329	0.701684	1.292529	2.132167	1.962500	0.552075	1.016942
13-14	0.741141	0.682086	0.298812	0.650393	0.975553	0.897820	0.393322	0.856103
14-15	0.362854	0.333873	0.317614	0.318324	0.322946	0.297152	0.282682	0.283314
15-16	0.286110	0.263226	0.250390	0.250951	0.208937	0.192226	0.182852	0.183262
16-17	0.255791	0.235304	0.223814	0.224316	0.341518	0.314164	0.298824	0.299494
17-18	0.054014	0.049686	0.047259	0.047365	0.042355	0.038961	0.037058	0.037141
2-19	0.160928	0.160419	0.160451	0.160351	0.153569	0.153083	0.153114	0.153018
19-20	0.832042	0.829405	0.829571	0.829051	0.749734	0.747357	0.747508	0.747039
20-21	0.100742	0.100422	0.100442	0.100379	0.117692	0.117319	0.117342	0.117269
21-22	0.043627	0.043489	0.043498	0.043470	0.057684	0.057501	0.057512	0.057476
3-23	2.450763	2.400864	2.403984	0.597502	1.674579	1.640484	1.642616	0.408267
23-24	3.845433	3.767014	3.771918	3.732765	3.036522	2.974599	2.978471	2.947554
24-25	0.962028	0.942353	0.943583	0.933761	0.752765	0.737370	0.738333	0.730647
6-26	2.598521	2.397647	2.409576	1.251810	1.323581	1.221265	1.227341	0.637621
26-27	3.325946	3.068302	3.083599	1.700017	1.693400	1.562221	1.570010	0.865561
27-28	11.290487	10.414083	10.466111	6.198627	9.954606	9.181898	9.227770	5.465211
28-29	7.826154	7.218018	7.254117	7.147505	6.817960	6.288166	6.319615	6.226737
29-30	3.892086	3.589295	3.607267	3.554190	1.982471	1.828242	1.837396	1.810361
30-31	1.592164	1.467684	1.475069	1.453258	1.573537	1.450512	1.457812	1.436256
31-32	0.212998	0.196333	0.197322	0.194402	0.248257	0.228834	0.229987	0.226584
32-33	0.013156	0.012127	0.012188	0.012007	0.020456	0.018855	0.018950	0.018669

	line flow limit index				Voltage stability index			
	original	One PV	two PV	three PV	original	One PV	two PV	three PV
1-2	40.932508	23.330789	23.965932	21.893077	0.011518	0.005269	0.005665	0.004427
2-3	36.284642	19.885238	20.300947	18.935815	0.054079	0.020978	0.023092	0.016486
3-4	26.999017	15.400938	15.055311	15.546685	0.030277	0.005115	0.006711	0.011899
4-5	25.651946	14.918049	14.441324	14.565104	0.030171	0.003710	0.005383	0.010836
5-6	25.028541	14.805915	14.263584	14.186584	0.074777	0.017373	0.020895	0.032631
6-7	11.752961	11.289326	6.387603	5.693458	0.014767	0.013650	0.010400	0.009864
7-8	9.598614	9.215981	4.527007	4.050874	0.051891	0.047994	0.023131	0.018306
8-9	7.415941	7.117501	2.964756	3.019239	0.026880	0.024815	0.007492	0.004089
9-10	6.798442	6.524110	2.736180	3.025441	0.025114	0.023170	0.005482	0.002000
10-11	6.177784	5.927984	2.624961	3.147028	0.003758	0.003464	0.000116	-0.00054
11-12	5.647373	5.418602	2.499650	3.204565	0.006555	0.006041	-0.00032	-0.00158
12-13	4.961226	4.759739	2.524511	3.426309	0.026630	0.024545	-0.00040	-0.00536
13-14	4.271500	4.097789	2.712247	4.001456	0.009983	0.009193	-0.00012	0.008768
14-15	2.861155	2.744518	2.676857	2.679851	0.006242	0.005745	0.005467	0.005479
15-16	2.260888	2.168588	2.115052	2.117421	0.006055	0.005573	0.005302	0.005314
16-17	1.626619	1.560118	1.521551	1.523257	0.008981	0.008264	0.007862	0.007880
17-18	0.991900	0.951327	0.927799	0.928840	0.002700	0.002484	0.002362	0.002368
2-19	3.617126	3.611400	3.611762	3.610631	0.002118	0.002112	0.002112	0.002111
19-20	2.715747	2.711439	2.711711	2.710860	0.014307	0.014262	0.014265	0.014256
20-21	1.811120	1.808245	1.808427	1.807859	0.002836	0.002827	0.002827	0.002825
21-22	0.905851	0.904412	0.904503	0.904219	0.002568	0.002559	0.002560	0.002558
3-23	8.510116	8.423034	8.428506	4.201985	0.012694	0.012439	0.012455	-0.00010
23-24	7.556206	7.478764	7.483630	7.444688	0.023191	0.022724	0.022753	0.022520
24-25	3.783634	3.744744	3.747188	3.727633	0.011659	0.011422	0.011437	0.011319
6-26	13.064249	12.549141	12.580319	9.067561	0.008102	0.007493	0.007529	0.001910
26-27	12.491510	11.997931	12.027803	8.930668	0.010782	0.009969	0.010017	0.002124
27-28	11.922786	11.450697	11.479264	8.834239	0.047785	0.044204	0.044417	0.015098
28-29	11.390992	10.939471	10.966792	10.885906	0.034864	0.032213	0.032370	0.031904
29-30	10.112128	9.710820	9.735102	9.663216	0.015302	0.014122	0.014192	0.013985
30-31	4.667611	4.481433	4.492695	4.459356	0.017956	0.016564	0.016647	0.016403
31-32	3.024309	2.903592	2.910894	2.889278	0.003982	0.003671	0.003689	0.003635
32-33	0.717234	0.688593	0.690325	0.685197	0.001236	0.001139	0.001145	0.001128

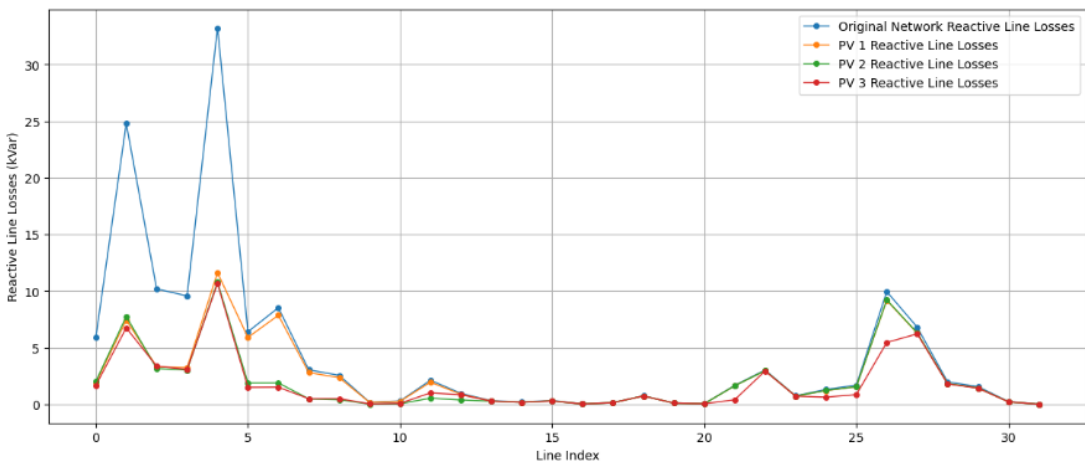
Appendix I

IEEE 33 Results Summary of distribution of PV Using Conventional ABC

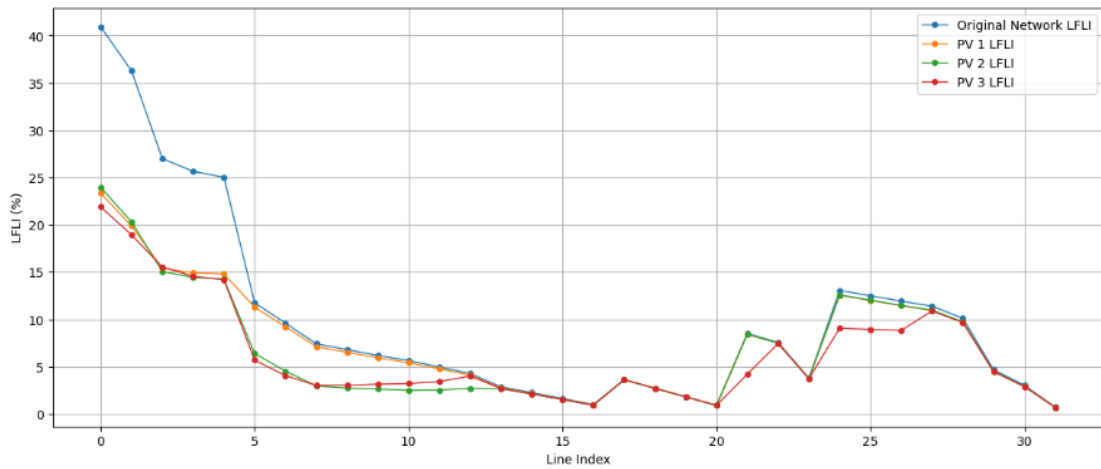
Comparison of IEEE 33 active power losses in Each line using conventional ABC



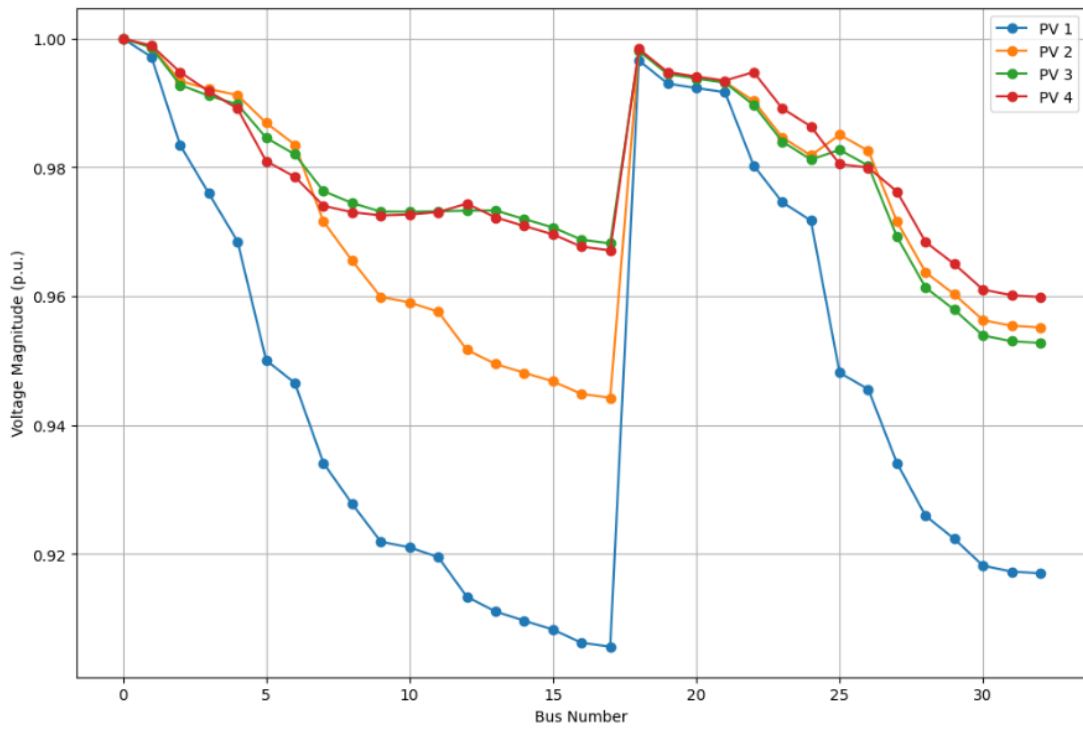
Comparison of IEEE 33 Reactive Power Losses in Each Line Using Conventional ABC



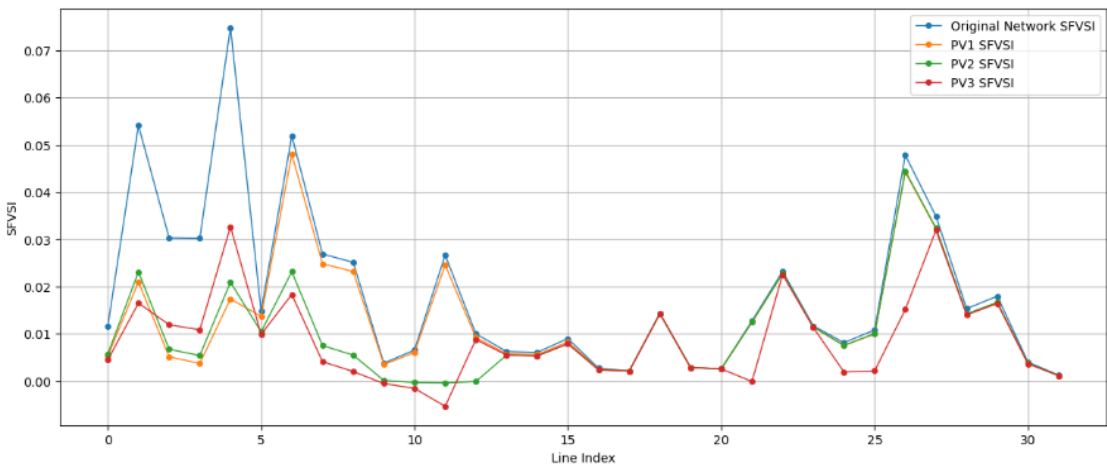
Comparison of IEEE 33-line Flow Limit Index in Each Line Using Conventional ABC



Comparison for IEEE 33 Voltage Profile in Each bus Using Conventional ABC



Comparison for IEEE 33 Voltage Profile in Each bus Using Conventional ABC



Appendix J

IEEE 33 Improved ABC Python Code

Import Libraries

```
import pandapower as pp
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
import numba
```

Network Data

```
# Read data from Excel for Buses
excel_file = 'networkData3.xlsx'
buses = pd.read_excel(excel_file, sheet_name='Buses', engine='openpyxl')
# Display the DataFrame to ensure it's loaded correctly
#buses

# Read data from Excel for Lines
lines = pd.read_excel(excel_file, sheet_name='Lines', engine='openpyxl')
# Display the DataFrame to ensure it's loaded correctly
#lines
```

Chaotic Bees Algorithm

Building the Network

```
def build_network(buses_df, lines_df):
    # Create an empty pandapower network
    net = pp.create_empty_network()
    # Create buses and attach loads, create slack bus if necessary
    for _, row in buses_df.iterrows():
```

```

# Create a bus

bus = pp.create_bus(net, vn_kv=6.6, name=row['name'], zone=1.0,
max_vm_pu=1.1, min_vm_pu=0.9)

# Create a load at the bus

pp.create_load(net, bus=bus, p_mw=row['p_kw']/1000,
q_mvar=row['q_kvar']/1000)

# Check if the bus should be a slack bus and create it if so

if row.get('is_slack', False):

    pp.create_ext_grid(net, bus=bus, vm_pu=1.0)

    slack_bus_created = True

    print(f"Slack bus created at Bus {bus}, Name: {row['name']}")

# Check if at least one slack bus was created, print a message if not

if 'slack_bus_created' not in locals():

    print("No slack bus was created from the Excel data!")

# Create lines between buses

for _, row in lines_df.iterrows():

    # Create a line between two buses

    pp.create_line_from_parameters(net, from_bus=row['from_bus']-1,
to_bus=row['to_bus']-1,
length_km=1, r_ohm_per_km=row['r'],
x_ohm_per_km=row['x'],
c_nf_per_km=0, max_i_ka=1)

return net

newTest = build_network(buses, lines

Slack bus created at Bus 0, Name: Bus 1

Run the Load Flow Analysis

```

```

def run_power_flow_analysis(net):
    try:
        pp.runpp(net, algorithm='bfs', max_iteration=100, numba= True)
        print("Power flow executed successfully!")
        real_power_loss = net.res_line.pl_mw.sum() * 1000 # Convert from MW to kW
        reactive_power_loss = net.res_line ql_mvar.sum() * 1000 # Convert from MVar to
kVar
        return real_power_loss, reactive_power_loss, net.converged
    except Exception as e:
        print("Error running power flow:", e)
        return None, None, False

fitness = newTest.res_line.pl_mw.sum() * 1000
starting_loss = fitness

# Before starting the ABC algorithm, calculate the base losses
net = build_network(buses, lines)
pp.runpp(net) # Run the initial power flow to determine the base losses
base_real_loss = net.res_line.pl_mw.sum() * 1000 # Convert from MW to kW
base_reactive_loss = net.res_line ql_mvar.sum() * 1000 # Convert from MVar to kVar

Slack bus created at Bus 0, Name: Bus 1

Fitness Function

def run_load_flow_and_calculate_fitness(net, pv_config, base_real_loss,
base_reactive_loss):
    # Backup original Loads
    original_loads = net.load.p_mw.copy()
    # Subtract PV sizes from the loads at corresponding buses
    for bus_id, pv_size in pv_config.items():

```

```

load_indices = net.load[net.load.bus == bus_id].index

for load_index in load_indices:

    net.load.at[load_index, 'p_mw'] = max(net.load.at[load_index, 'p_mw'] - pv_size
/ 1000, 0)

# Run the power flow analysis

try:

    pp.runpp(net, algorithm='bfs', max_iteration=100) # Use backward/forward
sweep algorithm

# Calculate real and reactive power losses in kW and kVar

real_power_loss = net.res_line.pl_mw.sum() * 1000 # Convert from MW to kW

reactive_power_loss = net.res_line.ql_mvar.sum() * 1000 # Convert from MVar to
kVar

# Calculate SFVSI for each line using the provided formula

vs = net.res_bus.loc[net.line['from_bus'], 'vm_pu'].values

vr = net.res_bus.loc[net.line['to_bus'], 'vm_pu'].values

sfvsi = (4 * vr / vs) * (1 - (vr / vs))

max_SFVSI = sfvsi.max()

# Calculate IVR for each bus

net.res_bus['IVR'] = (net.bus['max_vm_pu'] - net.res_bus['vm_pu']) /
net.bus['max_vm_pu']

max_IVR = net.res_bus['IVR'].max()

# Calculate LFLI for the network

current_loading_percent = np.abs(net.res_line.loading_percent) / 100 # Convert
percentage to a decimal

line_capacity_percent = np.ones_like(current_loading_percent) # Assume line
capacity is 100% for all lines

lfl_i = np.max(current_loading_percent / line_capacity_percent) # Compute LFLI

```

```

# Combine all metrics into a fitness value

weights = {'w1': 0.35, 'w2': 0.15, 'w3': 0.15, 'w4': 0.25, 'w5': 0.1}

fitness_value = (

    weights['w1'] * real_power_loss / base_real_loss +

    weights['w2'] * reactive_power_loss / base_reactive_loss +

    weights['w3'] * max_SFVSI +

    weights['w4'] * max_IVR +

    weights['w5'] * Ifli

)

except Exception as e:

    print("Error running load flow:", e)

    fitness_value = float('inf') # Assign a high cost to an invalid configuration

# Restore original Loads

net.load.p_mw = original_loads

return fitness_value

Chaotic Bees Functions

def logistic_map(r, x):

    return r * x * (1 - x)

def tent_map(mu, x):

    return mu * min(x, 1 - x)

chaotic_maps = {

    'logistic': logistic_map,

    'tent': tent_map,

}

# User selects the chaotic map

chaotic_map_name = "logistic"

```

```

chaotic_map = chaotic_maps[chaotic_map_name]

def hybrid_chaotic_map(r, x, iteration, lb, ub):
    if iteration % 2 == 0:
        return 4 * x * (1 - x) # Logistic map equation provided
    else:
        return min(r * x, r * (1 - x)) # Tent map

def chaotic_local_search(net, pv_config, base_real_loss, base_reactive_loss,
chaotic_iterations, lb, ub, r_value=4, mu_value=2):
    current_fitness = run_load_flow_and_calculate_fitness(net, pv_config,
base_real_loss, base_reactive_loss)

    best_pv_config = pv_config.copy()

    x_values = {bus_id: 0.5 for bus_id in pv_config} # Initial values for the chaotic maps
    for i in range(chaotic_iterations):
        for bus_id in pv_config.keys():
            # Convert decision variables to chaotic variables using equation 3.71
            c_xk = (pv_config[bus_id] - lb[bus_id]) / (ub[bus_id] - lb[bus_id])

            # Determine the chaotic variable using the hybrid chaotic map and equation 3.70
            c_xk_next = hybrid_chaotic_map(r_value, c_xk, i, lb[bus_id], ub[bus_id])

            # Convert chaotic variables back to decision variables using equation 3.72
            pv_config[bus_id] = lb[bus_id] + c_xk_next * (ub[bus_id] - lb[bus_id])

        # Calculate the new fitness
        new_fitness = run_load_flow_and_calculate_fitness(net, pv_config, base_real_loss,
base_reactive_loss)

        # Greedy selection to keep the better solution
        if new_fitness < current_fitness:
            current_fitness = new_fitness

```

```

        best_pv_config = pv_config.copy()

    return best_pv_config, current_fitness

Algorithm Parameters

# Constants and Parameters

PV_SIZE_RANGE = (100, 2000)

NUMBER_OF_PV_LOCATIONS = 3

NUMBER_OF_BEES = 50

NUMBER_OF_ITERATIONS = 100

PENETRATION_FACTOR = 1

SCOUT_BEE_LIMIT = 10

Algorithm Functions

def initialize_population(net, num_bees, num_pv_locations, pv_size_range,
penetration_factor):

    population = []

    total_pv_capacity_kw = calculate_total_pv_capacity(net, penetration_factor)

    print(total_pv_capacity_kw)

    for _ in range(num_bees):

        bee_solution = {}

        available_buses = list(net.bus.index)

        used_capacity_kw = 0

        for _ in range(num_pv_locations):

            if not available_buses or used_capacity_kw >= total_pv_capacity_kw:

                break # Exit if no buses are left or capacity is reached

            bus_id = random.choice(available_buses)

            available_buses.remove(bus_id) # Remove the selected bus to avoid re-selection

            # Adjust the maximum allowable PV size for this bus

```

```

max_pv_size_for_bus = min(
    pv_size_range[1],
    net.load.loc[net.load.bus == bus_id, 'p_mw'].iloc[0] * 1000, # Convert MW to
kW
    total_pv_capacity_kw - used_capacity_kw # Respect the total remaining
capacity
)
# Choose a PV size within the allowable range
pv_size = random.uniform(pv_size_range[0], max_pv_size_for_bus)
bee_solution[bus_id] = pv_size
used_capacity_kw += pv_size # Update the used capacity
# Check after filling the solution if we have exceeded the capacity due to rounding
and correct it
if used_capacity_kw > total_pv_capacity_kw:
    # Scale down the sizes proportionally to fit the total capacity
    scale_factor = total_pv_capacity_kw / used_capacity_kw
    for bus_id in bee_solution:
        bee_solution[bus_id] *= scale_factor
    used_capacity_kw = total_pv_capacity_kw # Correct the used capacity
    population.append(bee_solution)
return population
def employed_bee_phase(net, population, pv_size_range, base_real_loss,
base_reactive_loss):
    new_population = []
    for bee_solution in population:
        new_solution = {}
        for bus_id, pv_size in bee_solution.items():

```

```

# Generate a new PV size using a random neighbor's PV size
partner_solution = random.choice(population)

partner_pv_size = partner_solution.get(bus_id, pv_size_range[0])

phi = random.uniform(-1, 1) # Random number between -1 and 1

new_pv_size = pv_size + phi * (partner_pv_size - pv_size)

# Ensure new PV size is within bounds

new_pv_size = max(min(new_pv_size, pv_size_range[1]), pv_size_range[0])

new_solution[bus_id] = new_pv_size

# Evaluate the fitness of the new solution

fitness = run_load_flow_and_calculate_fitness(net, new_solution, base_real_loss,
base_reactive_loss)

current_fitness = run_load_flow_and_calculate_fitness(net, bee_solution,
base_real_loss, base_reactive_loss)

# Choose the solution with better fitness

if fitness < current_fitness:

    new_population.append(new_solution)

else:

    new_population.append(bee_solution)

return new_population

def onlooker_bee_phase(net, population, fitnesses, pv_size_range, base_real_loss,
base_reactive_loss):

    new_population = []

    # Replace NaN or inf fitness values with a high cost

    fitnesses = [f if not (np.isnan(f) or np.isinf(f)) else float('inf') for f in fitnesses]

    # Calculate the total fitness to normalize probabilities

    total_fitness = sum(fitnesses)

```

```

# Handle the case where the total fitness is 0 to avoid division by zero
if total_fitness == 0:
    total_fitness = 1

# Calculate the probability of selecting each solution
selection_probabilities = [f / total_fitness for f in fitnesses]

# Handle the case where all selection probabilities are NaN or inf
if any(np.isnan(selection_probabilities)) or any(np.isinf(selection_probabilities)):
    selection_probabilities = [1.0 / len(fitnesses) for _ in fitnesses]

# Iterate over the full population size
for _ in range(len(population)):
    # Select a solution based on fitness
    selected_index = np.random.choice(range(len(population)),
p=selection_probabilities)

    selected_solution = population[selected_index]

    # Perturb the selected solution
    partner_solution = random.choice(population)
    new_solution = { }

    for bus_id, pv_size in selected_solution.items():
        phi = random.uniform(-1, 1) # Random number between -1 and 1
        partner_pv_size = partner_solution.get(bus_id, pv_size)

        # Calculate new PV size and ensure it is within the specified size range
        new_pv_size = pv_size + phi * (partner_pv_size - pv_size)
        new_pv_size = max(min(new_pv_size, pv_size_range[1]), pv_size_range[0])

        new_solution[bus_id] = new_pv_size

    # Evaluate the fitness of the new solution

```

```

    fitness = run_load_flow_and_calculate_fitness(net, new_solution, base_real_loss,
base_reactive_loss)

    # Choose the solution with better fitness

    if fitness < fitnesses[selected_index]:

        new_population.append(new_solution)

        fitnesses[selected_index] = fitness # Update fitness value

    else:

        new_population.append(selected_solution)

# Ensure new_population has the same size as the original population

while len(new_population) < len(population):

    new_population.append(random.choice(population))

return new_population

def scout_bee_phase(net, population, trials, limit, pv_size_range, num_pv_locations):

    for i, trial_count in enumerate(trials):

        if trial_count > limit: # If a solution hasn't improved for a certain number of trials

            # Generate a new random solution

            new_solution = {}

            remaining_capacity_kw = calculate_total_pv_capacity(net,
PENETRATION_FACTOR)

            while remaining_capacity_kw > 0 and len(new_solution) < num_pv_locations:

                bus_id = random.choice(net.bus.index)

                pv_size = random.uniform(pv_size_range[0], pv_size_range[1])

                new_solution[bus_id] = pv_size

                remaining_capacity_kw -= pv_size

            population[i] = new_solution

            trials[i] = 0 # Reset the trial count for this new solution

```

```

    return population

bus_ids = net.load.bus.unique()

lower_bounds = {bus_id: PV_SIZE_RANGE[0] for bus_id in bus_ids}

upper_bounds = {bus_id: PV_SIZE_RANGE[1] for bus_id in bus_ids}

def abc_algorithm_full(net, num_iterations, pv_size_range, num_pv_locations,
penetration_factor, num_bees, scout_bee_limit, base_real_loss, base_reactive_loss, lb,
ub):

    # Initialize bee population

    population = initialize_population(net, num_bees, num_pv_locations, pv_size_range,
penetration_factor)

    # Initialize the best solution and its fitness

    best_solution = None

    best_fitness = float('inf')

    # Initialize a list to track the best fitness value in each iteration

    best_fitness_values = []

    # Initialize a list to track all fitness values in each iteration

    iteration_fitness_values = []

    # Initialize trial counters for scout bees

    trials = [0] * len(population)

    for i in range(num_iterations):

        # Employed Bee Phase

        population = employed_bee_phase(net, population, pv_size_range, base_real_loss,
base_reactive_loss)

        # Onlooker Bee Phase

        fitnesses = [run_load_flow_and_calculate_fitness(net, sol, base_real_loss,
base_reactive_loss) for sol in population]

```

```

    population = onlooker_bee_phase(net, population, fitnesses, pv_size_range,
base_real_loss, base_reactive_loss)

    # Chaotic Local Search

    for idx, pv_config in enumerate(population):

        chaotic_solution, chaotic_fitness = chaotic_local_search(

            net, pv_config, base_real_loss, base_reactive_loss, chaotic_iterations=10,
lb=lb, ub=ub

        )

        # Greedy selection to potentially replace the solution in the population

        if chaotic_fitness < fitnesses[idx]:

            population[idx] = chaotic_solution

            fitnesses[idx] = chaotic_fitness

    # Scout Bee Phase

    population = scout_bee_phase(net, population, trials, scout_bee_limit,
pv_size_range, num_pv_locations)

    # Update fitnesses after scout bee phase

    new_fitnesses = [run_load_flow_and_calculate_fitness(net, sol, base_real_loss,
base_reactive_loss) for sol in population]

    iteration_fitness_values.append(new_fitnesses)

    # Find the best solution and its fitness in the current iteration

    for idx, fitness in enumerate(new_fitnesses):

        if fitness < best_fitness:

            best_fitness = fitness

            best_solution = population[idx]

    # If there is no improvement, increment the trial counter

    if fitness > fitnesses[idx]:

        trials[idx] += 1

```

```

else:
    trials[idx] = 0 # Reset the trial counter if there was an improvement

    # Store the best fitness of this iteration

    best_fitness_values.append(best_fitness)

return best_solution, best_fitness_values, iteration_fitness_values
best_solution, best_fitness_values, iteration_fitness_values = abc_algorithm_full(
    net=newTest,
    num_iterations=NUMBER_OF_ITERATIONS,
    pv_size_range=PV_SIZE_RANGE,
    num_pv_locations=NUMBER_OF_PV_LOCATIONS,
    penetration_factor=PENETRATION_FACTOR,
    num_bees=NUMBER_OF_BEES,
    scout_bee_limit=SCOUT_BEE_LIMIT,
    base_real_loss=base_real_loss,
    base_reactive_loss=base_reactive_loss,
    lb=lower_bounds,
    ub=upper_bounds

```

Testing and Debugging

```

# Create a DataFrame for iteration fitness values

df_iterations = pd.DataFrame(iteration_fitness_values).transpose()

df_iterations.columns = [f"Iteration {i+1}" for i in range(len(iteration_fitness_values))]

# Display the DataFrame

df_iterations

# Convert the nested list into a pandas DataFrame for easier handling

df_iterations = pd.DataFrame(iteration_fitness_values).transpose()

df_iterations.columns = [f"Iteration {i+1}" for i in range(len(iteration_fitness_values))]

```

```

# Calculate the mean fitness for each iteration (ignoring NaN values)
mean_fitness_values = df_iterations.mean()

Plots

# Add the starting_loss to the beginning of the best_fitness_values list
complete_fitness_values = [starting_loss] + best_fitness_values

# Create a new list for the x-axis that starts from 0
iterations = list(range(len(complete_fitness_values)))

# Create the plot
plt.figure(figsize=(10, 5))

plt.plot(iterations, complete_fitness_values, marker='o', linestyle='-', color='green',
label='Best Fitness Value')

# Add a red point for the starting loss
plt.scatter(0, starting_loss, marker='o', color='red',)

# Add labels and title
plt.title('Best Fitness Value Over Iterations')

plt.xlabel('Iteration')

plt.ylabel('Best Fitness Value')

plt.grid(True)

plt.legend()

plt.show()

# Calculate the best fitness value in each iteration
best_in_each_iteration = df_iterations.min()

# Plot for the best fitness value of each iteration
plt.figure(figsize=(10, 5))

plt.plot(best_in_each_iteration.index, best_in_each_iteration.values, marker='o',
linestyle='-', color='blue', label='Best Iteration Fitness')

```

```

plt.title('Best Fitness Value Found in Each Iteration')

plt.xlabel('Iteration')

plt.ylabel('Best Fitness Value')

plt.grid(True, which="both", linestyle='--')

plt.legend()

plt.show()

import pandas as pd

# Run the initial power flow for the original case

pp.runpp(net)

# Save the results for the original case

original_res_bus = net.res_bus.copy()

original_res_line = net.res_line.copy()

# Apply the best solution's PV sizes to the network

for bus_id, pv_size in best_solution.items():

    load_idx = net.load[net.load.bus == bus_id].index[0]

    net.load.at[load_idx, 'p_mw'] -= pv_size / 1000 # Assuming pv_size is in kW

# Run the power flow for the best solution case

pp.runpp(net)

# Save the results for the best solution case

best_res_bus = net.res_bus.copy()

best_res_line = net.res_line.copy()

# Create DataFrames

original_bus_df = pd.DataFrame(original_res_bus)

original_line_df = pd.DataFrame(original_res_line)

best_bus_df = pd.DataFrame(best_res_bus)

best_line_df = pd.DataFrame(best_res_line)

```

```

# Reset the changes made for the best solution run

for bus_id, pv_size in best_solution.items():

    load_idx = net.load[net.load.bus == bus_id].index[0]

    net.load.at[load_idx, 'p_mw'] += pv_size / 1000 # Reset the changes

# Now you have DataFrames original_bus_df, original_line_df, best_bus_df, and
best_line_df

# You can display them using:

print("Original Bus Results:")

print(original_bus_df)

print("Original Line Results:")

print(original_line_df)

print("Best Solution Bus Results:")

print(best_bus_df)

print("Best Solution Line Results:")

print(best_line_df)

import matplotlib.pyplot as plt

# Assuming original_bus_df and best_bus_df are already defined and contain 'q_mvar'
for reactive power

# for each bus in the original and best solution scenarios respectively.

# Plotting

plt.figure(figsize=(14, 7))

# Plot original reactive power

plt.plot(original_bus_df.index, original_bus_df['q_mvar'], marker='o', linestyle='-',
color='blue', label='Original Reactive Power')

# Plot best solution reactive power

plt.plot(best_bus_df.index, best_bus_df['q_mvar'], marker='x', linestyle='--', color='red',
label='Best Solution Reactive Power')

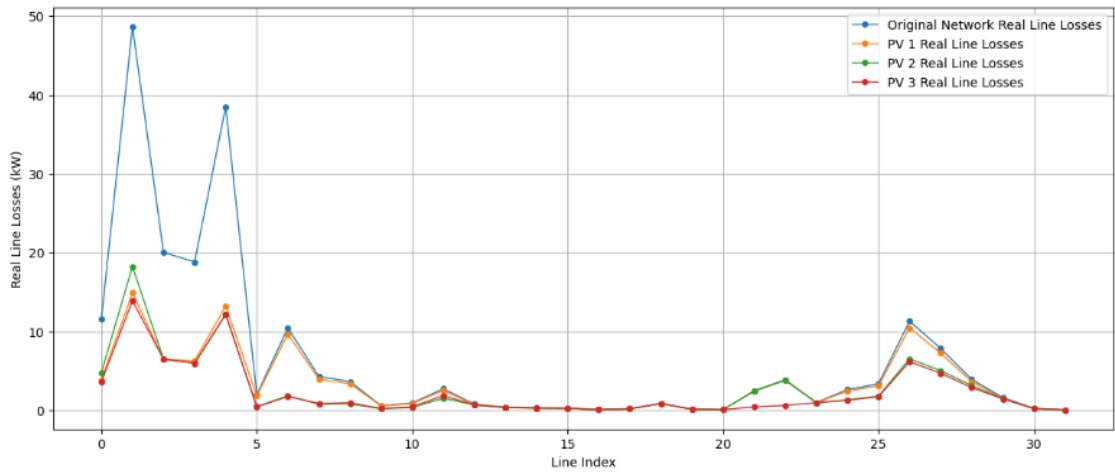
```

```
# Adding labels and title
plt.xlabel('Bus Index')
plt.ylabel('Reactive Power (MVar)')
plt.title('Comparison of Reactive Power Between Original and Best Solution')
plt.legend()
plt.grid(True)
# Show the plot
plt.show()
```

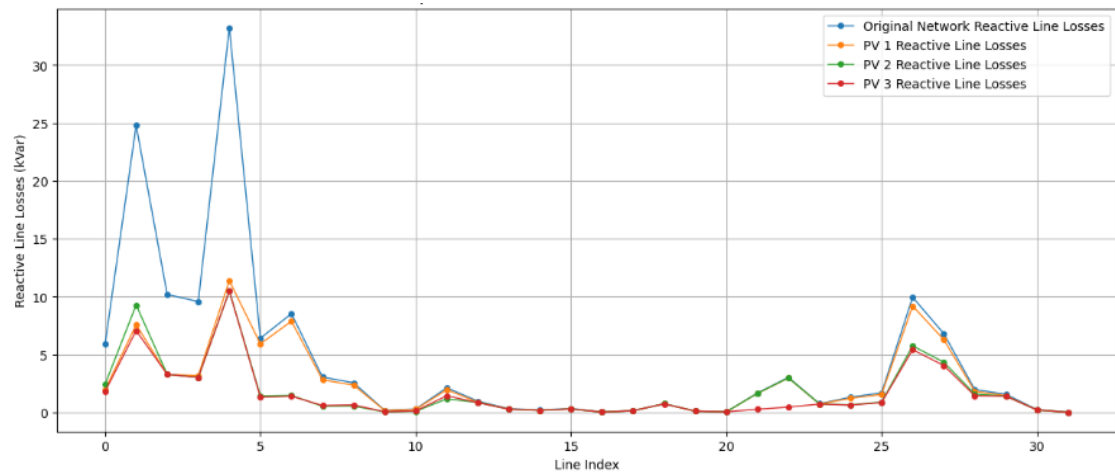
Appendix K

IEEE 33 Results Summary of random distribution of PV Using Improved ABC

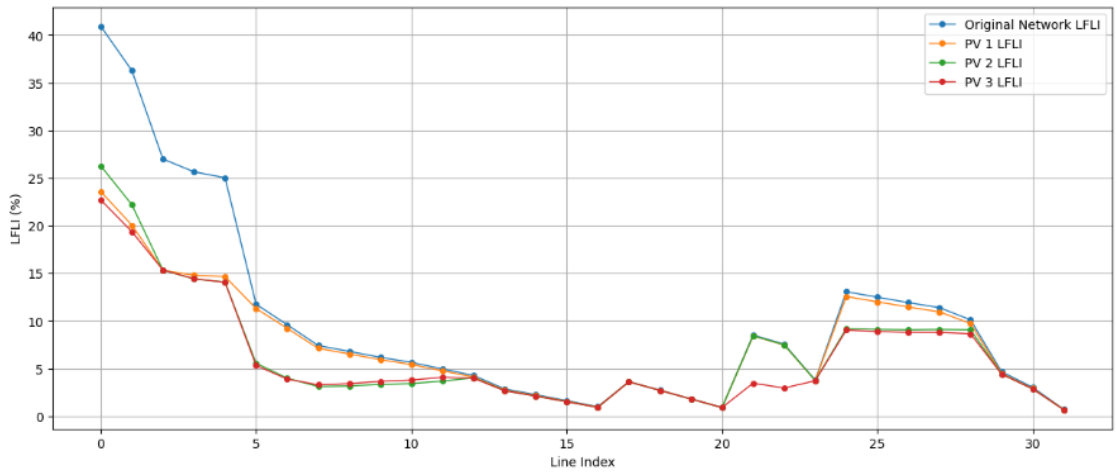
Comparison for IEEE 33 Active Power Losses in Each Line Using Improved ABC



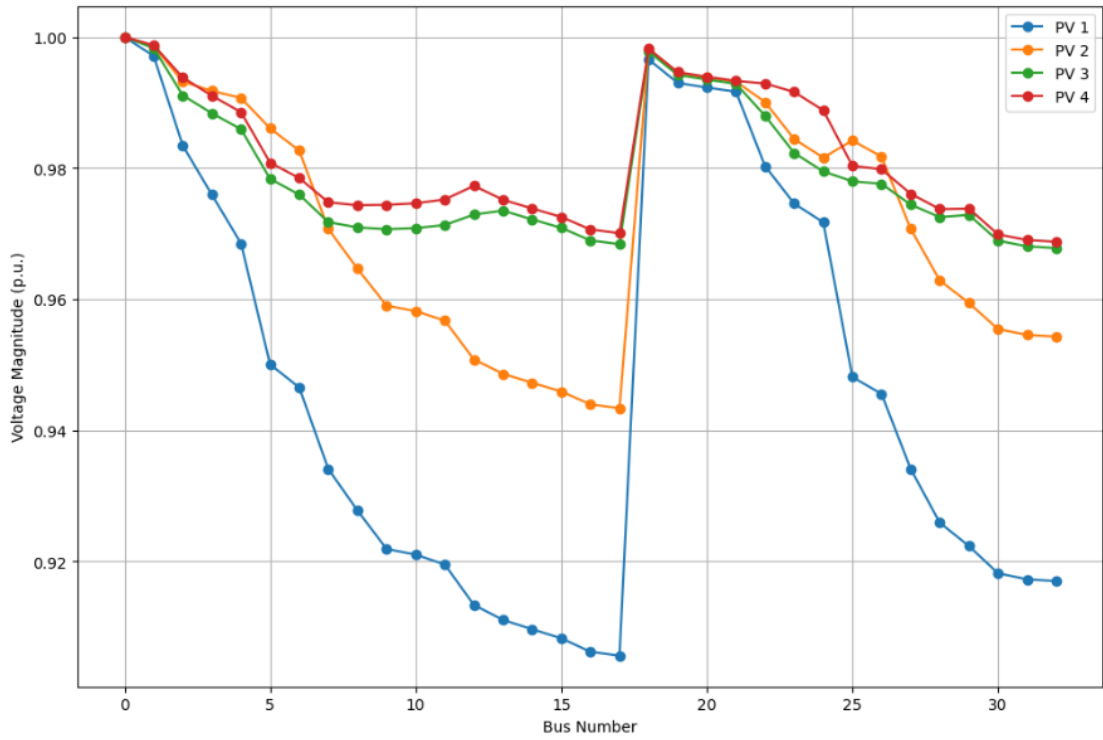
Comparison of IEEE 33 Reactive Power Losses in Each Line Using Improved ABC



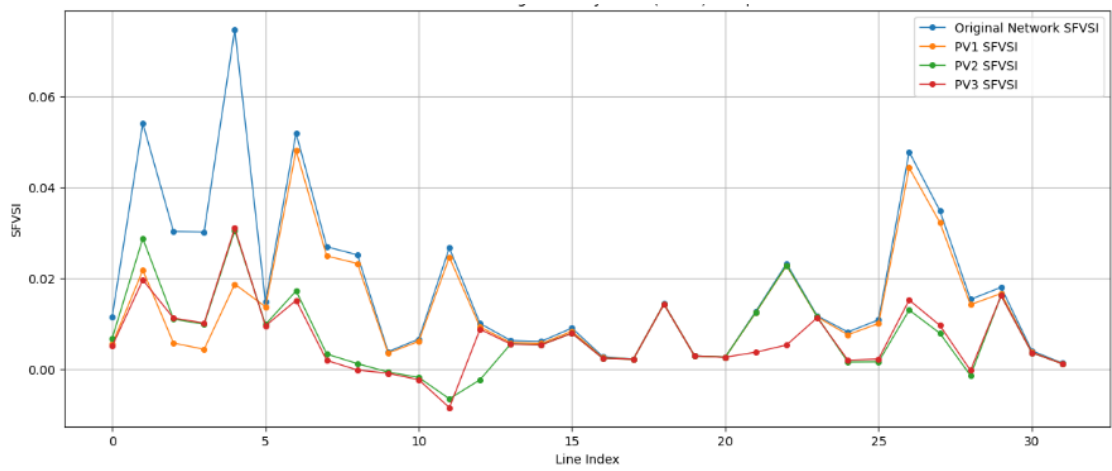
Comparison Of IEEE 33 Reactive Power Losses in Each Line Using Improved ABC



Comparison of IEEE 33 Voltage Profile in Each Bus Using Improved ABC



Comparison of IEEE 33 Voltage Stability Index in Each Bus Using Improved ABC



Appendix L

IEEE 33 Network Parameters After Installing PV Using improved ABC Algorithm

	Voltage profile			
	original	One PV	two PV	three PV
1	1.000000	1.000000	1.000000	1.000000
2	0.997112	0.998646	0.998317	0.998747
3	0.983444	0.993193	0.991100	0.993834
4	0.975943	0.991781	0.988379	0.991049
5	0.968525	0.990714	0.985944	0.988546
6	0.950068	0.986072	0.978365	0.980827
7	0.946547	0.982690	0.975971	0.978498
8	0.934104	0.970733	0.971763	0.974796
9	0.927784	0.964662	0.970971	0.974343
10	0.921922	0.959032	0.970694	0.974410
11	0.921055	0.958199	0.970867	0.974646
12	0.919543	0.956747	0.971328	0.975230
13	0.913380	0.950829	0.972931	0.977311
14	0.911095	0.948635	0.973517	0.975177
15	0.909671	0.947268	0.972185	0.973848
16	0.908292	0.945944	0.970896	0.972560
17	0.906248	0.943982	0.968984	0.970652
18	0.905636	0.943395	0.968412	0.970080
19	0.996584	0.998118	0.997789	0.998219
20	0.993006	0.994547	0.994216	0.994648
21	0.992302	0.993843	0.993513	0.993945
22	0.991665	0.993207	0.992876	0.993309
23	0.980313	0.990093	0.987994	0.992922
24	0.974596	0.984433	0.982322	0.991618
25	0.971747	0.981613	0.979496	0.988818
26	0.948139	0.984218	0.977999	0.980354
27	0.945577	0.981755	0.977621	0.979828
28	0.934142	0.970764	0.974445	0.976101
29	0.925928	0.962868	0.972528	0.973759
30	0.922372	0.959451	0.972885	0.973843
31	0.918213	0.955454	0.968944	0.969906
32	0.917298	0.954575	0.968077	0.969040
33	0.917014	0.954302	0.967809	0.968771

	Active losses				Reactive losses			
	original	One PV	two PV	three PV	original	One PV	two PV	three PV
1-2	11.585876	3.844674	4.775007	3.564236	5.906032	1.959866	2.434114	1.816910
2-3	48.680369	14.878577	18.224373	13.875763	24.794403	7.578115	9.282232	7.067351
3-4	20.009593	6.440420	6.458402	6.472224	10.190678	3.280039	3.289197	3.296236
4-5	18.807924	6.252215	5.939419	5.940061	9.579160	3.184348	3.025036	3.025363

5-6	38.478331	13.193380	12.161423	12.146497	33.216337	11.389157	10.498322	10.485438
6-7	1.939375	1.792552	0.433684	0.397963	6.410710	5.925380	1.433567	1.315489
7-8	10.443779	9.644958	1.801780	1.723981	8.534545	7.881757	1.472395	1.408819
8-9	4.248455	3.920463	0.746981	0.835245	3.052288	2.816643	0.536666	0.600079
9-10	3.618933	3.338798	0.786105	0.917612	2.565144	2.366581	0.557201	0.650414
10-11	0.562743	0.519094	0.164060	0.196681	0.186054	0.171623	0.054242	0.065027
11-12	0.895550	0.825963	0.329742	0.403111	0.296125	0.273115	0.109033	0.133293
12-13	2.709975	2.498875	1.492060	1.837684	2.132167	1.966076	1.173930	1.445862
13-14	0.741141	0.683331	0.659094	0.646452	0.975553	0.899458	0.867555	0.850915
14-15	0.362854	0.334483	0.317479	0.316391	0.322946	0.297696	0.282561	0.281593
15-16	0.286110	0.263708	0.250284	0.249425	0.208937	0.192578	0.182774	0.182147
16-17	0.255791	0.235735	0.223719	0.222950	0.341518	0.314741	0.298697	0.297670
17-18	0.054014	0.049777	0.047238	0.047076	0.042355	0.039033	0.037042	0.036915
2-19	0.160928	0.160431	0.160537	0.160398	0.153569	0.153094	0.153195	0.153063
19-20	0.832042	0.829463	0.830015	0.829294	0.749734	0.747410	0.747908	0.747258
20-21	0.100742	0.100429	0.100496	0.100409	0.117692	0.117327	0.117405	0.117303
21-22	0.043627	0.043492	0.043521	0.043483	0.057684	0.057505	0.057543	0.057493
3-23	2.450763	2.401964	2.412313	0.406564	1.674579	1.641235	1.648307	0.277801
23-24	3.845433	3.768743	3.785007	0.589080	3.036522	2.975964	2.988806	0.465164
24-25	0.962028	0.942787	0.946867	0.929097	0.752765	0.737710	0.740902	0.726997
6-26	2.598521	2.401892	1.288859	1.244125	1.323581	1.223427	0.656492	0.633707
26-27	3.325946	3.073746	1.769273	1.688549	1.693400	1.564993	0.900823	0.859722
27-28	11.290487	10.432598	6.521948	6.153250	9.954606	9.198222	5.750276	5.425203
28-29	7.826154	7.230864	4.995697	4.663021	6.817960	6.299358	4.352133	4.062314
29-30	3.892086	3.595690	3.126184	2.849362	1.982471	1.831499	1.592352	1.451350
30-31	1.592164	1.470312	1.429606	1.426769	1.573537	1.453110	1.412880	1.410076
31-32	0.212998	0.196685	0.191236	0.190856	0.248257	0.229244	0.222893	0.222451
32-33	0.013156	0.012148	0.011812	0.011788	0.020456	0.018889	0.018365	0.018329

	line flow limit index				Voltage stability index			
	original	One PV	two PV	three PV	original	One PV	two PV	three PV
1-2	40.932508	23.579450	26.277916	22.703203	0.011518	0.005409	0.006720	0.005006
2-3	36.284642	20.059801	22.200983	19.371995	0.054079	0.021724	0.028707	0.019578
3-4	26.999017	15.317430	15.338799	15.355204	0.030277	0.005677	0.010953	0.011179
4-5	25.651946	14.789955	14.415240	14.416019	0.030171	0.004299	0.009831	0.010078
5-6	25.028541	14.655662	14.070827	14.062190	0.074777	0.018654	0.030510	0.030990
6-7	11.752961	11.299320	5.557803	5.323997	0.014767	0.013674	0.009766	0.009473
7-8	9.598614	9.224224	3.986856	3.899832	0.051891	0.048076	0.017170	0.015079
8-9	7.415941	7.123927	3.109607	3.288197	0.026880	0.024858	0.003257	0.001856
9-10	6.798442	6.530015	3.168541	3.423328	0.025114	0.023211	0.001143	-0.00027
10-11	6.177784	5.933361	3.335636	3.652240	0.003758	0.003470	-0.00071	-0.00097
11-12	5.647373	5.423525	3.426800	3.788903	0.006555	0.006052	-0.00190	-0.00239
12-13	4.961226	4.764075	3.681285	4.085468	0.026630	0.024589	-0.00661	-0.00855
13-14	4.271500	4.101527	4.028132	3.989312	0.009983	0.009209	-0.00241	0.008715
14-15	2.861155	2.747027	2.676289	2.671700	0.006242	0.005756	0.005465	0.005446
15-16	2.260888	2.170573	2.114603	2.110972	0.006055	0.005583	0.005300	0.005282
16-17	1.626619	1.561548	1.521227	1.518612	0.008981	0.008280	0.007859	0.007832
17-18	0.991900	0.952200	0.927602	0.926006	0.002700	0.002488	0.002361	0.002353
2-19	3.617126	3.611528	3.612726	3.611160	0.002118	0.002112	0.002113	0.002111
19-20	2.715747	2.711535	2.712437	2.711258	0.014307	0.014263	0.014272	0.014260
20-21	1.811120	1.808309	1.808911	1.808124	0.002836	0.002827	0.002829	0.002826
21-22	0.905851	0.904444	0.904745	0.904352	0.002568	0.002560	0.002561	0.002559
3-23	8.510116	8.424964	8.443094	3.466165	0.012694	0.012444	0.012497	0.003666
23-24	7.556206	7.480480	7.496603	2.957457	0.023191	0.022734	0.022831	0.005247
24-25	3.783634	3.745606	3.753702	3.718312	0.011659	0.011427	0.011477	0.011262

6-26	13.064249	12.560245	9.200766	9.039687	0.008102	0.007506	0.001499	0.001925
26-27	12.491510	12.008570	9.110763	8.900495	0.010782	0.009986	0.001545	0.002146
27-28	11.922786	11.460871	9.061708	8.801844	0.047785	0.044280	0.012950	0.015157
28-29	11.390992	10.949201	9.100922	8.792675	0.034864	0.032269	0.007853	0.009576
29-30	10.112128	9.719468	9.062722	8.652173	0.015302	0.014147	-0.00146	-0.00034
30-31	4.667611	4.485444	4.422918	4.418527	0.017956	0.016594	0.016138	0.016106
31-32	3.024309	2.906192	2.865653	2.862806	0.003982	0.003678	0.003576	0.003569
32-33	0.717234	0.689210	0.679591	0.678916	0.001236	0.001141	0.001110	0.001107

Appendix M
Part of the Nablus network

Sbase = 100MVA

Vbase = 6.6KV

from	to	R(ohm)	X(ohm)	Length (Km)	Load P(KW)	Load Q(KVAR)	Imax(A)
1	2	0.03946	0.028852	1	0	0	370
2	3	0.1072	0.05687	1	160	52.58947	370
3	4	0.1956	0.17153	0.47	160	52.58947	340
4	5	0.1956	0.17515	0.14	160	52.58947	340
5	6	0.1956	0.1036	0.1	100	32.86842	340
6	7	0.41	0.175	0.08	0	0	370
7	8	0.16266	0.12	0.365	160	52.58947	380
8	9	0.16266	0.12	0.27	252	82.82842	380
7	10	0.1956	0.103	0.29	0	0	340
10	11	0.16266	0.12	0.35	160	52.58947	380
11	12	0.16266	0.12	0.27	252	82.82842	380
10	13	0.13062	0.0671	1	160	52.58947	340
13	14	0.1956	0.1036	0.22	252	82.82842	340
2	15	0.18647	0.0786	1	160	52.58947	370
15	16	0.4938	0.18755	0.25	100	32.86842	200
16	17	0.4938	0.18755	0.33	160	52.58947	200
17	18	0.4938	0.18755	0.21	100	32.86842	200
15	19	0.325	0.17153	0.43	160	52.58947	270
19	20	0.325	0.17153	0.53	160	52.58947	270
20	21	0.1614	0.07857	1	252	82.82842	330
19	22	0.05584	0.036	1	100	32.86842	360
22	23	0.0216	0.0137	1	252	82.82842	360
23	24	0.1956	0.1036	0.17	252	82.82842	340
1	25	0.03584	0.02804	1	0	0	380
25	26	0.16266	0	0.14	100	32.86842	340
26	27	0.2842	0.12	1	0	0	380
27	28	0.325	0.1715	0.25	160	52.58947	270
27	29	0.16266	0.12	0.32	160	52.58947	380
29	30	0.1956	0.1036	0.22	64	21.03579	340
29	31	0.1956	0.1036	0.21	160	52.58947	340
31	32	0.1588	0.1683	0.55	0	0	360
32	33	0.1956	0.1036	0.42	252	82.82842	340
32	34	0.1956	0.1036	0.19	252	82.82842	340
34	35	0.1956	0.1036	0.53	160	52.58947	340
35	36	0.1956	0.1036	0.57	160	52.58947	340
25	37	0.1956	0.1036	0.23	0	0	340
37	38	0.1588	0.1683	0.17	160	52.58947	360
38	39	0.134	0.14	1	100	32.86842	360
39	40	0.081	0.06	1	252	82.82842	360
40	41	0.1	0.08	1	0	0	360
41	42	0.036	0.03	1	252	82.82842	340

41	43	0.068	0.032	1	400	131.4737	340
43	44	0.1956	0.1036	0.25	252	82.82842	340
44	45	0.1956	0.1036	0.17	160	52.58947	340
37	46	0.1956	0.1036	0.2	0	0	340
46	47	0.4103	0.175	0.01	252	82.82842	370
47	48	0.16266	0.12	0.12	160	52.58947	380
46	49	0.4103	0.175	0.36	160	52.58947	370
49	50	0.1956	0.1036	0.35	160	52.58947	340
49	51	0.4103	0.175	0.17	252	82.82842	370
51	52	0.1956	0.1036	0.5	160	52.58947	340
25	53	0.3051	0.1114	1	0	0	340
53	54	0.1956	0.1036	0.18	400	131.4737	340
53	55	0.1956	0.1036	0.32	252	82.82842	340
55	56	0.1956	0.1036	0.35	252	82.82842	340
55	57	0.2114	0.09	0.53	252	82.82842	330
57	58	0.1956	0.1036	0.35	0	0	340
58	59	0.16266	0.12	0.48	100	32.86842	380
58	60	0.1956	0.1036	0.18	160	52.58947	340
60	61	0.16266	0.12	0.1	160	52.58947	380
57	62	0.0392	0.0158	1	160	52.58947	330
62	63	0.0618	0.0264	1	160	52.58947	330
63	64	0.12865	0.062	1	160	52.58947	330
64	65	0.16266	0.12	0.51	100	32.86842	380
64	66	0.1	0.053	1	100	32.86842	330
1	67	0.04832	0.03244	1	0	0	360
67	68	0.1956	0.1713	0.58	160	52.58947	340
68	69	0.1956	0.1036	0.84	160	52.58947	340
69	70	0.0553	0.039	1	160	52.58947	340
70	71	0.16266	0.12	0.07	160	52.58947	380
71	72	0.06112	0.0416	1	160	52.58947	340
69	73	0.1956	0.1036	0.68	160	52.58947	340
73	74	0.1956	0.1036	0.2	100	32.86842	340
67	75	0.1956	0.01713	0.6	160	52.58947	340
75	76	0.1956	0.1036	1.22	160	52.58947	340
76	77	0.1956	0.1036	0.5	160	52.58947	340
67	78	0.1956	0.1036	0.75	160	52.58947	340
78	79	0.1956	0.1713	0.15	252	82.82842	340
79	80	0.1956	0.1036	0.18	160	52.58947	340
80	81	0.16266	0.12	0.07	252	82.82842	380
79	82	0.0234	0.014	1	252	82.82842	340
82	83	0.043	0.024	1	160	52.58947	340
83	84	0.1956	0.1036	0.23	100	32.86842	340
84	85	0.16266	0.12	0.22	100	32.86842	380
84	86	0.0333	0.02	1	252	82.82842	340
86	87	0.023	0.016	1	252	82.82842	340
87	88	0.1588	0.1039	0.1	252	82.82842	360
67	89	0.1956	0.1036	0.5	0	0	340
89	90	0.1956	0.1036	0.18	160	52.58947	340
89	91	0.049	0.026	1	100	32.86842	340

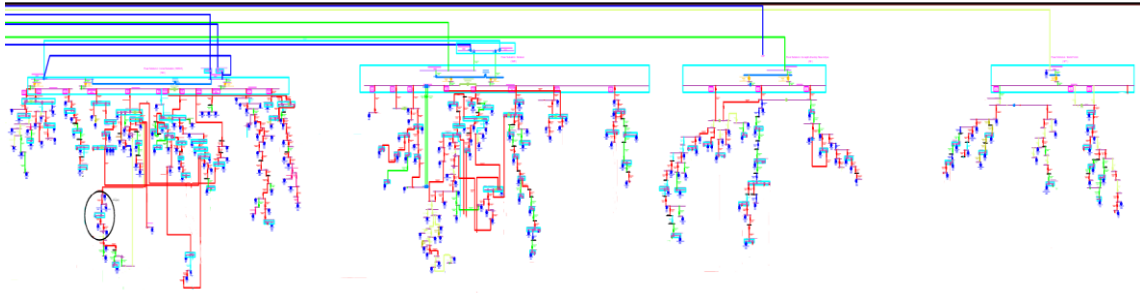
91	92	0.0195	0.01	1	252	82.82842	340
92	93	0.211	0.096	0.35	252	82.82842	330
93	94	0.211	0.096	0.4	160	52.58947	330
94	95	0.4103	0.1753	0.04	160	52.58947	370
94	96	0.325	0.171	0.65	160	52.58947	270
92	97	0.1956	0.1036	0.18	252	82.82842	340
97	98	0.1956	0.1036	0.2	100	32.86842	340
98	99	0.1956	0.1036	0.18	400	131.4737	340
92	100	0.1956	0.1036	0.29	100	32.86842	340
100	101	0.1956	0.1036	0.31	252	82.82842	340
101	102	0.1956	0.1036	0.16	0	0	340
102	103	0.1956	0.1713	0.05	252	82.82842	340
102	104	0.1956	0.1713	0.43	160	52.58947	340
104	105	0.1956	0.1713	0.3	252	82.82842	340
105	106	0.1956	0.1036	0.31	0	0	340
106	107	0.1956	0.1036	0.56	-5000	82.82842	340
107	108	0.1956	0.1036	0.77	160	52.58947	340
108	109	0.1956	0.1036	0.55	100	32.86842	340
107	110	0.325	0.171	0.88	100	32.86842	270
107	111	0.16266	0.12	0.93	100	32.86842	380
106	112	0.4102	0.1753	0.32	0	0	370
112	113	0.4102	0.1753	0.24	100	32.86842	370
112	114	0.4102	0.1753	0.87	64	21.03579	370
114	115	0.4102	0.1753	1.2	100	32.86842	370
115	116	0.4938	0.8187	0.4	160	52.58947	200
114	117	0.4102	0.1753	1	0	0	370
117	118	0.4102	0.1753	1.2	160	52.58947	370
117	119	0.4102	0.1753	0.73	100	32.86842	370
119	120	0.4102	0.1753	0.17	0	0	370
120	121	0.4938	0.187	0.35	100	32.86842	200
120	122	0.13	0.06	1	160	52.58947	340
122	123	0.169	0.08	1	160	52.58947	340
123	124	0.4102	0.1753	0.85	160	52.58947	370
124	125	0.4102	0.1753	1.26	64	21.03579	370
125	126	0.206	0.1	1	160	52.58947	370
67	127	0.00508	0.00656	3.7	0	0	360
127	128	0.25	0.1715	0.18	100	32.86842	270
128	129	0.1956	0.1036	0.17	252	82.82842	340
128	130	0.1956	0.1036	0.24	160	52.58947	340
130	131	0.1956	0.1715	0.22	252	82.82842	340
131	132	0.1956	0.1036	0.07	160	52.58947	340
132	133	0.1956	0.1036	0.19	252	82.82842	340
130	134	0.1956	0.1036	0.04	252	82.82842	340
134	135	0.1956	0.1715	0.05	160	52.58947	340
135	136	0.16266	0.12	0.09	100	32.86842	370
134	137	0.1956	0.1715	0.07	160	52.58947	340
137	138	0.1956	0.1715	0.03	160	52.58947	340
127	139	0.1956	0.1036	0.08	252	82.82842	340
127	140	0.1956	0.1036	0.62	252	82.82842	340

140	141	0.1956	0.1715	0.4	160	52.58947	340
141	142	0.1956	0.1036	0.64	0	0	340
142	143	0.1956	0.1036	0.4	160	52.58947	340
142	144	0.1956	0.1715	0.56	160	52.58947	340
144	145	0.1956	0.1715	0.43	252	82.82842	340
67	146	0.0196	0.0148	1	0	0	410
146	147	0.1956	0.1036	0.97	0	0	340
147	148	0.1956	0.1036	0.18	400	131.4737	340
147	149	0.1956	0.1036	0.26	0	0	340
149	150	0.1956	0.1036	0.15	252	82.82842	340
149	151	0.0674	0.033	1	252	82.82842	340
151	152	0.325	0.1715	0.73	160	52.58947	270
152	153	0.325	0.1715	0.65	100	32.86842	270
153	154	0.359	0.13	1	252	82.82842	340
154	155	0.325	0.1715	0.5	160	52.58947	270
155	156	0.822	0.187	0.74	0	0	210
156	157	0.822	0.187	0.18	100	32.86842	210
156	158	0.822	0.187	0.18	160	52.58947	210
158	159	0.236	0.077	1	160	52.58947	340
146	160	0.1956	0.1036	0.28	160	52.58947	340
160	161	0.2114	0.096	0.2	100	32.86842	340
161	162	0.1956	0.1036	0.55	100	32.86842	330
162	163	0.16266	0.12	0.48	252	82.82842	340
163	164	0.325	0.1715	0.54	160	52.58947	270
164	165	0.325	0.1715	0.23	0	0	270
165	166	0.325	0.1715	0.4	160	52.58947	270
165	167	0.325	0.1715	0.3	252	82.82842	270
167	168	0.325	0.1715	0.35	160	52.58947	270
168	169	0.1956	0.1036	0.31	160	52.58947	340
169	170	0.1956	0.1036	0.29	160	52.58947	340
170	171	0.1956	0.1036	0.18	0	0	340
171	172	0.16266	0.12	0.76	64	21.03579	380
171	173	0.1956	0.1715	0.45	160	52.58947	340
173	174	0.1956	0.1715	0.65	160	52.58947	340
174	175	0.1956	0.1715	0.29	160	52.58947	340
173	176	0.64	0.26	1	64	21.03579	210
146	177	0.114	0.076	1	1200	394.4211	340
146	178	0.1956	0.1036	0.79	0	0	340
178	179	0.494	0.118	0.02	252	82.82842	200
178	180	0.211	0.096	0.11	64	21.03579	330
178	181	0.1956	0.1036	0.38	160	52.58947	340
181	182	0.1956	0.1036	0.32	252	82.82842	340
182	183	0.1956	0.1036	0.46	252	82.82842	340
183	184	0.1956	0.1036	0.3	252	82.82842	340
184	185	0.1956	0.1036	0.33	400	131.4737	340
185	186	0.1956	0.1036	0.11	252	82.82842	340
186	187	0.1956	0.1036	0.24	252	82.82842	340
187	188	0.236	0.075	1	400	131.4737	340
1	189	0.00635	0.0064	1.93	0	0	360

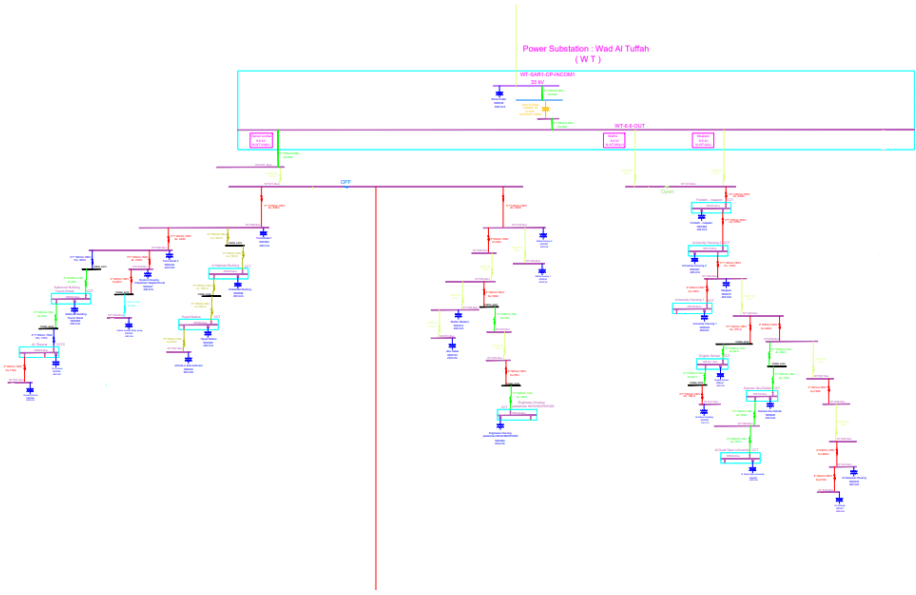
189	190	0.221	0.096	0.45	160	52.58947	330
190	191	0.1785	0.094	1	100	32.86842	340
191	192	0.0151	0.008	1	160	52.58947	340
192	193	0.1956	0.1036	0.55	0	0	340
193	194	0.16266	0.12	0.3	252	82.82842	380
193	195	0.494	0.187	0.02	160	52.58947	200
193	196	0.1956	0.1036	0.34	160	52.58947	340
193	197	0.1956	0.1715	0.16	0	0	340
197	198	0.1956	0.1715	0.02	160	52.58947	340
197	199	0.085	0.059	1	252	82.82842	340
199	200	0.1956	0.1036	0.38	252	82.82842	340
200	201	0.1956	0.1036	0.2	252	82.82842	340
189	202	0.1956	0.1715	1.73	252	82.82842	340
202	203	0.1184	0.06	1	160	52.58947	290
203	204	0.1956	0.1715	0.33	160	52.58947	340
202	205	0.16266	0.12	0.89	160	52.58947	380
205	206	0.1956	0.1036	0.36	252	82.82842	340
202	207	0.1956	0.1715	0.88	0	0	340
207	208	0.1956	0.1715	0.03	252	82.82842	340
207	209	0.04	0.011	1	160	52.58947	340
209	210	0.073	0.048	1	160	52.58947	340
210	211	0.1956	0.1036	0.13	400	131.4737	340
207	212	0.1956	0.1036	0.46	252	82.82842	340
207	213	0.1956	0.1715	0.45	100	32.86842	340
213	214	0.1956	0.1036	0.365	160	52.58947	340
214	215	0.1825	0.08	1	252	82.82842	340
215	216	0.1956	0.1036	0.14	160	52.58947	340
216	217	0.098	0.048	1	160	52.58947	340
217	218	0.16266	0.12	0.83	160	52.58947	380
217	219	0.11	0.08	1	160	52.58947	340
189	220	0.1956	0.1715	0.2	0	0	340
220	221	0.1956	0.1715	0.01	252	82.82842	340
221	222	0.494	0.187	0.07	400	131.4737	200
220	223	0.1956	0.1036	2.84	160	52.58947	340
223	224	0.1956	0.1036	0.15	1200	394.4211	340
189	225	0.1956	0.1036	0.19	252	82.82842	340
225	226	0.1956	0.1036	0.32	160	52.58947	340
226	227	0.1956	0.1036	0.17	252	82.82842	340
227	228	0.1956	0.1715	0.16	100	32.86842	340
228	229	0.1956	0.1715	0.3	252	82.82842	340
229	230	0.015	0.01	1	100	32.86842	360
189	231	0.1956	0.1036	0.25	0	0	340
231	232	0.1588	0.16	0.05	252	82.82842	360
231	233	0.2114	0.09	0.6	252	82.82842	330
233	234	0.1956	0.1715	0.35	160	52.58947	340
234	235	0.1956	0.1715	0.55	100	32.86842	340
233	236	0.1956	0.1036	0.85	252	82.82842	340
236	237	0.1956	0.1036	0.42	160	52.58947	340
189	238	0.325	0.1715	0.68	252	82.82842	270

238	239	0.325	0.1715	0.45	252	82.82842	270
239	240	0.4103	0.175	0.04	0	0	370
240	241	0.4103	0.175	0.1	252	82.82842	370
241	242	0.16266	0.12	0.08	160	52.58947	380
240	243	0.4103	0.175	0.1	160	52.58947	370
243	244	0.4103	0.175	0.06	0	0	370
244	245	0.4103	0.175	0.21	252	82.82842	370
244	246	0.16266	0.12	0.27	160	52.58947	380
246	247	0.16266	0.12	0.22	160	52.58947	380

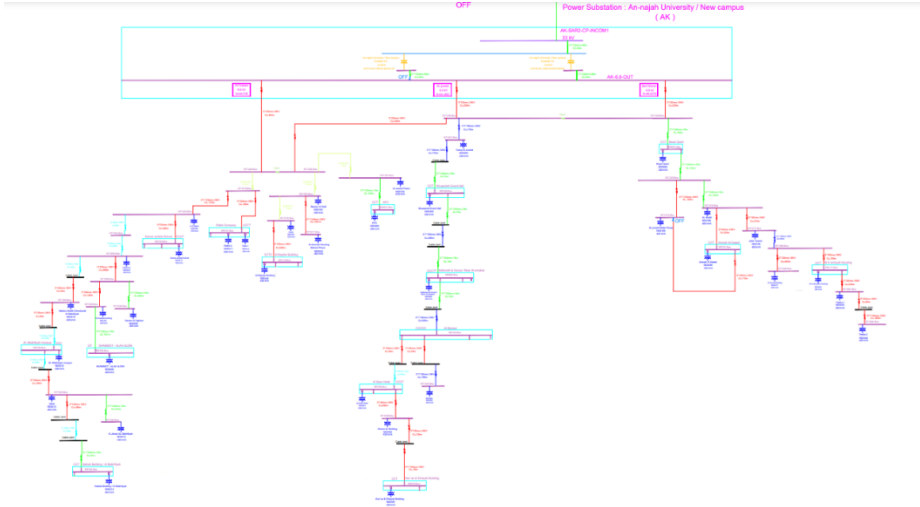
Appendix N
Substations single line diagrams



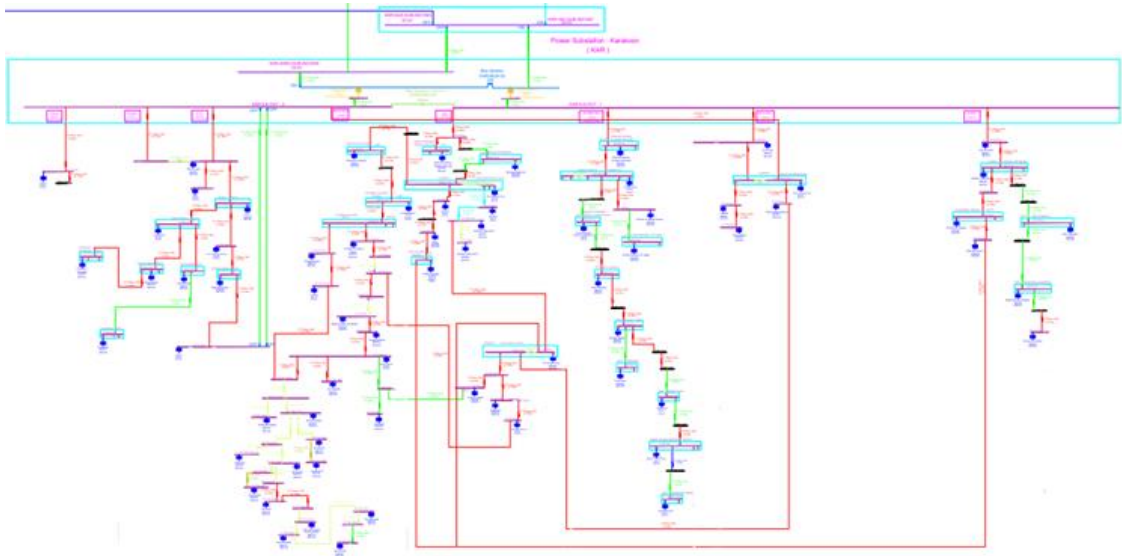
The Four substations



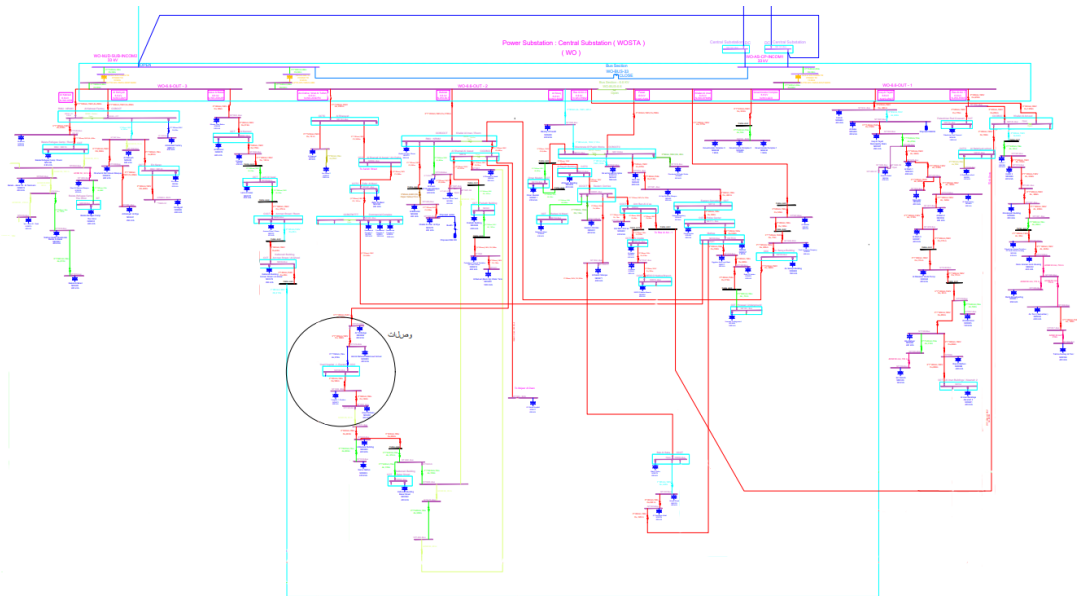
Wad AL-Tuffah Substation



An- Najah National University Substation



Karakoon Substation



Central Substation

Appendix O

Part of Nablus transformers names

Transformer number	Transformers capacity (KVA)	Name of transformer
1	0	Slak bus
2	0	-
3	400	Fraitek
4	400	University housing 2
5	400	Majeen
6	250	University housing 1
7	0	-
8	400	English School
9	630	Al Zahra'a
10	0	-
11	400	Sameer abu esha
12	630	Al Quds university
13	400	Nemsawy housing
14	630	Al zahrawi
15	400	Tunis street 1
16	250	Al Nakheel bulding
17	400	Hayat Nablus
18	250	Alf Leyla w lela
19	400	Tunis street 2
20	400	Doctors housing
21	630	Kamal junblat
22	250	Kalbuneh tunis
23	630	Al ruzana
24	630	Church
25	0	-
26	250	Basel Qadri
27	0	-
28	400	Al jenad pump
29	400	Al khilily building
30	160	Amena al saeed
31	400	Jalal yaseen
32	0	-
33	630	Al fureki
34	630	Al ameryah
35	400	Teabah 1
36	400	Teabah 2
37	0	-
38	400	Tareq al jenady
39	250	Khwairah
40	630	Kalbunah ,suosa
41	0	-
42	630	bulebleh
43	1000	Al qaser hotel
44	630	Shana'a building
45	400	Kanaan & shtayah building
46	0	-

47	630	Junade presion
48	400	KFC
49	400	Bashar al saife
50	400	Al shunner housing
51	630	Orabi
52	400	Al byader building
53	0	-
54	630+400	Paltel 1+2
55	630	Al farduse
56	630	Kamal junblat pump
57	630	Affunah
58	0	-
59	250	Hamzah al aghbar
60	400	Al khateeb building
61	400	Shammut -ala'a al din
62	400	Health center
63	400	masged Fatima
64	400	Jaber
65	250	Al johari
66	250	Salhab buiding
67	0	-
68	400	-
69	400	Yosra abu zant
70	400	Al dardouk
71	400	Sami al shaka'a
72	400	Salhab & baarah
73	400	Johari street 15
74	250	Masjed Al shuhada'a
75	400	Ain al asal
76	400	Kshekah
77	400	Kroom Ashour
78	400	Kalbunah old najah
79	630	Jawwal
80	400	masjed Al rawnaq
81	630	Al afuori building
82	630	Al dimashqi
83	400	Meri building
84	250	Amena / salhab
85	250	Shadi abdo
86	630	Rafidia tower
87	630	Maher zagha building
88	630	Al leddawi building
89	0	-
90	400	Khalbunah building
91	250	Mansour khwireh
92	630	24 street pump
93	630	24 street
94	400	Southern tank 1
95	400	Southern tank 2

96	400	Al tawon Crossroads
97	630	Abu raeed
98	250	Plaza
99	1000	Nablus specialist hospital
100	250	Retirement Authority
101	630	Najah parking
102	0	-
103	630	Najah library
104	400	Masged al nour
105	630	Tel street
106	0	-
107	630	Nablus al jadedah
108	400	Hashem albarqawi
109	250	Zeid al alem
110	250	Al Bahar neighborhood
111	250	Al andalos
112	0	-
113	250	Masjed al barqawi
114	160	Ain alfawwar
115	250	Iraq Burin 1
116	400	Iraq Burin 2
117	0	-
118	400	Tel farm
119	250	Tel east
120	0	-
121	250	Tel alluhuf
122	400	Tel downtown
123	400	Tel olive squwisser
124	400	Tel – sarra street
125	160	Tel tank
126	400	Sarra downtown
127	0	-
128	250	Othman bank
129	630	Rivoli
130	400	B.o.p
131	630	Aluol & abu salha
132	400	Sukutian building
133	630	Al Koni
134	630	Touqan building
135	400	Naya building
136	250	Al awartani
137	400	Diya'a alfori
138	400	Al shaka'a 1
139	630	Al karakon
140	630	Ras Al Ain pump
141	400	Masjed salah al din
142	0	-
143	400	Al amal hospital
144	400	Al rabyah

145	630	Electricity housing
146	0	-
147	0	-
148	1000	Red Cross
149	0	-
150	630	Al salahiya school
151	630	Al hurush 1
152	400	Al hurush 2
153	250	Khudarya building
154	630	Trade Federation
155	400	Al qahera amman bank
156	0	-
157	250	Podcast Radio
158	400	Al tur
159	400	Al tur Tahini
160	400	Fattoh
161	250	Al baba
162	250	Iraq al tayeh
163	630	Askar al balad
164	400	Water street 1
165	0	-
166	400	Stone saw
167	630	Al safa factory
168	400	Water Street 2
169	400	Water Street 3
170	400	Al masaken pump
171	0	-
172	160	Koza
173	400	Abu rabe
174	400	Ras al badan
175	400	Eman Alawneh Building 2
176	160	Ein kakoub
177	3000	Governorate 1+2+3
178	0	-
179	630	public works
180	160	Governorate out side
181	400	Hala car
182	630	Mazoz al masri
183	630	Eastern Cemetery
184	630	Police center
185	1000	Malhas
186	630	Arab Egypt bank
187	630	Jordan bank
188	1000	Al huwwari
189	0	-
190	400	Ministry of Health
191	250	Omar mosleh
192	400	Al sayeh building
193	0	-

194	630	Badawi al masri
195	400	Eastern Complex
196	400	Masged al salahi
197	0	-
198	400	Amro ibn al aas
199	630	Al aqabah
200	630	Al bash
201	630	NEDCO
202	630	Khala al eman room
203	400	Al ballotat
204	400	Al khala al olya
205	400	Masjed al imam al Husse
206	630	Northern tank
207	0	-
208	630	Middle eastern
209	400	Fratik building
210	400	Trade Federation 2
211	1000	Reserve tank
212	630	National Hospital
213	250	Titi bridge
214	400	Abd alrahem school
215	630	Al Etihad hospital
216	400	Emad al din
217	400	Al Badawi building
218	400	Sama Nablus
219	400	Kalbunah building
220	0	-
221	630	Al sharayet
222	1000	Ein difna
223	400	Zafer al masri
224	3000	Commercial complex 1+2+3
225	630	Maser station
226	400	Al nammor
227	630	Al barakah
228	250	Lami al esawi
229	630	Amman street room
230	250	Kalbunah building
231	0	-
232	630	Sulfur factory
233	630	Al daheya
234	400	Masged omahat al momenen
235	250	Al daheya al olya
236	630	Ein sreen
237	400	Hindiyah
238	630	Sharaab
239	630	Balata camp
240	0	-
241	630	Masged abd al Rahman
242	400	Balata the clinic

243	400	Balata -askar al tammam
244	0	-
245	630	Balata -askar qadri
246	400	Balata health center
247	400	Balata al bald

Appendix P

Part of Nablus Network Parameters Before Installing PV

Bus number	Voltage magnitude (P.U)	Line from-to	Line load index	active power losses(KW)	Reactive power losses(KVAR)	Voltage stability index
1	1.000000	1-2	33.509177	13.292475	9.719070	0.016327
2	0.995901	2-3	17.521483	9.873196	5.237767	0.022004
3	0.990393	3-4	16.033970	15.085934	13.229501	0.040416
4	0.980282	4-5	14.531070	12.390399	11.095172	0.037160
5	0.971090	5-6	13.013915	9.938161	5.263771	0.030472
6	0.963635	6-7	12.058355	17.884682	7.633706	0.057500
7	0.949577	7-8	4.006913	0.783469	0.577993	0.008488
8	0.947558	8-9	2.452072	0.293406	0.216456	0.005209
9	0.946322	7-10	8.051442	3.803973	2.003115	0.019311
10	0.944971	10-11	4.026562	0.791172	0.583675	0.008571
11	0.942941	11-12	2.464109	0.296293	0.218586	0.005260
12	0.941700	10-13	4.024880	0.634800	0.326099	0.006475
13	0.943439	13-14	2.463250	0.356047	0.188581	0.005971
14	0.942028	2-15	15.988000	14.299423	6.027429	0.033706
15	0.987438	15-16	3.382783	1.695199	0.643853	0.018886
16	0.982753	16-17	2.445810	0.886171	0.336576	0.013738
17	0.979366	17-18	0.941467	0.131305	0.049871	0.005318
18	0.978063	15-19	11.113178	12.041516	6.355327	0.042356
19	0.976869	19-20	3.901123	1.483829	0.783142	0.015132
20	0.973159	20-21	2.387185	0.275929	0.134323	0.004575
21	0.972045	19-22	5.703867	0.545011	0.351368	0.003937
22	0.975906	22-23	4.760319	0.146841	0.093135	0.001269
23	0.975596	23-24	2.381824	0.332897	0.176320	0.005584
24	0.974233	1-25	61.752108	41.000847	32.082248	0.027619
25	0.993047	25-26	14.014725	9.584538	0.000000	0.022733
26	0.987371	26-27	13.082144	14.591609	6.161130	0.042321
27	0.976811	27-28	1.510498	0.222457	0.117389	0.005873
28	0.975375	27-29	11.571660	6.534213	4.820519	0.023762
29	0.970973	29-30	0.607156	0.021632	0.011457	0.001432
30	0.970626	29-31	9.447168	5.237131	2.773859	0.022170
31	0.965562	31-32	7.921324	2.989285	3.168115	0.017458
32	0.961329	32-33	2.417275	0.342880	0.181607	0.005751
33	0.959945	32-34	5.504050	1.777685	0.941555	0.013070
34	0.958177	34-35	3.082316	0.557499	0.295281	0.007354
35	0.956413	35-36	1.541869	0.139504	0.073888	0.003689
36	0.955530	25-37	26.086562	39.932251	21.150210	0.059330
37	0.978092	37-38	15.146922	10.930010	11.583883	0.032921
38	0.969975	38-39	13.628036	7.466079	7.800382	0.025115
39	0.963846	39-40	12.672685	3.902506	2.890745	0.013171
40	0.960662	40-41	10.257217	3.156315	2.525052	0.013407
41	0.957431	41-42	2.424323	0.063475	0.052896	0.001158
42	0.957154	41-43	7.832894	1.251626	0.589001	0.006396
43	0.955898	43-44	3.979704	0.929377	0.492246	0.009512
44	0.953619	44-45	1.546394	0.140324	0.074323	0.003710
45	0.952734	37-46	10.939820	7.022802	3.719644	0.025457

46	0.971827	46-47	3.923411	1.894743	0.808141	0.018746
47	0.967251	47-48	1.524395	0.113396	0.083656	0.003174
48	0.966483	46-49	7.016409	6.059720	2.584575	0.033401
49	0.963643	49-50	1.530279	0.137414	0.072782	0.003634
50	0.962767	49-51	3.957243	1.927562	0.822138	0.019067
51	0.959028	51-52	1.537657	0.138742	0.073485	0.003669
52	0.958148	25-53	21.651091	42.906495	15.666285	0.072923
53	0.974600	53-54	3.787676	0.841852	0.445889	0.008881
54	0.972432	53-55	17.863423	18.724898	9.917686	0.041548
55	0.964370	55-56	2.409631	0.340715	0.180461	0.005714
56	0.962990	55-57	13.047609	10.796627	4.596483	0.032252
57	0.956530	57-58	4.060532	0.967511	0.512444	0.009699
58	0.954205	58-59	0.965498	0.045489	0.033559	0.002038
59	0.953719	58-60	3.095033	0.562109	0.297722	0.007415
60	0.952433	60-61	1.548151	0.116958	0.086284	0.003274
61	0.951653	57-62	6.561174	0.506256	0.204052	0.003034
62	0.955804	62-63	5.019747	0.467168	0.199567	0.003688
63	0.954922	63-64	3.476896	0.466567	0.224852	0.005404
64	0.953630	64-65	0.966081	0.045544	0.033599	0.002041
65	0.953143	64-66	0.965874	0.027987	0.014833	0.001186
66	0.953347	1-67	41.287692	617.773675	414.747062	0.118303
67	0.969494	67-68	10.192846	6.096507	5.339119	0.026264
68	0.963085	68-69	8.663075	4.403867	2.332519	0.020497
69	0.958125	69-70	4.622331	0.354461	0.249982	0.003275
70	0.957340	70-71	3.083377	0.463933	0.342260	0.006481
71	0.955786	71-72	1.541920	0.043594	0.029671	0.001203
72	0.955498	69-73	2.503050	0.367645	0.194724	0.005974
73	0.956692	73-74	0.963053	0.054424	0.028826	0.002304
74	0.956140	67-75	4.576708	1.229126	0.107643	0.009451
75	0.967197	75-76	3.053439	0.547102	0.289774	0.007217
76	0.965449	76-77	1.527411	0.136900	0.072509	0.003620
77	0.964575	67-78	21.453560	27.007778	14.304733	0.050096
78	0.957196	78-79	19.914405	23.271522	20.380428	0.051663
79	0.944669	79-80	4.029043	0.952564	0.504528	0.009745
80	0.942362	80-81	2.465629	0.296659	0.218856	0.005267
81	0.941119	79-82	13.428999	1.265973	0.757420	0.003967
82	0.943731	82-83	10.970196	1.552453	0.866485	0.005892
83	0.942339	83-84	9.406744	5.192408	2.750171	0.022734
84	0.936952	84-85	0.983296	0.047181	0.034807	0.002114
85	0.936457	84-86	7.440673	0.553083	0.332182	0.003155
86	0.936213	86-87	4.962122	0.169896	0.118189	0.001493
87	0.935863	87-88	2.482645	0.293630	0.192117	0.005098
88	0.934669	67-89	57.674861	195.192544	103.384190	0.132104
89	0.936341	89-90	1.574984	0.145560	0.077096	0.003849
90	0.935440	89-91	56.100278	46.264446	24.548481	0.034209
91	0.928264	91-92	55.108542	17.766166	9.110854	0.013488
92	0.925123	92-93	7.342261	3.412427	1.552573	0.019106
93	0.920683	93-94	4.821903	1.471772	0.669622	0.012629
94	0.917767	94-95	1.608601	0.318507	0.136082	0.008162
95	0.915890	94-96	1.607990	0.252099	0.132643	0.006650
96	0.916239	92-97	7.545866	3.341245	1.769698	0.018596
97	0.920802	97-98	5.025834	1.482199	0.785050	0.012462

98	0.917924	98-99	4.022686	0.949560	0.502937	0.010012
99	0.915621	92-100	37.717606	83.479211	44.214960	0.091603
100	0.903428	100-101	36.698769	79.030201	41.858532	0.091244
101	0.882327	101-102	34.069751	68.112688	36.076046	0.086819
102	0.862742	102-103	2.694912	0.426166	0.373222	0.007833
103	0.861049	102-104	31.375621	57.766329	50.589837	0.090041
104	0.842863	104-105	29.627904	51.510047	45.110793	0.086814
105	0.824155	105-106	26.812461	42.185526	22.343663	0.073212
106	0.808784	106-107	8.163022	3.910138	2.071014	0.022984
107	0.804109	107-108	2.984634	0.522724	0.276862	0.008483
108	0.802400	108-109	1.148515	0.077404	0.040997	0.003275
109	0.801743	107-110	1.146689	0.128202	0.067454	0.005414
110	0.803019	107-111	1.145958	0.064083	0.047276	0.002870
111	0.803532	106-112	18.649586	42.801138	18.291174	0.104745
112	0.787019	112-113	1.172038	0.169044	0.072241	0.006936
113	0.785652	112-114	17.477572	37.590586	16.064431	0.100966
114	0.766625	114-115	3.147356	1.219014	0.520949	0.019066
115	0.762953	115-116	1.940455	0.557801	0.924812	0.019290
116	0.759256	114-117	13.561509	22.632521	9.672064	0.080854
117	0.750802	117-118	1.968323	0.476771	0.203749	0.012193
118	0.748506	117-119	11.593198	16.539538	7.068213	0.070759
119	0.737277	119-120	10.344263	13.167885	5.627329	0.064396
120	0.725210	120-121	1.272812	0.239994	0.090885	0.009695
121	0.723448	120-122	9.071452	3.209358	1.481242	0.018592
122	0.721824	122-123	7.030370	2.505903	1.186226	0.018881
123	0.718400	123-124	4.979560	3.051398	1.304023	0.032078
124	0.712592	124-125	2.912035	1.043542	0.445960	0.018975
125	0.709196	125-126	2.081064	0.267645	0.129925	0.006978
126	0.707956	67-127	31.496148	0.297602	1.488011	0.000865
127	0.969284	127-128	19.632072	28.906368	19.829768	0.061011
128	0.954267	128-129	2.435216	0.347989	0.184313	0.005836
129	0.952873	128-130	16.231926	15.460737	8.188816	0.038592
130	0.944970	130-131	6.512104	2.488472	2.181866	0.017251
131	0.940877	131-132	4.045840	0.960522	0.508743	0.009824
132	0.938560	132-133	2.476092	0.359769	0.190553	0.006033
133	0.937142	130-134	8.160725	3.907937	2.069848	0.019684
134	0.940296	134-135	2.550994	0.381864	0.334814	0.006807
135	0.938694	135-136	0.981470	0.047006	0.034678	0.002106
136	0.938199	134-137	3.141945	0.579278	0.507905	0.008381
137	0.938322	137-138	1.571800	0.144972	0.127110	0.004205
138	0.937335	127-139	2.397380	0.337259	0.178630	0.005657
139	0.967911	127-140	9.466780	5.258897	2.785387	0.022254
140	0.963861	140-141	7.059336	2.924273	2.563971	0.018337
141	0.959422	141-142	5.523724	1.790416	0.948298	0.013145
142	0.956259	142-143	1.542118	0.139549	0.073912	0.003690
143	0.955376	142-144	3.981607	0.930265	0.815647	0.010439
144	0.953757	144-145	2.436871	0.348462	0.305528	0.006411
145	0.952226	67-146	81.052219	38.628438	29.168412	0.020364
146	0.964532	146-147	19.884576	23.201859	12.288919	0.046692
147	0.953139	147-148	3.873356	0.880369	0.466290	0.009285
148	0.950921	147-149	16.011245	15.043203	7.967668	0.038124
149	0.943966	149-150	2.461869	0.355648	0.188370	0.005964

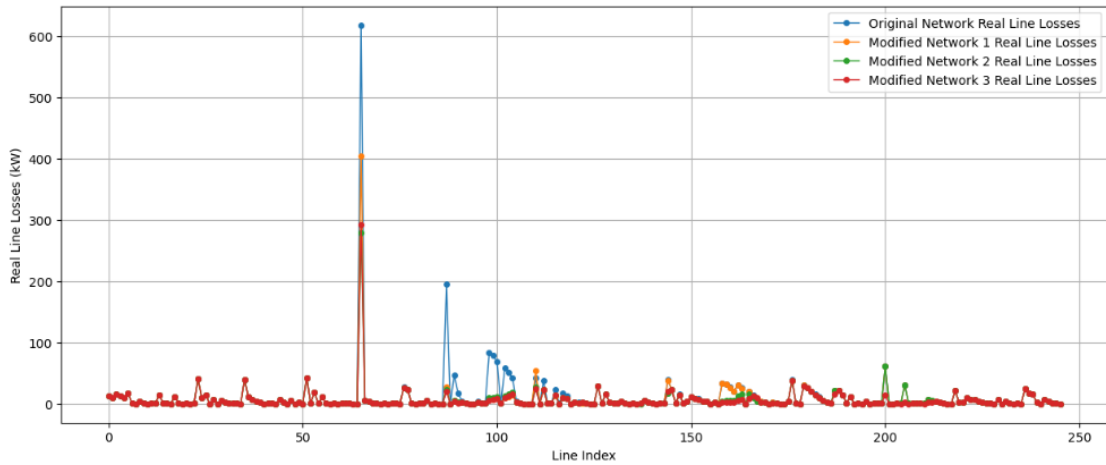
150	0.942557	149-151	13.549385	3.712106	1.817500	0.011174
151	0.941322	151-152	11.084299	11.979014	6.321234	0.044300
152	0.930779	152-153	9.501431	8.802025	4.644761	0.038453
153	0.921743	153-154	8.502439	7.785791	2.819367	0.036613
154	0.913228	154-155	5.961505	3.465105	1.828509	0.024670
155	0.907560	155-156	4.338140	4.640879	1.055772	0.041674
156	0.898004	156-157	1.027993	0.260599	0.059285	0.010060
157	0.895740	156-158	3.310148	2.702015	0.614692	0.032213
158	0.890713	158-159	1.656077	0.194175	0.063354	0.004839
159	0.889634	146-160	23.925585	33.590406	17.791237	0.056144
160	0.950799	160-161	22.376222	31.754096	14.420025	0.056291
161	0.937225	161-162	21.393837	26.857617	14.225200	0.051711
162	0.924948	162-163	20.398393	20.304577	14.979400	0.044119
163	0.914630	163-164	17.861502	31.105741	16.414260	0.073009
164	0.897620	164-165	16.220223	25.651824	13.536270	0.067630
165	0.882178	165-166	1.673091	0.272925	0.144021	0.007200
166	0.880587	165-167	14.547168	20.632959	10.887854	0.061791
167	0.868333	167-168	11.874897	13.748785	7.255128	0.051373
168	0.857034	168-169	10.155838	6.052316	3.205624	0.026970
169	0.851216	169-170	8.425026	4.165168	2.206091	0.022550
170	0.846390	170-171	6.684342	2.621848	1.388668	0.018014
171	0.842561	171-172	0.699733	0.023893	0.017627	0.001673
172	0.842208	171-173	5.984611	2.101658	1.842711	0.017782
173	0.838798	173-174	3.524523	0.728939	0.639126	0.010534
174	0.836583	174-175	1.763429	0.182476	0.159993	0.005290
175	0.835476	173-176	0.703645	0.095062	0.038619	0.006060
176	0.837526	146-177	11.503632	4.525807	3.017205	0.016464
177	0.960546	146-178	25.739634	38.877189	20.591395	0.060269
178	0.949774	178-179	2.451567	0.890709	0.212760	0.013668
179	0.946517	178-180	0.620731	0.024390	0.011097	0.001580
180	0.949398	178-181	22.667440	30.150538	15.969304	0.053977
181	0.936780	181-182	21.094743	26.111907	13.830233	0.050954
182	0.924690	182-183	18.585338	20.268939	10.735491	0.045533
183	0.914042	183-184	16.046683	15.109867	8.002977	0.039822
184	0.904850	184-185	13.482229	10.666294	5.649428	0.033843
185	0.897128	185-186	9.376626	5.159211	2.732588	0.023798
186	0.891759	186-187	6.774519	2.693067	1.426389	0.017323
187	0.887880	187-188	4.161045	1.225852	0.389572	0.012145
188	0.885176	1-189	53.384239	21.722902	21.887055	0.009022
189	0.997739	189-190	18.178828	21.910148	9.517530	0.045381
190	0.986288	190-191	16.685058	14.907847	7.850631	0.035036
191	0.977572	191-192	15.743124	1.122742	0.594830	0.002847
192	0.976876	192-193	14.234956	11.890561	6.297864	0.033110
193	0.968722	193-194	2.398365	0.280694	0.207078	0.004984
194	0.967513	193-195	1.524193	0.344293	0.130330	0.008697
195	0.966611	193-196	1.522242	0.135974	0.072019	0.003596
196	0.967850	193-197	8.790172	4.534034	3.975393	0.022691
197	0.963195	197-198	1.531128	0.137567	0.120617	0.003991
198	0.962233	197-199	7.259044	1.343690	0.932679	0.007832
199	0.961306	199-200	4.845191	1.377564	0.729630	0.011510
200	0.958531	200-201	2.424351	0.344890	0.182672	0.005784
201	0.957143	189-202	32.253284	61.043299	53.522115	0.079930

202	0.977387	202-203	3.019456	0.323840	0.164108	0.004251
203	0.976347	203-204	1.510462	0.133878	0.117383	0.003884
204	0.975398	202-205	3.892663	0.739428	0.545502	0.008012
205	0.975425	205-206	2.382243	0.333014	0.176382	0.005586
206	0.974061	202-207	22.967263	30.953417	27.139626	0.058314
207	0.962924	207-208	2.413597	0.341837	0.299719	0.006289
208	0.961408	207-209	6.905933	0.572303	0.157383	0.003117
209	0.962173	209-210	5.374710	0.632636	0.415980	0.004940
210	0.960984	210-211	3.841591	0.865989	0.458673	0.009134
211	0.958784	207-212	2.413259	0.341742	0.181004	0.005731
212	0.961542	207-213	11.234532	7.406279	6.493747	0.029143
213	0.955857	213-214	10.271195	6.190590	3.278861	0.024464
214	0.949975	214-215	8.720310	4.163398	1.825051	0.019024
215	0.945435	215-216	6.265936	2.303892	1.220262	0.015122
216	0.941847	216-217	4.701667	0.649907	0.318322	0.005656
217	0.940513	217-218	1.567804	0.119946	0.088488	0.003357
218	0.939723	217-219	1.567375	0.081070	0.058960	0.002264
219	0.939981	189-220	19.009492	21.204650	18.592012	0.047393
220	0.985774	220-221	6.134314	2.208117	1.936053	0.015582
221	0.981919	221-222	3.771135	2.107620	0.797824	0.021161
222	0.976697	220-223	12.875179	9.727397	5.152139	0.029697
223	0.978401	223-224	11.369352	7.585104	4.017468	0.026438
224	0.971891	189-225	10.443281	6.399765	3.389650	0.023837
225	0.991758	225-226	8.103550	3.853370	2.040947	0.018631
226	0.987117	226-227	6.611021	2.564645	1.358370	0.015283
227	0.983331	227-228	4.251237	1.060524	0.929857	0.010840
228	0.980659	228-229	3.312262	0.643783	0.564462	0.008471
229	0.978577	229-230	0.941013	0.003985	0.002657	0.000175
230	0.978535	189-231	10.984525	7.080317	3.750107	0.025060
231	0.991449	231-232	2.343384	0.261613	0.263590	0.004977
232	0.990214	231-233	8.641141	4.735529	2.016072	0.020837
233	0.986257	233-234	2.431802	0.347014	0.304258	0.006187
234	0.984729	234-235	0.935652	0.051371	0.045042	0.002386
235	0.984141	233-236	3.856555	0.872749	0.462253	0.008936
236	0.984049	236-237	1.498490	0.131764	0.069789	0.003484
237	0.983191	189-238	15.900303	24.649914	13.007570	0.059737
238	0.982609	238-239	13.538801	17.871666	9.430741	0.051741
239	0.969730	239-240	11.145924	15.291670	6.522160	0.052911
240	0.956729	240-241	3.985971	1.955650	0.834118	0.019343
241	0.952080	241-242	1.548726	0.117045	0.086348	0.003276
242	0.951299	240-243	7.159954	6.310202	2.691409	0.034613
243	0.948377	243-244	5.606456	3.869008	1.650198	0.027391
244	0.941838	244-245	2.471312	0.751758	0.320638	0.012204
245	0.938955	244-246	3.135144	0.479642	0.353849	0.006699
246	0.940258	246-247	1.568231	0.120011	0.088537	0.003359
247	0.939467					

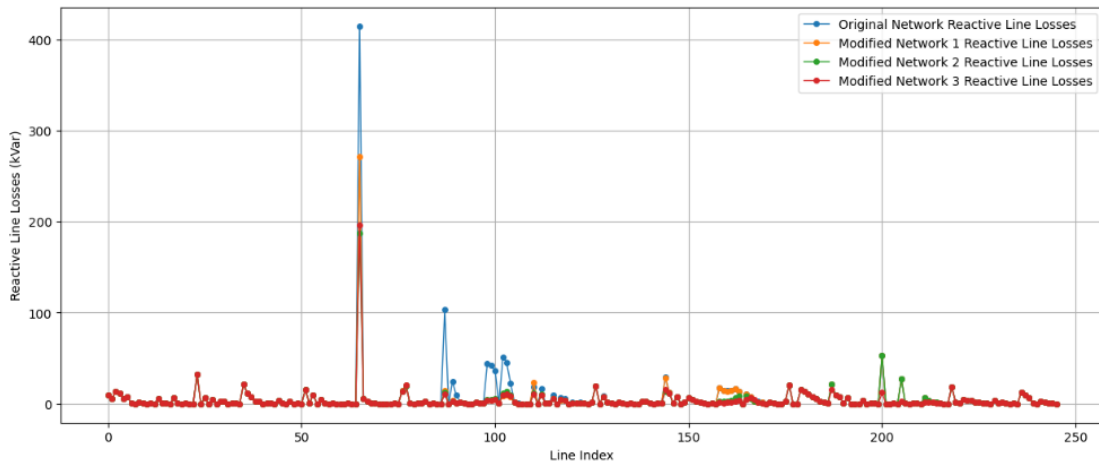
Appendix Q

Nablus Network Results Summary of random distribution of PV Using Improved ABC

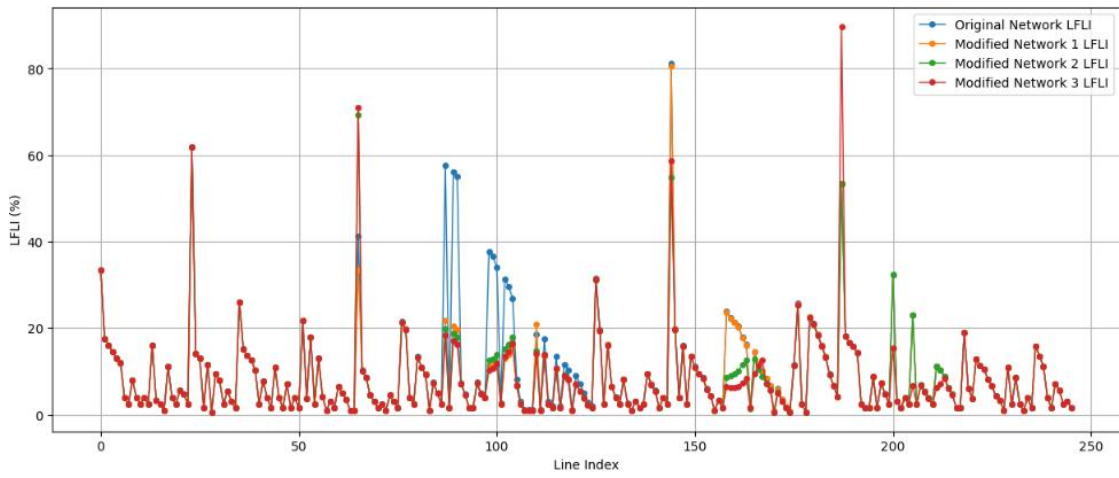
Comparison for Nablus Network Active Power Losses in Each Line Using Improved ABC



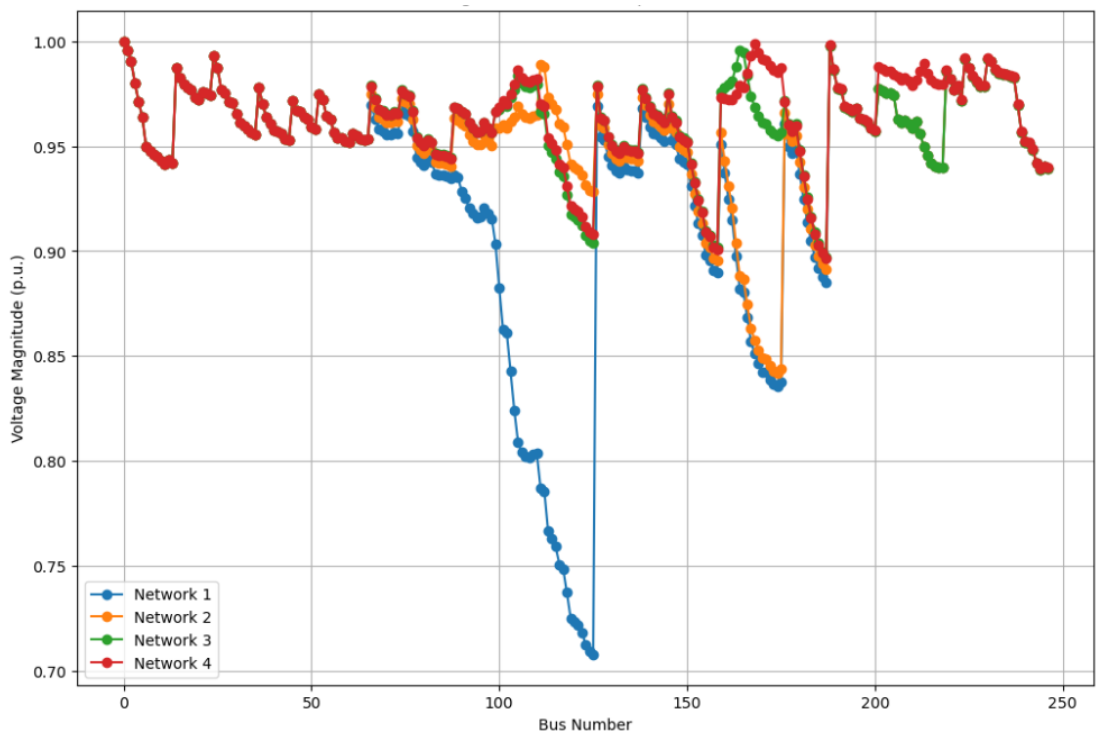
Comparison of Nablus Network Reactive Power Losses in Each Line Using Improved ABC



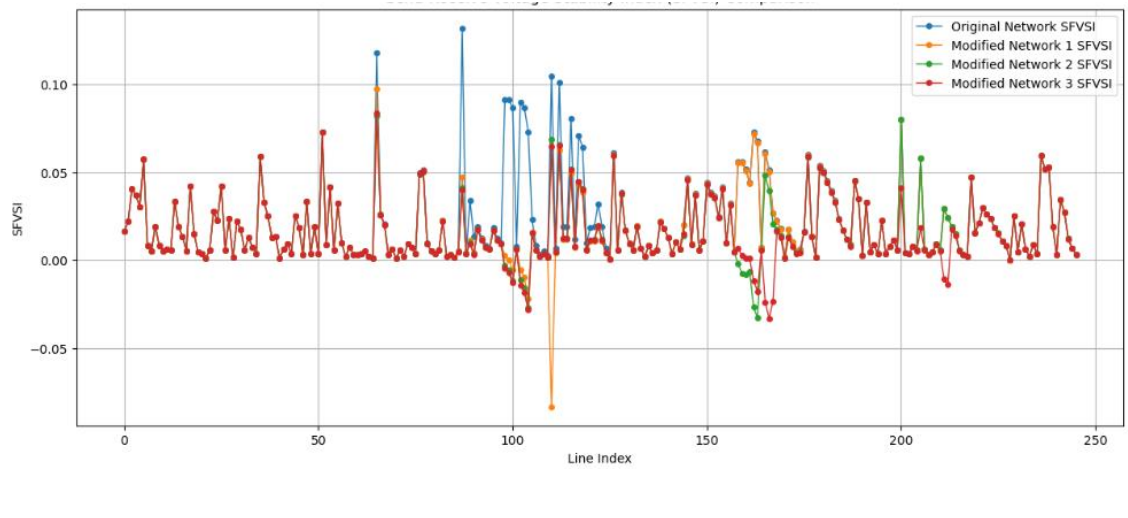
Comparison of Nablus network - Line Flow Limit Index in Each Line Using Improved ABC



Comparison of IEEE 33 Voltage Profile in Each Bus Using Improved ABC



Comparison of IEEE 33 Voltage Stability Index in Each Bus Using Improved ABC



Appendix R

Part of Nablus Network Parameters After Installing PV Using improved ABC Algorithm

Bus number	Voltage profile			
	original	One PV	two PV	three PV
1				
2	1.000000	1.000000	1.000000	1.000000
3	0.995901	0.995901	0.995901	0.995901
4	0.990393	0.990393	0.990393	0.990393
5	0.980282	0.980282	0.980282	0.980282
6	0.971090	0.971090	0.971090	0.971090
7	0.963635	0.963635	0.963635	0.963635
8	0.949577	0.949577	0.949577	0.949577
9	0.947558	0.947558	0.947558	0.947558
10	0.946322	0.946322	0.946322	0.946322
11	0.944971	0.944971	0.944971	0.944971
12	0.942941	0.942941	0.942941	0.942941
13	0.941700	0.941700	0.941700	0.941700
14	0.943439	0.943439	0.943439	0.943439
15	0.942028	0.942028	0.942028	0.942028
16	0.987438	0.987438	0.987438	0.987438
17	0.982753	0.982753	0.982753	0.982753
18	0.979366	0.979366	0.979366	0.979366
19	0.978063	0.978063	0.978063	0.978063
20	0.976869	0.976869	0.976869	0.976869
21	0.973159	0.973159	0.973159	0.973159
22	0.972045	0.972045	0.972045	0.972045
23	0.975906	0.975906	0.975906	0.975906
24	0.975596	0.975596	0.975596	0.975596
25	0.974233	0.974233	0.974233	0.974233
26	0.993047	0.993047	0.993047	0.993047
27	0.987371	0.987371	0.987371	0.987371
28	0.976811	0.976811	0.976811	0.976811
29	0.975375	0.975375	0.975375	0.975375
30	0.970973	0.970973	0.970973	0.970973
31	0.970626	0.970626	0.970626	0.970626
32	0.965562	0.965562	0.965562	0.965562
33	0.961329	0.961329	0.961329	0.961329
34	0.959945	0.959945	0.959945	0.959945
35	0.958177	0.958177	0.958177	0.958177
36	0.956413	0.956413	0.956413	0.956413
37	0.955530	0.955530	0.955530	0.955530
38	0.978092	0.978092	0.978092	0.978092
39	0.969975	0.969975	0.969975	0.969975
40	0.963846	0.963846	0.963846	0.963846
41	0.960662	0.960662	0.960662	0.960662
42	0.957431	0.957431	0.957431	0.957431
43	0.957154	0.957154	0.957154	0.957154
44	0.955898	0.955898	0.955898	0.955898
45	0.953619	0.953619	0.953619	0.953619
46	0.952734	0.952734	0.952734	0.952734

47	0.971827	0.971827	0.971827	0.971827
48	0.967251	0.967251	0.967251	0.967251
49	0.966483	0.966483	0.966483	0.966483
50	0.963643	0.963643	0.963643	0.963643
51	0.962767	0.962767	0.962767	0.962767
52	0.959028	0.959028	0.959028	0.959028
53	0.958148	0.958148	0.958148	0.958148
54	0.974600	0.974600	0.974600	0.974600
55	0.972432	0.972432	0.972432	0.972432
56	0.964370	0.964370	0.964370	0.964370
57	0.962990	0.962990	0.962990	0.962990
58	0.956530	0.956530	0.956530	0.956530
59	0.954205	0.954205	0.954205	0.954205
60	0.953719	0.953719	0.953719	0.953719
61	0.952433	0.952433	0.952433	0.952433
62	0.951653	0.951653	0.951653	0.951653
63	0.955804	0.955804	0.955804	0.955804
64	0.954922	0.954922	0.954922	0.954922
65	0.953630	0.953630	0.953630	0.953630
66	0.953143	0.953143	0.953143	0.953143
67	0.953347	0.953347	0.953347	0.953347
68	0.969494	0.975002	0.978977	0.978570
69	0.963085	0.968631	0.972633	0.972223
70	0.958125	0.963699	0.967722	0.967309
71	0.957340	0.962919	0.966944	0.966531
72	0.955786	0.961374	0.965406	0.964993
73	0.955498	0.961088	0.965121	0.964708
74	0.956692	0.962274	0.966303	0.965890
75	0.956140	0.961726	0.965757	0.965343
76	0.967197	0.972719	0.976704	0.976295
77	0.965449	0.970981	0.974972	0.974563
78	0.964575	0.970111	0.974106	0.973697
79	0.957196	0.962778	0.966806	0.966393
80	0.944669	0.950327	0.954409	0.953990
81	0.942362	0.948034	0.952126	0.951706
82	0.941119	0.946799	0.950896	0.950476
83	0.943731	0.949395	0.953481	0.953062
84	0.942339	0.948011	0.952103	0.951684
85	0.936952	0.942657	0.946773	0.946351
86	0.936457	0.942165	0.946282	0.945860
87	0.936213	0.941922	0.946041	0.945618
88	0.935863	0.941574	0.945695	0.945272
89	0.934669	0.940387	0.944513	0.944090
90	0.936341	0.963284	0.968734	0.968697
91	0.935440	0.962408	0.967862	0.967825
92	0.928264	0.960570	0.966389	0.966443
93	0.925123	0.959559	0.965525	0.965613
94	0.920683	0.955280	0.961273	0.961362
95	0.917767	0.952471	0.958481	0.958570
96	0.915890	0.950663	0.956685	0.956774
97	0.916239	0.950999	0.957018	0.957107
98	0.920802	0.955395	0.961387	0.961476

99	0.917924	0.952623	0.958632	0.958721
100	0.915621	0.950404	0.956427	0.956516
101	0.903428	0.958908	0.966291	0.966728
102	0.882327	0.958851	0.967658	0.968425
103	0.862742	0.960228	0.970454	0.971531
104	0.861049	0.958707	0.968950	0.970029
105	0.842863	0.961540	0.973205	0.974981
106	0.824155	0.963899	0.977014	0.979458
107	0.808784	0.969104	0.983634	0.986312
108	0.804109	0.965211	0.979799	0.982488
109	0.802400	0.963789	0.978398	0.981091
110	0.801743	0.963241	0.977859	0.980553
111	0.803019	0.964304	0.978905	0.981596
112	0.803532	0.964730	0.979325	0.982016
113	0.787019	0.988885	0.966355	0.970138
114	0.785652	0.987797	0.965243	0.969030
115	0.766625	0.973125	0.950191	0.954044
116	0.762953	0.970241	0.947237	0.951102
117	0.759256	0.967341	0.944266	0.948143
118	0.750802	0.960957	0.937707	0.941613
119	0.748506	0.959166	0.935870	0.939785
120	0.737277	0.950582	0.927059	0.931012
121	0.725210	0.941337	0.917570	0.921566
122	0.723448	0.939980	0.916179	0.920180
123	0.721824	0.938743	0.914909	0.918916
124	0.718400	0.936125	0.912222	0.916240
125	0.712592	0.931690	0.907668	0.911707
126	0.709196	0.929099	0.905008	0.909059
127	0.707956	0.928153	0.904037	0.908092
128	0.969284	0.974794	0.978770	0.978362
129	0.954267	0.959867	0.963908	0.963493
130	0.952873	0.958481	0.962527	0.962112
131	0.944970	0.950626	0.954706	0.954288
132	0.940877	0.946557	0.950655	0.950235
133	0.938560	0.944255	0.948363	0.947942
134	0.937142	0.942845	0.946960	0.946538
135	0.940296	0.945980	0.950081	0.949661
136	0.938694	0.944387	0.948495	0.948074
137	0.938199	0.943896	0.948005	0.947584
138	0.938322	0.944018	0.948127	0.947706
139	0.937335	0.943037	0.947150	0.946728
140	0.967911	0.973429	0.977410	0.977002
141	0.963861	0.969402	0.973401	0.972991
142	0.959422	0.964989	0.969006	0.968594
143	0.956259	0.961844	0.965875	0.965461
144	0.955376	0.960967	0.965001	0.964587
145	0.953757	0.959357	0.963398	0.962983
146	0.952226	0.957835	0.961882	0.961467
147	0.964532	0.970073	0.975495	0.974872
148	0.953139	0.958751	0.964241	0.963611
149	0.950921	0.956546	0.962049	0.961417
150	0.943966	0.949636	0.955183	0.954546

151	0.942557	0.948235	0.953790	0.953152
152	0.941322	0.947009	0.952572	0.951933
153	0.930779	0.936534	0.942163	0.941517
154	0.921743	0.927558	0.933243	0.932591
155	0.913228	0.919098	0.924838	0.924179
156	0.907560	0.913468	0.919244	0.918581
157	0.898004	0.903976	0.909814	0.909144
158	0.895740	0.901727	0.907579	0.906907
159	0.890713	0.896734	0.902619	0.901943
160	0.889634	0.895662	0.901554	0.900878
161	0.950799	0.956437	0.976001	0.973182
162	0.937225	0.942960	0.977788	0.972559
163	0.924948	0.930772	0.979751	0.972304
164	0.914630	0.920530	0.981325	0.972071
165	0.897620	0.903645	0.987849	0.974882
166	0.882178	0.888318	0.995875	0.979175
167	0.880587	0.886738	0.994466	0.977743
168	0.868333	0.874577	0.983722	0.984950
169	0.857034	0.863364	0.973816	0.993017
170	0.851216	0.857590	0.968718	0.998745
171	0.846390	0.852801	0.964492	0.994648
172	0.842561	0.849001	0.961140	0.991400
173	0.842208	0.848651	0.960831	0.991101
174	0.838798	0.845268	0.957848	0.988209
175	0.836583	0.843070	0.955910	0.986331
176	0.835476	0.841971	0.954940	0.985392
177	0.837526	0.844005	0.956733	0.987129
178	0.960546	0.966110	0.971554	0.970928
179	0.949774	0.955411	0.960925	0.960292
180	0.946517	0.952174	0.957707	0.957071
181	0.949398	0.955038	0.960554	0.959921
182	0.936780	0.942502	0.948099	0.947456
183	0.924690	0.930493	0.936167	0.935515
184	0.914042	0.919915	0.925657	0.924998
185	0.904850	0.910784	0.916586	0.915920
186	0.897128	0.903115	0.908967	0.908295
187	0.891759	0.897782	0.903669	0.902993
188	0.887880	0.893929	0.899841	0.899163
189	0.885176	0.891244	0.897174	0.896493
190	0.997739	0.997739	0.997739	0.998073
191	0.986288	0.986288	0.986288	0.986626
192	0.977572	0.977572	0.977572	0.977913
193	0.976876	0.976876	0.976876	0.977217
194	0.968722	0.968722	0.968722	0.969066
195	0.967513	0.967513	0.967513	0.967858
196	0.966611	0.966611	0.966611	0.966956
197	0.967850	0.967850	0.967850	0.968195
198	0.963195	0.963195	0.963195	0.963541
199	0.962233	0.962233	0.962233	0.962579
200	0.961306	0.961306	0.961306	0.961652
201	0.958531	0.958531	0.958531	0.958879
202	0.957143	0.957143	0.957143	0.957491

203	0.977387	0.977387	0.977387	0.987738
204	0.976347	0.976347	0.976347	0.986709
205	0.975398	0.975398	0.975398	0.985770
206	0.975425	0.975425	0.975425	0.985797
207	0.974061	0.974061	0.974061	0.984448
208	0.962924	0.962924	0.962924	0.983179
209	0.961408	0.961408	0.961408	0.981694
210	0.962173	0.962173	0.962173	0.982444
211	0.960984	0.960984	0.960984	0.981279
212	0.958784	0.958784	0.958784	0.979125
213	0.961542	0.961542	0.961542	0.981826
214	0.955857	0.955857	0.955857	0.985845
215	0.949975	0.949975	0.949975	0.989261
216	0.945435	0.945435	0.945435	0.984905
217	0.941847	0.941847	0.941847	0.981462
218	0.940513	0.940513	0.940513	0.980183
219	0.939723	0.939723	0.939723	0.979425
220	0.939981	0.939981	0.939981	0.979672
221	0.985774	0.985774	0.985774	0.986112
222	0.981919	0.981919	0.981919	0.982258
223	0.976697	0.976697	0.976697	0.977038
224	0.978401	0.978401	0.978401	0.978741
225	0.971891	0.971891	0.971891	0.972233
226	0.991758	0.991758	0.991758	0.992093
227	0.987117	0.987117	0.987117	0.987454
228	0.983331	0.983331	0.983331	0.983669
229	0.980659	0.980659	0.980659	0.980998
230	0.978577	0.978577	0.978577	0.978918
231	0.978535	0.978535	0.978535	0.978875
232	0.991449	0.991449	0.991449	0.991784
233	0.990214	0.990214	0.990214	0.990550
234	0.986257	0.986257	0.986257	0.986594
235	0.984729	0.984729	0.984729	0.985067
236	0.984141	0.984141	0.984141	0.984479
237	0.984049	0.984049	0.984049	0.984387
238	0.983191	0.983191	0.983191	0.983529
239	0.982609	0.982609	0.982609	0.982948
240	0.969730	0.969730	0.969730	0.970074
241	0.956729	0.956729	0.956729	0.957077
242	0.952080	0.952080	0.952080	0.952430
243	0.951299	0.951299	0.951299	0.951650
244	0.948377	0.948377	0.948377	0.948729
245	0.941838	0.941838	0.941838	0.942192
246	0.938955	0.938955	0.938955	0.939310
247	0.940258	0.940258	0.940258	0.940612

	Active losses				Reactive losses			
	original	One PV	two PV	three PV	original	One PV	two PV	three PV
1-2	13.292475	13.292475	13.292475	13.292475	9.719070	9.719070	9.719070	9.719070
2-3	9.873196	9.873196	9.873196	9.873196	5.237767	5.237767	5.237767	5.237767
3-4	15.085934	15.085934	15.085934	15.085934	13.22950	13.22950	13.22950	13.22950
4-5	12.390399	12.390399	12.390399	12.390399	11.09517	11.09517	11.09517	11.09517
5-6	9.938161	9.938161	9.938161	9.938161	5.263771	5.263771	5.263771	5.263771
6-7	17.884682	17.884682	17.884682	17.884682	7.633706	7.633706	7.633706	7.633706
7-8	0.783469	0.783469	0.783469	0.783469	0.577993	0.577993	0.577993	0.577993
8-9	0.293406	0.293406	0.293406	0.293406	0.216456	0.216456	0.216456	0.216456
7-10	3.803973	3.803973	3.803973	3.803973	2.003115	2.003115	2.003115	2.003115
10-11	0.791172	0.791172	0.791172	0.791172	0.583675	0.583675	0.583675	0.583675
11-12	0.296293	0.296293	0.296293	0.296293	0.218586	0.218586	0.218586	0.218586
10-13	0.634800	0.634800	0.634800	0.634800	0.326099	0.326099	0.326099	0.326099
13-14	0.356047	0.356047	0.356047	0.356047	0.188581	0.188581	0.188581	0.188581
2-15	14.299423	14.299423	14.299423	14.299423	6.027429	6.027429	6.027429	6.027429
15-16	1.695199	1.695199	1.695199	1.695199	0.643853	0.643853	0.643853	0.643853
16-17	0.886171	0.886171	0.886171	0.886171	0.336576	0.336576	0.336576	0.336576
17-18	0.131305	0.131305	0.131305	0.131305	0.049871	0.049871	0.049871	0.049871
15-19	12.041516	12.041516	12.041516	12.041516	6.355327	6.355327	6.355327	6.355327
19-20	1.483829	1.483829	1.483829	1.483829	0.783142	0.783142	0.783142	0.783142
20-21	0.275929	0.275929	0.275929	0.275929	0.134323	0.134323	0.134323	0.134323
19-22	0.545011	0.545011	0.545011	0.545011	0.351368	0.351368	0.351368	0.351368
22-23	0.146841	0.146841	0.146841	0.146841	0.093135	0.093135	0.093135	0.093135
23-24	0.332897	0.332897	0.332897	0.332897	0.176320	0.176320	0.176320	0.176320
1-25	41.000847	41.000847	41.000847	41.000847	32.08224	32.08224	32.08224	32.08224
25-26	9.584538	9.584538	9.584538	9.584538	0.000000	0.000000	0.000000	0.000000
26-27	14.591609	14.591609	14.591609	14.591609	6.161130	6.161130	6.161130	6.161130
27-28	0.222457	0.222457	0.222457	0.222457	0.117389	0.117389	0.117389	0.117389
27-29	6.534213	6.534213	6.534213	6.534213	4.820519	4.820519	4.820519	4.820519
29-30	0.021632	0.021632	0.021632	0.021632	0.011457	0.011457	0.011457	0.011457
29-31	5.237131	5.237131	5.237131	5.237131	2.773859	2.773859	2.773859	2.773859
31-32	2.989285	2.989285	2.989285	2.989285	3.168115	3.168115	3.168115	3.168115
32-33	0.342880	0.342880	0.342880	0.342880	0.181607	0.181607	0.181607	0.181607
32-34	1.777685	1.777685	1.777685	1.777685	0.941555	0.941555	0.941555	0.941555
34-35	0.557499	0.557499	0.557499	0.557499	0.295281	0.295281	0.295281	0.295281
35-36	0.139504	0.139504	0.139504	0.139504	0.073888	0.073888	0.073888	0.073888
25-37	39.932251	39.932251	39.932251	39.932251	21.15021	21.15021	21.15021	21.15021
37-38	10.930010	10.930010	10.930010	10.930010	11.583883	11.583883	11.583883	11.583883
38-39	7.466079	7.466079	7.466079	7.466079	7.800382	7.800382	7.800382	7.800382
39-40	3.902506	3.902506	3.902506	3.902506	2.890745	2.890745	2.890745	2.890745
40-41	3.156315	3.156315	3.156315	3.156315	2.525052	2.525052	2.525052	2.525052
41-42	0.063475	0.063475	0.063475	0.063475	0.052896	0.052896	0.052896	0.052896
41-43	1.251626	1.251626	1.251626	1.251626	0.589001	0.589001	0.589001	0.589001
43-44	0.929377	0.929377	0.929377	0.929377	0.492246	0.492246	0.492246	0.492246
44-45	0.140324	0.140324	0.140324	0.140324	0.074323	0.074323	0.074323	0.074323
37-46	7.022802	7.022802	7.022802	7.022802	3.719644	3.719644	3.719644	3.719644
46-47	1.894743	1.894743	1.894743	1.894743	0.808141	0.808141	0.808141	0.808141
47-48	0.113396	0.113396	0.113396	0.113396	0.083656	0.083656	0.083656	0.083656
46-49	6.059720	6.059720	6.059720	6.059720	2.584575	2.584575	2.584575	2.584575
49-50	0.137414	0.137414	0.137414	0.137414	0.072782	0.072782	0.072782	0.072782
49-51	1.927562	1.927562	1.927562	1.927562	0.822138	0.822138	0.822138	0.822138
51-52	0.138742	0.138742	0.138742	0.138742	0.073485	0.073485	0.073485	0.073485
25-53	42.906495	42.906495	42.906495	42.906495	15.666285	15.666285	15.666285	15.666285
53-54	0.841852	0.841852	0.841852	0.841852	0.445889	0.445889	0.445889	0.445889
53-55	18.724898	18.724898	18.724898	18.724898	9.917686	9.917686	9.917686	9.917686
55-56	0.340715	0.340715	0.340715	0.340715	0.180461	0.180461	0.180461	0.180461
55-57	10.796627	10.796627	10.796627	10.796627	4.596483	4.596483	4.596483	4.596483
57-58	0.967511	0.967511	0.967511	0.967511	0.512444	0.512444	0.512444	0.512444
58-59	0.045489	0.045489	0.045489	0.045489	0.033559	0.033559	0.033559	0.033559
58-60	0.562109	0.562109	0.562109	0.562109	0.297722	0.297722	0.297722	0.297722
60-61	0.116958	0.116958	0.116958	0.116958	0.086284	0.086284	0.086284	0.086284
57-62	0.506256	0.506256	0.506256	0.506256	0.204052	0.204052	0.204052	0.204052

62-63	0.467168	0.467168	0.467168	0.467168	0.199567	0.199567	0.199567	0.199567
63-64	0.466567	0.466567	0.466567	0.466567	0.224852	0.224852	0.224852	0.224852
64-65	0.045544	0.045544	0.045544	0.045544	0.033599	0.033599	0.033599	0.033599
64-66	0.027987	0.027987	0.027987	0.027987	0.014833	0.014833	0.014833	0.014833
1-67	617.773675	403.696827	278.648516	292.335183	414.747062	271.024939	187.072804	196.261452
67-68	6.096507	6.026104	5.976057	5.981160	5.339119	5.277462	5.233633	5.238102
68-69	4.403867	4.352909	4.316687	4.320380	2.332519	2.305529	2.286343	2.288300
69-70	0.354461	0.350356	0.347438	0.347735	0.249982	0.247086	0.245029	0.245238
70-71	0.463933	0.458554	0.454730	0.455120	0.342260	0.338291	0.335470	0.335758
71-72	0.043594	0.043089	0.042729	0.042766	0.029671	0.029327	0.029083	0.029108
69-73	0.367645	0.363390	0.360365	0.360674	0.194724	0.192470	0.190868	0.191032
73-74	0.054424	0.053794	0.053346	0.053391	0.028826	0.028492	0.028255	0.028279
67-75	1.229126	1.215170	1.205246	1.206258	0.107643	0.106421	0.105551	0.105640
75-76	0.547102	0.540881	0.536457	0.536908	0.289774	0.286479	0.284136	0.284375
76-77	0.136900	0.135341	0.134234	0.134346	0.072509	0.071684	0.071097	0.071157
67-78	27.007778	26.684508	26.454883	26.478289	14.304733	14.133513	14.011891	14.024288
78-79	23.271522	22.992241	22.793872	22.814092	20.380428	20.135843	19.962118	19.979826
79-80	0.952564	0.941182	0.933096	0.933920	0.504528	0.498499	0.494217	0.494653
80-81	0.296659	0.293111	0.290590	0.290847	0.218856	0.216238	0.214379	0.214568
79-82	1.265973	1.250730	1.239905	1.241008	0.757420	0.748300	0.741823	0.742483
82-83	1.552453	1.533711	1.520400	1.521757	0.866485	0.856025	0.848596	0.849353
83-84	5.192408	5.129600	5.084995	5.089542	2.750171	2.716904	2.693280	2.695688
84-85	0.047181	0.046611	0.046207	0.046248	0.034807	0.034387	0.034088	0.034119
84-86	0.553083	0.546389	0.541636	0.542120	0.332182	0.328162	0.325307	0.325598
86-87	0.169896	0.167839	0.166378	0.166527	0.118189	0.116757	0.115741	0.115845
87-88	0.293630	0.290070	0.287542	0.287799	0.192117	0.189788	0.188133	0.188302
67-89	195.192544	27.551431	23.179857	19.683532	103.384190	14.592680	12.277266	10.425429
89-90	0.145560	0.137517	0.135971	0.135981	0.077096	0.072836	0.072017	0.072023
89-91	46.264446	6.102674	5.133604	4.267588	24.548481	3.238154	2.723953	2.264434
91-92	17.766166	2.243574	1.889274	1.547023	9.110854	1.150551	0.968858	0.793345
92-93	3.412427	3.168411	3.128820	3.128241	1.552573	1.441552	1.423539	1.423276
93-94	1.471772	1.366233	1.349113	1.348863	0.669622	0.621603	0.613815	0.613701
94-95	0.318507	0.295633	0.291923	0.291869	0.136082	0.126309	0.124724	0.124701
94-96	0.252099	0.234007	0.231072	0.231029	0.132643	0.123124	0.121580	0.121557
92-97	3.341245	3.102140	3.063348	3.062781	1.769698	1.643056	1.622509	1.622209
97-98	1.482199	1.375792	1.358533	1.358281	0.785050	0.728691	0.719550	0.719417
98-99	0.949560	0.881328	0.870262	0.870100	0.502937	0.466797	0.460936	0.460850
92-100	83.479211	7.471297	9.207208	6.189550	44.214960	3.957190	4.876619	3.278310
100-101	79.030201	7.618648	9.645972	6.647895	41.858532	4.035235	5.109012	3.521073
101-102	68.112688	8.492069	11.247248	8.288307	36.076046	4.497844	5.957131	4.389921
102-103	0.426166	0.343766	0.336537	0.335788	0.373222	0.301059	0.294728	0.294072
102-104	57.766329	10.073727	13.543901	10.613436	50.589837	8.822237	11.861300	9.294896
104-105	51.510047	11.450236	15.366091	12.444140	45.110793	10.027737	13.457113	10.898166
105-106	42.185526	14.199547	18.801791	15.877480	22.343663	7.520824	9.958413	8.409545
106-107	3.910138	2.711827	2.631549	2.617142	2.071014	1.436325	1.393806	1.386175
107-108	0.522724	0.362249	0.351507	0.349579	0.276862	0.191866	0.186176	0.185155
108-109	0.077404	0.053625	0.052033	0.051748	0.040997	0.028402	0.027560	0.027408
107-110	0.128202	0.088904	0.086271	0.085799	0.067454	0.046777	0.045392	0.045143
107-111	0.064083	0.044456	0.043141	0.042905	0.047276	0.032797	0.031827	0.031653
106-112	42.801138	53.886488	26.986217	24.796395	18.291174	23.028526	11.532627	10.596801
112-113	0.169044	0.106936	0.111992	0.111119	0.072241	0.045699	0.047860	0.047487
112-114	37.590586	22.455063	23.621314	23.419020	16.064431	9.596228	10.094628	10.008177
114-115	1.219014	0.752052	0.789167	0.782743	0.520949	0.321391	0.337253	0.334507
115-116	0.557801	0.343634	0.360634	0.357691	0.924812	0.569731	0.597917	0.593038
114-117	22.632521	13.388478	14.094446	13.971916	9.672064	5.721600	6.023297	5.970933
117-118	0.476771	0.290344	0.304978	0.302443	0.203749	0.124079	0.130333	0.129250
117-119	16.539538	9.735598	10.252867	10.163060	7.068213	4.160532	4.381589	4.343209
119-120	13.167885	7.730505	8.142887	8.071278	5.627329	3.303651	3.479883	3.449281
120-121	0.239994	0.142160	0.149643	0.148344	0.090885	0.053836	0.056669	0.056177
120-122	3.209358	1.881756	1.982327	1.964862	1.481242	0.868503	0.914920	0.906859
122-123	2.505903	1.465732	1.544353	1.530698	1.186226	0.693838	0.731055	0.724591
123-124	3.051398	1.779759	1.875625	1.858972	1.304023	0.760585	0.801553	0.794436
124-125	1.043542	0.607386	0.640203	0.634502	0.445960	0.259568	0.273592	0.271156

125-126	0.267645	0.155716	0.164134	0.162672	0.129925	0.075590	0.079677	0.078967
67-127	0.297602	0.294097	0.291606	0.291860	1.488011	1.470483	1.458028	1.459298
127-128	28.906368	28.561121	28.315873	28.340872	19.829768	19.592929	19.424689	19.441838
128-129	0.347989	0.343928	0.341043	0.341337	0.184313	0.182162	0.180634	0.180790
128-130	15.460737	15.275155	15.143339	15.156776	8.188816	8.090522	8.020705	8.027822
130-131	2.488472	2.458569	2.437330	2.439495	2.181866	2.155647	2.137025	2.138923
131-132	0.960522	0.948951	0.940732	0.941570	0.508743	0.502614	0.498261	0.498705
132-133	0.359769	0.355430	0.352348	0.352662	0.190553	0.188254	0.186622	0.186788
130-134	3.907937	3.860965	3.827603	3.831004	2.069848	2.044969	2.027299	2.029100
134-135	0.381864	0.377272	0.374010	0.374342	0.334814	0.330788	0.327928	0.328219
135-136	0.047006	0.046441	0.046039	0.046080	0.034678	0.034261	0.033964	0.033995
134-137	0.579278	0.572302	0.567347	0.567852	0.507905	0.501788	0.497443	0.497886
137-138	0.144972	0.143224	0.141983	0.142109	0.127110	0.125578	0.124489	0.124600
127-139	0.337259	0.333447	0.330736	0.331012	0.178630	0.176611	0.175175	0.175321
127-140	5.258897	5.198102	5.154886	5.159292	2.785387	2.753187	2.730297	2.732631
140-141	2.924273	2.890306	2.866163	2.868624	2.563971	2.534189	2.513021	2.515179
141-142	1.790416	1.769565	1.754745	1.756256	0.948298	0.937254	0.929405	0.930205
142-143	0.139549	0.137930	0.136779	0.136896	0.073912	0.073055	0.072445	0.072507
142-144	0.930265	0.919415	0.911704	0.912490	0.815647	0.806134	0.799372	0.800062
144-145	0.348462	0.344393	0.341501	0.341795	0.305528	0.301960	0.299424	0.299683
67-146	38.628438	38.128287	17.677724	20.281944	29.168412	28.790747	13.348485	15.314938
146-147	23.201859	22.913108	22.635939	22.667505	12.288919	12.135982	11.989178	12.005897
147-148	0.880369	0.870045	0.860120	0.861251	0.466290	0.460822	0.455565	0.456164
147-149	15.043203	14.853373	14.671221	14.691962	7.967668	7.867124	7.770647	7.781632
149-150	0.355648	0.351401	0.347320	0.347785	0.188370	0.186120	0.183959	0.184205
149-151	3.712106	3.664806	3.619430	3.624596	1.817500	1.794341	1.772124	1.774654
151-152	11.979014	11.824332	11.675992	11.692879	6.321234	6.239609	6.161331	6.170242
152-153	8.802025	8.687400	8.577498	8.590008	4.644761	4.584274	4.526279	4.532881
153-154	7.785791	7.683920	7.586258	7.597374	2.819367	2.782478	2.747113	2.751138
154-155	3.465105	3.419249	3.375301	3.380302	1.828509	1.804311	1.781120	1.783760
155-156	4.640879	4.578872	4.519460	4.526221	1.055772	1.041666	1.028150	1.029688
156-157	0.260599	0.257150	0.253845	0.254221	0.059285	0.058500	0.057748	0.057834
156-158	2.702015	2.665808	2.631119	2.635066	0.614692	0.606455	0.598564	0.599462
158-159	0.194175	0.191570	0.189075	0.189359	0.063354	0.062504	0.061690	0.061782
146-160	33.590406	33.117347	4.293237	2.494454	17.791237	17.540681	2.273923	1.321194
160-161	31.754096	31.301771	5.227883	2.481611	14.420025	14.214617	2.374062	1.126938
161-162	26.857617	26.472424	5.316807	2.311501	14.225200	14.021182	2.816059	1.224292
162-163	20.304577	20.011516	4.909777	2.024694	14.979400	14.763198	3.622115	1.493688
163-164	31.105741	30.649455	13.046171	5.347620	16.414260	16.173482	6.884364	2.821898
164-165	25.651824	25.271961	15.669806	6.758832	13.536270	13.335819	8.268836	3.566583
165-166	0.272925	0.269152	0.213997	0.221380	0.144021	0.142029	0.112925	0.116821
165-167	20.632959	20.325083	15.903799	8.621509	10.887854	10.725390	8.392313	4.549504
167-168	13.748785	13.541485	10.571684	12.441846	7.255128	7.145737	5.578596	6.565466
168-169	6.052316	5.960581	4.647984	9.278186	3.205624	3.157036	2.461816	4.914213
169-170	4.165168	4.101740	3.195166	3.002038	2.206091	2.172496	1.692327	1.590036
170-171	2.621848	2.581751	2.009228	1.887402	1.388668	1.367430	1.064192	0.999667
171-172	0.023893	0.023531	0.018357	0.017253	0.017627	0.017360	0.013543	0.012728
171-173	2.101658	2.069475	1.610097	1.512381	1.842711	1.814494	1.411716	1.326040
173-174	0.728939	0.717750	0.558137	0.524206	0.639126	0.629316	0.489368	0.459618
174-175	0.182476	0.179672	0.139676	0.131176	0.159993	0.157534	0.122466	0.115014
173-176	0.095062	0.093608	0.072849	0.068432	0.038619	0.038028	0.029595	0.027800
146-177	4.525807	4.473829	4.423834	4.429533	3.017205	2.982552	2.949223	2.953022
146-178	38.877189	38.374264	37.892005	37.946904	20.591395	20.325019	20.069590	20.098667
178-179	0.890709	0.880158	0.870018	0.871173	0.212760	0.210240	0.207818	0.208094
178-180	0.024390	0.024103	0.023827	0.023858	0.011097	0.010966	0.010841	0.010855
178-181	30.150538	29.755982	29.377749	29.420800	15.969304	15.760326	15.559994	15.582796
181-182	26.111907	25.768291	25.438938	25.476424	13.830233	13.648236	13.473793	13.493648
182-183	20.268939	20.000223	19.742710	19.772017	10.735491	10.593165	10.456773	10.472295
183-184	15.109867	14.908283	14.715134	14.737114	8.002977	7.896207	7.793905	7.805547
184-185	10.666294	10.523279	10.386267	10.401858	5.649428	5.573679	5.501111	5.509369
185-186	5.159211	5.089598	5.022918	5.030505	2.732588	2.695717	2.660400	2.664419
186-187	2.693067	2.656605	2.621683	2.625656	1.426389	1.407077	1.388581	1.390685
187-188	1.225852	1.209217	1.193285	1.195097	0.389572	0.384285	0.379222	0.379798

1-189	21.722902	21.722902	21.722902	15.329470	21.887055	21.887055	21.887055	15.445310
189-190	21.910148	21.910148	21.910148	21.894528	9.517530	9.517530	9.517530	9.510745
190-191	14.907847	14.907847	14.907847	14.897181	7.850631	7.850631	7.850631	7.845014
191-192	1.122742	1.122742	1.122742	1.121938	0.594830	0.594830	0.594830	0.594404
192-193	11.890561	11.890561	11.890561	11.882016	6.297864	6.297864	6.297864	6.293338
193-194	0.280694	0.280694	0.280694	0.280494	0.207078	0.207078	0.207078	0.206930
193-195	0.344293	0.344293	0.344293	0.344048	0.130330	0.130330	0.130330	0.130237
193-196	0.135974	0.135974	0.135974	0.135878	0.072019	0.072019	0.072019	0.071968
193-197	4.534034	4.534034	4.534034	4.530756	3.975393	3.975393	3.975393	3.972518
197-198	0.137567	0.137567	0.137567	0.137468	0.120617	0.120617	0.120617	0.120530
197-199	1.343690	1.343690	1.343690	1.342717	0.932679	0.932679	0.932679	0.932004
199-200	1.377564	1.377564	1.377564	1.376564	0.729630	0.729630	0.729630	0.729101
200-201	0.344890	0.344890	0.344890	0.344640	0.182672	0.182672	0.182672	0.182539
189-202	61.043299	61.043299	61.043299	14.005799	53.522115	53.522115	53.522115	12.280135
202-203	0.323840	0.323840	0.323840	0.317067	0.164108	0.164108	0.164108	0.160676
203-204	0.133878	0.133878	0.133878	0.131076	0.117383	0.117383	0.117383	0.114926
202-205	0.739428	0.739428	0.739428	0.723924	0.545502	0.545502	0.545502	0.534064
205-206	0.333014	0.333014	0.333014	0.326024	0.176382	0.176382	0.176382	0.172679
202-207	30.953417	30.953417	30.953417	2.631386	27.139626	27.139626	27.139626	2.307172
207-208	0.341837	0.341837	0.341837	0.327856	0.299719	0.299719	0.299719	0.287460
207-209	0.572303	0.572303	0.572303	0.548829	0.157383	0.157383	0.157383	0.150928
209-210	0.632636	0.632636	0.632636	0.606656	0.415980	0.415980	0.415980	0.398897
210-211	0.865989	0.865989	0.865989	0.830382	0.458673	0.458673	0.458673	0.439814
207-212	0.341742	0.341742	0.341742	0.327768	0.181004	0.181004	0.181004	0.173603
207-213	7.406279	7.406279	7.406279	2.351800	6.493747	6.493747	6.493747	2.062033
213-214	6.190590	6.190590	6.190590	3.011321	3.278861	3.278861	3.278861	1.594954
214-215	4.163398	4.163398	4.163398	3.834112	1.825051	1.825051	1.825051	1.680707
215-216	2.303892	2.303892	2.303892	2.121182	1.220262	1.220262	1.220262	1.123489
216-217	0.649907	0.649907	0.649907	0.598321	0.318322	0.318322	0.318322	0.293055
217-218	0.119946	0.119946	0.119946	0.110419	0.088488	0.088488	0.088488	0.081460
217-219	0.081070	0.081070	0.081070	0.074634	0.058960	0.058960	0.058960	0.054279
189-220	21.204650	21.204650	21.204650	21.189795	18.592012	18.592012	18.592012	18.578987
220-221	2.208117	2.208117	2.208117	2.206583	1.936053	1.936053	1.936053	1.934709
221-222	2.107620	2.107620	2.107620	2.106150	0.797824	0.797824	0.797824	0.797267
220-223	9.727397	9.727397	9.727397	9.720555	5.152139	5.152139	5.152139	5.148515
223-224	7.585104	7.585104	7.585104	7.579761	4.017468	4.017468	4.017468	4.014638
189-225	6.399765	6.399765	6.399765	6.395368	3.389650	3.389650	3.389650	3.387322
225-226	3.853370	3.853370	3.853370	3.850711	2.040947	2.040947	2.040947	2.039538
226-227	2.564645	2.564645	2.564645	2.562871	1.358370	1.358370	1.358370	1.357430
227-228	1.060524	1.060524	1.060524	1.059789	0.929857	0.929857	0.929857	0.929211
228-229	0.643783	0.643783	0.643783	0.643336	0.564462	0.564462	0.564462	0.564070
229-230	0.003985	0.003985	0.003985	0.003982	0.002657	0.002657	0.002657	0.002655
189-231	7.080317	7.080317	7.080317	7.075471	3.750107	3.750107	3.750107	3.747540
231-232	0.261613	0.261613	0.261613	0.261435	0.263590	0.263590	0.263590	0.263411
231-233	4.735529	4.735529	4.735529	4.732280	2.016072	2.016072	2.016072	2.014689
233-234	0.347014	0.347014	0.347014	0.346775	0.304258	0.304258	0.304258	0.304049
234-235	0.051371	0.051371	0.051371	0.051336	0.045042	0.045042	0.045042	0.045011
233-236	0.872749	0.872749	0.872749	0.872149	0.462253	0.462253	0.462253	0.461936
236-237	0.131764	0.131764	0.131764	0.131674	0.069789	0.069789	0.069789	0.069741
189-238	24.649914	24.649914	24.649914	24.631862	13.007570	13.007570	13.007570	12.998044
238-239	17.871666	17.871666	17.871666	17.858444	9.430741	9.430741	9.430741	9.423763
239-240	15.291670	15.291670	15.291670	15.280254	6.522160	6.522160	6.522160	6.517291
240-241	1.955650	1.955650	1.955650	1.954211	0.834118	0.834118	0.834118	0.833504
241-242	0.117045	0.117045	0.117045	0.116958	0.086348	0.086348	0.086348	0.086284
240-243	6.310202	6.310202	6.310202	6.305453	2.691409	2.691409	2.691409	2.689384
243-244	3.869008	3.869008	3.869008	3.866085	1.650198	1.650198	1.650198	1.648952
244-245	0.751758	0.751758	0.751758	0.751189	0.320638	0.320638	0.320638	0.320395
244-246	0.479642	0.479642	0.479642	0.479280	0.353849	0.353849	0.353849	0.353582
246-247	0.120011	0.120011	0.120011	0.119921	0.088537	0.088537	0.088537	0.088470

	line flow limit index	Voltage stability index
--	-----------------------	-------------------------

	original	One PV	two PV	three PV	original	One PV	two PV	three PV
1-2	33.509177	33.509177	33.509177	33.509177	0.016327	0.016327	0.016327	0.016327
2-3	17.521483	17.521483	17.521483	17.521483	0.022004	0.022004	0.022004	0.022004
3-4	16.033970	16.033970	16.033970	16.033970	0.040416	0.040416	0.040416	0.040416
4-5	14.531070	14.531070	14.531070	14.531070	0.037160	0.037160	0.037160	0.037160
5-6	13.013915	13.013915	13.013915	13.013915	0.030472	0.030472	0.030472	0.030472
6-7	12.058355	12.058355	12.058355	12.058355	0.057500	0.057500	0.057500	0.057500
7-8	4.006913	4.006913	4.006913	4.006913	0.008488	0.008488	0.008488	0.008488
8-9	2.452072	2.452072	2.452072	2.452072	0.005209	0.005209	0.005209	0.005209
7-10	8.051442	8.051442	8.051442	8.051442	0.019311	0.019311	0.019311	0.019311
10-11	4.026562	4.026562	4.026562	4.026562	0.008571	0.008571	0.008571	0.008571
11-12	2.464109	2.464109	2.464109	2.464109	0.005260	0.005260	0.005260	0.005260
10-13	4.024880	4.024880	4.024880	4.024880	0.006475	0.006475	0.006475	0.006475
13-14	2.463250	2.463250	2.463250	2.463250	0.005971	0.005971	0.005971	0.005971
2-15	15.988000	15.988000	15.988000	15.988000	0.033706	0.033706	0.033706	0.033706
15-16	3.382783	3.382783	3.382783	3.382783	0.018886	0.018886	0.018886	0.018886
16-17	2.445810	2.445810	2.445810	2.445810	0.013738	0.013738	0.013738	0.013738
17-18	0.941467	0.941467	0.941467	0.941467	0.005318	0.005318	0.005318	0.005318
15-19	11.113178	11.113178	11.113178	11.113178	0.042356	0.042356	0.042356	0.042356
19-20	3.901123	3.901123	3.901123	3.901123	0.015132	0.015132	0.015132	0.015132
20-21	2.387185	2.387185	2.387185	2.387185	0.004575	0.004575	0.004575	0.004575
19-22	5.703867	5.703867	5.703867	5.703867	0.003937	0.003937	0.003937	0.003937
22-23	4.760319	4.760319	4.760319	4.760319	0.001269	0.001269	0.001269	0.001269
23-24	2.381824	2.381824	2.381824	2.381824	0.005584	0.005584	0.005584	0.005584
1-25	61.752108	61.752108	61.752108	61.752108	0.027619	0.027619	0.027619	0.027619
25-26	14.014725	14.014725	14.014725	14.014725	0.022733	0.022733	0.022733	0.022733
26-27	13.082144	13.082144	13.082144	13.082144	0.042321	0.042321	0.042321	0.042321
27-28	1.510498	1.510498	1.510498	1.510498	0.005873	0.005873	0.005873	0.005873
27-29	11.571660	11.571660	11.571660	11.571660	0.023762	0.023762	0.023762	0.023762
29-30	0.607156	0.607156	0.607156	0.607156	0.001432	0.001432	0.001432	0.001432
29-31	9.447168	9.447168	9.447168	9.447168	0.022170	0.022170	0.022170	0.022170
31-32	7.921324	7.921324	7.921324	7.921324	0.017458	0.017458	0.017458	0.017458
32-33	2.417275	2.417275	2.417275	2.417275	0.005751	0.005751	0.005751	0.005751
32-34	5.504050	5.504050	5.504050	5.504050	0.013070	0.013070	0.013070	0.013070
34-35	3.082316	3.082316	3.082316	3.082316	0.007354	0.007354	0.007354	0.007354
35-36	1.541869	1.541869	1.541869	1.541869	0.003689	0.003689	0.003689	0.003689
25-37	26.086562	26.086562	26.086562	26.086562	0.059330	0.059330	0.059330	0.059330
37-38	15.146922	15.146922	15.146922	15.146922	0.032921	0.032921	0.032921	0.032921
38-39	13.628036	13.628036	13.628036	13.628036	0.025115	0.025115	0.025115	0.025115
39-40	12.672685	12.672685	12.672685	12.672685	0.013171	0.013171	0.013171	0.013171
40-41	10.257217	10.257217	10.257217	10.257217	0.013407	0.013407	0.013407	0.013407
41-42	2.424323	2.424323	2.424323	2.424323	0.001158	0.001158	0.001158	0.001158
41-43	7.832894	7.832894	7.832894	7.832894	0.006396	0.006396	0.006396	0.006396
43-44	3.979704	3.979704	3.979704	3.979704	0.009512	0.009512	0.009512	0.009512
44-45	1.546394	1.546394	1.546394	1.546394	0.003710	0.003710	0.003710	0.003710
37-46	10.939820	10.939820	10.939820	10.939820	0.025457	0.025457	0.025457	0.025457
46-47	3.923411	3.923411	3.923411	3.923411	0.018746	0.018746	0.018746	0.018746
47-48	1.524395	1.524395	1.524395	1.524395	0.003174	0.003174	0.003174	0.003174
46-49	7.016409	7.016409	7.016409	7.016409	0.033401	0.033401	0.033401	0.033401
49-50	1.530279	1.530279	1.530279	1.530279	0.003634	0.003634	0.003634	0.003634
49-51	3.957243	3.957243	3.957243	3.957243	0.019067	0.019067	0.019067	0.019067
51-52	1.537657	1.537657	1.537657	1.537657	0.003669	0.003669	0.003669	0.003669
25-53	21.651091	21.651091	21.651091	21.651091	0.072923	0.072923	0.072923	0.072923
53-54	3.787676	3.787676	3.787676	3.787676	0.008881	0.008881	0.008881	0.008881
53-55	17.863423	17.863423	17.863423	17.863423	0.041548	0.041548	0.041548	0.041548
55-56	2.409631	2.409631	2.409631	2.409631	0.005714	0.005714	0.005714	0.005714
55-57	13.047609	13.047609	13.047609	13.047609	0.032252	0.032252	0.032252	0.032252
57-58	4.060532	4.060532	4.060532	4.060532	0.009699	0.009699	0.009699	0.009699
58-59	0.965498	0.965498	0.965498	0.965498	0.002038	0.002038	0.002038	0.002038
58-60	3.095033	3.095033	3.095033	3.095033	0.007415	0.007415	0.007415	0.007415
60-61	1.548151	1.548151	1.548151	1.548151	0.003274	0.003274	0.003274	0.003274
57-62	6.561174	6.561174	6.561174	6.561174	0.003034	0.003034	0.003034	0.003034
62-63	5.019747	5.019747	5.019747	5.019747	0.003688	0.003688	0.003688	0.003688


63-64	3.476896	3.476896	3.476896	3.476896	0.005404	0.005404	0.005404	0.005404
64-65	0.966081	0.966081	0.966081	0.966081	0.002041	0.002041	0.002041	0.002041
64-66	0.965874	0.965874	0.965874	0.965874	0.001186	0.001186	0.001186	0.001186
1-67	41.287692	33.375945	69.322510	71.004595	0.118303	0.097492	0.082323	0.083884
67-68	10.192846	10.133821	10.091653	10.095961	0.026264	0.025966	0.025754	0.025776
68-69	8.663075	8.612808	8.576898	8.580566	0.020497	0.020263	0.020096	0.020113
69-70	4.622331	4.595487	4.576310	4.578269	0.003275	0.003237	0.003210	0.003213
70-71	3.083377	3.065449	3.052642	3.053950	0.006481	0.006407	0.006353	0.006359
71-72	1.541920	1.532952	1.526546	1.527201	0.001203	0.001189	0.001179	0.001180
69-73	2.503050	2.488522	2.478143	2.479204	0.005974	0.005905	0.005856	0.005861
73-74	0.963053	0.957460	0.953464	0.953872	0.002304	0.002277	0.002258	0.002260
67-75	4.576708	4.550651	4.532031	4.533933	0.009451	0.009344	0.009268	0.009276
75-76	3.053439	3.036028	3.023587	3.024858	0.007217	0.007136	0.007077	0.007083
76-77	1.527411	1.518694	1.512465	1.513102	0.003620	0.003579	0.003550	0.003553
67-78	21.453560	21.324780	21.232829	21.242220	0.050096	0.049520	0.049111	0.049153
78-79	19.914405	19.794548	19.708973	19.717713	0.051663	0.051062	0.050634	0.050678
79-80	4.029043	4.004900	3.987661	3.989421	0.009745	0.009629	0.009547	0.009555
80-81	2.465629	2.450839	2.440278	2.441357	0.005267	0.005204	0.005159	0.005164
79-82	13.428999	13.347911	13.290018	13.295931	0.003967	0.003919	0.003886	0.003889
82-83	10.970196	10.903776	10.856357	10.861200	0.005892	0.005821	0.005771	0.005776
83-84	9.406744	9.349679	9.308940	9.313100	0.022734	0.022462	0.022269	0.022289
84-85	0.983296	0.977338	0.973086	0.973520	0.002114	0.002089	0.002070	0.002072
84-86	7.440673	7.395512	7.363272	7.366565	0.003155	0.003117	0.003090	0.003093
86-87	4.962122	4.931985	4.910470	4.912667	0.001493	0.001475	0.001462	0.001464
87-88	2.482645	2.467548	2.456770	2.457871	0.005098	0.005036	0.004993	0.004997
67-89	57.674861	21.668409	19.875146	18.314981	0.132104	0.047496	0.041415	0.039948
89-90	1.574984	1.530850	1.522223	1.522281	0.003849	0.003636	0.003595	0.003596
89-91	56.100278	20.375179	18.687556	17.038547	0.034209	0.011240	0.009660	0.009285
91-92	55.108542	19.583589	17.970888	16.261870	0.013488	0.004205	0.003573	0.003433
92-93	7.342261	7.074877	7.030536	7.029885	0.019106	0.017756	0.017536	0.017533
93-94	4.821903	4.645799	4.616601	4.616173	0.012629	0.011729	0.011583	0.011581
94-95	1.608601	1.549763	1.540008	1.539865	0.008162	0.007578	0.007483	0.007482
94-96	1.607990	1.549216	1.539471	1.539329	0.006650	0.006174	0.006097	0.006096
92-97	7.545866	7.270857	7.225254	7.224585	0.018596	0.017281	0.017067	0.017064
97-98	5.025834	4.842072	4.811606	4.811159	0.012462	0.011574	0.011430	0.011428
98-99	4.022686	3.875463	3.851057	3.850699	0.010012	0.009296	0.009179	0.009178
92-100	37.717606	11.283737	12.526188	10.270332	0.091603	0.002713	-0.003177	-0.004626
100-101	36.698769	11.394464	12.821178	10.643808	0.091244	0.000236	-0.005665	-0.007033
101-102	34.069751	12.029891	13.844526	11.884690	0.086819	-0.005750	-0.011594	-0.012872
102-103	2.694912	2.420396	2.394810	2.392146	0.007833	0.006323	0.006190	0.006177
102-104	31.375621	13.102375	15.192416	13.448782	0.090041	-0.005474	-0.011372	-0.014254
104-105	29.627904	13.968899	16.182166	14.562549	0.086814	-0.009839	-0.015715	-0.018449
105-106	26.812461	15.555794	17.900063	16.449236	0.073212	-0.021715	-0.027285	-0.028189
106-107	8.163022	6.798074	6.696698	6.678341	0.022984	0.016002	0.015532	0.015448
107-108	2.984634	2.484610	2.447495	2.440775	0.008483	0.005887	0.005713	0.005682
108-109	1.148515	0.955953	0.941664	0.939076	0.003275	0.002270	0.002203	0.002191
107-110	1.146689	0.954900	0.940657	0.938078	0.005414	0.003757	0.003646	0.003626
107-111	1.145958	0.954478	0.940253	0.937677	0.002870	0.001992	0.001933	0.001923
106-112	18.649586	20.925772	14.808550	14.195014	0.104745	-0.083312	0.069029	0.064518
112-113	1.172038	0.932189	0.953971	0.950243	0.006936	0.004393	0.004600	0.004564
112-114	17.477572	13.508238	13.854587	13.795133	0.100966	0.062731	0.065787	0.065258
114-115	3.147356	2.472097	2.532363	2.522035	0.019066	0.011818	0.012397	0.012297
115-116	1.940455	1.523042	1.560261	1.553882	0.019290	0.011920	0.012507	0.012405
114-117	13.561509	10.430548	10.702015	10.655394	0.080854	0.049391	0.051867	0.051438
117-118	1.968323	1.536025	1.574259	1.567702	0.012193	0.007443	0.007817	0.007752
117-119	11.593198	8.894527	9.127760	9.087696	0.070759	0.042722	0.044905	0.044526
119-120	10.344263	7.925843	8.134498	8.098651	0.064396	0.038524	0.040521	0.040175
120-121	1.272812	0.979610	1.005059	1.000689	0.009695	0.005754	0.006056	0.006004
120-122	9.071452	6.946234	7.129439	7.097963	0.018592	0.010988	0.011569	0.011468
122-123	7.030370	5.376794	5.519113	5.494659	0.018881	0.011125	0.011715	0.011612
123-124	4.979560	3.802964	3.904043	3.886672	0.032078	0.018863	0.019867	0.019693
124-125	2.912035	2.221641	2.280870	2.270690	0.018975	0.011094	0.011689	0.011586
125-126	2.081064	1.587348	1.629692	1.622414	0.006978	0.004066	0.004285	0.004247

67-127	31.496148	31.310092	31.177212	31.190785	0.000865	0.000855	0.000848	0.000848
127-128	19.632072	19.514481	19.430517	19.439092	0.061011	0.060313	0.059816	0.059867
128-129	2.435216	2.420968	2.410790	2.411830	0.005836	0.005768	0.005720	0.005725
128-130	16.231926	16.134212	16.064447	16.071572	0.038592	0.038140	0.037819	0.037852
130-131	6.512104	6.472859	6.444839	6.447701	0.017251	0.017045	0.016900	0.016914
131-132	4.045840	4.021395	4.003943	4.005726	0.009824	0.009707	0.009623	0.009632
132-133	2.476092	2.461115	2.450422	2.451514	0.006033	0.005960	0.005909	0.005914
130-134	8.160725	8.111532	8.076411	8.079998	0.019684	0.019450	0.019284	0.019301
134-135	2.550994	2.535607	2.524622	2.525744	0.006807	0.006725	0.006667	0.006673
135-136	0.981470	0.975546	0.971317	0.971749	0.002106	0.002081	0.002063	0.002065
134-137	3.141945	3.122967	3.109418	3.110802	0.008381	0.008280	0.008209	0.008216
137-138	1.571800	1.562296	1.555511	1.556204	0.004205	0.004155	0.004119	0.004122
127-139	2.397380	2.383791	2.374080	2.375072	0.005657	0.005593	0.005547	0.005552
127-140	9.466780	9.411901	9.372695	9.376700	0.022254	0.022001	0.021821	0.021839
140-141	7.059336	7.018218	6.988845	6.991845	0.018337	0.018126	0.017977	0.017992
141-142	5.523724	5.491464	5.468421	5.470774	0.013145	0.012993	0.012886	0.012897
142-143	1.542118	1.533146	1.526737	1.527392	0.003690	0.003647	0.003617	0.003620
142-144	3.981607	3.958319	3.941684	3.943384	0.010439	0.010318	0.010232	0.010241
144-145	2.436871	2.422601	2.412407	2.413448	0.006411	0.006336	0.006283	0.006288
67-146	81.052219	80.525788	54.830791	58.730829	0.020364	0.020118	0.014178	0.015056
146-147	19.884576	19.760455	19.640575	19.654265	0.046692	0.046142	0.045613	0.045673
147-148	3.873356	3.850577	3.828552	3.831069	0.009285	0.009177	0.009073	0.009085
147-149	16.011245	15.909902	15.812046	15.823219	0.038124	0.037665	0.037224	0.037275
149-150	2.461869	2.447126	2.432874	2.434502	0.005964	0.005893	0.005825	0.005833
149-151	13.549385	13.462785	13.379180	13.388726	0.011174	0.011036	0.010904	0.010919
151-152	11.084299	11.012502	10.943206	10.951117	0.044300	0.043755	0.043231	0.043291
152-153	9.501431	9.439361	9.379463	9.386301	0.038453	0.037971	0.037509	0.037562
153-154	8.502439	8.446632	8.392782	8.398929	0.036613	0.036149	0.035704	0.035755
154-155	5.961505	5.921927	5.883746	5.888104	0.024670	0.024352	0.024047	0.024081
155-156	4.338140	4.309062	4.281015	4.284216	0.041674	0.041133	0.040614	0.040673
156-157	1.027993	1.021167	1.014583	1.015334	0.010060	0.009928	0.009801	0.009815
156-158	3.310148	3.287895	3.266433	3.268882	0.032213	0.031789	0.031382	0.031428
158-159	1.656077	1.644930	1.634180	1.635407	0.004839	0.004774	0.004712	0.004719
146-160	23.925585	23.756514	8.553569	6.519927	0.056144	0.055437	-0.002077	0.006924
160-161	22.376222	22.216280	9.079246	6.255381	0.056291	0.055569	-0.007335	0.002560
161-162	21.393837	21.239868	9.518761	6.276275	0.051711	0.051034	-0.008047	0.001049
162-163	20.398393	20.250651	10.030669	6.441377	0.044119	0.043530	-0.006438	0.000957
163-164	17.861502	17.730014	11.567493	7.405902	0.073009	0.072024	-0.026768	-0.011600
164-165	16.220223	16.099677	12.677380	8.325944	0.067630	0.066695	-0.032764	-0.017694
165-166	1.673091	1.661484	1.481501	1.506841	0.007200	0.007101	0.005650	0.005844
165-167	14.547168	14.438227	12.771683	9.403496	0.061791	0.060917	0.048216	-0.023729
167-168	11.874897	11.785034	10.412854	11.296401	0.051373	0.050629	0.039875	-0.033028
168-169	10.155838	10.078578	8.899944	12.574377	0.026970	0.026571	0.020830	-0.023207
169-170	8.425026	8.360630	7.379070	7.152585	0.022550	0.022214	0.017376	0.016340
170-171	6.684342	6.633032	5.851531	5.671359	0.018014	0.017742	0.013851	0.013020
171-172	0.699733	0.694421	0.613345	0.594613	0.001673	0.001648	0.001286	0.001208
171-173	5.984611	5.938613	5.238187	5.076748	0.017782	0.017512	0.013655	0.012833
173-174	3.524523	3.497371	3.084078	2.988863	0.010534	0.010373	0.008076	0.007587
174-175	1.763429	1.749826	1.542821	1.495143	0.005290	0.005209	0.004052	0.003806
173-176	0.703645	0.698243	0.615972	0.597005	0.006060	0.005967	0.004647	0.004366
146-177	11.503632	11.437382	11.373297	11.380620	0.016464	0.016277	0.016096	0.016117
146-178	25.739634	25.572605	25.411409	25.429810	0.060269	0.059546	0.058851	0.058930
178-179	2.451567	2.437004	2.422925	2.424533	0.013668	0.013507	0.013352	0.013370
178-180	0.620731	0.617066	0.613522	0.613927	0.001580	0.001562	0.001544	0.001546
178-181	22.667440	22.518636	22.375060	22.391448	0.053977	0.053315	0.052679	0.052751
181-182	21.094743	20.955487	20.821137	20.836472	0.050954	0.050318	0.049708	0.049777
182-183	18.585338	18.461729	18.342492	18.356101	0.045533	0.044955	0.044399	0.044463
183-184	16.046683	15.939282	15.835692	15.847515	0.039822	0.039307	0.038814	0.038871
184-185	13.482229	13.391539	13.304075	13.314057	0.033843	0.033400	0.032975	0.033023
185-186	9.376626	9.313152	9.251944	9.258929	0.023798	0.023482	0.023179	0.023213
186-187	6.774519	6.728503	6.684132	6.689196	0.017323	0.017091	0.016869	0.016894
187-188	4.161045	4.132715	4.105399	4.108516	0.012145	0.011981	0.011824	0.011842
1-189	53.384239	53.384239	53.384239	89.690760	0.009022	0.009022	0.009022	0.007694

189-190	18.178828	18.178828	18.178828	18.172347	0.045381	0.045381	0.045381	0.045350
190-191	16.685058	16.685058	16.685058	16.679088	0.035036	0.035036	0.035036	0.035012
191-192	15.743124	15.743124	15.743124	15.737482	0.002847	0.002847	0.002847	0.002845
192-193	14.234956	14.234956	14.234956	14.229840	0.033110	0.033110	0.033110	0.033087
193-194	2.398365	2.398365	2.398365	2.397512	0.004984	0.004984	0.004984	0.004980
193-195	1.524193	1.524193	1.524193	1.523650	0.008697	0.008697	0.008697	0.008690
193-196	1.522242	1.522242	1.522242	1.521700	0.003596	0.003596	0.003596	0.003593
193-197	8.790172	8.790172	8.790172	8.786993	0.022691	0.022691	0.022691	0.022675
197-198	1.531128	1.531128	1.531128	1.530577	0.003991	0.003991	0.003991	0.003988
197-199	7.259044	7.259044	7.259044	7.256416	0.007832	0.007832	0.007832	0.007826
199-200	4.845191	4.845191	4.845191	4.843432	0.011510	0.011510	0.011510	0.011502
200-201	2.424351	2.424351	2.424351	2.423470	0.005784	0.005784	0.005784	0.005780
189-202	32.253284	32.253284	32.253284	15.449303	0.079930	0.079930	0.079930	0.040990
202-203	3.019456	3.019456	3.019456	2.987716	0.004251	0.004251	0.004251	0.004162
203-204	1.510462	1.510462	1.510462	1.494569	0.003884	0.003884	0.003884	0.003803
202-205	3.892663	3.892663	3.892663	3.851638	0.008012	0.008012	0.008012	0.007845
205-206	2.382243	2.382243	2.382243	2.357110	0.005586	0.005586	0.005586	0.005469
202-207	22.967263	22.967263	22.967263	6.696491	0.058314	0.058314	0.058314	0.018378
207-208	2.413597	2.413597	2.413597	2.363721	0.006289	0.006289	0.006289	0.006033
207-209	6.905933	6.905933	6.905933	6.762824	0.003117	0.003117	0.003117	0.002990
209-210	5.374710	5.374710	5.374710	5.263194	0.004940	0.004940	0.004940	0.004738
210-211	3.841591	3.841591	3.841591	3.761783	0.009134	0.009134	0.009134	0.008761
207-212	2.413259	2.413259	2.413259	2.363404	0.005731	0.005731	0.005731	0.005498
207-213	11.234532	11.234532	11.234532	6.330749	0.029143	0.029143	0.029143	-0.010876
213-214	10.271195	10.271195	10.271195	7.163636	0.024464	0.024464	0.024464	-0.013910
214-215	8.720310	8.720310	8.720310	8.368359	0.019024	0.019024	0.019024	0.017537
215-216	6.265936	6.265936	6.265936	6.012345	0.015122	0.015122	0.015122	0.013933
216-217	4.701667	4.701667	4.701667	4.511215	0.005656	0.005656	0.005656	0.005209
217-218	1.567804	1.567804	1.567804	1.504253	0.003357	0.003357	0.003357	0.003091
217-219	1.567375	1.567375	1.567375	1.503873	0.002264	0.002264	0.002264	0.002084
189-220	19.009492	19.009492	19.009492	19.002832	0.047393	0.047393	0.047393	0.047361
220-221	6.134314	6.134314	6.134314	6.132183	0.015582	0.015582	0.015582	0.015571
221-222	3.771135	3.771135	3.771135	3.769819	0.021161	0.021161	0.021161	0.021147
220-223	12.875179	12.875179	12.875179	12.870650	0.029697	0.029697	0.029697	0.029676
223-224	11.369352	11.369352	11.369352	11.365347	0.026438	0.026438	0.026438	0.026420
189-225	10.443281	10.443281	10.443281	10.439693	0.023837	0.023837	0.023837	0.023821
225-226	8.103550	8.103550	8.103550	8.100754	0.018631	0.018631	0.018631	0.018619
226-227	6.611021	6.611021	6.611021	6.608734	0.015283	0.015283	0.015283	0.015273
227-228	4.251237	4.251237	4.251237	4.249761	0.010840	0.010840	0.010840	0.010832
228-229	3.312262	3.312262	3.312262	3.311112	0.008471	0.008471	0.008471	0.008465
229-230	0.941013	0.941013	0.941013	0.940686	0.000175	0.000175	0.000175	0.000175
189-231	10.984525	10.984525	10.984525	10.980766	0.025060	0.025060	0.025060	0.025043
231-232	2.343384	2.343384	2.343384	2.342589	0.004977	0.004977	0.004977	0.004974
231-233	8.641141	8.641141	8.641141	8.638177	0.020837	0.020837	0.020837	0.020823
233-234	2.431802	2.431802	2.431802	2.430967	0.006187	0.006187	0.006187	0.006183
234-235	0.935652	0.935652	0.935652	0.935331	0.002386	0.002386	0.002386	0.002385
233-236	3.856555	3.856555	3.856555	3.855229	0.008936	0.008936	0.008936	0.008930
236-237	1.498490	1.498490	1.498490	1.497975	0.003484	0.003484	0.003484	0.003482
189-238	15.900303	15.900303	15.900303	15.894479	0.059737	0.059737	0.059737	0.059696
238-239	13.538801	13.538801	13.538801	13.533792	0.051741	0.051741	0.051741	0.051705
239-240	11.145924	11.145924	11.145924	11.141762	0.052911	0.052911	0.052911	0.052873
240-241	3.985971	3.985971	3.985971	3.984504	0.019343	0.019343	0.019343	0.019329
241-242	1.548726	1.548726	1.548726	1.548156	0.003276	0.003276	0.003276	0.003274
240-243	7.159954	7.159954	7.159954	7.157260	0.034613	0.034613	0.034613	0.034587
243-244	5.606456	5.606456	5.606456	5.604338	0.027391	0.027391	0.027391	0.027371
244-245	2.471312	2.471312	2.471312	2.470377	0.012204	0.012204	0.012204	0.012195
244-246	3.135144	3.135144	3.135144	3.133961	0.006699	0.006699	0.006699	0.006694
246-247	1.568231	1.568231	1.568231	1.567639	0.003359	0.003359	0.003359	0.003356

Appendix S

Cables and over headlines specifications

Medium Voltage Cables												
8.7/15 (17.5) kV												
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="width: 60%;"> <p>Single & Three Cores Copper Conductors, XLPE Insulated and PVC Sheathed</p> <p>Description</p> <ul style="list-style-type: none"> Stranded circular compacted Copper conductor, semiconducting layer as conductor screen, XLPE insulated, semiconducting layer as non metallic insulation screen, Copper tape or wire as metallic insulation screen, three cores assembled together with non hygroscopic Polypropylene fillers, wrapped with binder tape and PVC sheathed. Cables are produced according to IEC 60502 or BS 6622. <p>Application</p> <ul style="list-style-type: none"> These cables are generally suitable for direct burial or for installation on trays or in ducts. </div> <div style="width: 35%; text-align: center;">  </div> </div>												
Product - Code	Nominal Cross Sectional Area mm ²	Max. Conductor Resistance		Operating Capacitance µf/km	Inductance		Current Rating				Approx. Overall Diameter mm	Approx. Weight kg/km
		DC at 20 °C Ω/km	AC at 90 °C Ω/km		Trefoil mh/km	Flat mh/km	Laid in Ground		Laid in Free Air			
					⊗	⊙	⊗	⊙	⊗	⊙		
						A	A	A	A			
Single Core Cable												
CX3-T101-U12	25	0.7270	0.9271	0.165	0.469	0.654	139	145	143	173	22.0	729
CX3-T101-U13	35	0.5240	0.6683	0.183	0.446	0.631	166	174	174	211	23.3	846
CX3-T101-U14	50	0.3870	0.4937	0.204	0.412	0.597	197	206	209	255	24.7	999
CX3-T101-U15	70	0.2680	0.3420	0.234	0.387	0.572	242	252	262	319	26.5	1230
CX3-T101-U16	95	0.1930	0.2465	0.253	0.373	0.558	287	299	317	384	27.7	1479
CX3-T101-U17	120	0.1530	0.1956	0.275	0.361	0.546	326	339	366	443	29.3	1761
CX3-T101-U18	150	0.1240	0.1588	0.297	0.351	0.536	364	375	416	498	30.9	2111
CX3-T101-U19	185	0.0991	0.1272	0.324	0.338	0.522	411	422	478	570	32.8	2512
CX3-T101-U20	240	0.0754	0.0973	0.360	0.324	0.509	475	484	565	669	35.4	3083
CX3-T101-U30	300	0.0601	0.0781	0.402	0.313	0.498	535	542	650	765	38.0	3718
CX3-T101-U40	400	0.0470	0.0618	0.442	0.304	0.489	599	591	745	853	41.0	4661
CX3-T101-U50	500	0.0366	0.0490	0.495	0.293	0.478	674	659	859	975	44.6	5808
CX3-T101-U60	630	0.0283	0.0391	0.548	0.285	0.470	752	728	981	1103	48.3	7127
CX3-T101-U70	800	0.0221	0.0319	0.616	0.277	0.462	869	979	1160	1480	53.2	9010
CX3-T101-U80	1000	0.0176	0.0234	0.759	0.274	0.459	1084	1188	1515	1864	63.6	11341
Three Core Cables												
CX3-T103-U12	25	0.7270	0.9271	0.165	0.431	-	140	-	141	-	42.9	2105
CX3-T103-U13	35	0.5240	0.6684	0.183	0.408	-	167	-	171	-	45.5	2485
CX3-T103-U14	50	0.3870	0.4938	0.204	0.376	-	197	-	206	-	48.3	2990
CX3-T103-U15	70	0.2680	0.3423	0.234	0.354	-	241	-	257	-	52.3	3725
CX3-T103-U16	95	0.1930	0.2469	0.253	0.340	-	286	-	309	-	55.1	4505
CX3-T103-U17	120	0.1530	0.1961	0.275	0.329	-	325	-	356	-	58.4	5410
CX3-T103-U18	150	0.1240	0.1595	0.297	0.320	-	364	-	405	-	62.0	6415
CX3-T103-U19	185	0.0991	0.1282	0.324	0.308	-	410	-	463	-	65.9	7685
CX3-T103-U20	240	0.0754	0.0986	0.360	0.295	-	475	-	546	-	71.0	9540
CX3-T103-U30	300	0.0601	0.0799	0.402	0.288	-	535	-	626	-	76.9	11590

3.2

8.7/15 (17.5) kV

Single & Three Cores Aluminium Conductors,
XLPE Insulated and PVC Sheathed



Description

- Stranded circular compacted Aluminium conductor, semiconducting layer as conductor screen, XLPE insulated, semiconducting layer as non metallic insulation screen, Copper tape or wire as metallic insulation screen, three cores assembled together with non hygroscopic Polypropylene fillers, wrapped with binder tape and PVC sheathed.
- Cables are produced according to IEC 60502 or BS 6622.

Application

- These cables are generally suitable for direct burial or for installation on trays or in ducts.

Product - Code	Nominal Cross Sectional Area mm ²	Max. Conductor Resistance		Operating Capacitance µf/km	Inductance		Current Rating				Approx. Overall Diameter mm	Approx. Weight kg/km
		DC at 20 °C Ω/km	AC at 90 °C Ω/km		Trefoil mh/km	Flat mh/km	Laid in Ground		Laid in Free Air			
							Trefoil A	Flat A	Trefoil A	Flat A		
Single Core Cable												
AX3-T101-U12	25	1.2000	1.5390	0.165	0.469	0.654	108	113	111	135	22.0	575
AX3-T101-U13	35	0.8680	1.1130	0.183	0.446	0.631	129	135	135	164	23.3	635
AX3-T101-U14	50	0.6410	0.8220	0.204	0.412	0.597	153	160	162	198	24.6	700
AX3-T101-U15	70	0.4430	0.5681	0.234	0.387	0.572	188	196	204	248	26.2	800
AX3-T101-U16	95	0.3200	0.4105	0.253	0.373	0.558	222	233	246	299	27.7	905
AX3-T101-U17	120	0.2530	0.3247	0.275	0.361	0.546	254	265	284	346	29.3	1020
AX3-T101-U18	150	0.2060	0.2645	0.297	0.351	0.536	283	295	324	391	30.9	1215
AX3-T101-U19	185	0.1640	0.2107	0.324	0.338	0.522	321	333	373	449	32.8	1370
AX3-T101-U20	240	0.1250	0.1610	0.360	0.324	0.509	372	385	443	530	35.3	1595
AX3-T101-U30	300	0.1000	0.1291	0.402	0.313	0.498	421	433	511	608	37.7	1825
AX3-T101-U40	400	0.0778	0.1009	0.442	0.304	0.489	478	483	593	693	41.0	2270
AX3-T101-U50	500	0.0605	0.0791	0.495	0.293	0.478	544	545	693	802	44.6	2670
AX3-T101-U60	630	0.0469	0.0621	0.548	0.288	0.470	617	612	803	921	48.2	3180
AX3-T101-U70	800	0.0367	0.0495	0.616	0.277	0.462	715	786	954	1188	53.2	3870
AX3-T101-U80	1000	0.0291	0.0376	0.759	0.274	0.459	862	937	1205	1468	63.6	4990
Three Core Cables												
AX3-T103-U12	25	1.2000	1.5390	0.165	0.431	-	108	-	110	-	42.5	1605
AX3-T103-U13	35	0.8680	1.1130	0.183	0.408	-	129	-	133	-	45.1	1815
AX3-T103-U14	50	0.6410	0.8220	0.204	0.376	-	153	-	160	-	47.6	2040
AX3-T103-U15	70	0.4430	0.5683	0.234	0.354	-	187	-	200	-	51.3	2375
AX3-T103-U16	95	0.3200	0.4107	0.253	0.340	-	222	-	240	-	54.7	2730
AX3-T103-U17	120	0.2530	0.3250	0.275	0.329	-	253	-	277	-	58.0	3120
AX3-T103-U18	150	0.2060	0.2649	0.297	0.320	-	283	-	314	-	61.6	3645
AX3-T103-U19	185	0.1640	0.2114	0.324	0.308	-	320	-	361	-	65.7	4205
AX3-T103-U20	240	0.1250	0.1618	0.360	0.295	-	371	-	427	-	70.4	4975
AX3-T103-U30	300	0.1000	0.1302	0.402	0.288	-	420	-	491	-	76.0	5775

Nominal cross sectional area	Nominal cross sectional area		Conductor construction		Approx. overall diameter	Approx. overall weight	Calculated DC Resistance at 20 °C	Ampacity (*)	Rated strength	AES Code
	Al	St	Al	St						
	mm ²		No. x Ø (mm)							
16 / 2.5	15.27	2.54	6 x 1.80	1 x 1.80	5.4	62	1.87930	105	5.81	S02F700070DMX
25 / 4	23.86	3.98	6 x 2.25	1 x 2.25	6.8	97	1.20280	140	9.02	S02F800070DMX
35 / 6	34.35	5.73	6 x 2.70	1 x 2.70	8.1	140	0.83530	170	12.70	S02F900070DMX
44 / 32	43.98	31.67	14 x 2.00	7 x 2.40	11.2	373	0.65730	195	45.46	S02G000210DMX
50 / 8	48.25	8.04	6 x 3.20	1 x 3.20	9.6	196	0.59460	210	17.18	S02G100070DMX
50 / 30	51.17	29.85	12 x 2.33	7 x 2.33	11.7	378	0.56440	213	44.28	S02G200190DMX
70 / 12	69.89	11.40	26 x 1.85	7 x 1.44	11.7	284	0.41300	290	26.31	S02G300330DMX
95 / 15	94.39	15.33	26 x 2.15	7 x 1.67	13.6	383	0.30580	350	35.17	S02G400330DMX
95 / 55	96.61	66.30	12 x 3.20	7 x 3.20	16.0	714	0.29920	367	80.20	S02G500190DMX
105 / 75	105.67	75.55	14 x 3.10	19 x 2.25	17.5	899	0.27360	394	106.69	S02G600330DMX
120 / 20	121.57	19.85	26 x 2.44	7 x 1.90	15.5	494	0.23740	410	44.94	S02G700330DMX
120 / 70	122.15	71.25	12 x 3.60	7 x 3.60	18.0	904	0.23640	724	98.16	S02G800190DMX
125 / 30	127.92	29.85	30 x 2.33	7 x 2.33	16.3	590	0.22590	425	57.86	S02G900370DMX
150 / 25	148.86	24.25	26 x 2.70	7 x 2.10	17.1	604	0.19390	470	54.37	S02H000330DMX
170 / 40	171.77	40.08	30 x 2.70	7 x 2.70	18.9	794	0.16820	520	77.01	S02H100370DMX
185 / 30	183.78	29.85	26 x 3.00	7 x 2.33	19.0	744	0.15710	535	66.28	S02H200330DMX

Appendix T

Tables

Table 3.4

Literature and Improved ABC Results Comparison of the IEEE 33 Buss System

Simulation case	method	Optimal location (bus number)	DER optimal output (MW)	Total capacity added (MW)	Power losses (MW)
1 DER unit	ABC [64]	6	2.5775	2.5775	0.1050
	Our ABC	6	2598.6	2598.6	0.1054
	PSO-CFA [46]	6	2.5752	2.5752	0.1039
	ACO-ABC [46]	6	2.5753	2.5753	0.1039
	Prupused IABC	6	2.5476	2547.6	0.10382
2 DER units	ABC	6	1.9707	2.5464	0.0899
		15	0.5757		
	Our ABC	6	1764.2	2388.7	0.08692
		14	624.5		
	PSO-CFA	14	0.7876	2.0363	0.0862
		29	1.2487		
	ACO-ABC [46]	13	0.8464	2.0052	0.0859
		30	1.1588		
	Prupused IABC	14	0.7854	1.9321	0.08242
		30	1.146.7		
3 DER units	ABC [64]	6	1.7569	3.1152	0.0792
		15	0.7826		
		25	0.5757		
	Our ABC	13	753.6	2923.9	0.0765
		23	1100		
		28	1070.3		
	PSO-CFA [46]	10	1.0491	2.7326	0.0760
		25	0.8786		
		33	0.8049		
	ACO-ABC [46]	14	0.7547	2.9260	0.0714
		24	1.0999		
		30	1.0714		
	Prupused IABC	13	0.8374	2.6639	0.07108
24		0.7637			
30		1.0628			

Table 3.5*Improved ABC on Nablus Network Results Summary*

	Original case	PV in one location	PV in two locations	PV in three locations
Bus number	-	112	106, 165	106, 169, 214
PV size (KWP)	-	3857.5	4156.2, 2887.4	4073.7, 2347.6, 1863.1
Total PV capacity	-	3857.5	7043.6	8284.4
Min bus voltage	0.7079	0.844	0.897	0.896
Max bus voltage	1.00	1.00	1.00	1.00
total active power losses (KW)	2759.86	1958.12	1668.05	1540.45
Percentage of losses active reduction	0%	29.05%	39.56%	44.18%
Total reactive power losses (KVAR)	1666.6	1183.93	1011.54	913.4
Percentage of reactive losses reduction	0%	28.96%	39.3%	45.19%
Max Line limit index	81.05%	80.52%	69.32%	89.69%
Max voltage stability index	0.132104	0.097492	0.082323	0.083884
Best Fitness function value	0.7955	0.5804	0.4873	0.4658

Appendix U

Figures

Figure 2.1

Typical Buses of Power System

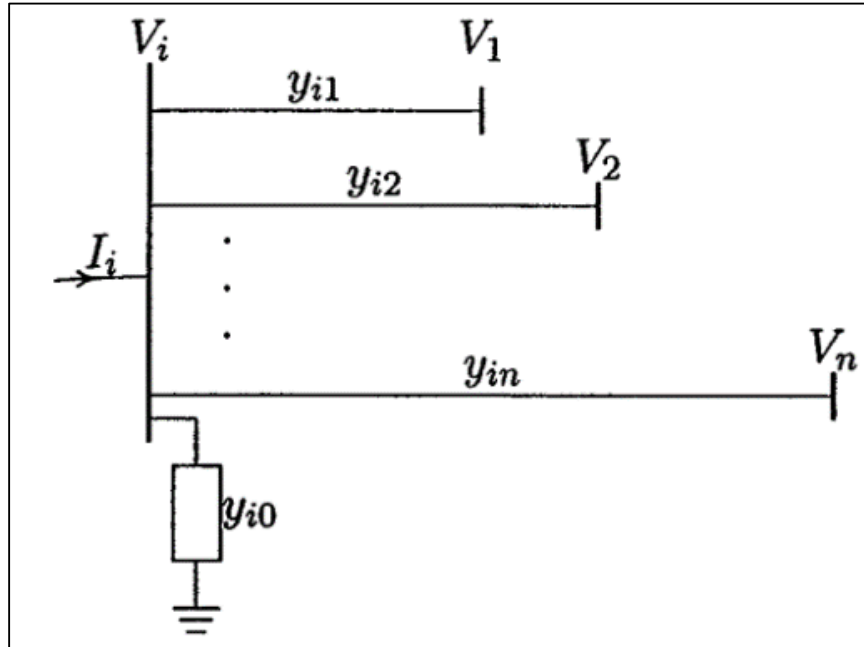


Figure 2.2

Flowchart of ABC

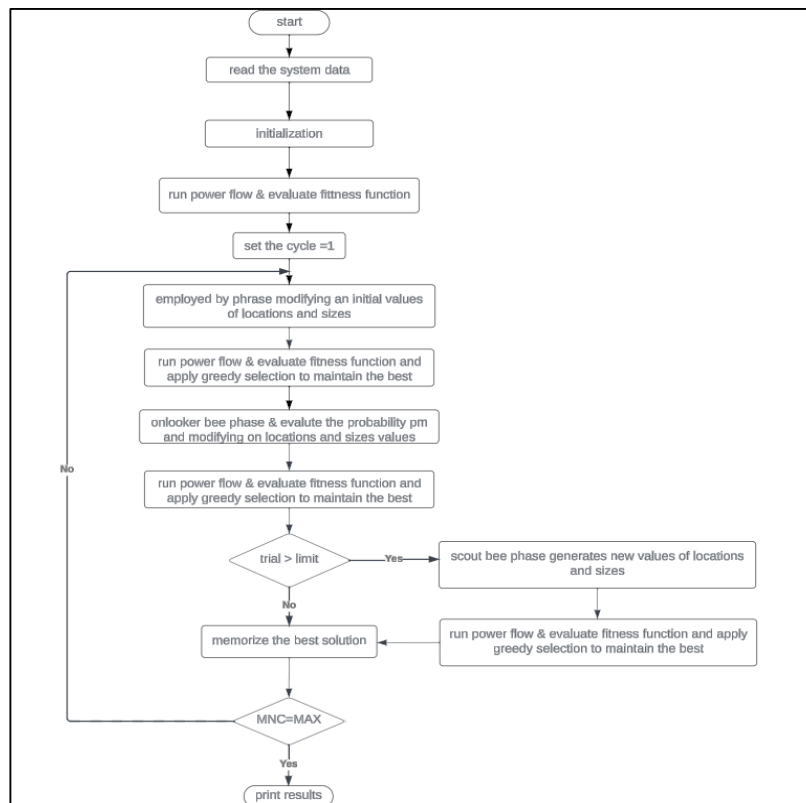


Figure 2.3

Flowchart of Improved ABC

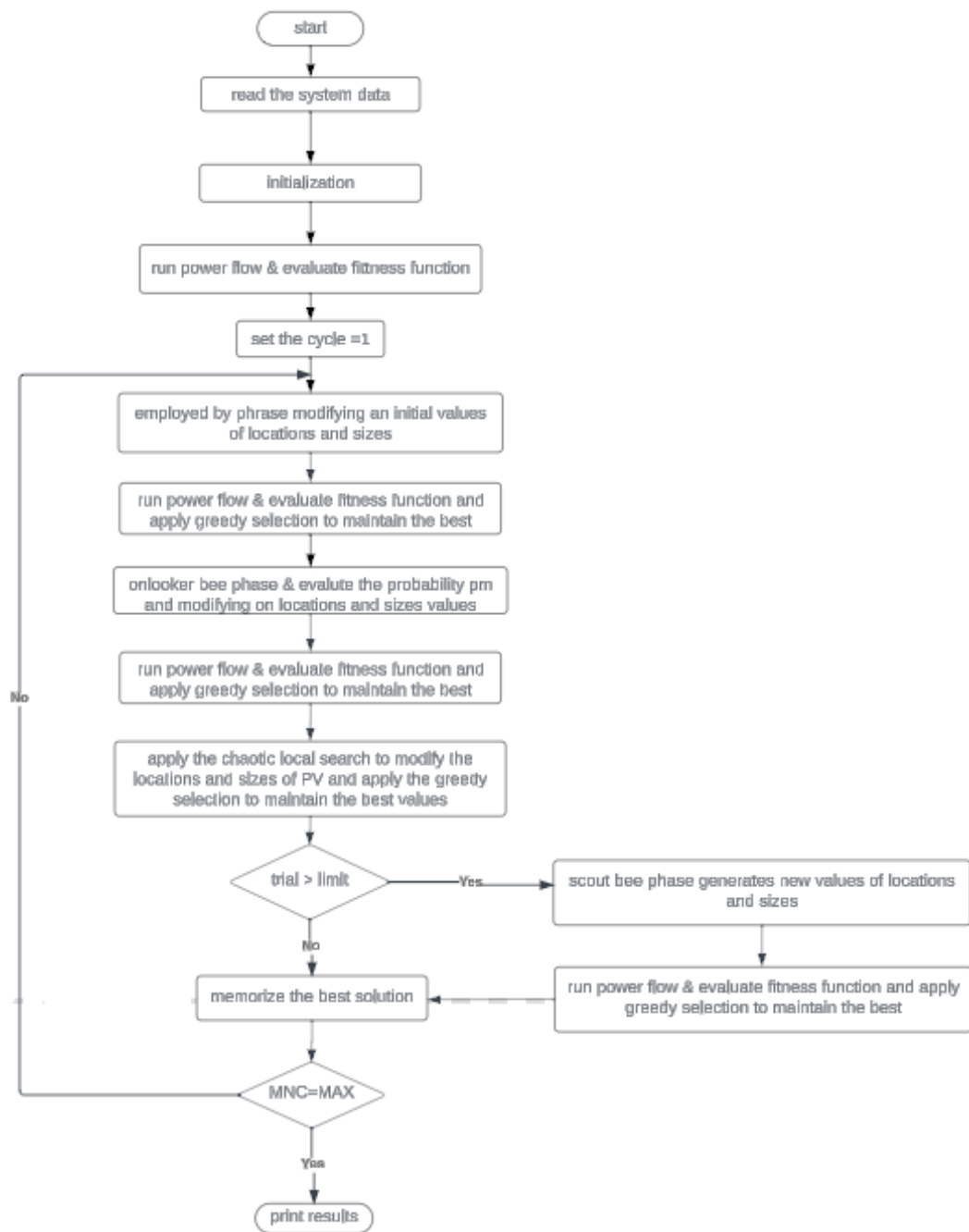


Figure 3.1
IEEE 33 Buss System Diagram

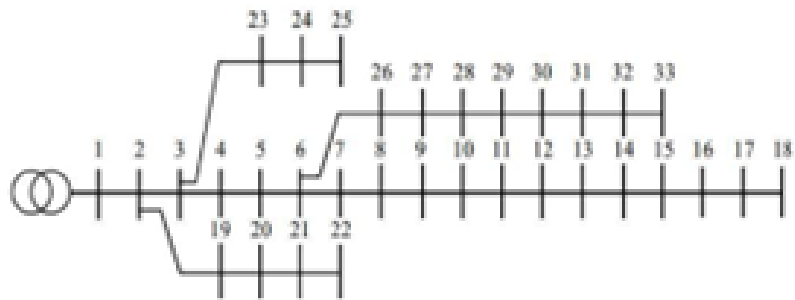


Figure 3.2
Flowchart of the ABC Algorithm

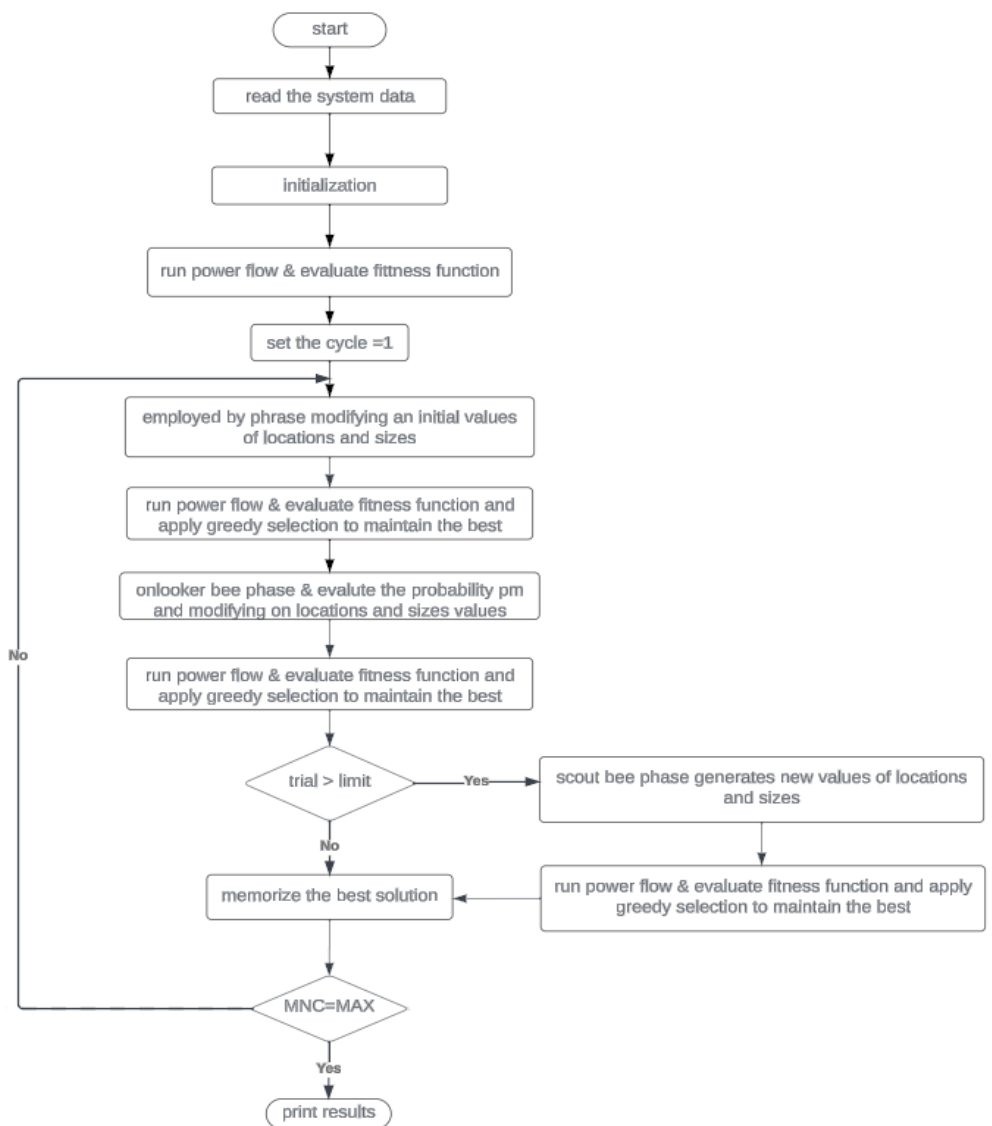


Figure 3.3

Fitness and Convergence Analysis for Different Configurations

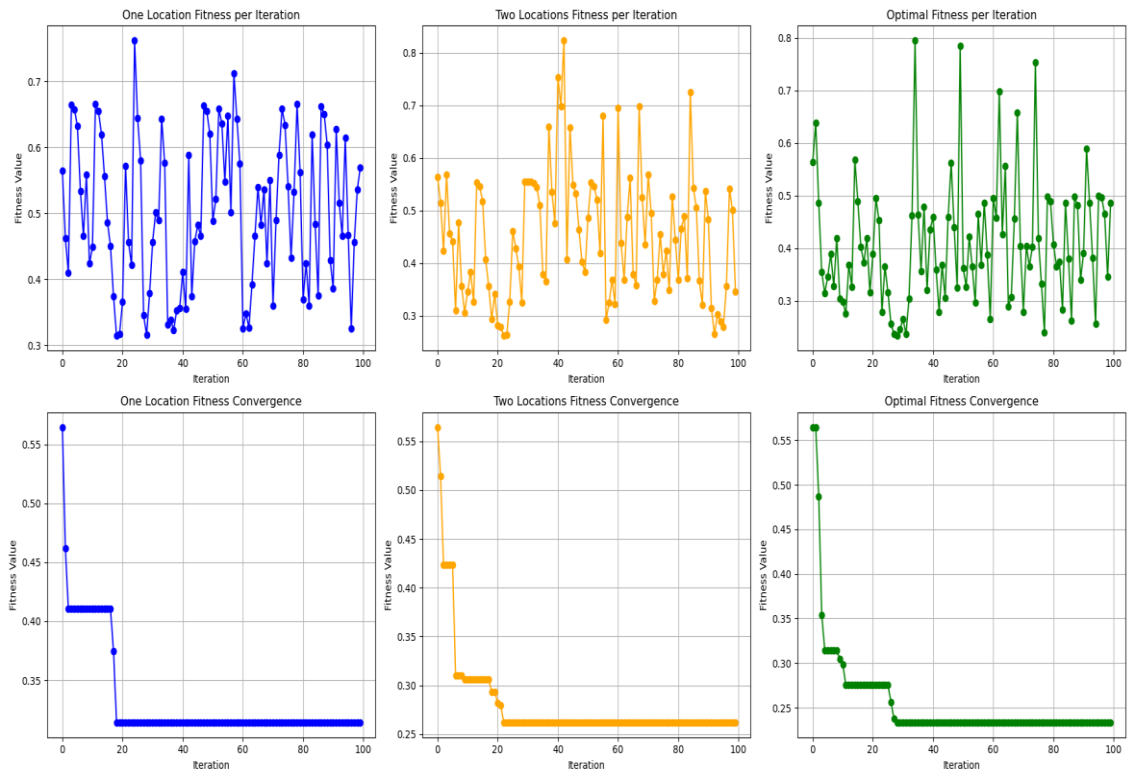


Figure 3.4

Flowchart of the Improved ABC

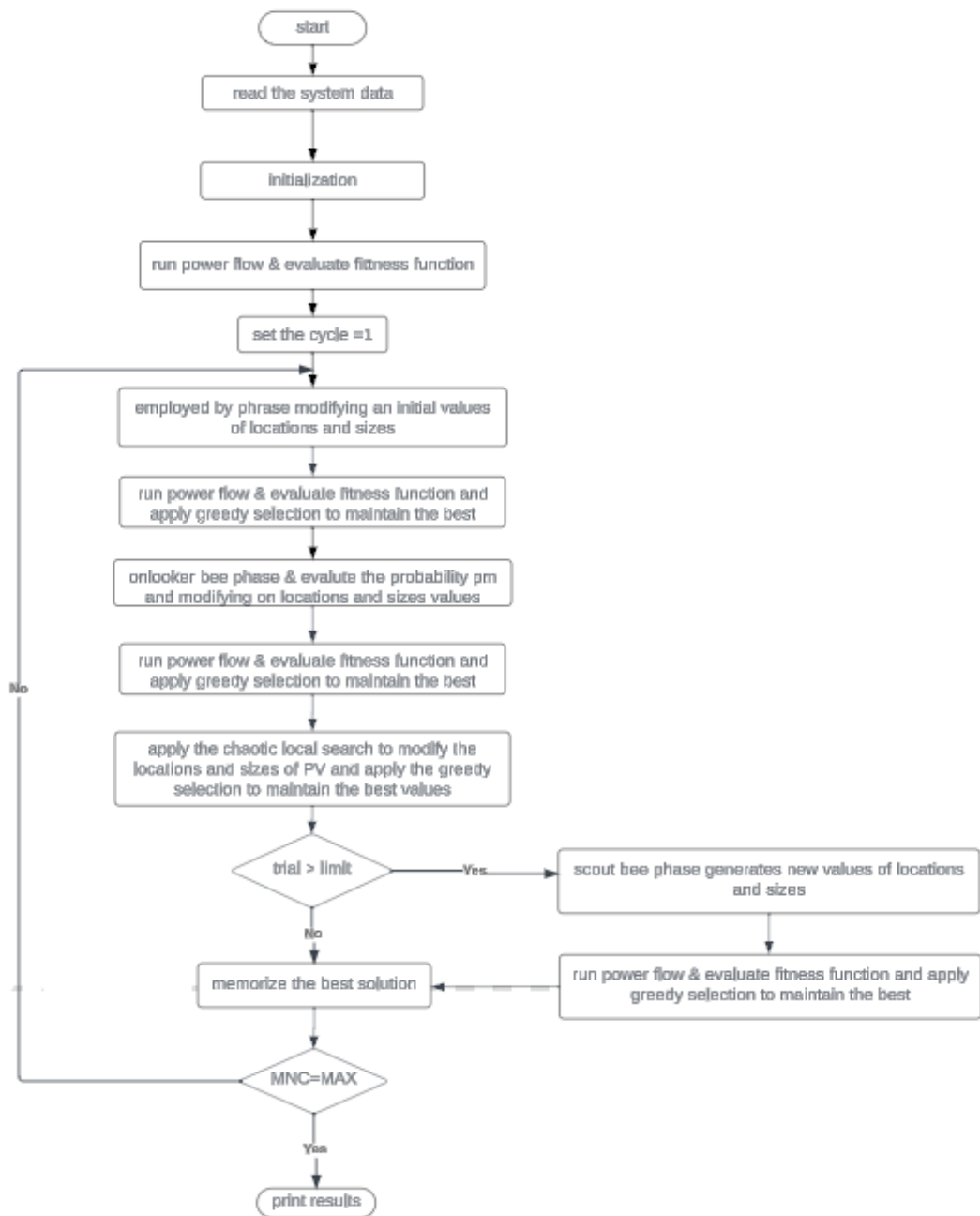


Figure 3.5

Fitness and Convergence Analysis for Different Configurations - IEEE 33 Chaotic Bees Algorithm

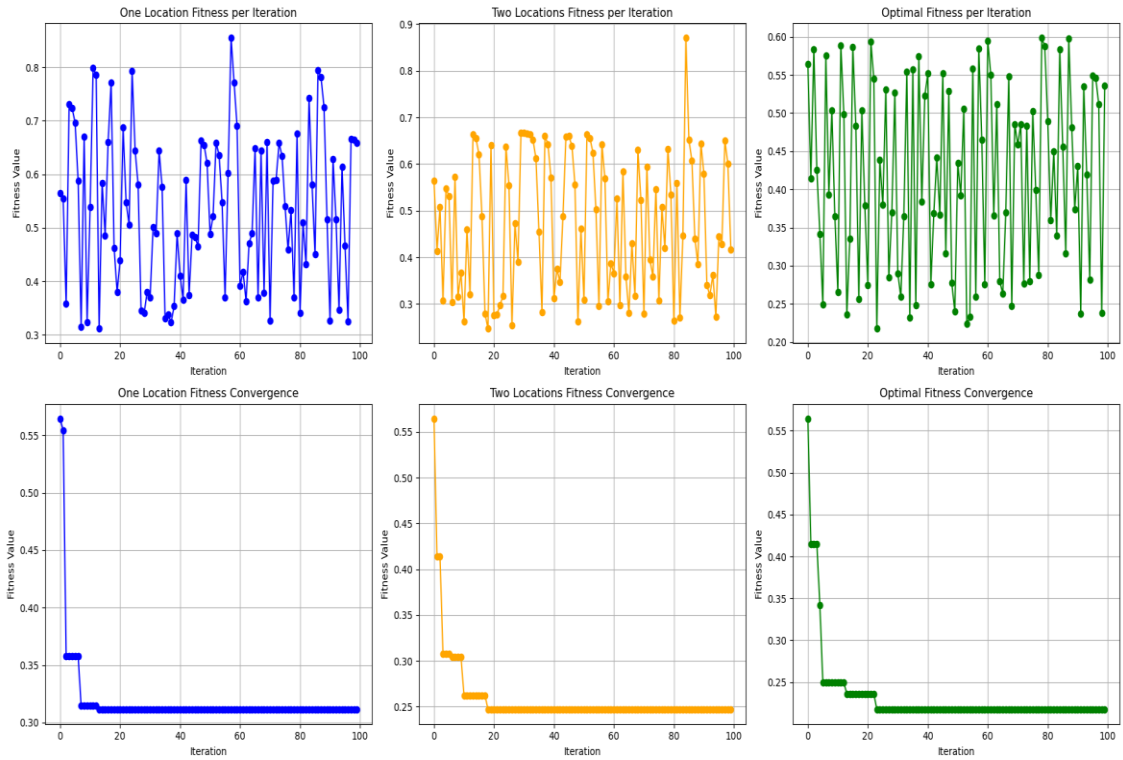
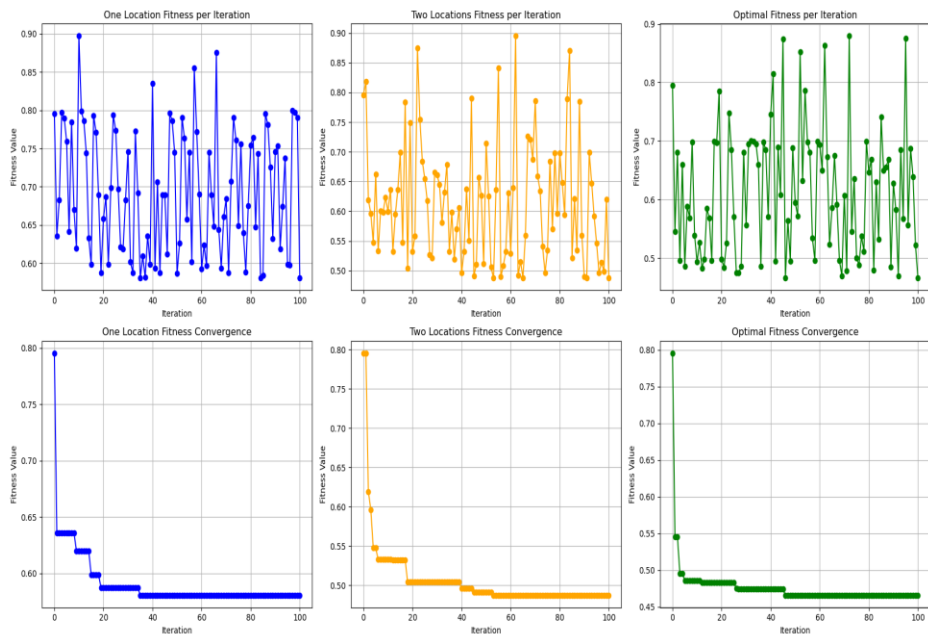


Figure 3.6

The Convergence Characteristic of the Optimum Iteration Using Improved ABC on the Nablus network





جامعة النجاح الوطنية
كلية الدراسات العليا

التحجيم والتوزيع الامثل للأنظمة الكهروضوئية لتحسين اداء شبكات
الكهرباء - دراسة حالة شركة الشمال

إعداد

عمر نجم الدين صلاح بريك

إشراف

د. عماد بريك

قدمت هذه الرسالة استكمالاً لمتطلبات الحصول على درجة الماجستير في هندسة القوى الكهربائية، من كلية الدراسات العليا، في جامعة النجاح الوطنية، نابلس - فلسطين.

2024

التحجيم والتوزيع الأمثل للأنظمة الكهروضوئية لتحسين أداء شبكات الكهرباء - دراسة حالة شركة الشمال

إعداد

عمر نجم الدين صلاح بريك

إشراف

د. عماد بريك

الملخص

تقترح هذه الأطروحة حل المواقع والاحجام المثلى لانظمة الطاقة الشمسية الكهروضوئية في شبكات توزيع الكهرباء حيث ان اضافة انظمة طاقة شمسية امر ذو اهمية كبيرة في الاونة الاخيرة وان وضع هذه الانظمة يجب ان يكون بناءا على معايير ودراسات لتعظيم الاستفادة منها.

في هذا البحث، تم استخدام خوارزمية النحل الفوضوي لتحديد الحجم والموقع الأمثل لأنظمة الطاقة الشمسية الكهروضوئية، تحل خوارزمية النحل التقليدية في معظم الدراسات مشكلة وضع وحجم الألواح الشمسية الكهروضوئية الأمثل من خلال التركيز على تقليل الخسائر مع تحسين استقرار الجهد فقط على الأنظمة والشبكات الصغيرة، لقد قمنا بتعديل الخوارزمية بحيث يزيد الشكل المعدل للخوارزمية باستخدام رسم الخرائط الفوضوي من مساحة البحث ويعيد نتائج أفضل بشكل أسرع ويتعامل مع الشبكات الأكبر.

التي تضم 33 نقطة ربط كمثال من اجل تطبيق خوارزمية النحل التقليدية لتقليل الفاقد ودعم IEEE 33 تم اختيار شبكة اختبار استقرارية وتحسين ملف الجهد ثم تم تطوير الخوارزمية باضافة خارطة عشوائية للمساعدة في توسيع مناطق البحث عن حلول واخراج حلول افضل ومن ثم تمت مقارنة كفاءة النتائج للشبكة لكل من الحالات التالية وهي :عدم وجود انظمه كهروضوئية، التوزيع العشوائي والتوزيع باستخدام خوارزمية النحل التقليدية والتوزيع باستخدام خوارزمية النحل المطورة وقد اثبتت الخوارزمية المطورة نجاحها باخراج افضل الحلول وقامت بافضل التحسينات للشبكة.

من ثم تم اختبار هذه الخوارزمية على شبكة اوسع وهي جزء من شبكة نابلس والتي تضمنت 247 نقطة اتصال و 213 محوّلًا، بناءً على المعلومات الواردة من مهندس التشغيل، فإننا نأخذ في الاعتبار عامل الحمل 0.4 وعامل القدرة 0.95 للمحولات لأننا لا نمتلك بيانات مقاسة في جميع المحولات الـ 213. ومن ثم الحصول على افضل الاماكن والاحجام للانظمة الكهروضوئية في الشبكة.

الذي يمكن من دراسة وتحليل الشبكات الكهربائية وتطبيق Jupyter notebook مع برنامج Python تم اعتماد لغة البرمجة كافة الخوارزميات عليها.

الكلمات المفتاحية: التحسين، أنظمة الطاقة الكهروضوئية، الشبكة الكهربائية.