Electrical Power Energy Optimization at Hydrocarbon Industrial Plant Using Intelligent Algorithms

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

by

Muhammad T. Al-Hajri

Department of Electronics and Computer Engineering
College of Engineering, Design and Physical Science

Brunel University

London, UK

May, 2016

Abstract

In this work, the potential of intelligent algorithms for optimizing the real power loss and enhancing the grid connection power factor in a real hydrocarbon facility electrical system is assessed. Namely, genetic algorithm (GA), improve strength Pareto evolutionary algorithm (SPEA2) and differential evolutionary algorithm (DEA) are developed and implemented. The economic impact associated with these objectives optimization is highlighted. The optimization of the subject objectives is addressed as single and multi-objective constrained nonlinear problems. Different generation modes and system injected reactive power cases are evaluated. The studied electrical system constraints and parameters are all real values.

The uniqueness of this thesis is that none of the previous literature studies addressed the technical and economic impacts of optimizing the aforementioned objectives for real hydrocarbon facility electrical system. All the economic analyses in this thesis are performed based on real subsidized cost of energy for the kingdom of Saudi Arabia. The obtained results demonstrate the high potential of optimizing the studied system objectives and enhancing the economics of the utilized generation fuel via the application of intelligent algorithms.

Declaration of Authorship

This is to declare that this thesis has not been submitted for a degree in any university. All cited information has been referenced.

Muhammad Tami Al-Hajri
May, 2016
London, UK

Dedication

I would like to dedicate this thesis to the soul of my great mother Mounirah Al-Dossary. Special thanks for my father and my lovely kids Mounirah, Tami, Salma, Joud and Turki for their emotional support.

Particularly and most sincerely, I would like to pay my warmest praise to my darling wife Maha Al-Hajri for her patient and great emotion support during the PhD journey. With her endless love this achievement was possible.

Acknowledgement

I would like to acknowledge all those who supported me to complete this thesis. Acknowledgment is due to Brunel University for providing the support to carry out this research.

I really appreciate the excellent effort by Dr. Mohamed Darwish from Brunel University and Prof. Mohamed Abido from King Fahd University of Petroleum & Minerals for their technical and academic support and encouragement. The thanks is extended to my thesis committee members.

I would like to thank Saudi ARAMCO Oil Company specially the power systems admin area for providing the logistic support during my research.

Table of Contents

LIST OF ABRIVATIONS

Abbreviations	Description
ABC	Artificial Bee Colony
AC	Ant Colony
BAU	Business as Usual Case
BTU	British thermal unit
DEA	Differential Evolution Algorithm
DC	differential crossover
DEA	Differential Evolution Algorithm
ETAP	Electrical Transient Analysis Program
EA	Evolution Algorithm
ESP	Electrical Submersible Pumps
GA	Genetic Algorithm
GTG	Gas Turbine Generators
GIS	Gas Insulated Switchgear
GCPF	Grid Connection Power Factor
GEA	Generic Evolution Algorithm
GW	Gaga-Watts
HP	Horse Power (HP)
ISF	Industrial Services Facilities
IPSO	Improved Particle Swarm Optimization
IA	Intelligent Algorithm
IGA	Integer-Coded Genetic Algorithm
KWH	Kilowatts Hours
LP	Liner Programming
LV	Low Voltage
L_Index	The System Voltage Stability Index
MCC	Motor Control Center
MVAR	Mega-Voltage Ampere Reactive
MW	Mega-Watts
MMSCF	Million Standards Cubical Feet Of Gas
MMBTU	Million BTU
MV	Medium Voltage
MOEA	Multi-Objective Evolution Algorithm
NLP	Nonlinear Programming
NSGA-II	Non-Dominated Sorting Genetic Algorithm II
NB	Buses number
NG	Number of Generators
PSO	Particle Swarm Optimization
PMS	Power Management System
P_{Loss}	Real Power Loss
PR_{Inject}	Injected Real Power Into The Grid

P_{Gi}	Generator Real Power
P_{Di}	Load Real Power
P_{li}	Line Real Power Loss
QP	Quadratic Programming
Q_{Gi}	Generator Reactive Power
Q_{Di}	Load Reactive Power
Q_{li}	Line Reactive Power Loss
Q^{min}_{GTGi}	GTG Minimum Reactive Power Outputs
Q^{max}_{GTGi}	GTG Maximum Reactive Power Outputs
Q^{min}_{STG1}	Small STG Minimum Reactive Power Outputs
Q^{max}_{STG1}	Small STG Maximum Reactive Power Outputs
Q^{min}_{STG2}	Large STG Minimum Reactive Power Outputs
Q^{max}_{STG2}	Large STG Maximum Reactive Power Outputs
QC_{Synch}	Captive Synchronous Motors Reactive Power
Q_{Synch}	Sub#9 Synchronous Motors Reactive Power
RPL	Real Power System Loss
SPEA	Strength Pareto Evolution Algorithm
SPEA2	The Improved Strength Pareto Evolution Algorithm
STG	Steam Turbine Generators
SWG	Switchgear
SR	Saudi Riyal
SPEA2	Improve Strength Pareto Evolution Algorithm
UBV	Utility Bus Voltage
V^{min}_{GTGi}	GTG Minimum Terminal Voltage
V^{max}_{GTGi}	GTG Maximum Terminal Voltage
V^{min}_{STG1}	Small STG Minimum Terminal Voltage
V^{max}_{STG1}	Small STG Maximum Terminal Voltage
V^{min}_{STG2}	Large STG Minimum Terminal Voltage
V^{max}_{STG2}	Large STG Maximum Terminal Voltage
V_{Synch}	Synchronous Motors Terminal Voltage
WSCC	Western System Coordination Council

LIST OF TABELS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Electrical Generation Challenges

In most of the developing countries the electrical generation is very pressing social and economic issue. The annual exponential increase in the electrical demand, the cost of the electrical generation fuel and the generation low efficiency urged most of these countries to unleash nationwide initiatives to improve the generation fleet efficiency and optimize the electrical energy usage. For example as illustrated in Figure 1.1, the Kingdom of Saudi Arabia average annual increase of electricity demand is 7.4% [1].

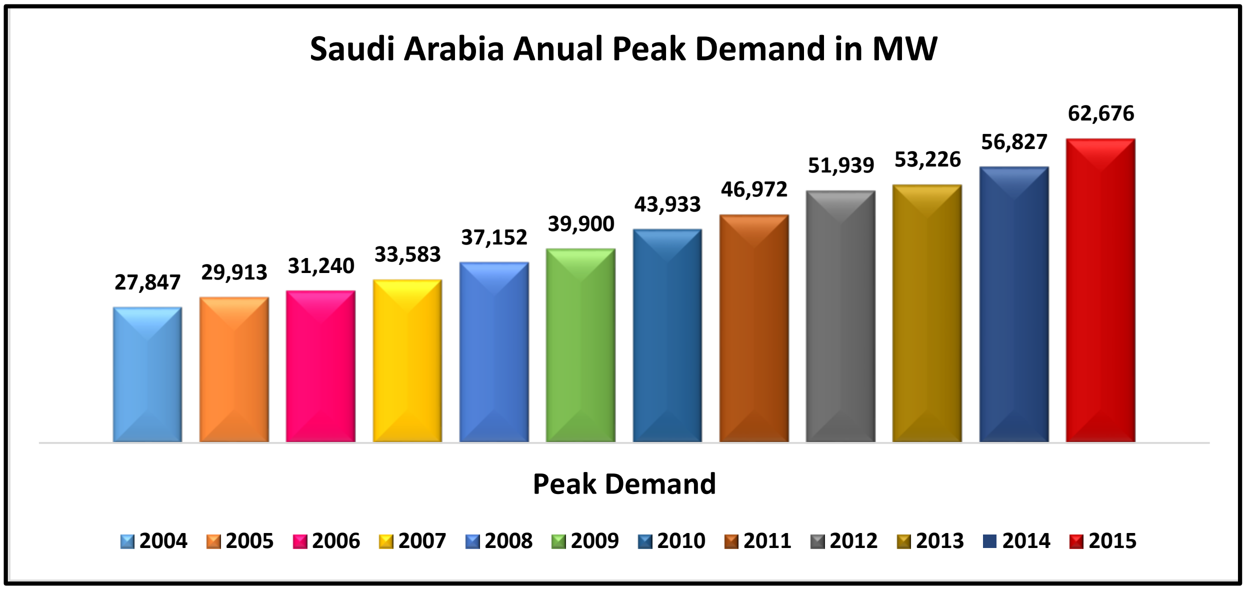

Fig. 1.1: *Saudi Arabia annual peak demand in MW*

In fact, in these countries, the high percentage of the electrical generation comes from low efficient power generation plants, such as the simple cycle steam turbine. This complicates the issue and creates burning platform for the utilization of the more efficient plant to the maximum and the reduction of the power system loss. In 2014, the distribution of plant capacity for electricity generation in Saudi Arabia by technology as posted in Figure 1.2

illustrates that the low efficient simple cycle steam turbine generators are making 34.5% of Saudi Arabia utility company generation fleet while the most efficient combined cycle generators is around 14.2% of the whole fleet [2].

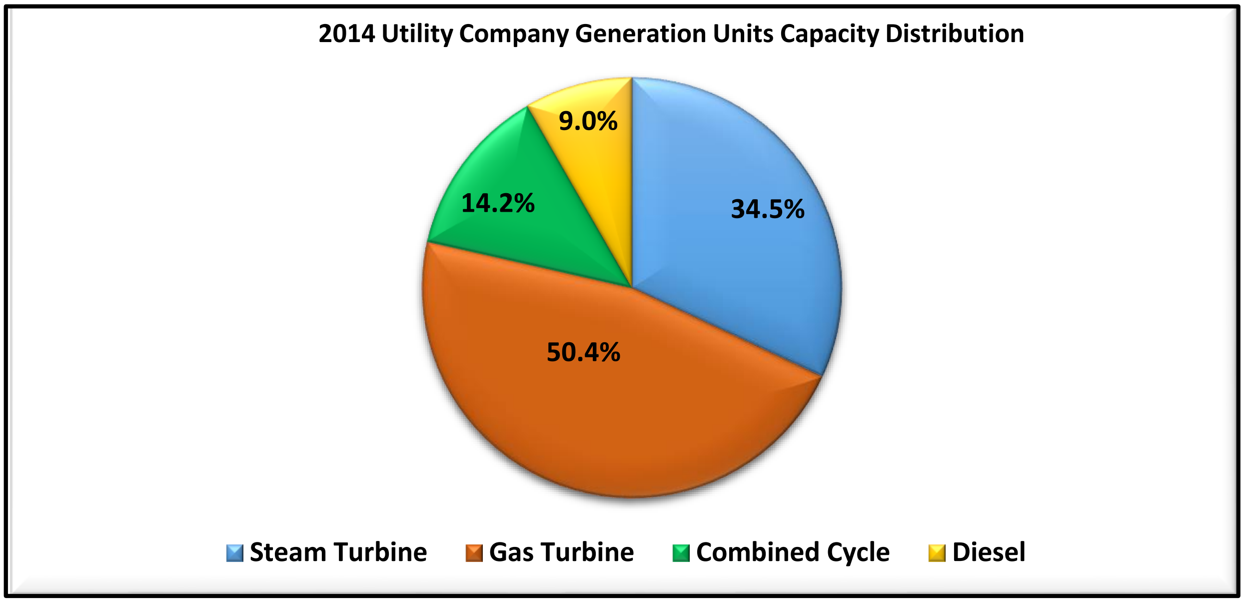

Fig. 1.2: *Saudi Arabia generation fleet mix*

1.2 Thesis Motivation

The kingdom of Saudi Arabia is very well known worldwide to be the biggest oil exporter to the world. Yet, lately the domestic demand for natural gas, heavy fuel oil, crude oil and diesel increased sharply due to the increase in electrical generation. For, example, as shown in Figure 1.3, the crude oil makes 32% of the average annual fuel used for electricity generation in Saudi Arabia [3]. This increase trend is very alarming as it develops a threat in jeopardizing the kingdom commitment to complement any shortage in the world oil supply. As a result, an environment of urgency was created to optimize the electrical power loss and produce power from the most efficient power generation plants. The motivation of this thesis is based on the fact that the hydrocarbon facilities in the kingdom of Saudi Arabia are bulk power demand hubs with possible high potential of power loss optimization. Most of these hydrocarbon facilities are equipped with their own high efficient co-generation and combined cycle power plants with 1.927 GW generation capacity [3], this unleashed the motivation of exploring the potential of optimizing real power loss within these plants and maximizing the power injected to the national grid from them.

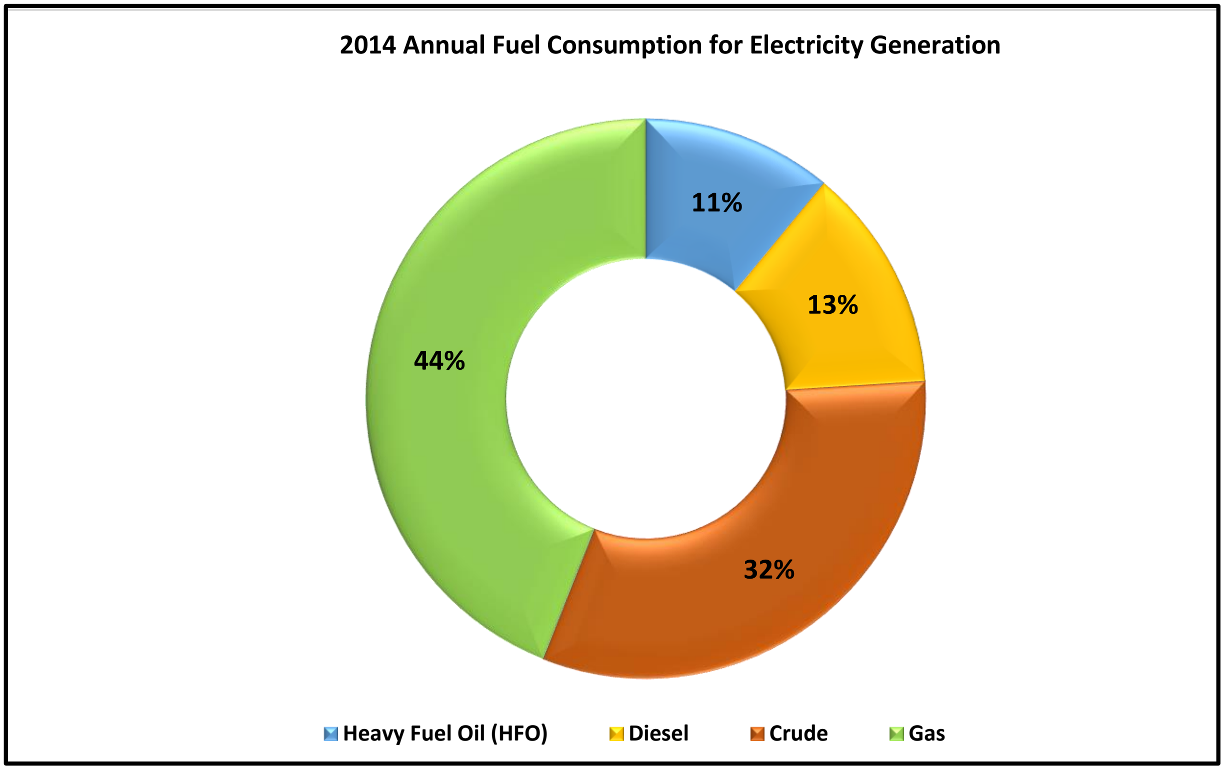

Fig. 1.3: *Saudi Arabia 2014 annual average fuel for electricity generation*

1.3 Thesis Objective

The thesis objective is to assess the potential of optimizing a real hydrocarbon facility power system real power loss using intelligent algorithms. The effects of treating this as single objective or coupled with other objectives, such as grid connection power factor and system voltage stability index is addressed in this thesis. Exploring the economic advantages when addressing the aforementioned objectives is another objective of this thesis.

1.4 Thesis Contribution

The utilization of real life hydrocarbon facility electrical system and its associated real system and equipment parameters and constraints for addressing the real power loss optimization via intelligent algorithms is the main contribution of this thesis. This is due to the fact that all the previous studies used the virtual IEEE, utility transmission system or distribution system electrical model in assessing the potential of intelligent algorithms for addressing the predefined system objectives. None of these studies utilized a hydrocarbon facility model with its unique system components. Such as, generation units, long HV cables, large synchronous and induction motors, utility loads, upstream production loads with very aggressive real constraints to be satisfied. Also, none of them considered the site condition and the efficiency and capacity curve of the generation units when addressing optimization problem. Another novelty of this thesis is that all the economic analyses are based on real subsidized fuel cost at the kingdom of Saudi Arabia.

1.5 Thesis Organization

In Chapter two (2) of this thesis, the literature will be reviewed with regard to the applications of intelligent algorithms in addressing power optimization problems and the usage of real system parameters in these applications. The real hydrocarbon facility electrical model development will be addressed in Chapter three (3). In Chapter four (4), the problem formulation will be developed. The problem programming using MATLAB software will be developed in Chapter five (5). The proposed solution methodology will be developed and implemented in Chapter six (6). In Chapter seven (7), the study cases and simulation results analysis will be highlighted. The conclusion and future work will be posted in Chapter eight (8).

CHAPTER 2: LITERATURE REVIEW

2.1 Conventional Optimization Methods

There are many conventional techniques which were applied to solve power system optimization problem such as the optimal power flow (OPF) as a single objective problem. Minimizing the power loss or reducing the generation fuel cost is the common objective for OPF problem. Among these techniques are the nonlinear programming (NLP), linear programing (LP), and quadratic programming (QP) and Interior Point (IP) method.

The NLP optimization method deals with problem involving nonlinear objectives and equality represented by the power flow equations and inequality constraints represented by the system limitations such as the bus voltage limitations. NLP has several drawbacks, including slow convergence, complexity, and difficulties involved in handling constraints [4]. Yet, its computation speed and accuracy for achieving the solution is out-perform the linear programming (LP) method [5].

LP treats problems with constraints and objective function formulated in linear forms with non-negative variables. In order for the linearized problem to accurately model the nonlinear problem, the movement of each variable must be restricted to a small region during each iteration. The LP is capable of handling inequality constraints and produce acceptable rates of convergence. The LP was subjected to tuning which enhances its capability to deal with large scale problems with many constraints. The incremental LP method is one method of tuning the LP traditional method. It can be precisely applied to a varied range of applications, and can produce the same results as NLP methods on power systems of all types and sizes. Several advantages of the incremental LP method over other existing methods include its reliability, its capability to quickly recognize problem infeasibility, its ability to handle a wide range of operating limits, a very fast computational speed, and better flexibility for trade-offs between computing speed and convergence accuracy [6, 7].

In the OPF problems, generator-fuel costs, and total system losses are usually expressed as quadratic functions of real and reactive power generation, it will be useful to use QP methods instead of LP [8]. The QP technique is a special form of nonlinear programming whose objective function is quadratic with linear constraints. Quadratic programming based techniques have some disadvantages associated with the piecewise quadratic cost approximation [6].

Interior Point (IP), among the traditional optimization techniques, is said to be computationally efficient but it suffers from sensitivity to the initial conditions and step size. The IP method converts the inequality constraints to equalities by introduction of non-negative slack variables. This method has been reported as computationally effective. Yet, if the step size is not chosen correctly, the sub-linear problem may have a solution that is infeasible in the original nonlinear domain. In additional, this method suffers from initial, premature termination, and optimality criteria and, in the most cases, is unable to solve nonlinear quadratic objective functions [9, 10].

In general, classical optimization techniques have several drawbacks related to complexity, approximation, non-convergent, linearization, differentiation, and convexity

2.2 Intelligent Optimization Algorithms

In the last two decades, the heuristic and intelligent algorithms to solve different discipline optimization problems were introduced. These algorithms were proven to be effective in identifying the problem global minima without much details about the addressed optimization problem. Among these algorithms is the genetic algorithm (GA) which simulates the natural selection of next generation genes, the strength Pareto evolutionary algorithm (SPEA) which is similar to GA, but used for multi-objective problem, the improved strength Pareto evolutionary algorithm (SPEA2) which is an improved version of SPEA , differential evolutionary algorithm (DEA) which is a population-based algorithm, particle swarm optimization (PSO) which simulates the organism behavior, such as fish schooling or bird flocking, artificial bee colony which simulates the bee colony behavior and ant colony which simulates the ant behavior in identifying the shortest path to the food location [11, 12, 13, 14].

In [15], an improved genetic algorithm the (IGA) was implemented to optimize the fuel transportation scheme. Decision support system based on GA for optimizing the operation planning of hydrothermal power systems was developed in [16]. The ant colony was developed to predicting software reliability in [17]. Among the ant colony security implementation is the improvement of segmentation for number plate recognition [18]. Adaptive artificial bee colony was implemented to solve traveling salesman problem [19]. In [20], DEA was developed to optimize the task scheduling in cloud computing. Also, DEA was developed to optimize the design of three-phase induction machine in term of reducing the manufacturing cost and the motor loss in [21]. In [22], particle swarm optimization was implemented to optimize the channel time allocation for gigabit multimedia wireless networks.

The intelligent algorithms were also applied in electrical power system optimization problem to address many objectives. The most objectives considered are the reactive power optimization, real power loss optimization, system voltage stability enhancement, voltage profile improvement, and generation cost reduction.

2.3 Single Objective Applications

The objectives addressed by the optimization algorithms in power systems were treated as single objective problems or multi-objective problems. In the single objective problem, there is one objective that guide the optimizations process while maintaining the constraints. Penalty can be assigned to those solutions who violate the constraints, this will give high priority for the solutions that satisfy the constraints to stay in the evolution process. Removing the solutions that does not meet the constraints from the evolution process is another method of maintaining feasible solutions. In some cases the constraints are integrated with the problem objective which guarantee that only feasible solutions stay in the evolution process [23].

In this paragraph, examples of intelligent algorithms applications in power system single objective and linearized- weighted- multi-objective problems will be highlighted. The application of repeated genetic algorithm for reactive power optimization using IEEE 30 bus system was addressed in [24]. A comparison between the particle swarm optimization (PSO) and modified POS performance in optimizing IEEE 3 and 6 bus systems reactive and active power was illustrated in [25]. The ant colony and DEA were merged in [26] for the optimization of distribution network reactive power. In [27], an accelerated partial swarm algorithm was implemented to optimize linearized multi-objective problems of real power loss reduction, capacitors cost optimization and system voltage stability enhancement. Introducing the weighted sum strategy in linearizing the objective with the system constraints was illustrated in [28]. In [29], multi-objective was converted into merged cost objective where the discrete particle swarm optimization (PSO) and genetic algorithm were implemented to identify the optimized objective value. This was implemented to optimize the cost associated with power loss, capital, maintenance and operation cost. Optimizing radial distribution system by reconfiguration and capacitor placement, using dedicated genetic algorithm was addressed in [30]. The objectives functions were combined in a single objective satisfying the system constraints in that study. In [31], the power loss and other objectives, generation cost minimization and voltage profile improvement, were assessed as a single objective. A radial system was incorporated for the verification of the anticipated algorithm in comparison with the other heuristic methods. The portfolio of assets optimization objective was treated as single objective in [32]. Differential evolutionary (DEA) algorithm was applied to identify the optimal solution for that study. In [33], the differential evolutionary and pattern research were implemented to solve multi-objectives of power loss, energy cost and capacitor cost by linearizing in single objective. Hybrid algorithm of differential evolution and evolutionary programming was implemented to overcome the lack of population diversity when applying the DEA for optimization of reactive power flow in [34]. In [35], the active and reactive power dispatching optimization when applying a genetic-fuzzy technique was presented. The dynamic self-adaptive differential evolutionary approach was implemented in [36] to optimize linearized multi-objectives power loss, voltage deviation and system voltage stability problem.

2.4 Multi-Objective Applications

The philosophies of multiobjective optimization are different from that in a single objective optimization. The main goal in a single objective optimization is to find the global optimal solution, resulting in the optimal value for the single objective function. Yet, in a multiobjective optimization problem, there are more than one objective function, each of which may have a different individual optimal solution. These objectives are represented in the following expression:

$$\{ J_1 ,\ J_2 ; \ldots\ldots\ldots\ldots , \ J_n \} \tag{1.1}$$

where J_1, J_2 and J_n are the different objective functions to be optimized in parallel and n is the number of objectives to be optimized.

If there is sufficient difference in the optimal solutions corresponding to different objectives, the objective functions are often known as conflicting to each other. Multiobjective optimization with such conflicting objective functions gives rise to a set of optimal solutions, instead of one optimal solution. The reason for the optimality of many solutions is that no one can be considered to be better than any other with respect to all objective functions. These optimal solutions have a special name of Pareto optimal solutions.

The multi-objective problem is usually addressed in two methods. The first method is to linearize the multi-objective via the weighted sum technique, where the objectives J_1, J_2 *and* J_n are combined into a single objective J. This can be expressed as follow:

$$J = w_1 * J_1 + w_2 * J_2 + \ldots\ldots\ldots\ldots + w_n * J_n \quad (1.2)$$

where w_1, w_2 and w_n are the weight coefficients of each objective.

The main disadvantages of this approach is that the weight coefficients play a critical role in having the overall objective dominated by the objective with higher weight coefficient. Also, it is difficult to identify the suitable weight coefficients [23]. The other approach is to handle the objectives simultaneously and extract the most compromise solution out of the front optimal Pareto-set. This technique treats all objectives in equal manner.

Application examples of intelligent algorithms in addressing power system multi-objective problems will be given in this paragraph. In [37] the authors included the application of the differential evolution algorithm (DEA) for resolving multi-objectives with economic and environmental objectives. A comprising of treating the active and reactive power optimization multi-objectives problem as linearized single objective or non-linearized objectives was presented in [38], when applying differential evolution algorithm. The diversity preservation technique was implemented in this analysis in order to produce well-distributed front Pareto-optimal set. In [39], performance benchmark was done between three multi-objectives algorithms. Namely, PSO, non-dominated sorting genetic algorithm II (NSGA-II) and SPEA2 in addressing multi-objective reactive power dispatch problem. In [40], an improved version of SPEA2 was implemented for optimizing economical and technical objectives.

Particle swarm optimization was implemented in [41] to address power loss and voltage deviation multi-objectives problem. Improved particle swarm optimization (IPSO) performance in optimizing the power loss, reduce cost, and enhancing the system voltage stability was benchmarked with others intelligent algorithms when applied to IEEE 30-bus system in [42].

2.5 Usage of Real Power System Parameters and Constraints

In most of the literature standard IEEE electrical system models are used for addressing optimization problems using intelligent algorithms. Real-life system models with real parameters are also used in some of the studies. GA was applied to control the bus voltages and reactive power loss for chiangmai distribution system considering transformer real tap values [43]. In [44], the evolutionary programming, particle swarm optimization, differential evolution, and hybrid differential evolution algorithms were implemented for solving multi-objective active-reactive power optimization problem. IEEE 30-bus and Taiwan Power Company 345kV simplified system were used to benchmark the performance of the selected four evolutionary computation algorithms in solving the problem. New England 39 bus system and Indian utility 62 bus system power loss optimization problem was addressed in [45]. Genetic algorithm and hybridized simulated annealing with pattern search was used in addressing the subject problem in that study. Particle swarm optimization was implemented in [41] to address power loss and voltage deviation multi-objectives problem. The Western System Coordination Council (WSCC) system was studied to determine the effectiveness of the presented approach. Mexican power system was used as the model for assessing the effectiveness of genetic algorithm in dealing with power loess and voltage stability multi-objective problems in [11]. In [46], transformer real taps positions were used in assessing the potential of using strength pareto evolutionary algorithm for solving power loss and voltage deviation multi-objective problems. The effectiveness of genetic algorithm in optimizing the reactive power in part of Italian distribution network is addressed in [28]. In [47], England and Wales transmission system was used to benchmark the performance of integer-coded genetic algorithm (IGA) and linear programming (LP) for reactive power optimization. The adaptive genetic algorithm was implemented to identify the optimal configuration of real life 70 bus and 136 bus distribution systems in [48].

None of the previous studies used real life hydrocarbon facility electrical model for addressing optimization problem using intelligent algorithms. The design of islanding scheme for real hydrocarbon facility with high load rejection was an example of applying conventional algorithm in real hydrocarbon facility [49]. The implementation of the power management system (PMS) in managing islanded real hydrocarbon facility power system against any disturbance by applying a predefined load shedding scheme and controlling the generator active and reactive powers is another example [50]. A third example is the application of PMS in enhancing the reliability of an upgraded electrical system due to the load demand increase as illustrated in [51].

2.6 Summary

In this chapter, a comprehensive survey of the literature with focus on papers published in 2009 forward was presented. The main drawbacks of the conventional methods such as, the nonlinear programming (NLP), the linear programming (LP) , quadratic programming (QP) and interior point when addressing optimization problem were highlighted. The most intelligent algorithms widely implemented for addressing power system and other discipline optimization problem have been discussed such as genetic algorithm (GA), particles swarm optimization (PSO), artificial bee colony (ABC), ant colony (AC), strength pareto evolution algorithm (SPEA), improved strength pareto evolution algorithm (SPEA2) and differential evolution algorithm (DEA). The concept of single objective and multi-objective problem formulation was illustrated. The application of the intelligent algorithm in addressing problems in their single and mult-objective formulation were given. A mix between well-tuned algorithms (GA and SPEA2) and relatively new and very promising DEA were selected to address the thesis objectives. The application of intellegnt algorithm for real hydrocarbon facility with real parmeters is unique.

CHAPTER 3: HYDROCARBON FACILITY POWER SYSTEM MODEL

3.1 The System Parameters Gathering Strategy

Developing credible and identical system parameters to the real life hydrocarbon facility electrical system mandates considering the following:

- The actual system parameters- lines and transformers impedances.
- The definite operation limitations.
- The weather temperature and its effect on the system efficiency.
- The generation operation mode- combined cycle.
- The utility regulations- connection power factor limitation.
- The large rotation equipment design limitations.

The system parameters were gathered by the author via different mechanisms as follows:

- Visiting the Plant.
- Contacting the utility company.
- Gathering the equipment manufacture datasheets and specifications.
- Reviewing the System Electrical Transient Analysis Program (ETAP) Model.
- Obtaining the Gas Turbine and Steam Turbine generators manufacturers' datasheets.
- Collecting the cable parameters - gathering the manufacturer datasheets.
- Studying the plant design documents.

3.2 The System Overall Description

The hydrocarbon facility electrical system was designed to be very reliable and secure system to support the production of around 900,000 barrels of oil per day and ship the oil associated natural gas to a nearby gas processing facility. The main components of the hydrocarbon facility electrical system are as follows:

- Two (2) incoming 115kV transmission lines from the Utility Company
- Two (2) Gas Turbine Generators (GTG).
- Two (2) Steam Turbine Generators (STG).
- One (1) 115kV Gas Insulated Switchgear (GIS).
- Two (2) 115kV cable feeders looping three (3) causeway upstream production 115kV/13.8kV step-down substations. These substations are located 25kM away from the main plant in artificial causeway islands.
- The causeway three substations have 13.8kV cable feeders feeding a total of twenty five (25) oil well sites in artificial islands in addition to six (6) oil offshore platforms.
- In the causeway islands, there are a total of 352 Electrical Submersible Pumps (ESP) each with 450KVA load.
- Downstream utility 115kV/13.8kV substation (Sub#2) feeding five (5) distribution substations - Sub#3, Sub#4, Sub#5, Sub#6, Sub#11 and the industrial service facilities (ISF) substation.
- Five (5) captive large synchronous motors, 25000 horse power (HP) are fed directly from the 115kV GIS via captive 115kV/13.8kV power transformers.
- Three (3) distribution 115kV/13.8kV substations (Sub#7, Sub#8 and Sub#9) are feeding motor control centers (MCC).
- Two (2) 13.8kV, 16000 HP synchronous motors switchgear is fed from Sub#9.

Simplified one line diagram of the hydrocarbon facility electrical system is shown in Figure 3.1. The hydrocarbon facility electrical system is a very complex system with many large equipment within small geographical location footprint. This fact challenges the idea of power loss optimization within this small plant in terms of geographical footprint. Yet, the huge loads mandate the flow of high current within this system which support exploring the potential of power loss optimization. Also, given that all the large power transformers within the plant are equipped with on load tap changers makes the idea of supporting the bus voltages and accordingly optimizing the power loss an easy idea to be implemented in real life. In addition, most of the generators and the synchronous motors can play a critical role in supporting the system voltage via reactive power control which again will result in real power loss optimization.

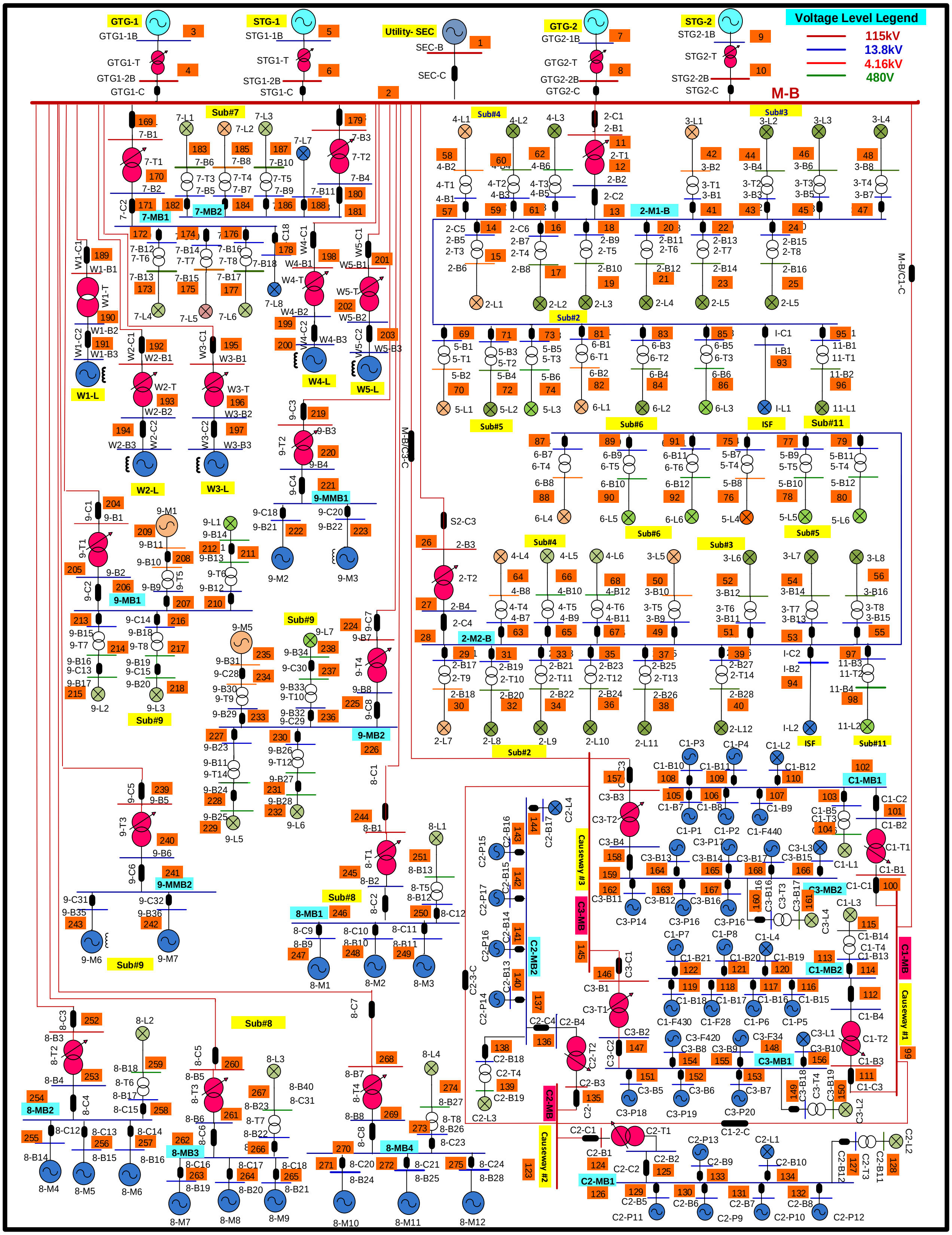

Fig. 3.1: Simplified one line diagram of the hydrocarbon facility electrical system

3.3 The Generation Mode and Utility Connection

The hydrocarbon facility power plant consists of two identical 154.45 MW gas turbine generators and two steam turbine generators with capacity of 86.87 MW and 43.29 MW integrated in combined cycle generation mode. There are two 115kV transmission lines coming to the facility from the national grid utility company.

3.3.1 The Utility Connection Parameters

The two 115kV transmission lines are sourcing from close 380kV/115kV utility company bulk power substation. These two lines are connected to the hydrocarbon facility substation 115kV GIS via two incomers GIS bays. Each line is capable to supply 360MVA of power (720MVA total). The main objective of these two lines is to be used as stand by power (backup power) for the facility. In addition, excess generated power within the facility can be injected into the national grid via these two lines.

3.3.2 The Gas Turbine Generator (GTG) Parameters

The two Gas Turbine Generators parameters are summarized in the Table 3.1. The GTGs operation philosophy is based on the following conditions:

1. The two generators will be working at their full capacity for satisfying the load needs.
2. The ambient temperature was chosen to be 95 ^{0}F (35 ^{0}C) as it is the average temperature between the winter average temperature (20 ^{0}C) and summer average temperature (50 ^{0}C).
3. The generators MVAR range is within a limit that enables delivering the rated MW for the selected capacity curve.

Table 3.1: Gas turbine generators parameters

GTG #	MW Rating	Voltage kV	MVAR Range		Gross Heat Rate Btu/kWh	Ambient Temperature in Deg.F	Load Percentage
			Min	Max			
GTG-1&2	154.45	16.5	-62.123	95.72	9,856	95	100%

Figure 3.2 shows the GTG capacity curves, the line with green highlight is the selected capacity curve borders- capacity curve at 95 ^{0}F. The GTG operating area shall be within the rectangalar area with red borders. This is to ensure that 154.45 MW can be generated as any delivered MVAR outside the +95.72 MVAR and -62.123 MVAR range will jeopardize the generated MW for the selected capacity curve.

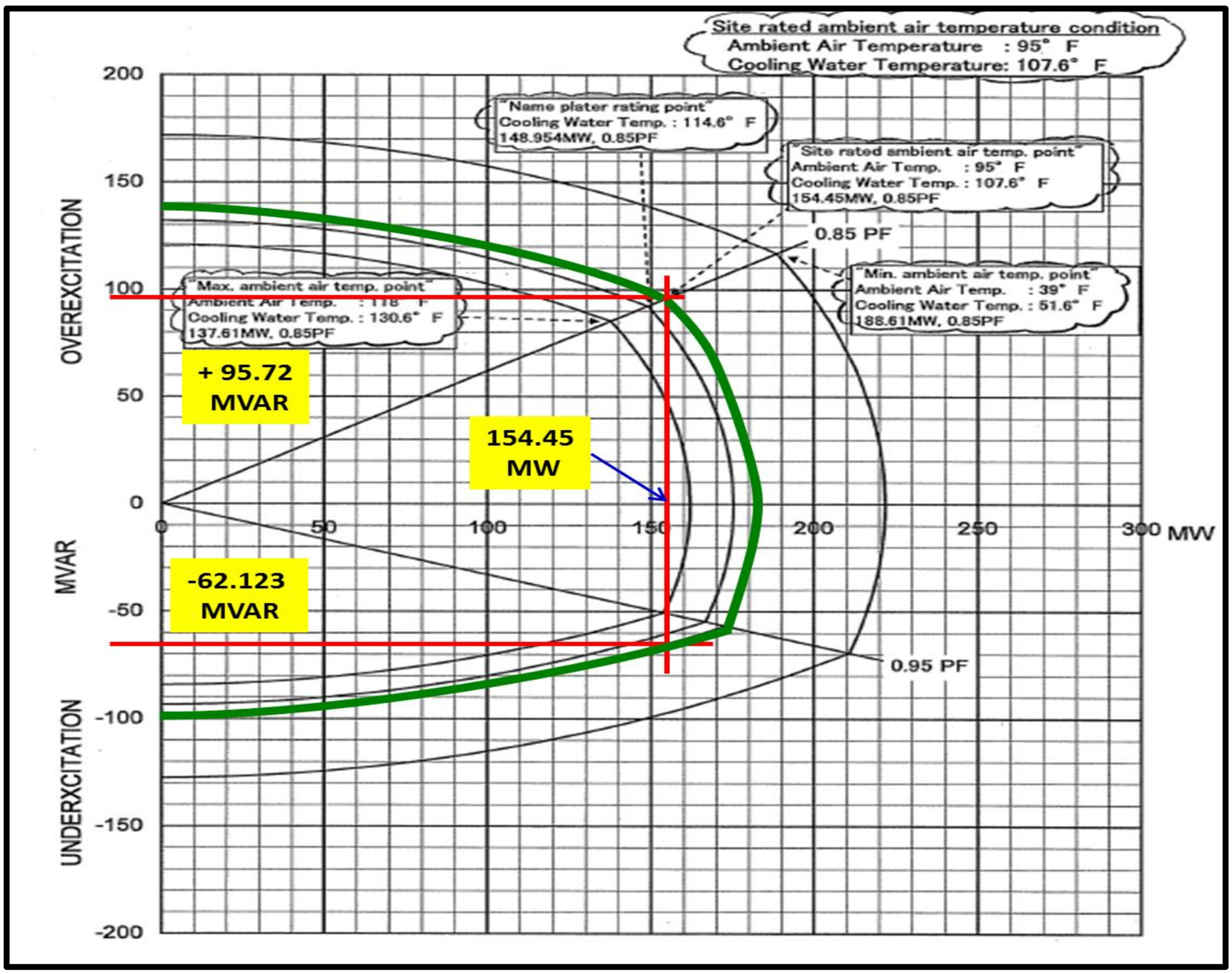

Fig. 3.2: GTG active and reactive power capacity curve

3.3.3 The Steam Turbine Generator (STG) Parameters

There are two Steam Turbine Generators at the hydrocarbon facility. These two generators are of combined cycle configuration with the two GTGs. The GTG exhaust heat is used to generate steam that drives the two STGs. Table 3.2 summarizes these two generators' parameters. The STGs operation philosophy is based on the following conditions:

1. The two generators will be working at their full capacity for satisfying the load need and pushing the combined cycle generation to its maximum efficiency.
2. The ambient temperature was chosen to be 95 ^{0}F (35 ^{0}C) as it is the average temperature between the winter average temperature (20 ^{0}C) and summer average temperature (50 ^{0}C).
3. The generators MVAR range is within a limit that enables delivering the rated MW for the selected capacity curve.

Table 3.2: Steam turbine generator parameters

STG #	MW Rating	Voltage kV	MVAR Range		Inlet Flow LBS/HR	Ambient Temperature in Deg.F	Load Percentage
			Min	Max			
STG-1	43.29	13.8	-22.4	21.919	346,000	95	100%
STG-2	86.87	13.8	-41.9	53.837	855,000	95	100%

Figure 3.3 shows the smaller STG capacity curve. The lines with red highlight determine the selected capacity curve borders. The area within the green highlight borders is the selected area of operation. This is to ensure that 43.29 MW can be delivered, as any delivered MVAR outside the +21.919 MVAR and -22.4 MVAR range will jeopardize the delivered MW for the selected operating area.

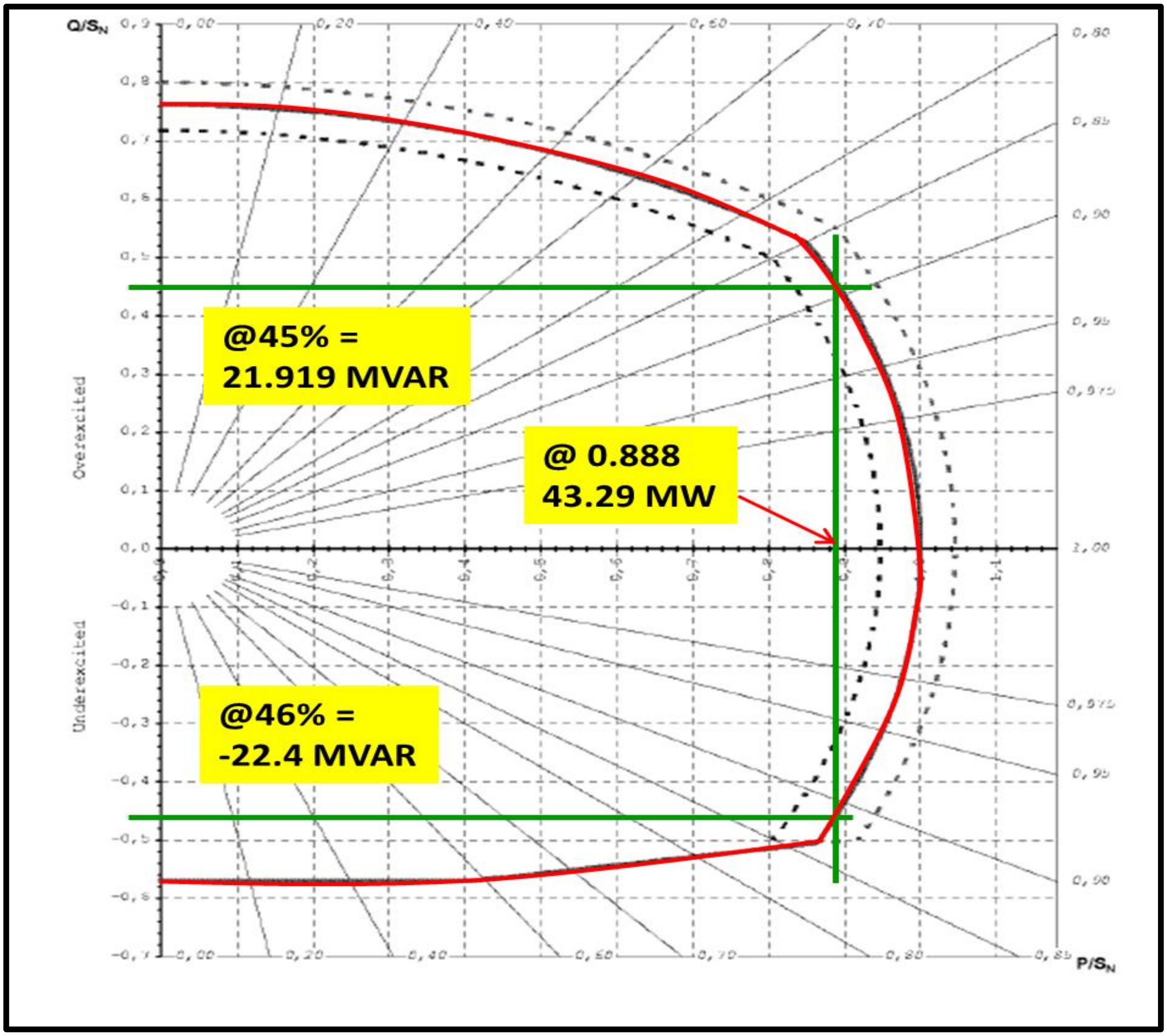

Fig. 3.3: *Small STG (STG-1) active and reactive power capacity curve*

Figure 3.4 shows the larger STG capacity curve. The red borders are the selected capacity curve. The area within the green highlight is selected to be the area of operation. In order to deliver 86.87 MW, the delivered MVAR shall be limited within +53.837 MVAR and -41.9 MVAR range.

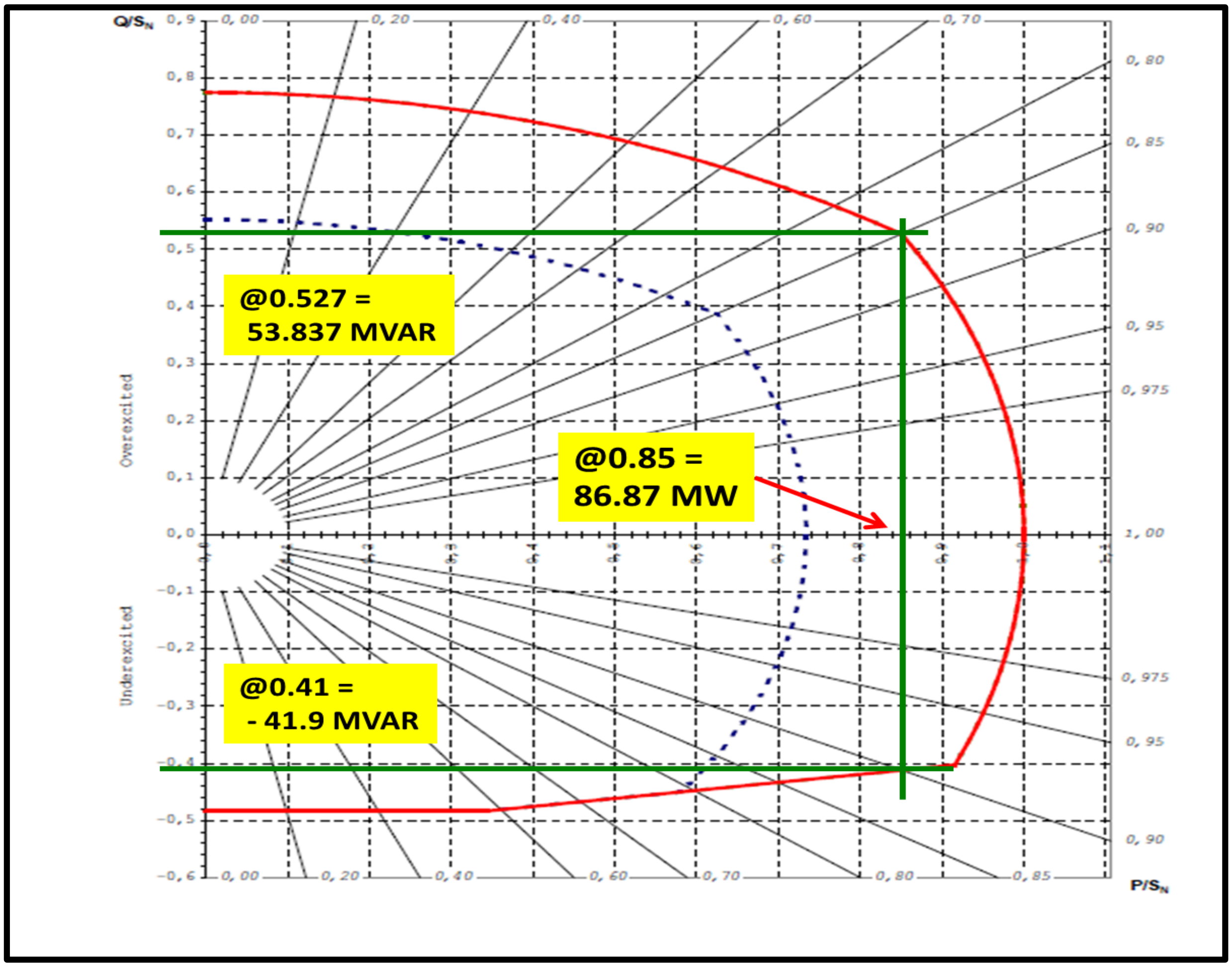

Fig. 3.4: *Large STG (STG-2) active and reactive power capacity curve*

3.4 The Large Synchronous Motor Model Parameters

The hydrocarbon facility has two types of synchronous motors. The five 25,000 HP motors at substation#7 which are fed via 115kV/13.8kV captive transformers and the two 16,000 HP motors at substation#9. The larger synchronous motors will be utilized as reactive power (MVAR) sources for supporting the system MVAR need. The 16,000HP motors will be utilized to support the reactive power in some of the considered sudy cases. This is due to the fact that the MVAR range of these motors' is very slim. The parameters of these motors giving their selected area of operation are summarized in Table 3.3 refrenced to their tag number posted in Figure 3.1.

Table 3.3: Synchronous motor parameters

Substation #	Tag#	Rated Voltage	Rated MW	Rated MVAR	
				Min	Max
Sub#7	W1-L/W2-L/W3-L/W4-L/W5-L	13.8kV	15.987	-6.96	16.42
Sub#9	9M-3/9M-6	13.8kV	10.254	-6.104	2.441

The capacity curve of the two 16,000 HP motors at substation#9 is shown in Figure 3.5. Each of these motors MW demand is 10.254 MW giving the range of their MVAR to be between -6.104 MVAR and 2.441 MVAR. As aforementioned, the range of the MVAR for these motors is very slim which limits their capability of supporting the system MVAR need.

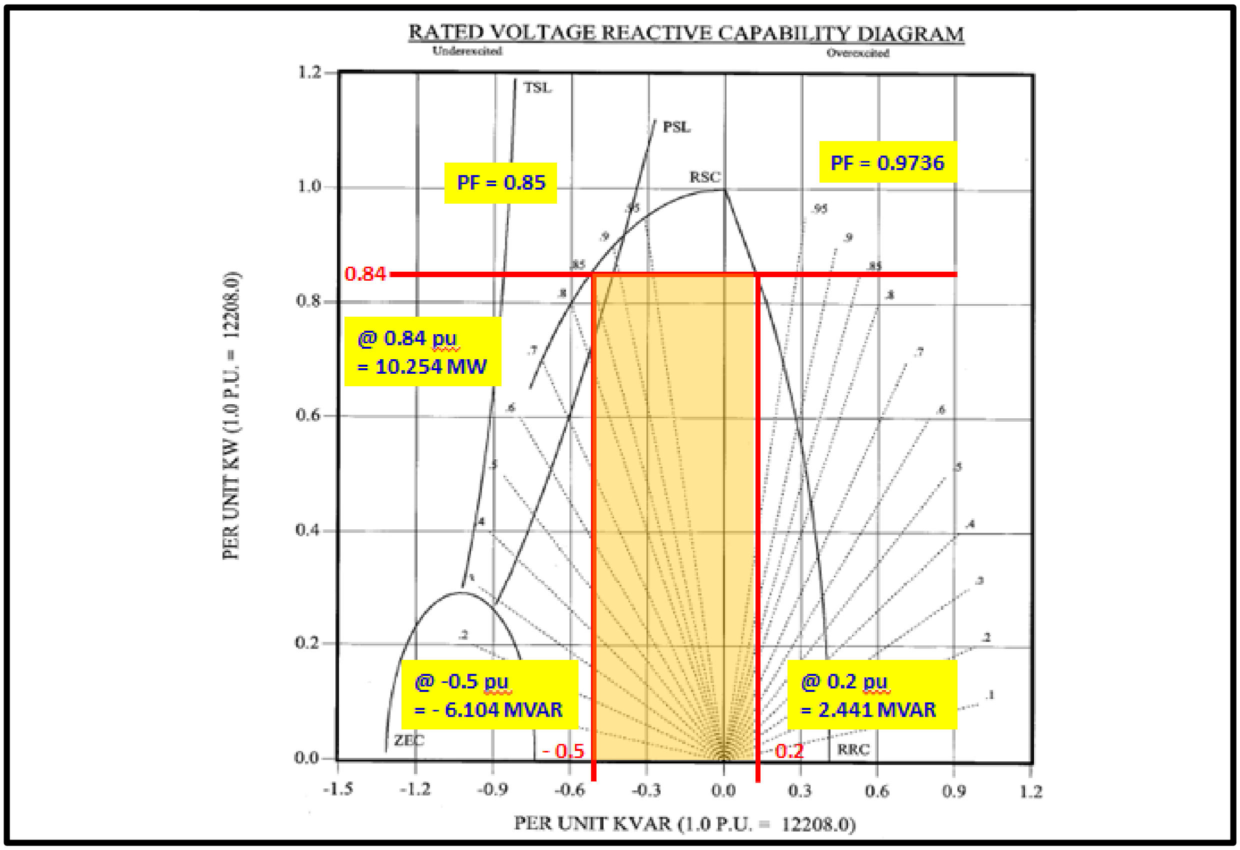

Fig. 3.5: 9M-3 & 9M-6 synchronous selected area of operation

Figure 3.6 shows the capability curve of the five 25,000 HP motors at substation#7. The MW demand of each of these motors is 15.987 MW within the MVAR range between -6.96 MVAR and 16.42 MVAR. The range of the MVAR for these motors is adequate for supporting the system MVAR need.

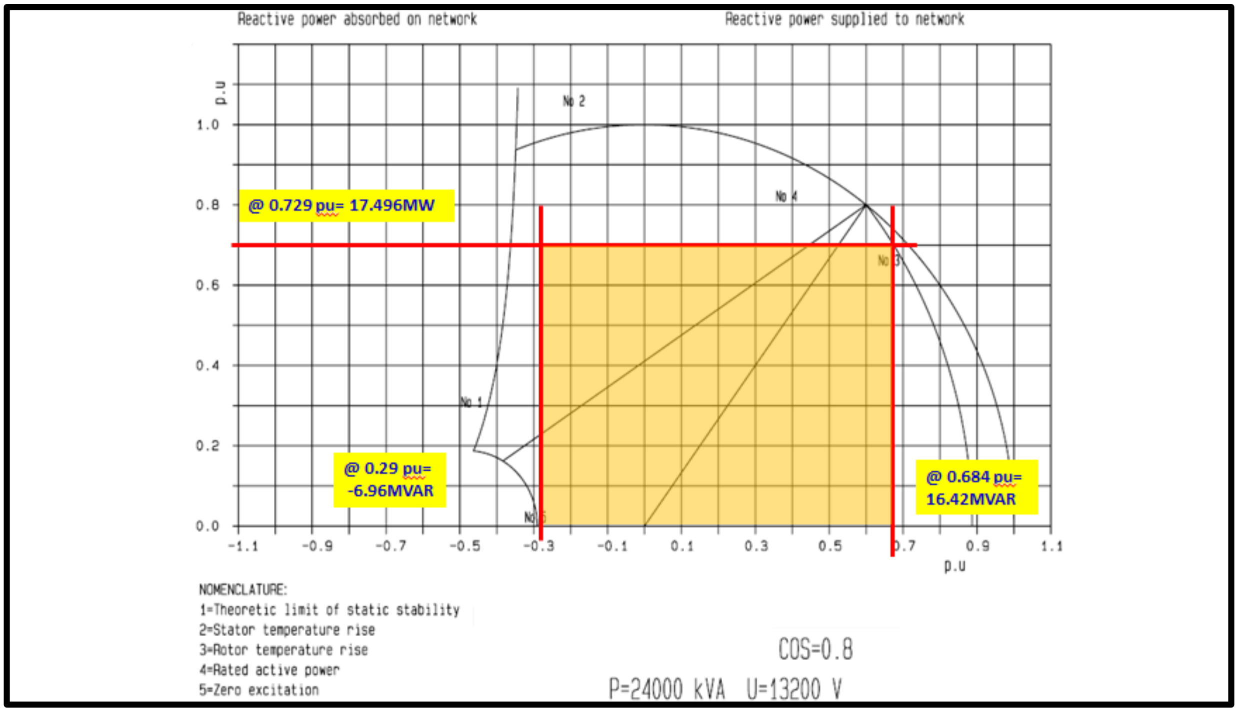

Fig. 3.6: *W1-L, W2-L, W3-L, W4-L & W5-L synchronous selected area of operation*

3.4.1 Above 5000 HP Medium Voltage (MV) Induction Motors Parameters

The hydrocarbon facility has couples of induction motors (> 5000 HP). These induction motors are fed from substation#8 and substation#9. Table 3.4 summarizes their parameters refrenced to their tag number posted in Figure 3.1.

Table 3.4: Induction motors (> 5000 HP) parameters

Substation #	Tag#	Rated Voltage	Rated MW	Rated MVAR
Sub#8	8M-1/8M-2/8M-3	13.8kV	4.801	2.202
Sub#8	8M-4/8M-5/8M-6/8M-7/8M-8/8M-9	13.8kV	4.801	1.872
Sub#8	8M-10/8M-11/8M-12	13.8kV	5.534	1.872
Sub#9	9M-2/9M-7	13.8kV	6.123	3.116

3.5 The Upstream (Causeway) Substations Model Parameters

There are three upstream substations called the causeway substations fed via 115kV closed loop cables sourcing from the facility main 115kV Gas Insulated Switchgear (GIS), refer to Table 3.5 for the GIS parameters. The 115kV GIS is located at the main generation substation. It receives power from the utility 115kV two lines and the local generators. 115kV feeders are sourcing from the GIS and feeding all the hydrocarbon facility upstream (causeway) and downstream substations.

Table 3.5: The hydrocarbon facility main GIS parameters

Configuration	Rated Voltage	Rated Current	Rated Short Circuit
Ring Type with double buses single breaker and two disconnects	115kV	3150A	40kA

In each one of the upstream substations, there are two 115kV/13.8kV step-down power transformers feeding 13.8kV switchgears, refer to Table 3.6 for the step-down transformer parameters. The 13.8kV switchgear feeders are feeding a total of twenty five oil well sites in artificial islands in addition to six oil offshore platforms.

Table 3.6: Upstream substations power transformer parameters

Substation #	Transformer Tag #	MVA Rating	Voltage in kV		Z %	Voltage Variance %	# of Taps	Each Tap Step %
			Primary	Secondary				
Causeway#1	C1-T1	50/66	115	13.8	9	±10%	±16	0.625
Causeway#1	C1-T3	1	13.8	0.48	5.75	±5%	±2	2.5
Causeway#1	C1-T2	50/66	115	13.8	9	±10%	±16	0.625
Causeway#1	C1-T4	1	13.8	0.48	5.75	±5%	±2	2.5
Causeway#2	C2-T1	50/66	115	13.8	9	±10%	±16	0.625
Causeway#2	C2-T3	0.75	13.8	0.48	5.75	±5%	±2	2.5
Causeway#2	C2-T2	50/66	115	13.8	9	±10%	±16	0.625
Causeway#2	C2-T4	0.75	13.8	0.48	5.75	±5%	±2	2.5
Causeway#3	C3-T1	50/66	115	13.8	9	±10%	±16	0.625
Causeway#3	C3-T3	1	13.8	0.48	5.75	±5%	±2	2.5
Causeway#3	C3-T2	50/66	115	13.8	9	±10%	±16	0.625
Causeway#3	C3-T4	1	13.8	0.48	5.75	±5%	±2	2.5

3.6 The Process Substations Model Parameters

There are nine downstream process substations. The largest downstream process substation is substation#2- the utility substation. Substation#2 is feeding a total six further downstream substations namely, substation 3, 4, 5, 6, 11 and industrial service facilities (ISF) substation. Substaion#2 is fed via two 115kV feeders sourcing from the facility main 115kV GIS. There are two power 115kV/13.8kV step-down transformers at substation#2 feeding the substation main 13.8kV switchgears. All the aforementioned further downstream substations are fed via 13.8kV feeders sourcing from substation#2 main 13.8kV switchgears.

3.7 The Generators Step-Up Power Transformer Parameters

There are four generator step-up transformers at the hydrocarbon facility, two for the GTGs and two for the STGs. A summary of these four transformers parameters is given in Table 3.7.

Table 3.7: Generator step-up power transformer parameters

Generator #	Transformer Tag #	MVA Rating	Voltage in kV		Z %	Voltage Variance %	# of Taps	Each Tap Step %
			Primary	Secondary				
GTG-1	GTG1-T	150/200	16.5	115	14	±10%	±8	1.25
GTG-2	GTG2-T	150/200	16.5	115	14	±10%	±8	1.25
STG-1	STG1-T	50/66	13.8	115	13	±10%	±8	1.25
STG-2	STG2-T	83/110	13.8	115	13	±10%	±8	1.25

3.8 The Main Step-Down Power and Distribution Transformer Parameters

There are many step down power and distribution transformers with different ratings at the facility downstream substations. Appendix I summarizes the parameters of these transformers and their locations.

3.9 Distribution Medium Voltage (MV) and Low Voltage (LV) Switchgear (SWG) and Motor Control Center (MCC) Parameters

The MV switchgears and motor control centers parameters and locations are summarized in Table 3.8.

Table 3.8: MV switchgear and motor control center parameters

Substation #	SWG Voltage	SWG Rating	SWG SC Rating	MCC Voltage	MCC Rating	MCC SC Rating
Sub#2	13.8kV	3000A	50kA	N/A	N/A	N/A
Sub#2	4.16kV	2000A	31.5kA	4.16kV	2000A	40kA
Sub#2	480	3200A	85kA	480V	1200A	65kA
Sub#3	4.16kV	2000A	29kA	4.16kV	1200A	41kA
Sub#3	480	3200A	85kA	480V	1600A	85kA
Sub#4	4.16kV	2000A	29kA	4.16kV	2000A	41kA
Sub#4	480	3200A	85kA	480V	800A	85kA
Sub#5	4.16kV	2000A	29kA	4.16kV	2000A	41kA
Sub#5	480	3200A	85kA	480V	800A	85kA
Sub#6	4.16kV	2000A	29kA	4.16kV	2000A	41kA
Sub#6	480	3200A	85kA	480V	800A	85kA
Sub# ISF	13.8kV	600A	25kA	N/A	N/A	N/A
Sub#11	480	3200A	65kA	N/A	N/A	N/A
Causeway# 1, 2 & 3	115kV	1200A	40kA	N/A	N/A	N/A
	13.8kV	3000A	37kA	N/A	N/A	N/A
	480	1200A	42kA	N/A	N/A	N/A
	13.8kV	1200A	21kA	N/A	N/A	N/A
Sub#7	13.8kV	3000A	37kA	N/A	N/A	N/A
Sub#7	4.16kV	2000A	29kA	4.16kV	1200A	41kA
Sub#7	480	3200A	85kA	480V	1600A	85kA
Sub#7/W1	13.8kV	1200A	37kA	N/A	N/A	N/A
Sub#7/W2	13.8kV	1200A	37kA	N/A	N/A	N/A
Sub#7/W3	13.8kV	1200A	37kA	N/A	N/A	N/A
Sub#7/W4	13.8kV	1200A	37kA	N/A	N/A	N/A
Sub#7/W5	13.8kV	1200A	37kA	N/A	N/A	N/A
Sub#8	13.8kV	3000A	37kA	N/A	N/A	N/A
Sub#8	480	3200A	65kA	480V	1200A	65kA
Sub#9	13.8kV	3000A	37kA	N/A	N/A	N/A
Sub#9	4.16kV	2000A	29kA	4.16kV	2000A	41kA
Sub#9	480	3200A	85kA	480V	800A	85kA

3.10 The Line (cable) Model Parameters

The nominal Π model is used to represent the cables, refer to Figure 3.7. Gving the fact that the HV lines in the hydrcabon facility are considered medium length lines, this model was selected.

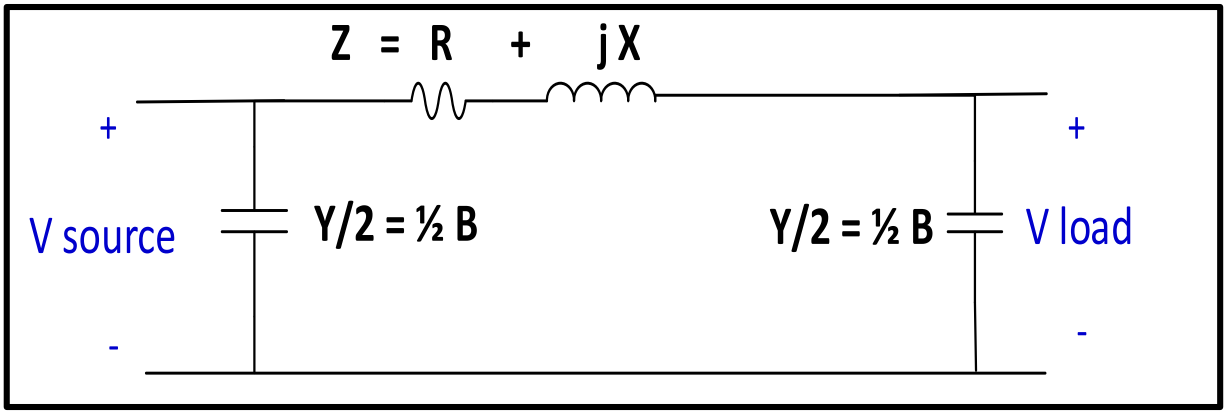

Fig. 3.7: Nominal Π model of the cable lines

In the transformer equipped with tap changing when the turn ratio between the transformer's two windings is one (1) - at the neutral tap - the transformer is represented by a series admittance Y_t. As the tap moves away from the neutral tap, the admittance must be modified to include the effect of the turn ratio (a) between the transformer's two windings. Therefore, the transformer with tap changer equivalent circuit is shown in Figure 3.8 [52,53, 54].

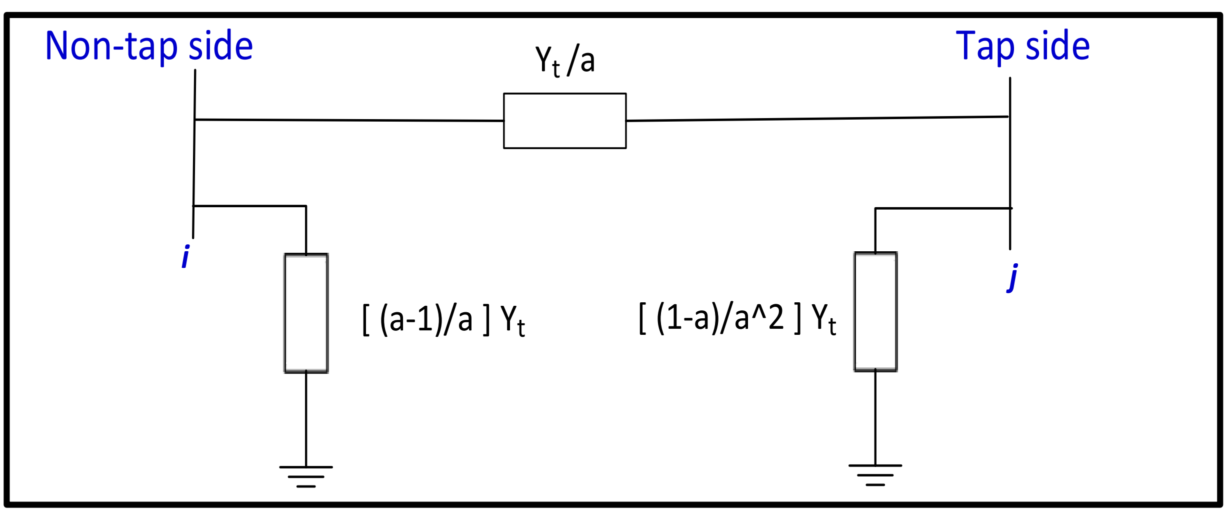

Fig. 3.8: Equivalent circuit for a tap changing transformer

3.11 The Load Model Profile

The MW load profile for each substation including the captive synchronous motors loads and the substations downstream of substation#2 is shown in Figure 3.8. Appendix II has a table of the load details per each bus excluding the synchronous motors.

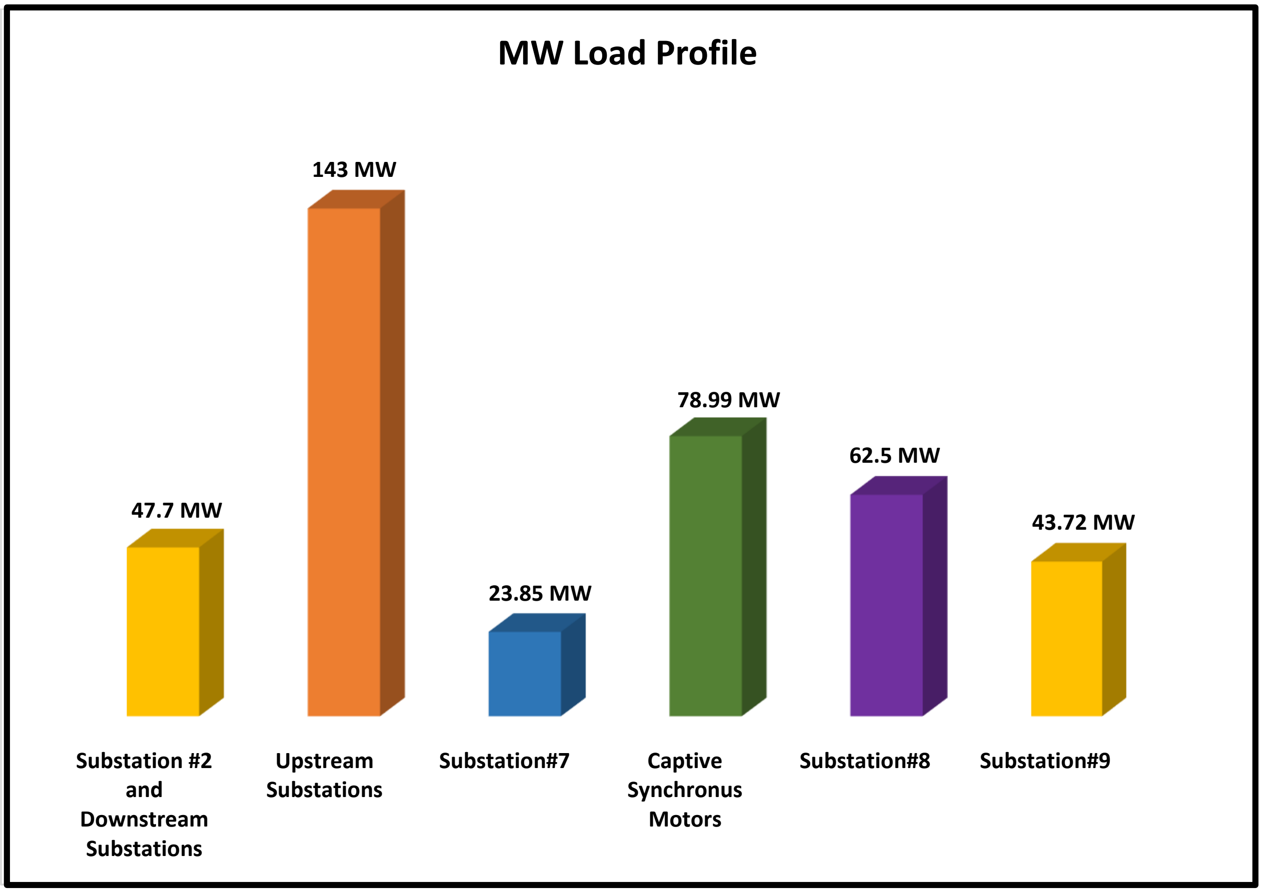

Fig. 3.8: *The MW load profile for each substation*

As shown in Figure 3.8, the dominant loads are the upstream substations – causeway substation#1, substation#2 and substation#3 – loads. These loads are mainly electrical submersiable pumps (ESP) at the oil wells. The five 25000 HP captive synchronous motors are the second dominate loads in the facility's electrical system. These motors are used to inject water into the oil field. The remaining loads are distributed between the utility loads – substation #2 and its downstream substations - and the process loads- substations# 7, 8 and 9.

3.12 Summary

In this chapter, the hydrocarbon facility model with real parameters has been developed. Firstly, the simplified one line diagram model has been developed. The utility connection parameters have been discussed. The gas and steam turbine generation units real capacity curves for different ambient temperatures have been collected and discussed. The parameters of synchronous motors as well as the high rating induction motors have been collected and listed. The upstream substations parameters and the generators step-up power transformers parameters have been highlighted. In addition, the main power transformers and distribution transformers parameters were enumerated. The MV and LV switchgears, motor control centers, line model parameters and the facility load profile were discussed.

CHAPTER 4: PROBLEM FORMULATION

4.1 System Equality Constraints

These constraints represent the balance between the active and reactive power in the system-power load flow equations. This mandates that the balance between the active power generated P_{Gi} , the active power demand P_D and the active power loss in tranmssion lines P_{loss} shall be equal to zero. This balance is applicable also to the reactive power Q_{Gi}, Q_D, and Q_{loss}. These balances are presented as follows:

$$P_{Gi} - P_D - P_{loss} = 0 \quad (4.1)$$

$$Q_{Gi} - Q_D - Q_{loss} = 0 \quad (4.2)$$

The active and reactive power losses for the lines between bus *i* and bus *j* can be detailed as follows:

$$P_{li} = V_i \sum_{j=1}^{NB} V_j \, [\, G_{ij} cos(\delta_i - \delta_j) + B_{ij} \sin(\, \delta_i - \delta_j)] = 0 \quad (4.3)$$

$$Q_{li} = V_i \sum_{j=1}^{NB} V_j \, [\, G_{ij} sin(\delta_i - \delta_j) - B_{ij} \cos(\, \delta_i - \delta_j)] = 0 \quad (4.4)$$

where i = 1, 2, NB; NB represents the bus number.

G_{ij} is the conductance and B_{ij} is the susceptance between bus *i* and bus *j*, respectively.

V_i and V_j are the voltage magnitudes at bus *i* and bus *j*, respectively.

δ_i and δ_j are the voltage angles at bus *i* and bus *j*, respectively.

4.2 System Inequality Constraints

These constraints represent the system operating constraints such as generator voltage (V_G), the synchronous motor voltage (V_{Synch}), the generators reactive power output (Q_G), the synchronous motors reactive power (Q_{Synch}), transformer tap and load bus voltage (V_L) constraints.

4.2.1 Generator Constraints

The gas turbine generator (GTG) terminal bus voltage bandwidth is with ± 5% window

$V^{min}_{GTGi} \leq V_{GTGi} \leq V^{max}_{GTGi}$ i= 1,.....,NG; NG represents the generators number.

$0.95 \leq V_{GTGi} \leq 1.05$ for both of the GTGs terminal voltage (4.5)

Although the steam turbine generator (STG) terminal bus voltage bandwidth is within ± 10% window, the STG terminal bus voltage was limited to ± 5% window to reduce the system voltage stress.

$V^{min}_{STGi} \leq V_{STGi} \leq V^{max}_{STGi}$ i= 1,.....,NG

$0.95 \leq V_{STGi} \leq 1.05$ for the STGs terminal bus voltage (4.6)

The gas turbine generator (GTG) reactive power outputs constraints are as follow:

$Q^{min}_{GTGi} \leq Q_{GTGi} \leq Q^{max}_{GTGi}$ i= 1,.....,NG

-62.123 MVAR $\leq Q_{GTGi} \leq$ 95.72 MVAR for both of the GTGs (4.7)

The small steam turbine generator (STG-1) reactive power outputs constraints are as follow:

$Q^{min}_{STG1} \leq Q_{STG1} \leq Q^{max}_{STG1}$

-22.4 MVAR $\leq V_{STG1} \leq$ 21.919 MVAR (4.8)

The large steam turbine generator (STG-2) reactive power outputs constraints are as follow:

$Q^{min}_{STG2} \leq Q_{STG2} \leq Q^{max}_{STG2}$

-41.9 MVAR $\leq V_{STG2} \leq$ 53.837 MVAR (4.9)

4.2.2 Synchronous Motor Constraints

The synchronous motor reactive power and terminal bus voltage constraints are posted in Table 4.1. The bandwidth of substation#9 synchronous motor reactive power is very small compared with the captive synchronous motor reactive power. This makes it very vulnerable for being violated for any minor change in the bus voltages. As a result, these synchronous motors will be treated as induction motors in some of the studied cases.

Table 4.1: Synchronous motor reactive power and terminal bus constraints

Description	Lower Limit	Upper Limit
Captive synch. motors reactive power (QC_{Synch})	-6.96 MVAR	16.42 MVAR
Sub#9 synch. motors reactive power (Q_{Synch})	-6.104 MVAR	2.441 MVAR
Synch. motors terminal voltage (V_{Synch})	90%	105%

4.2.3 Transformer and Load Bus Constraints

The transformers taps and the load buses voltage constraints are summarized in Table 4.2. In the main power transformers with 115kV primary voltage, each tap position can change the transformer bus voltage by ±0.625%. This means the ± 16 tap positions can regulate the transformer secondary bus by ±10% of the nominal voltage. In the distribution power transformer with < 115kV primary voltage, each tap position can change the transformer bus voltage by ±2.5%. This enables regulating the transformer secondary voltage by ±5% of the nominal voltage. In the generator step-up power tranformers, each tap position can change the transformer bus voltage by ±1.25%. This means that the ± 8 tap positions can regulate the transformer secondary bus by ±10% of the nominal voltage. All the load buses voltage shall be regulated within specific voltage range. An aggressive bus voltage limitation is applied for the causeway downstream buses.

Table 4.2: Generator, power and distribution transformers and load bus voltage constraints

Description	Lower Limit	Upper Limit
Main power transformer taps	-16 (-10%)	+16 (+10%)
Distribution power transformer taps	-2 (-5%)	+2 (+5%)
Generators' step-up transformer taps	-8 (-10%)	+8 (+10%)
Causeway downstream buses voltage	95%	105%
All load buses voltage	90%	105%

4.3 Objective Functions

Three objective functions are addressed in this thesis, namely real power system loss (RPL), grid connection power factor (GCPF) and the system voltage stability Index (L-Index). The problem formulated as single objective problem considering RPL objective only or coupled with one of the aforementioned two other objectives as a multi-objective problem. The problem is articulated to find the best optimized solution in the case of the single objective problem. This changes to find the best compromise solution between two competing objectives in the case of the multi-objective problem. In both cases, the system equality and inequality constraints' real life values shall be satisfied.

4.3.1 Real Power System Loss Objective Function

The RPL objective is achieved by minimizing (P_{Loss}), the real power loss within the transmission and distribution lines of the system. This objective function (J_1) can be expressed in terms of the power flow lines' loss between two buses, V_i and V_j as follows:

$$J_1 = P_{Loss} = \sum_{k=1}^{nl} g_k \left[V_i^2 + V_j^2 - 2\ V_i V_j \cos(\delta_i - \delta_i) \right] \tag{4.10}$$

where nl represents the transmission and distribution lines

g_k is the conductance of the kth line, $V_i\angle\delta_i$ and $V_j\angle\delta_j$ are the voltages at end buses i and j of the kth line, respectively [23, 38, 39, 55].

4.3.2 Grid Connection Power Factor (GCPF) Objective Function

The system regulator in a regulated electricity market mandates a penalty on any plant that has GCPF exceeding a certain limit. This is needed to regulate the reactive power absorbed or injected into the grid. The limit is usually set equal to 0.85 lagging or leading. Additional reactive power injected or absorbed will be subject to a penalty tariff. In this study, the GCPF will be pushed to unity [1.0 power factor (PF) = cos (θ)] and accordingly PF angle, θ will be pushed to zero . Therefore, it is aimed to minimize the GCPF angle index which can be expressed as follow:

$$LGCPF = |\theta| \tag{4.11}$$

Hence, the GCPF objective (J_2) is expressed as follows:

$$J_2 = LGCPF = |\theta| = (PF) \tag{4.12}$$

4.3.3 System Voltage Stability Index Objective Function

There are many indices for measuring the voltage stability of the system such as the jacobian eign-value index, deviation of bus voltages index and the L-index. The L-index which measures the strength of the link between the load buses and the generation buses. The L-index is adopted as a voltage stability index due to its credibility in reflecting the system voltage stability status. The system voltage stability index (L-Index) indicator varies in the range between 0 - the no load case- and 1 which corresponds to voltage collapse. The closer the L-Index value to the zero indicates that the system state is away from the unstability region. Therefore, the L-index is well known of being used to assess the voltage stability of the electrical system. It uses the bus voltage and network information provided by power flow program. The L-Index indicator can be calculated as given in [37,56,57,58].

Consider the simplest line mode as shown in Figure 4.1

where

G: Generator

V_1 and V_2: nodes voltages

I_1 and I_2: nodes currents

S_1 and S_2: complex powers

$$Y_{11} * V_1 + Y_{12} * V_2 = I_1 = \frac{S_1^*}{V_1^*} \tag{4.13}$$

The elements Y11, Y12, Y21, and Y22 are from the admittance matrix [Y] whereas S_1 is the complex power

$$S_1 = V_1 * I_1^* \tag{4.14}$$

Equation (4.13) can be brought into the form

$$V_1^2 + V_0 * V_1^* = \frac{S_1^*}{Y_{11}} = a1 + j\, b1 \qquad (4.15)$$

where $$V_0 = \frac{Y_{12}}{Y_{11}} * V_2 = -\frac{Y_L}{Y_L + Y_Q} * V_2 \qquad (4.16)$$

Y_L series admittance of $\boldsymbol{\Pi}$-equivalent

Y_Q Shunt admittance of $\boldsymbol{\Pi}$-equivalent

When this complex equation in V_1 is solved, then

$$V_1 = \sqrt{\frac{V_0^2}{2} + a_1 \pm \sqrt{\frac{V_0^4}{4} + a_1 V_0^2 - b_1^2}} \qquad (4.17)$$

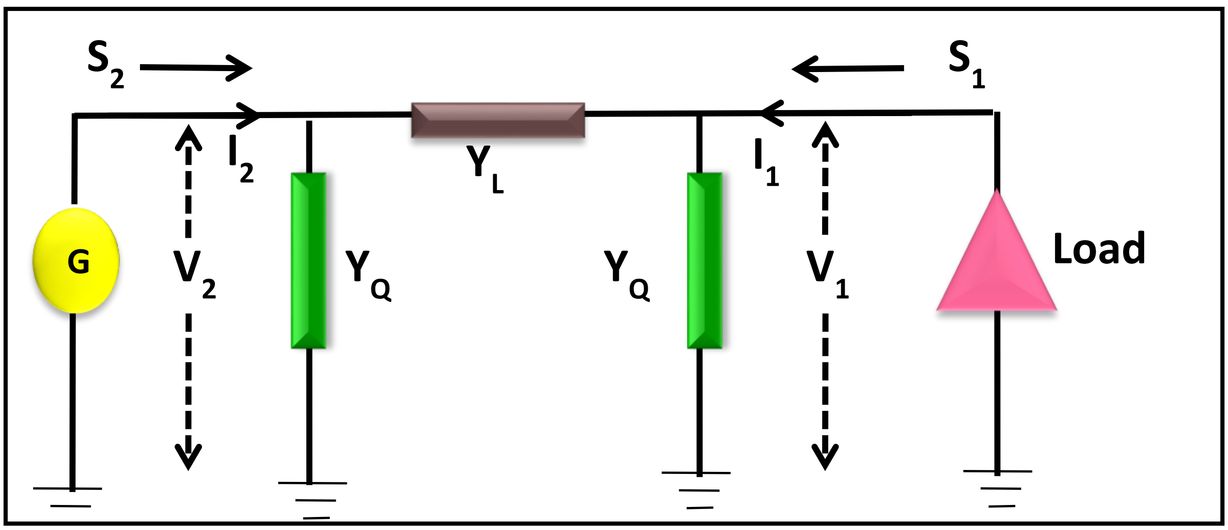

Fig. 4.1: *Line model with generator and load bus*

As in Figure 4.2, when varying V_1 in the permissible region which is defined by D= {V_1| $0 \le V_1 \le \infty$} a set of circles is produced whose merger area forms the feasible state space in S_1. The borderline envelope curve of this area is the desired stability limit of the two-node system in hand. Outside this borderline, no physical solutions are possible. The (+) relates to the feasible solution and the (-) relates to the un-feasible solution.

At the borderline, the two solutions of (4.17) must coincide, hence it must be true that

$$\pm\sqrt{\frac{V_0^4}{4} + a_1\, V_0^2 - \, b_1^2} \; = 0 \tag{4.18}$$

which can be expressed as

$$\left|1 + \frac{V_0}{V_1}\right| = \left|\frac{S_1}{Y_{11}^* \, V_1^2}\right| = 1 \tag{4.19}$$

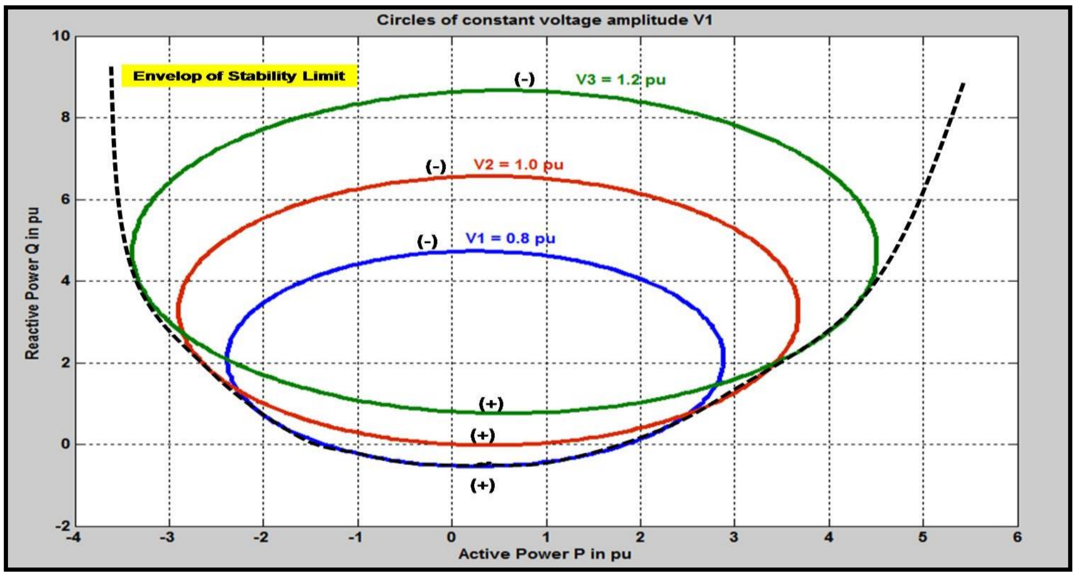

Fig. 4.2: *Circles of constant voltage amplitude V_1*

This relationship can be used to define a voltage stability indicator (L-Index). Its range is

R $=\{L - Index \mid 0 \leq L - Index \;\leq 1\}$ for feasible solution V_1

$$L\text{-}Index = \left| 1 - \frac{\frac{Y_L}{Y_L + Y_Q} V_2}{V_1} \right| \tag{4.20}$$

The above L-Index can be generalized for n nodes. A power system with n nodes can be represented in matrix format as follow:

$$\begin{bmatrix} I_L \\ I_G \end{bmatrix} = \begin{bmatrix} Y_1 & Y_2 \\ Y_3 & Y_4 \end{bmatrix} \begin{bmatrix} V_L \\ V_G \end{bmatrix} \tag{4.21}$$

$$\begin{bmatrix} V_L \\ I_G \end{bmatrix} = \begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix} \begin{bmatrix} I_L \\ V_G \end{bmatrix} = \begin{bmatrix} Z_{LL} & F_{LG} \\ K_{GL} & Y_{GG} \end{bmatrix} \begin{bmatrix} I_L \\ V_G \end{bmatrix} \tag{4.22}$$

where

V_L, I_L *load bus voltages and currents*

V_G, I_G *Generator bus voltages and currents*

H_1, H_2, H_3, H_4 *Submatrices generated from Y_{bus} Partial Inversion*

Z_{LL}, F_{LG}, K_{GL}, Y_{GG} *Submatrices of H-matrix*

Hence a local indicator L_j can be worked out for each node j similar to the line model

$$L_j = \left| 1 - \frac{\sum_{i \in \alpha G} F_{ji} V_i}{V_j} \right| \tag{4.23}$$

For a stable situation, the condition $L_j \leq 1$ must not be violated for any of the nodes j. Hence a global indicator L describing the voltage stability of the whole system is given by

$$J_3 = L\text{-}Index = \text{MAX}_{j \in \alpha L} \left| 1 - \frac{\sum_{i \in \alpha G} F_{ji} V_i}{V_j} \right| \tag{4.24}$$

where α_L is the set of load buses and α_G is the set of generator buses.

4.4 Single Objective Problem Formulation

Combining the objective function and the system constraints, the problem can be mathematically formulated as a nonlinear constrained single objective optimization problem as follows:

Minimize J_1 - in the minimization single objective problem (4.25)

$$\text{Subject to :} \begin{cases} g\ (x,u) = 0 \\ h\ (x,u) \leq 0 \end{cases} \tag{4.26}$$

where:

x: denotes the dependent variables vector involving load bus voltages V_{Li}, the generator reactive power outputs Q_{Gi}, and synchronous motor reactive power Q_{Synchi}. Hence, x is expressed as

$$x^T = [V_{Li} \dots V_{LN}, Q_{Gi} \dots Q_{GN}, Q_{Synchi} \dots Q_{SynchN}] \quad (4.27)$$

u: represents the control variables' vector consisting of the generators' real power (P_{Gi}), the generators terminal bus voltages (V_{Gi}), the synchronous motors terminal bus voltages (V_{Synchi}), and the transformer taps' settings (T_i). They can be transformed into the following expression:

$$u^T=[P_{Gi}, V_{Gi} \dots V_{GN}, V_{Synchi} \dots V_{SynchN}, T_i \dots T_{TN}] \quad (4.28)$$

g: are the equality constraints.

h: are the inequality constraints [23].

4.5 Multi-Objective Problem Formulation

Each day, there are many situations in which there are numerous objectives to be met. These objectives compete against each other. The solution for one objective may not provide a solution for another objective. The nature of the objectives calls for the development of numerous optimal solutions. The reason for this is that no single optimal solution can be regarded as superior over any other concerning every part of the objective functions. These type of solutions are named as Pareto-optimal solutions.

A multiobjective optimization problem subject to constraints is termed as follows:

$$Minimize\ f_i\ (x) \quad i = 1, \dots \dots, N_{obj} \quad (4.29)$$

$$Subject\ to: \begin{cases} g_j(x) = 0 & j = 1, \dots \dots, M \\ h_k(x) \leq 0 & k = 1, \dots \dots, K \end{cases} \tag{4.30}$$

where f_i is the i^{th} objective function

x denotes the decision vector representing a solution

N_{obj} represents the objective numbers

M represents the equality constraints, K represents the inequality constraints.

In any optimization problem, which involves multi-tasking, any two solutions such as x^1 and x^2 can have only two possibilities in terms of their mathematical relationship to each other. Either one can dominate the other, or none of them dominates each other. In the problem of minimization, deprived of the loss of generality, and under these conditions, a solution x^1 dominates x^2, if

$$1. \forall i \in \{1, 2, \dots, N_{obj}\} : f_i(x^1) \leq f_i(x^2) \tag{4.31}$$

$$2. \exists j \in \{1, 2, \dots, N_{obj}\} : f_j(x^1) < f_j(x^2) \tag{4.32}$$

Violation of any above-mentioned conditions, solution x^1 does not dominate the solution x^2. If x^1 dominates solution x^2 then, x^1, represents the non-dominated solution. Thus, Pareto-optimal is the non-dominated solution over all other solutions and constructs the set of Pareto-optimal or front Pareto-optimal [23].

Considering all the above objectives and constraints, the study problem may be precisely devised as a non-linear constrained optimization problem described as per the following equations:

Optimize (minimize) J_1 and J_2 (4.33)

$$\text{Subject to}: \begin{cases} g\ (x,u) = 0 \\ \\ h\ (x,u) \leq 0 \end{cases} \tag{4.34}$$

4.6 Economic Analysis Formulation

The Economic formulation for conducting the economic analysis of the optimization reults for the studied hydrocarbon facility electrical system will use actual generator parameters and defined energy cost at the kingdom of Saudi Arabia. In this section, all the economic equations will be developed.

4.6.1 The GTG's BTU to kWh Ratio

In reference to Figure 4.3 - the gas turbine generator data sheet- the British Thermal Unit (BTU) to the kilowatts hours (kWh) ratio at 95 ^{0}F is defined by the following equation

$$9856 \text{ BTU} = 1 \text{ kWh} \tag{4.35}$$

$$9856 \times 1000 \text{ BTU} = 1 \text{ MWh} \tag{4.36}$$

$$9.856 \text{ MMBTU} = 1 \text{ MWh} \tag{4.37}$$

CONDITION		1	2	3	4	5	
Ambient Temperature	deg.F	39	59	77	95	118	
Relative Humidity	%	30	30	30	30	30	
Barometric Pressure	psi	14.67	14.67	14.67	14.67	14.67	
Exhaust Loss (total)	inH2O	12.5	11.6	10.8	10.0	8.9	
PERFORMANCE							
Gross Output	kW	188,610	176,120	165,170	154,450	137,610	
Gross Heat Rate	Btu/kWh	9,361	9,514	9,671	9,856	10,208	
Fuel Flow	lb/h	96,790	91,850	87,560	83,440	77,000	
Injection Flow	lb/h	0	0	0	0	0	
Exhaust Flow	1000lb/h	3,812	3,665	3,525	3,382	3,194	
Exhaust Temperature	deg.F	1131	1143	1155	1169	1174	

Fig. 4.3: *GTG data sheet*

4.6.2 The MMBTU to MMSCF Ratio, Cost and Subsidize Price

The ratio between the million standards cubical feet (MMSCF) of gas and million BTU (MMBTU) is defined in the following equation

$$1 \text{ MMSCF} = 1050 \text{ MMBTU} \tag{4.38}$$

The following equation defines the cost per I MMBTU

$$1 \text{ MMBTU} = \$\ 3.5 \tag{4.39}$$

It is worth mentioning that the subsidized price of the MMBTU in Saudi Arabia is defined in equation (4.40).

$$\text{I MMBTU} = 0.75 \text{ cents (USD)} \tag{4.40}$$

4.6.3 The MMBTU to Oil Barrel Ratio, Oil Barrel to MMSCF Ratio and GTG Efficiency

The number of MMBTU per barrel of oil is defined by the following equation

1 Oil Barrel = 5.86 MMBTU (4.41)

In light of equations 4.38 and 4.41 above, the ratio between one barrel and the MMSCF of gas is given by the following equation.

1 Oil Barrel= 0.005581 MMSCF in term of MMBTU Equivalent (4.42)

The Gas Turbine Generator Efficiency interim of the total energy of the thermodynamic system (Enthalpy) is calculated as per the following equation.

(3142/9856)*100 = 31.8791 % (4.43)

where

3412 is for the enthalpy

9856 is the Gas Turbine Generator BTU to kWh ratio at 95 ^{0}F

4.6.4 Generation Cost

The hourly, monthly and annually generation cost is calculated as per the following steps:

1. Reading the MW out of the gas turbine generators (GTG-1 and GTG-2) buses. There is no generation cost associated with steam turbine generators (STG-1 and STG-2) as they are integrated with the GTGs in a combined cycle configuration.
2. Convert the generation MWh to equivalent MMBTU by multiplying by 9.856, refer to equation (4.37).
3. Calculating the Generation cost using the US Dollar ($) value per MMBTU, refer to equation (4.39).
4. The hourly generation cost calculated in the previous step is multiplied by 24 hours and 30 days to get the monthly cost.

5. The annually cost is calculated by multiplying the monthly cost calculated above by 12 (number of months per year).

4.6.5 The Real Power Loss (RPL) Cost

The hourly, monthly and annually cost associated with the system real power loss (MW) is calculated as per the following steps:

1. Reading the power system real power loss (RPL) in MW.
2. Convert the RPL MWh to equivalent MMBTU by multiplying by 9.856, refer to equation (4.37).
3. Calculating the hourly RPL cost using the USDolloar ($) value per MMBTU, refer to equation (4.39).
4. The hourly RPL cost calculated in the previous step is multiplied by 24 hours and 30 days to get the monthly cost.
5. The annually cost is calculated by multiplying the monthly cost calculated above by 12 (number of months per year)

4.6.6 Generation and Power loss Natural Gas Consumption

The hourly, monthly and annually Natural Gas Consumption for the system power electrical generation is calculated as per the following steps

1. Reading the MW out of the Gas Turbine Generators (GTG-1 and GTG-2) buses.
2. Convert the generation MWh to equivalent MMBTU by multiplying by 9.856, refer to equation (4.37).
3. Convert the MMBTU to equivalent MMSCF by dividing by 1050, refer to equation (4.38).
4. The hourly consumed MMSCF calculated in the previous step is multiplied by 24 hours and 30 days to get the monthly cost.

5. The annually consumed MMSCF is calculated by multiplying the monthly cost calculated above by 12 (number of months per year)

The natural gas consumption due to the system RPL is calculated in similar steps to the above ones. The only difference is that the MMBTU associated with real power loss is used to calculate the consumed gas in place of the generation needed MMBTU.

4.6.7 Grid Exported or Imported Power Cost

The subsidized electrical prices per kWh and MWh for industrial application in Saudi Arabia are given by the following equations:

Cost of 1 kWh = 14 Halalhs (100 Halalhs = 1 Saudi Riyal) (4.44)

Cost of 1 MWh = 140 Saudi Riyal (SR) (4.45)

I US $ = 3.75 SR (4.46)

Cost of 1 MWh = 37.7 US $ (4.47)

Therefore, the hourly, monthly and annually revenue by injecting excess power to the gird or expense by importing power from the grid is calculated as per the following steps:

1. Reading the system MW injected to or imported from the grid by reading the MW flow at the utility bus - bus number 1.
2. Calculate the hourly revenue in the case of injected power to the grid or the cost in the case of imported power from the grid by multiplying the MWh by $ 37.7, refer to equation 4.47.
3. The monthly revenue or cost calculated in the previous step is multiplied by 24 hours and 30 days to get the monthly revenue.
4. The annual revenue or cost calculated in the previous step is multiplied by 12 (number of months per year)

The new system operator regulation mandates penalties for maintaining the grid connection power factor more than 0.85. In case, the power factor is less than 0.85 an amount of $13 USD for each extra MVAR injected to or imported from the grid will be imposed.

4.6.8 Avoided Oil Barrels Burning

Any MW injected into the grid from the hydrocarbon facility combined cycle power generation plant is offsetting MW to be generated from one of the utility oil burning power plant. In other words, the MW produced from the hydrocarbon facility power plant resulted in avoiding burning oil in one of the utility company oil burning power plants. The hourly, monthly and annually number of avoided oil barrels burning at other generation facility (utility company), by injecting the excess power form the hydrocarbon facility to the grid are calculated as per the following steps:

1. Reading the system MW injected into the grid by reading the MW flow at the utility bus - bus number 1.
2. Calculate the Hourly equivalent MMBTU for the MW injected into the grid by multiplying the MWh injected into the grid by 9.856, refer to equation (4.37).
3. Calculate the hourly avoided number of oil barrels Burning at other generation facility (utility company) by dividing the calculated equivalent MMBTU for the MWh injected to the grid by 5.86, refer to equation (4.41)
4. The monthly avoided number of oil barrels burning due to power injection into the grid calculated in the previous step is multiplied by 24 hours and 30 days to get the monthly revenue.
5. The annually avoided number of oil barrels burning due to power injection into the grid calculated in the previous step is multiplied by 12 (number of months per year)

4.6.9 Avoided Oil Barrels Burning Cost

In reference to section 4.6.8 and the market price of the oil barrel as per equation (4.48) below, the cost of the avoided oil barrels burning is calculated as follows.

$$1 \text{ oil barrel} = \$\ 60 \text{ USD} \tag{4.48}$$

The $60 USD price of the oil barrel was identified as the best conservative oil barrel price at the research time as the average price during the subject period was $100 USD. The hourly, monthly and annually market value of the avoided oil barrels burning are calculated by multiplying the number of avoided oil barrels by the price of each barrel.

4.7 Summary

In this chapter, the optimal power flow problem has been formulated as an optimization problem. Initially, a set of objectives has been developed in order to reduce the real power loss, operational cost and improve the system performance. Namely, the objectives considered in this study are real power loss, grid connection PF enhancement, and voltage stability index. The equality constraints represented by the system power flow equations have been developed. In addition, the inequality constraints on control and dependent variables have been set. These inequality constraints include the generators terminal voltages and, reactive power, synchronous motors terminal voltages and reactive power, power transformers taps and the load buses voltages. Addressing the objectives as single objective and multi-objective was developed. The real power loss was addressed as single objective in the problem single objective formulation. On the other hand, both the real power loss and the grid connection PF enhancement were addressed as multi-objectives in the problem multi-objectives formulation. The bases for the economic analysis of the thesis study cases was addressed. The GTG kWh to BTU ratio for the ambient temperature close to the normal operational ambient temperature of the hydrocarbon facility was the fundamental base of all the economic analysis carried out in this thesis.

CHAPTER 5: PROBLEM PROGRAMMING

5.1 Research Software

The MATLAB software is utilized in conducting all the thesis studied cases due to its robustness and programming language flexibility. MATLAB is a numerical computing software equipped with fourth-generation programming language. It allows matrix manipulations, plotting of functions and data and implementation of algorithms. The MATLAB software has many built in tools (programs) to run different kind of intelligent algorithms including the GA and others. The user needs only to identify the problem objectives, its constraints and the algorithm parameters in the MATLAB optimization tools and the program will produce the optimized values for the inputted problem.

None of the MATALB built-in optimization tools was used in this thesis studied cases, all the power flow, data reading, constraints verifications, optimization and result analysis programming codes are self-developed. This enables full control and understanding of how the optimization evolution process is working. It also support integrating the optimization programs with other programs such as the power flow program and the economical analysis programs of the results. In the case of a multiobjective problem the self-coding gives the user the full control in developing the fitness assignment logic as per the problem in hand. This is another advantage over using the MATLAB built-in tools.

Debugging the self-coded programs is easier task compared to the ready-made optimization programs, as the source of the unexpected results cannot be traced. Also, the self-coded programs give the user the ability to adjust the fitness value such as including more terms in calculating it. Having hybrid optimization algorithms can only be achieved via the self-coding programs. A full privilege over how the reporting format of the results shall appear is achieved via the self-coding program. This include the numerical reports and the different results comparison graph figures.

5.2 Reading and Assigning the Electrical Model Parameters

In Figure 5.1, the MATLAB programming main steps for reading the hydrocarbon facility electrical system parameters, assigning the generators and synchronous motor bus voltage values, assigning the transformers tap values, calculating the system admittance matrix, performing the Newton Raphson power flow, verifying the voltages and reactive power constraints, performing the power line flow and storing the feasible solutions are illustrated. This MATLAB programming is used to identify the feasible solutions – system parameters- that meet all the system constraints. These solutions to be the initial feasible solutions to go through the evolution process for searching the solution space for the best objective function values.

5.2.1 Reading the Electrical Model Bus Data

Matrix format is used to facilitate the reading of each bus data in each row of the matrix. This matrix is called **busdata** matrix. The information to be read per each column of the matrix is posted in Table 5.1. Refer to the system one line digram in Figure 3.1.

Table 5.1: Bus data matrix format

Column number	Data Assigned
1	The Bus Number as per the system one line diagram
2	The Bus Code which identifies the bus nature - slack, generator or Synchronous motor or capacitor bus and load **Code 1** : Slack Bus **Code 0** : Load Bus **Code 2**: Generator or Synchronous motor or Capacitor Bus
3	The Bus voltage magnitude in per unit
4	The Bus phase angle in degree
5	The Bus Real Load in MW
6	The Bus Reactive Load in MVAR
7	The Generator Real Power in MW
8	The Generator Reactive Power in MVAR
9	The Generator minimum generated reactive power in MVAR
10	The Generator maximum generated reactive power in MVAR
11	The Bus injected MVAR of Shunt Capacitors

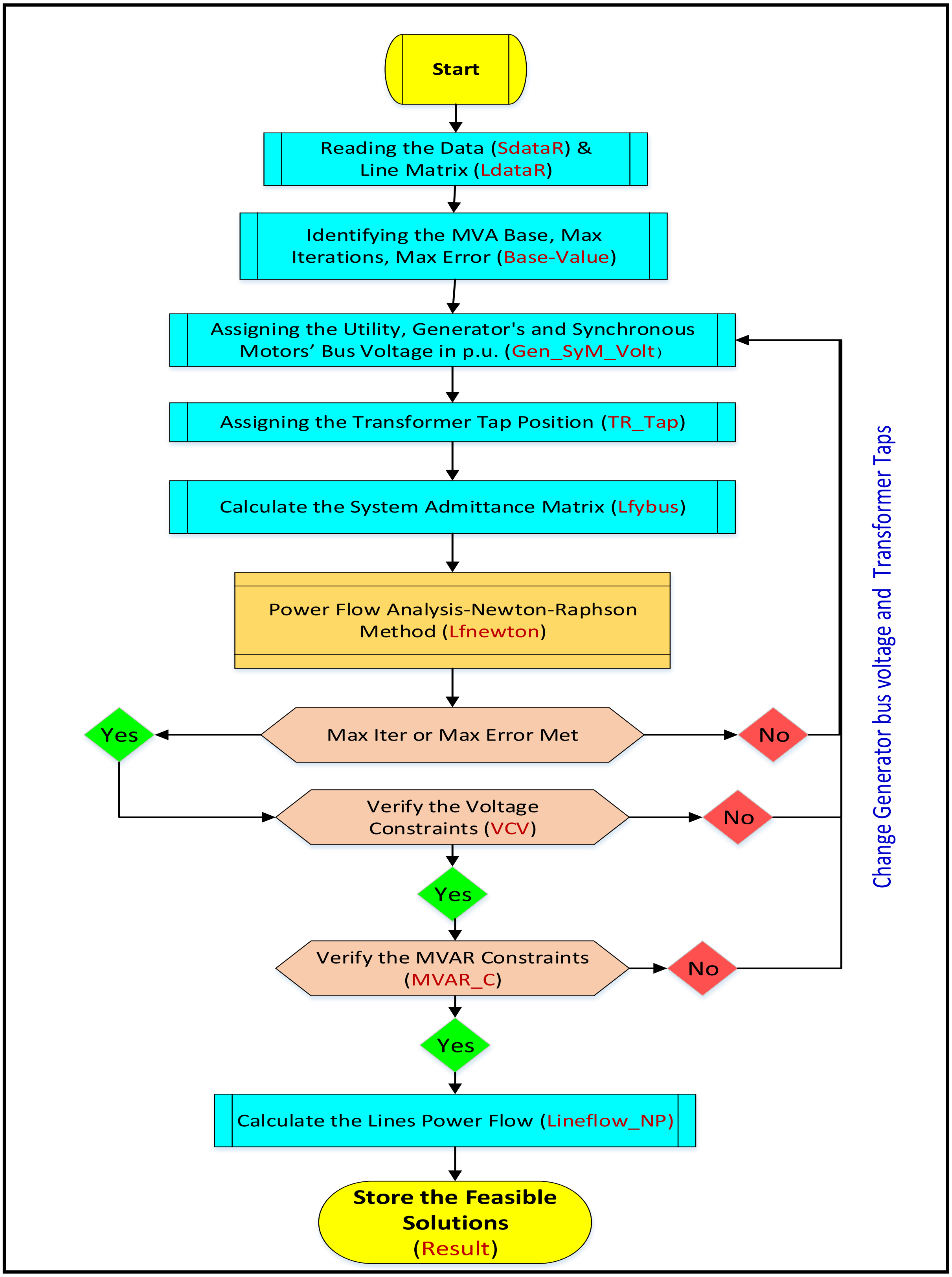

Fig. 5.1: *MATLAB programming loop for identifying the initial feasible solutions*

The system bus data was saved in excel file called System Data; within the excel file there are many sheets. Each sheet contains around twenty (20) bus data. Snap shot of one of the system data excel file sheet is shown in Figure 5.2.

Bus #	Bus Code	Bus Name	Voltage	Generation							Load		Shunt Capacitor
				Voltage Mag.	Voltage θ	MW Range			Mvar Range		MW	Mvar	
						Rated	Min	Max	Min	Max			Mvar
1	1	SEC-B	115	1	0	0	0	0	0	0	0	0	0
2	0	M-B	115	1	0	0	0	0	0	0	0	0	0
3	2	GTG1-1B	16.5	1	-24.3	154.45	0	154.45	-62.123	95.72	0	0	0
4	0	GTG1-2B	115	1	0	0	0	0	0	0	0	0	0
2	0	M-B	115	1	0	0	0	0	0	0	0	0	0
5	2	STG1-1B	13.8	1	-24.3	43.29	0	43.29	-22.4	21.919	0	0	0
6	0	STG1-2B	115	1	0	0	0	0	0	0	0	0	0
2	0	M-B	115	1	0	0	0	0	0	0	0	0	0
7	2	GTG2-1B	16.5	1	-24.3	154.45	0	154.45	-62.123	95.72	0	0	0
8	0	GTG2-2B	115	1	0	0	0	0	0	0	0	0	0
2	0	M-B	115	1	0	0	0	0	0	0	0	0	0
9	2	STG2-1B	13.8	1	-24.3	86.87	0	86.87	-41.9	53.837	0	0	0
10	0	STG2-2B	115	1	0	0	0	0	0	0	0	0	0
2	0	M-B	115	1	0	0	0	0	0	0	0	0	0
11	0	2-B1	115	1	0	0	0	0	0	0	0	0	0
12	0	2-B2	13.8	1	0	0	0	0	0	0	0	0	0
13	0	2-M1-B	13.8	1	0	0	0	0	0	0	0	0	0
14	0	2-B5	13.8	1	0	0	0	0	0	0	0	0	0
15	0	2-B6	4.16	1	0	0	0	0	0	0	2.464	1.33	0
13	0	2-M1-B	13.8	1	0	0	0	0	0	0	0	0	0
16	0	2-B7	13.8	1	0	0	0	0	0	0	0	0	0
17	0	2-B8	0.48	1	0	0	0	0	0	0	1.044	0.592	0
13	0	2-M1-B	13.8	1	0	0	0	0	0	0	0	0	0
18	0	2-B9	13.8	1	0	0	0	0	0	0	0	0	0
19	0	2-B10	0.48	1	0	0	0	0	0	0	0.875	0.484	0
13	0	2-M1-B	13.8	1	0	0	**0**	**0**	**0**	**0**	**0**	**0**	0
20	0	2-B11	13.8	1	0	0	0	0	0	0	0	0	0

Fig. 5.2: *Bus data sheet snap shot*

The system data excel file sheets has many posted data that are not required for the busdata matrix. In addition, the arrangement of the column is not aligned with the busdata matrix format highlighted above. This mandates writing MATLAB command codes (**SdataR.m**) to read the excel file, clear the unneeded information and arrange them in the requested format. The MATLAB command **xlsread** was used to read the line data from the excel file sheets. Other MATLAB codes were also implemented to eliminate unneeded information and arranging them in the requested format. The MATLAB codes of the SdataR program is posted in Appendix III.

5.2.2 Reading the Electrical Model Line Data

The hydrocarbon system line data is posted using matrix format, each row contains the parameters for the line between two buses. This matrix is called **linedata** matrix. The information to be read per each column of the **linedata** matrix is posted in Table 5.2.

Table 5.2: Line data matrix format

Column number	Data Assigned
1	The Source Bus (**nl**) number as per the system one line diagram
2	The Receiving Bus (**nr**) number as per the system one line diagram
3	The line between the Buses (nl & nr) Resistance in p.u.
4	The line between the Buses (nl & nr) Reactance in p.u.
5	One-half (1/2) of the total line Susceptance in p.u. for the line between Buses (nl & nr)
6	The transformer between the Buses (nl & nr) Taps Setting. If there is no Transformer, the number 1 must be entered which is the line code

The system line and transformer impedances data were saved in an excel file called Line Data, within the excel file there are many sheets. Each sheet contains around thirty (30) line data. Snap shot of one of the line data excel file sheet is showing in Figure 5.3. The system line and transformer parameters are posted in Appendix IV.

The system line and transformer impedances data excel file sheets have many posted data that are are not required for the **linedata** matrix and they are not arranged in the format needed. This dictates writing MATLAB command codes (**LdataR.m**) to read the excel file, clear the unneeded information and arrange them in the requested format. The MATLAB command **xlsread** was used to read the line date from each of the Line Data excel file sheets. The MATLAB codes of the LdataR program is posted in Appendix V.

Bus nl	Bus nr	R/KM	R p.u.	X/KM	X p.u.	Y/KM	½ B p.u.	Line Code	Transformer Tap Setting
1	2	0	0.002609	0	0.0660068	0	0	1	
3	4	0	0	0	0.09333	0	0	8	±10% - ±8 Steps (1.25)
4	2	0.0404	0.00003055	0.150857	0.000114	0.000067886	0.00404	1	
5	6	0	0	0	0.26	0	0	8	±10% - ±8 Steps (1.25)
6	2	0.0404	0.000128	0.150857	0.000479	0.000067886	0.001885	1	
7	8	0	0	0	0.09333	0	0	8	±10% - ±8 Steps (1.25)
8	2	0.0404	0.00002189	0.150857	0.00008175	0.000067886	0.002895	1	
9	10	0	0	0	0.1566	0	0	8	±10% - ±8 Steps (1.25)
10	2	0.0404	0.00005804	0.150857	0.0002167	0.000067886	0.0034116	1	
2	11	0.0404	0.0001054	0.150857	0.000393	0.000067886	0.001548	1	
11	12	0	0	0	0.15	0	0	16	±10% - ±16 Steps (0.625)
12	13	0.047	0.000148	0.123316	0.00038855	0.00016217	0.00001482	1	
13	14	0.124	0.0022789	0.14218	0.002613	0.00010937	0.00001458	1	
14	15	0	0	0	0.65	0	0	5	±5% - ±2 Steps (2.5)
13	16	0.124	0.0026045	0.14218	0.002986	0.00010937	0.00001666	1	
16	17	0	0	0	2.3	0	0	5	±5% - ±2 Steps (2.5)
13	18	0.124	0.0026045	0.14218	0.002986	0.00010937	0.00001666	1	
18	19	0	0	0	2.3	0	0	5	±5% - ±2 Steps (2.5)
13	20	0.124	0.0026045	0.14218	0.002986	0.00010937	0.00001666	1	
20	21	0	0	0	2.3	0	0	5	±5% - ±2 Steps (2.5)
13	22	0.124	0.0026045	0.14218	0.002986	0.00010937	0.00001666	1	
22	23	0	0	0	2.3	0	0	5	±5% - ±2 Steps (2.5)
13	24	0.124	0.0026045	0.14218	0.002986	0.00010937	0.00001666	1	
24	25	0	0	0	2.3	0	0	5	±5% - ±2 Steps (2.5)
2	26	0.0404	0.0000669	0.150857	0.0002498	6.78857E-05	0.000983	1	
26	27	0	0	0	0.15	0	0	16	±10% - ±16 Steps (0.625)
27	28	0.047	0.000148	0.123316	0.00038855	0.00016217	0.00001482	1	
28	29	0.124	0.0022789	0.14218	0.002613	0.00010937	0.00001458	1	

Fig. 5.3: *Lines data sheet snap shot*

5.2.3 Reading the Electrical Model Base Values

The power flow Newton Raphson method – section 5.3 - required the identification of the followings base values:

1. The MVA base (**mvabase**) to be used for calculating the per unit value of all real and reactive power to be prepared to go through the Newton – Raphson power flow iterations calculation. Also, the MVA base value is needed for converting the results per unit real and reactive power values back to their actual values.
2. The maximum number of iterations (**maxiter**) to be used as a stopping criterion if the Newton Raphson iterations process does not coverage.

3. The acceptable maximum mismatching (**accuracy**) between the scheduled and calculated real and reactive powers. The MATLAB program codes for reading the base values is posted in Appendix VI.

5.2.4 Assigning the Generator, Utility and Synchronous Motor Bus Voltages

The utility bus voltage (UBV) – bus number 1- was assigned to be less than 1.0 p.u. (0.97 p.u.) to minimize the reactive power imported from the national grid. The objective of this minimization will be explained in detail later in the research. The utility bus voltage should operate between +105% and -95% of the nominal bus voltage. The GTGs' terminal bus can tolerate ±5% voltage bandwidth. The generator exciter can adjust the GTGs' terminal voltage within this bandwidth in steps (±5 steps) each step can adjust the voltage by 1% (0.01 p.u.). The STGs' terminal bus can tolerate ±10% voltage bandwidth. Yet, to reduce the system voltage stress on the system, the STGs' terminal voltage was limited to ±5% voltage bandwidth. The generator exciter can adjust the STG terminal voltage within this bandwidth in steps similar to the GTGs' terminal bus. The synchronous motor terminal bus can tolerate ±5% voltage bandwidth. The motor exciter can adjust the synchronous motor terminal voltage bus within this bandwidth in steps (±5 steps) each step can adjust the voltage by 1% (0.01 p.u.). The (**Gen_SyM_Volt.m**) is the MATLAB program for assigning the utility, GTGs, STGs and synchronous motor terminal bus voltages. The MATLAB codes of the Gen_SyM_Volt program is posted in Appendix VII.

5.2.5 Assigning the Transformer Taps' Parameters

There are three (3) types of transformers in the hydrocarbon facility electrical system with regard to their tap changer (voltage regulation mechanism) design. The first type, the power transformer with 115kV primary voltage. These transformers include all the substations main transformers. They are equipped with automated voltage regulator (the transformer tap changes its position automatically to regulate the voltage to the nominal system voltage). The tap changers of these transformers have ± 16 tap positions, and each tap position can regulate the voltage by 0.625% of the nominal voltage. This means that the voltage regulator can adjust the voltage within a bandwidth of ± 10% of the nominal bus voltage. The adjustment of the bus voltage of this type of transformer has no operation impact, as the voltage can be regulated while the transformers are on-line.

The second type, the power transformer with < 115kV primary voltage. These transformers include all the substations distribution transformers downstream of the main transformers. These transformers equipped with off–line voltage regulator (the transformer tap position is changed manually after being de-energized to regulate the voltage to the nominal system voltage). The tap changers of these transformers have ± 2 tap positions and each tap position can regulate the voltage by 2.5% of the nominal voltage. This means that the voltage manual regulator can adjust the voltage within a bandwidth of ± 5% of the nominal bus voltage. The adjustment of the bus voltage of this type of transformer has an operation impact as the voltage can be regulated only when the transformers are off-line (de-energized).

The third type, the generators step-up power transformers. There are four (4) power transformers associated with each of the hydrocarbon facility four (4) generators. These transformers post the voltage from the generator terminal voltage (16.5kV in the case of the GTGs and 13.8kV in the case of the STGs) to 115kV. These transformers equipped with off – line voltage regulators (the transformer tap position is changed manually after being de-energized to regulate the voltage to the nominal system voltage). The tap changers of these transformers have ± 8 tap positions and each tap position can regulate the voltage by 1.25% of the nominal voltage. This means that the voltage manual regulator can adjust the voltage within a bandwidth of ± 10% of the nominal bus voltage. The adjustment of the bus voltage of this type of transformer has a large operation impact, as the voltage can be regulated only when the transformers are off-line (de-energized) which mandates shutting down the generator.

The MATLAB program (**TR_Tap.m)** is implemented for assigning the transformer taps positions depend on its type. Appendix VIII posts the MATLAB codes of the TR_Tap program.

5.2.6 Calculating the System Admittance Matrix

The (**Lfybus.m**) MATLAB program read the system impedances matrix linedata and then calculates the system admittances. The admittance matrix is needed to perform the Newton Raphson power flow.

5.3 Performing Newton Raphson Power Flow

The Newton- Raphson method is implemented to solve the power flow equations of the hydrocarbon facility electrical system model. The Newton-Raphson method is less vulnerable to divergence in the case of a weak power system, because of its quadratic convergence. Also, the number of iterations needed to converge to the power flow solution when using the Newton-Raphson method is independent of the system size [52, 53, 54]. The MATLAB codes for conducting electrical system power flow using Newton-Raphson in [53] were modified to perform the power flow of the hydrocarbon facility electrical system model.

5.4 Verifying the Constraints

In spite of the aforementioned system equality and inequality constraints, there are operational system constraints that need to be met all the time (global constraints). These constraints are as follow:

1. Having the two GTGs running at their full capacity due to the following:
   a. Increase the generation efficiency.
   b. Produce enough exhaust heat from the GTGs to run the two STGs.
   c. Maximize the injected real power to the utility system.

2. Minimize the reactive power imported or exported to the utility grid. This constraint will be addressed in one of the studied cases as will be detailed in Chapter 6.

5.4.1 Verifying the Voltage Constraints

A MATLAB program (**VCV.m**) is used to verify that all the bus voltage magnitudes produced by the power flow program are within the predefined constraints as posted in sections 4.2.1, 4.2.2 and 4.2.3. The MATLAB codes for the VCV program is posted in Appendix IX.

5.4.2 Verifying the Reactive Power Constraints

A MATLAB program (**MVAR_C.m**) is utilized to verify that all the GTGs, STGs, captive synchronous motor and synchronous motors reactive powers are within the predefined constraints posted in sections 4.2.1 and 4.2.2. Appendix X has the MATLAB codes for the MVAR verification.

5.5 Storing Feasible Solutions

In the event that the voltage and reactive power constraints are met, the system parameters that met all the constraints are considered feasible solutions and stored as the initial feasible solution that will go through the evolution process as mentioned earlier. In this case, lines power flow is performed to capture critical values such as generators outputs real and reactive powers associated with the current feasible solutions. In case that some of the constraints are violated, the system assigned parameters are changed as shown in Figure 5.1, and the process start again until the pre-identified number of initial feasible solutions is obtained.

5.6 Summary

In this chapter, a set of codes to compute different objectives have been developed, validated, and tested .The MTALAB software was used to develop all the programs and sub-programs for addressing the thesis optimization problem. All the codes used were self-coded (developed by the author) and none of the MATLAB built-in solvers were used. This gives flexibility in addressing the subject problem over the MATLAB built-in solvers. The self-developed MATLAB program contains several modules such as reading the model parameters and identifying the initial feasible solutions (populations) mechanism. It was observed that starting the evolution process for identifying the optimal value of the objective function with feasible solutions has critical role in minimizing the evolution process time. The programs of assigning the generators, utility, synchronous motors bus voltages, transformers taps values, the model admittance matrix calculation, the Newton Raphson method for power flow implementation have been discussed and developed. The program used to ensure that the system constraints are met and the program for storing the feasible solutions have been elaborated and discussed.

CHAPTER 6: PROPOSED SOLUTION METHODOLOGY

6.1 Evolutionary Algorithm (EA)

Evolution is an iterative process that improves the situation over time, it never makes big improvement suddenly. The EA idea originated from the facts that most of the animals and plants genes went through evolution development that enables them to adapt to the environments around them. For example, the cows living in cold areas have very long and thick hair to help them adapt to the cold weather. This implies that the genes responsible for growing the long thick hair evolved over the years are dominant genes. The EA is also called intelligent algorithm (IA), which goes through an iterative process that works on a set of individuals (population). Each individual of the population represents a feasible solution of the problem to be solved. Initially, the population is randomly selected and verified to be feasible solutions of the problem in hand. Then, a second set –second population- of feasible solutions is created out of the original population by random selection based on each individual fitness values. The fitness value of an individual represents the goodness of the individual to be a solution of the problem in hand. Each two selected individuals – parent- genes are recombined to produce new individual with shared genes of his parent. After that, the new individual – offspring – genes are perturbed stochastically, refer to Figure 6.1.

This process is repeated until the new offspring population size is met. This means, that the individuals with better fitness values have more chance of passing their genes to the next generation. The aforementioned iterations are repeated until the stopping criteria is met. These criteria can be a predefined maximum number of generations or specific tolerence between the current generation and previous generation best solutions. The EA proves its effectiveness in searching the solution space over the traditional algorithm as the solution space becomes large. The EA main challenges are setting its parameters to be able to reach the global optimal solution within the search space. In addition, selecting the optimal population size and number of generations are other challenges of the EA as both of them dramatically affect the evolution process time. The EA main evolutionary steps are posted in Figure 6.2 [37, 59, 60, 61].

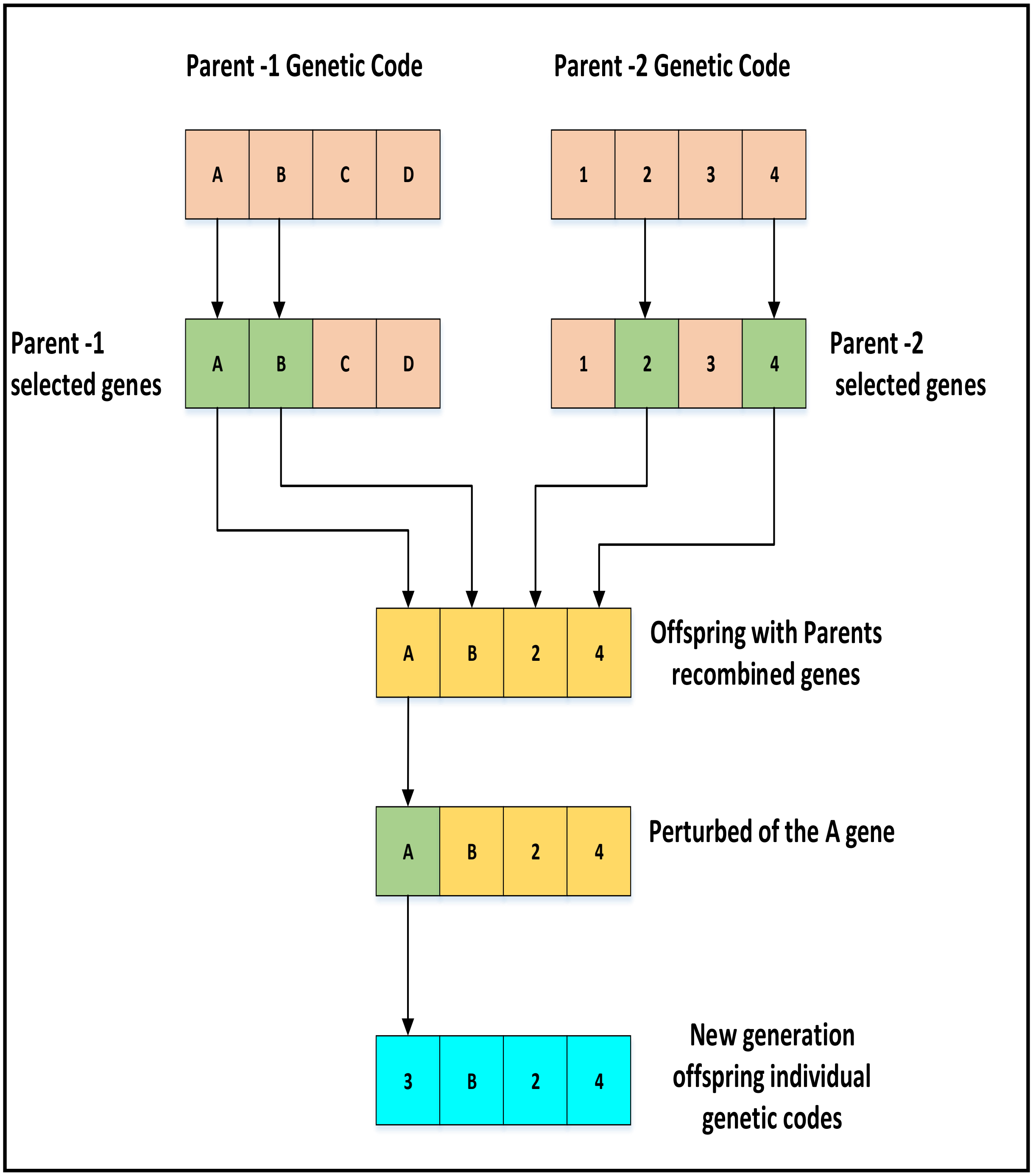

Fig. 6.1: *EA recombination and perturbed of genes*

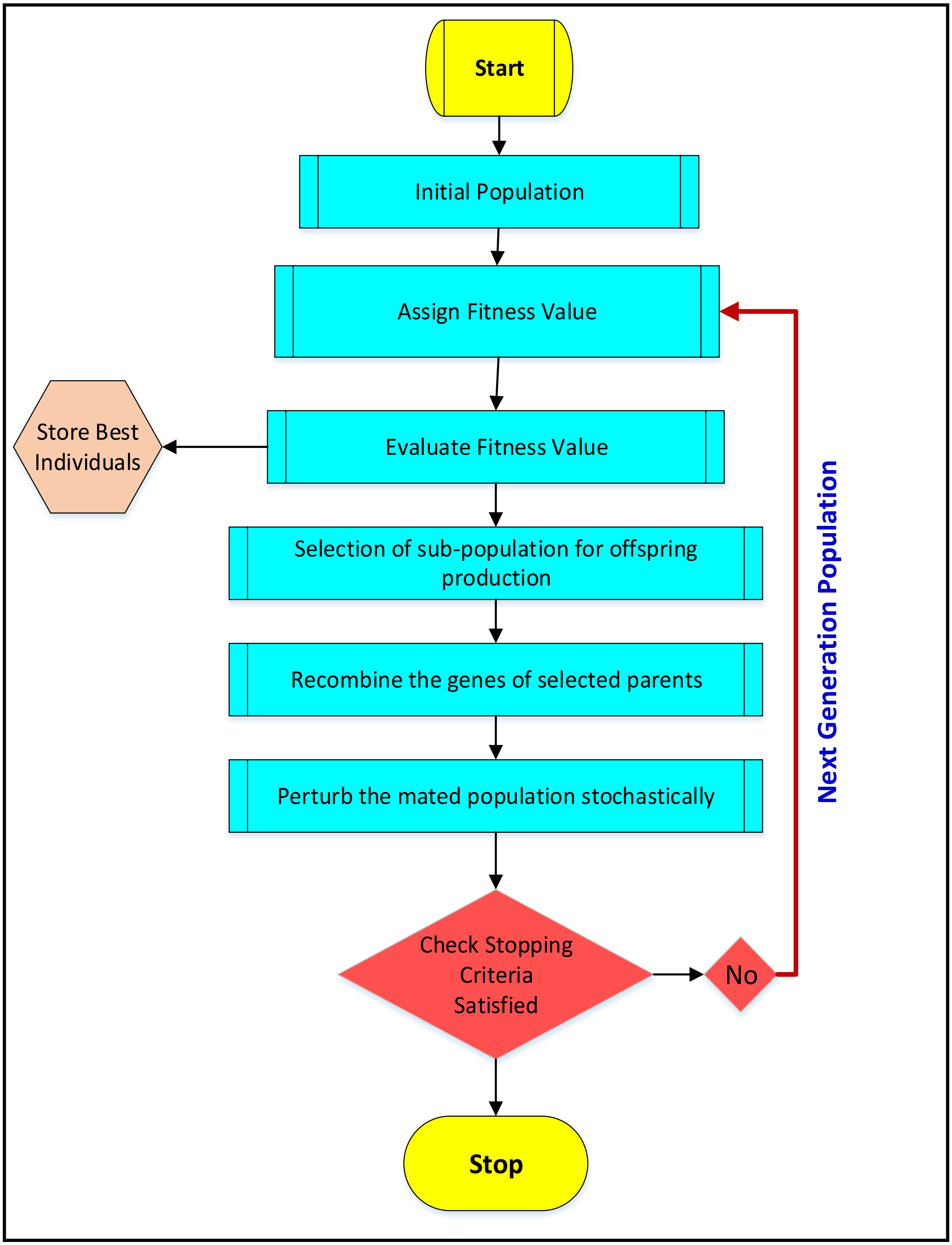

Fig. 6.2: *EA main process steps*

6.2 Genetic Algorithm (GA)

Genetic Algorithm (GA) has been gradually introduced as a powerful tool to handle complex, single and multi-objective optimization problems. Like nature does to its living things, GA tends to develop a group of initial poorly generated solution via selection, crossover and mutation techniques to a set of acceptable solutions through successive generations. In the course of genetic evolution, fitter specimens are given greater opportunities to reproduce. This selection pressure is counterbalanced by mutation and crossover operations. The major advantages of GA lies in its computation simplicity, powerful search ability to reach the global optimum and being extremely robust with respect to the complexity of the problem [62] . GA has the following advantages over other traditional optimization techniques:

1. GA works on both a coding of the parameters to be optimized or the parameters themselves- real values.
2. GA searches the problem space using a population of trials representing solutions to the problem, not a single point, i.e. GA has implicit parallelism. This property ensures GA to be less vulnerable to getting trapped in local minima.
3. GA uses an objective function assessment to guide the search in the problem space.
4. GA uses probabilistic rules to make a decision.
5. It can be used with non-continuous objective functions.
6. It does not require a lot of information about the optimized problem.

The GA was selected to be applied in this thesis due to the fact that it is a well-established algorithm with tuned operators. In addition, GA has been applied to complex problems in different disciplines with impressive success [23, 24, 28].

6.2.1 Genetic Algorithm Process

The main steps of the GA process are summarized in the following steps:

1. Identify the population size and the number of generations.

2. Generate an initial population of chromosomes - each chromosome consists of genes and each of these genes represents the GTGs and STGs terminal voltages, the synchronous motors terminal voltages and the transformer taps settings. The shunt capacitor MVAR value can be represented as gene if MVAR injection via shunt capacitator is considered. The structure of each chromosome can be represented as shown in equation 6.1[23, 61].

$$Chromosome_i = [V_{GTGi}\ldots\ldots V_{GTGN}, V_{STGi}\ldots\ldots V_{STGN}, V_{Synchi}\ldots\ldots V_{SynchN}, T_i\ldots\ldots T_{TN}] \quad (6.1)$$

3. Assign fitness value to each chromosome as follows:
   a. Calculate the objective function value for each chromosome.
   b. Identify if all the equality and inequality constraints are satisfied. If not, disregard the chromosome and replace it with a new one from the initial population.
   c. Assign fitness values to the chromosomes that meet all the constraints. This can be the objective function. For example, in electrical systems real power loss (RPL) optimization, the fitness value can be the RPL value.
4. Identify the best chromosome that has the minimum fitness value- in the RPL minimization optimization problem.
5. Select the chromosome parents that will go to the mating pool for producing the next generation. The following two common selection methods are used in this work:
   a. The tournament selection method.
   b. The random selection method.
6. Perform genes crossover for the selected parents. The following two crossover methods are considered:
   a. Simple crossover method.
   b. Differential crossover method.
7. Perform genes mutation on the selected parents after been crossovered. The used mutation methods are as follow:
   a. Random mutation.
   b. Non-uniform mutation method.

8. Go to step#5 and repeat the process until the number of the offspring chromosomes meet the predefined population size.

9. Go to Step#3 and repeat the above steps with the new chromosomes generation generated from the original chromosomes parents after being crossovered and mutated.

10. In each time identify the best chromosome and compare its fitness with the stored one, if it is better replace the best chromosome with the new one.

11. The loop of generation is repeated until the best chromosome in term of minimum real power loss is identified.

The afermentioned steps are summarized in Figure 6.3. It is worth mentioning that the type of problem in hand influences which selection, crossover and mutation methods will be used. In some problems, one set of these operators' mix would work better than other one. Also, the rates of the crossover and the mutation do have their influence on the convergent of the GA evolution process to the optimal solution.

Depending on the size of the problem, the population size and the generation number have to be tuned to give enough time for the GA evolution process to search the solution space for the optimal solution. Yet, as the population size becomes bigger and the same apply for the number of generations, the evolution process takes longer time, which prevents the decision makers to take the required decisions in a short time. This implies that, depending on the problem size, the selection of the population size and the number of generations will change.

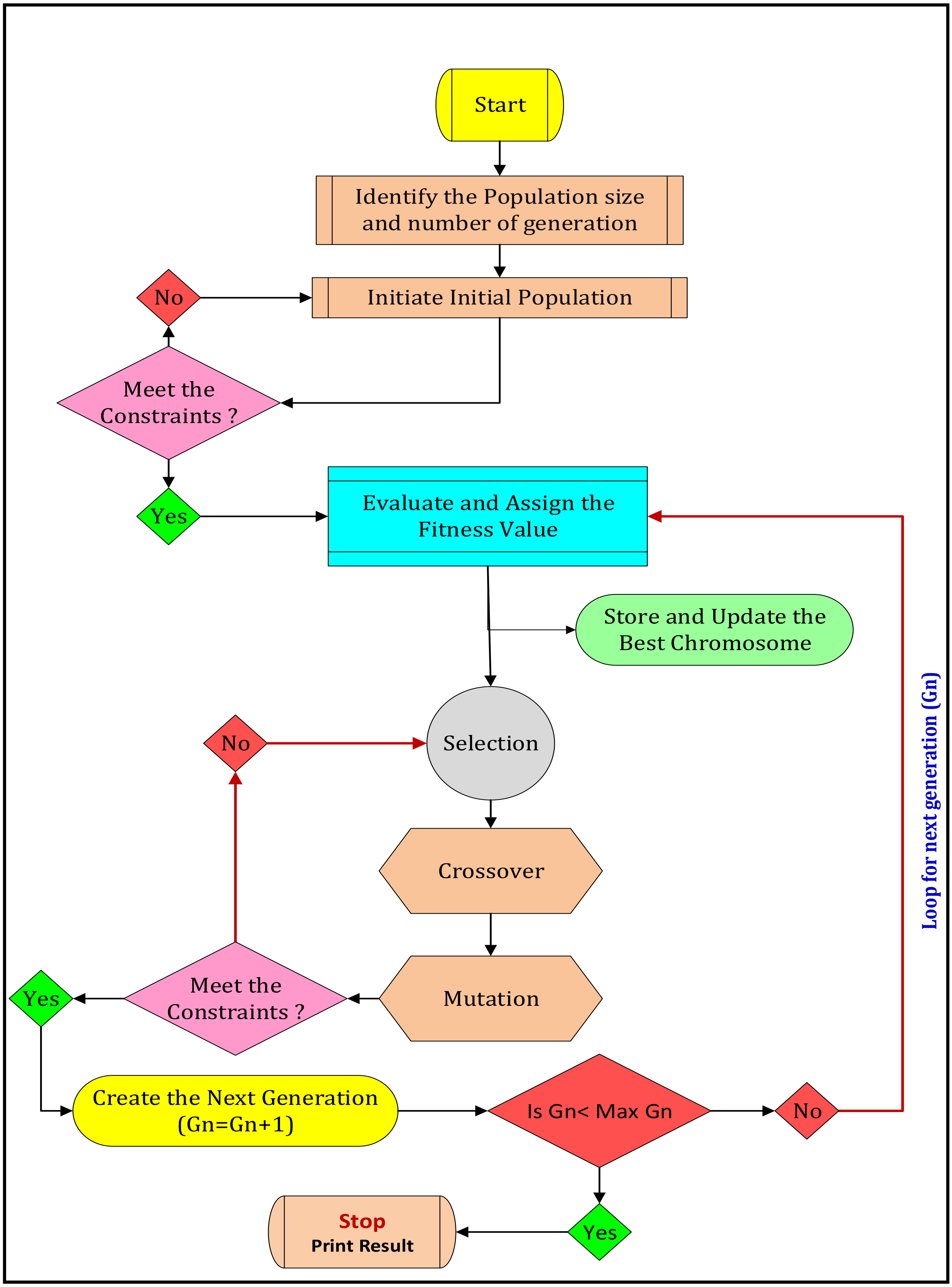

Fig. 6.3: *GA evolutionary process steps*

6.2.2 GA Selection Methods

There are many GA selection methods, yet among the most popular ones are tournament and random methods [63].

a. The tournament selection method

The tournament selection method does compare three chromosomes each time and allows the one with better fitness value – for example, less RPL MW value (J_1) to copy itself twice in the mating pool and the second one with respect to fitness value to copy itself only once in the mating pool. This will be repeated until the population size is met. Figure 6.4 demonstrates how the tournament selection method works, giving that each chromosome consists of GTG (V_{GTGi}), STG (V_{STGi}) and Synchronous motors (V_{Synchi}) terminal voltages and shunt capacitor MVAR value ($MAVR_{Capi}$) genes.

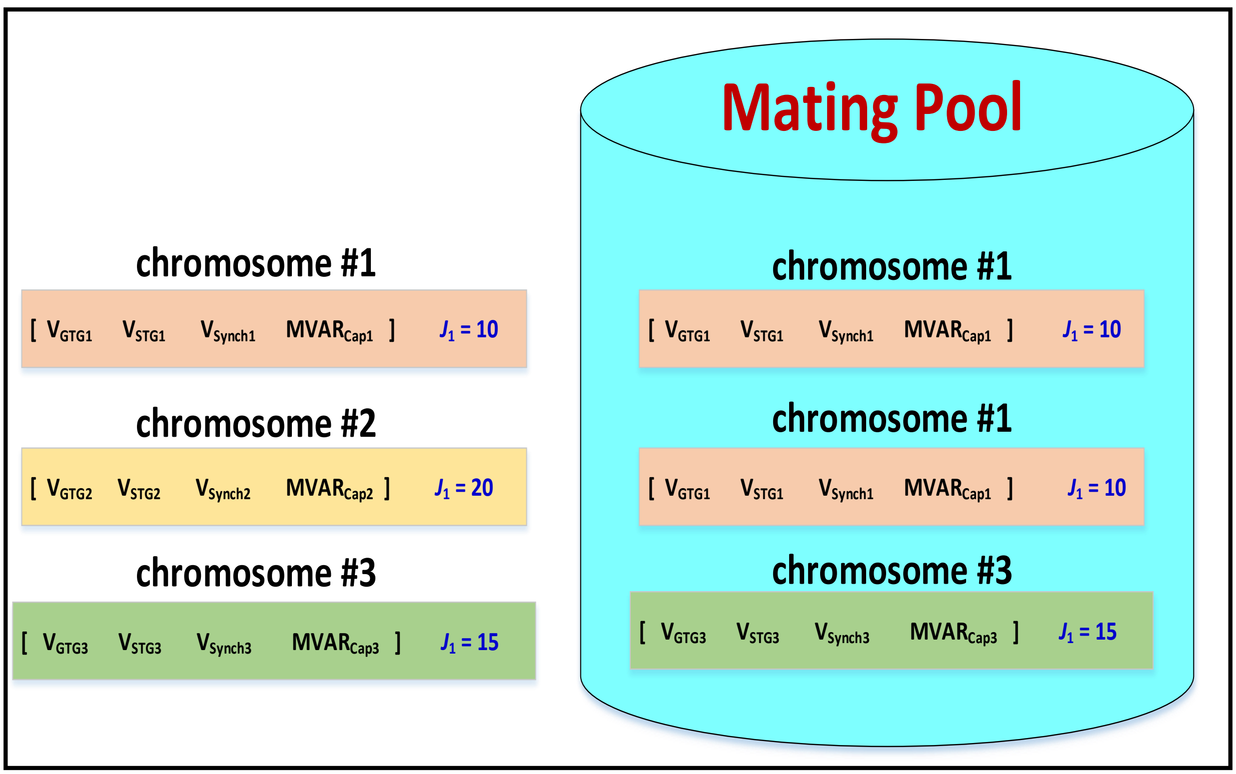

Fig. 6.4: *GA* tournament *selection method*

b. The random selection method

The random selection method works by randomly selecting two chromosomes. Then, these two randomly selected chromosomes fitness values are compared and the one with the better fitness value will go into the mating pool. This randomly selected chromosomes mechanism will be repeated until the population size in the mating pool is equal to the predefined population size. The random selection method is demonstrated in Figure 6.5.

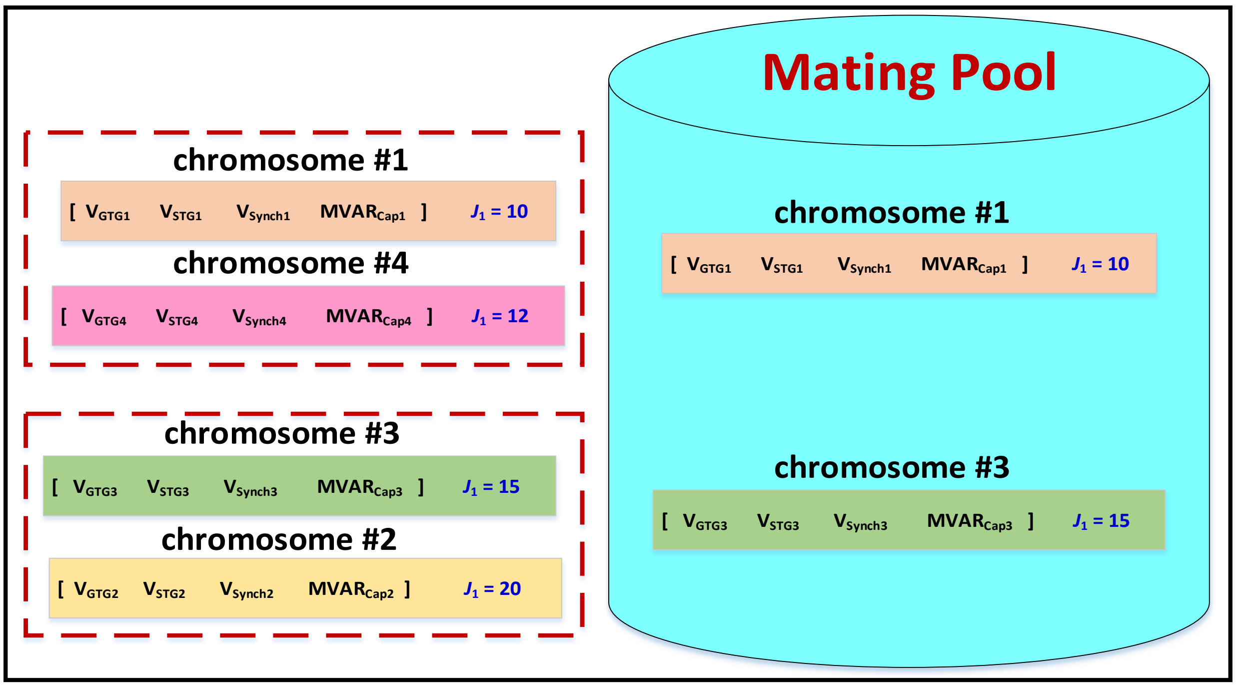

Fig. 6.5: *GA random selection method*

6.2.3 GA Crossover Methods

The crossover is a method for sharing genes information between chromosomes. It combines the genes features of the two parent chromosomes to produce the offspring chromosomes. This has the probability that better chromosomes may be produced. Performing the crossover depends on the crossover rate. For example, if the crossover rate is 90%, then there is a possibility that 90% of the selected parents will be subjected to crossover operation and the opposite apply when the crossover rate is very low. The two types of crossover used in this thesis are the simple crossover and differential crossover – differential mutation [63, 64].

a. Simple crossover method

The offspring H = (h_1,….h_i,….h_n) is generated in the simple crossover method by establishing a vertical crossover position then, the two new offspring chromosomes are built. Figure 6.6 demonstrates this crossover method.

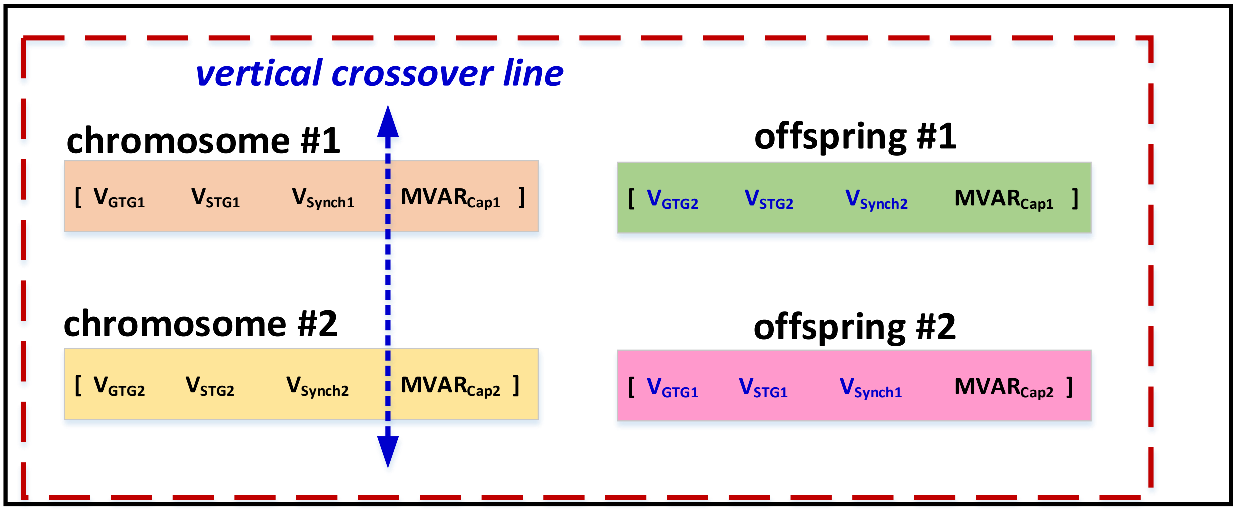

Fig. 6.6: *GA simple crossover method*

b. The differential crossover method

The differential crossover (DC) applies a similar technique to the BLX crossover method [63, 66] where chromosome genes are crossovered via the difference of the parent genes. The DC works by selecting three chromosomes. Then, a scalar number β scales the difference of any two of the three chromosomes genes and the scaled difference is added to the third chromosome gene to produce the offspring chromosome. The DC process of producing the offspring chromosomes from the parents chromosomes C_1, C_2 and C_3 is demonstrated in the equations below [37,66].

$$C_{1\text{-Offspring}} = C_1 + \beta\,(C_2 - C_3) \tag{6.2}$$

$$C_{2\text{-Offspring}} = C_2 + \beta\,(C_3 - C_1) \tag{6.3}$$

$$C_{3\text{-Offspring}} = C_3 + \beta\,(C_2 - C_1) \tag{6.4}$$

6.2.4 GA Mutation Methods

The mutation alters one or more of the chromosome genes randomly. This enhances the genes features of the chromosomes as it usually brings new gene codes. It helps in exploring new areas of the solution space, accordingly exploring most of the solution space. Similar to the crossover, the mutation is applied with predefined rate. There are two types of mutation rates considered in this work, the fixed rate and the dynamic rate or non-uniform rate. In the fixed mutation rate, the mutation rate is fixed usually set low – less than 10% - to ensure convergent. The second type of mutation rate is the dynamic rate that is gradually reduced with increase in the number of generations. The dynamic rate allows high mutation rate to explore as much as possible the solution space at the early generations. Then, it reduces the mutation rate toward the end of the evolution process to ensure convergent [63, 64]. In this work two, types of mutation methods are considered as follows:

a. Random mutation method

In the random mutation method, the new gene is generated randomly from the gene domain. For example the new h^1_1 gene, which is the gas turbine generator terminal voltage chromosome gene, is generated randomly from V_{GTG} domain (V_{GTG}^{min} –V_{GTG}^{max}).

b. Non-uniform mutation method

This method uses the following equations for mutation

$$g_i' = \begin{cases} g_i + (b_i - g_i) * \left[1 - r^{\left(1-\frac{t}{Gmax}\right)^d}\right] & \text{when } \tau = 0 \\ g_i + (g_i - a_i) * \left[1 - r^{\left(1-\frac{t}{Gmax}\right)^d}\right] & \text{when } \tau = 1 \end{cases} \quad (6.5)$$

Given the followings:

1) τ is arandomly generated binary number 0 or 1.

2) r is a randomly generated real number from the interval (0-1)

3) d is an integer number, d can be 1 to 10.

4) Gmax is the maximum number of Generations

5) t is the current Generation number.

6) a_i & b_i is the domain of the gene. For example, in the GTG terminal bus voltage gene case $a_i = V_{GTG}^{min}$ and $b_i = V_{GTG}^{max}$.

7) g_i is the gene current value.

The non-uniform mutation suppresses the mutation rate as the generation number is reaching the maximum number of the generation. Yet, the rate of the suppression depends on the value of d. At d =0, there is full mutation (100%), at d=3, the mutation rate is decreased.

Rastrigin function is used to demonstrate the effect of the non-uniform mutation d value. The two variables Rastrigin function is represented in equation (6.6).

$$Ras(x) = 20 + x_1^2 + x_1^2 - 10\,(\cos 2\pi x_1 + \cos 2\pi x_2) \quad (6.6)$$

The searching domain of the Rastrigin function is posted in equation (6.7)

$$-5.12 \le x_i \le 5.12 \quad (6.7)$$

The Rastrigin function has many local minima but only one global minima where Ras(x) = 0 for $x_1 = x_2 = 0$. Three (3D) dimensions representation of the Restring function is shown in Figure 6.7. The Rastrigin function is a challenging problem for the genetic algorithm to find its global minima as it has a large number of local minima.

As shown in Figure 6.8, as the value of d increases the search becomes more focused around the parent gene values which supports guiding the evolution more precisely. The smaller d value (d = 0), pushes the mutation of the offspring generation genes value away from the parents gene values. This affects the convergence of the evolution. The bigger the d values are, the more fine tuning search around the parent gene will be, this results in convergence to local minima. An average value of d (d=3) supports a little slower convergence but with a better optimized value.

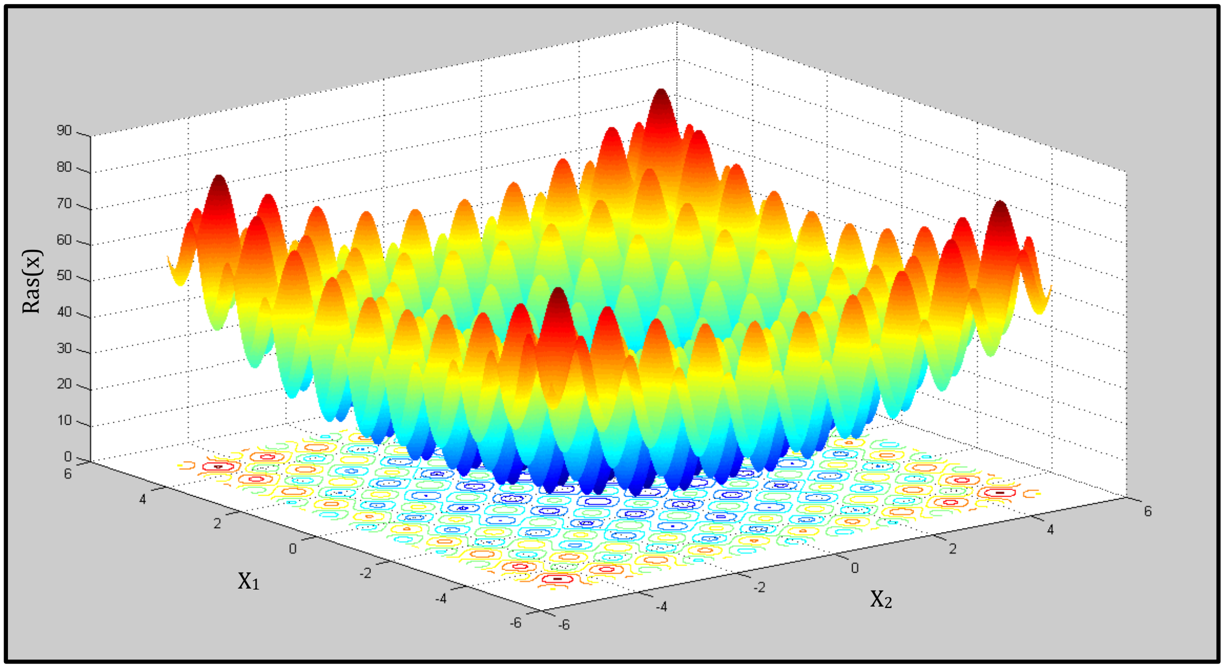

Fig. 6.7: *3D representation of Rastrigin function*

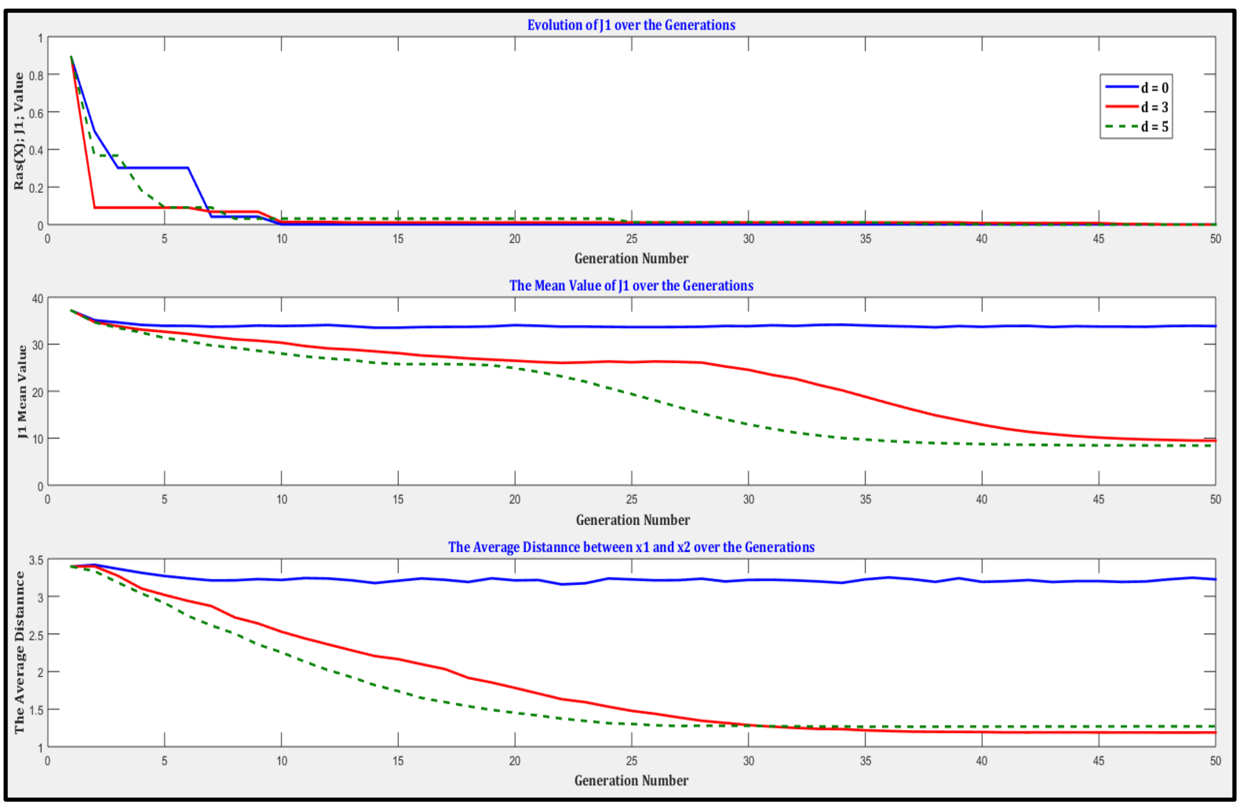

Fig. 6.8: *Behavior of Rastrigin function for different non-uniform mutation d value*

6.2.5 GA MATLAB Programming Codes

The GA MATLAB programming codes structure is posted in Figure 6.9. The text in red color is the name of the MATALB sub-program codes. Most of these MATLAB sub-program codes are posted in the appendices [52,53, 54]. All these sub-programs codes are customly developed.

The assigned fitness will depend on the objective function. For example, the system real power loss will be the fitness value of each chromosome in case the objective function is the real power loss (RPL) optimization. This implies that a chromosome with a low fitness value has a higher probability of passing its genes to the next generation. The fitness value can be changed depending on the problem in hand.

The tournament and the random selection methods MATLAB sub-program codes are illustrated in Appendix XI, refer to section 6.2.2. As posted in section 6.2.3, the simple and differential crossover methods MATLAB sub-program codes are posted in Appendix XII. The random and non-uniform mutation methods addressed in section 6.2.4 MATLAB sub-program codes are posted in Appendix XIII.

The green texts within the MATLAB sub-program codes posted in the aforementioned appendices are very important. These texts help the user to understand the following code steps. For example, in Appendix XIII – mutation methods MATLAB codes - selected feasible transformer tap values are used out of the full range values. This is explained as part of the green text above the code lines.

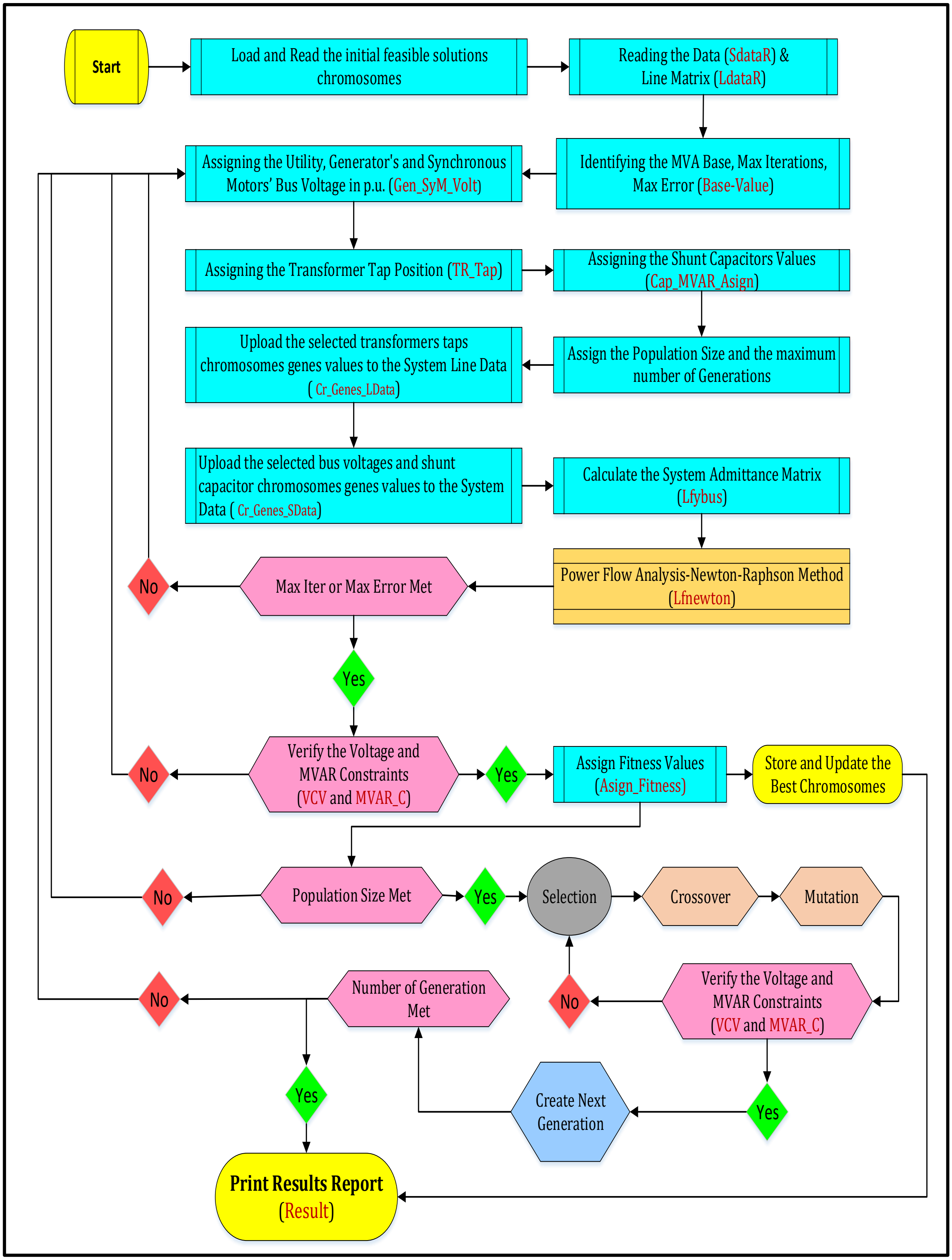

Fig. 6.9: *GA MATLAB programming codes structure*

6.3 The Multi-Objective Evolutionary Algorithm (MOEA)

Recently, the evolutionary multi-objective algorithms have been proved to be very robust in eliminating the common difficulties associated with the classical methods as listed below [67]:

1. Most classical algorithms require some knowledge about the problem being solved.
2. The classical method has to be applied many times to find multiple Pareto optimal solutions.
3. The classical algorithms are sensitive to the shape of the Pareto optimal front.

In general, the goal of a multiobjective optimization algorithm is not only guiding the search towards the Pareto optimal front while maintaining population diversity in the set of the nondominated solutions. This cannot be maintained with the classical method.

MOEA is designed to solve multiobjective optimization problems as they deal simultaneously with a set of possible solutions called population. This allows finding an entire set of Pareto optimal solutions in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Also, MOEAs are less vulnerable to the shape or continuity of the Pareto front. These are the serious difficulties associated with the classical methods.

The advantages of MOEA algorithms over the classical methods can be summarized as follows.

(a) Since MOEA algorithms use a population of solutions in their search, multiple Pareto optimal solutions can, in principle, be found in one single run.

(b) The use of diversity preserving mechanisms can be added to the evolutionary search algorithms to find widely different Pareto optimal solutions.

(c) MOEA algorithms do not require any knowledge in advance about the problem being solved.

(d) MOEA algorithms are insensitive to the shape of the Pareto optimal front.

(e) There is no restriction about the number of the objective functions being optimized.

In this research, two of the most capable MOEA techniques will be developed and implemented to solve the multi-objective optimization problem of RPL. Namely, the improved strength Pareto evolutionary algorithm (SPEA2) and the differential evolutionary algorithm (DEA).

6.4 Improved Strength Pareto Evolutionary Algorithm (SPEA2)

The SPEA2 is an improved version of the strength Pareto evolutionary algorithm (SPEA). The SPEA based approach has the following features [68, 69]:

- It stores externally those individuals that represent a nondominated front among all solutions considered so far.
- It uses the concept of Pareto dominance in order to assign scalar fitness values to individuals.
- It performs clustering to reduce the number of individuals externally stored without destroying the characteristics of the trade-off front.

The SPEA2 main differences from the SPEA are [70]:

- It uses an improved fitness assignment scheme which takes for each individual into account how many individuals it dominates and it is dominated by.
- The archive truncation method guarantees the preservation of boundary solutions.
- A nearest neighbor density estimation technique is incorporated which allows a more precise guidance of the search process.

The following steps describe the SPEA2 evolutionary process [12].

Step 1: Initialization
The population P_0 is generated with K size and a vacant annals (external) Pareto-optimal set $\overline{P}_0$ with $\overline{K}$ size.

Step 2: Updating of external Pareto set

In order to bring an updated set of the external Pareto optimal set, the following steps are to be followed:

(a) The population of non-dominated individuals are highlighted and reproduced to the external Pareto set.

(b) Look for the set of external Pareto, designed for the non-dominated individuals.

(c) If condition ($\overline{P_{t+1}}$) <$\overline{K}$ is satisfied, keep the individuals with higher fitness values until | $\overline{P_{t+1}}$| = $\overline{K}$ is satisfied.

(d) If ($\overline{P_{t+1}}$) >$\overline{K}$, truncation procedure is implemented to remove individuals from ($\overline{P_{t+1}}$) in anticipation of | $\overline{P_{t+1}}$| = $\overline{K}$.

Step 3: Assignment of fitness

The fitness values of the individuals are calculated in both the external Pareto set $\overline{P}_t$ and the population P_t *as* follows:

(a) *St*(*i*)- strength value, is assigned to all individuals *i* inside the external Pareto set $\overline{P}_t$ and the population P_t. *St*(*i*) signifies the unit – number of individuals - which *i* dominates and is expressed as follows:

$$St(i) = | \{ j, j \in P_t + \overline{P}_t , \wedge\, i > j \} | \tag{6.8}$$

Then, the raw fitness $R_w(i)$ with respect to an individual *i* can be measured as follows:

$$R_w(i) = \sum_{j \epsilon P_t + \bar{P}_t , j > i} St(j) \tag{6.9}$$

The raw fitness of an individual is obtained with respect to the strength of its dominators in the archive and population.

(b) The distances between an individual *i* and the entire *j* individuals, in the course of external and population sets, are listed. Then, the list is sorted in a cumulative manner, the distance to the *m*th individual, consequently, $m = \sqrt{\overline{K} + K}$ is represented as σ_i^m. Then, D(*i*) (density) is calculated for each *i:*

$$D(i) = \frac{1}{\sigma_i^m + 2} \tag{6.10}$$

The addition of integer 2 is made in the denominator to certify that the value of D (*i*) is larger than zero and is < 1. The fitness value *i* of an individual *is* expressed as follows:

$$F(i) = R_W(i) + D(i) \tag{6.11}$$

Step 4: Selection

Two individuals have been selected on a random basis from the restructured external set $\overline{P_{t+1}}$. In light of their fitness values, the best one is selected and is copied to the mating pool.

Step 5: Crossover and Mutation

The crossover operation and mutation operation are performed on the basis of their probabilities for the generation of the novel population.

Step 6: Looping back

The stopping criteria is checked. If not satisfied, then, copy the new offspring population to the old one.

Step 7: Termination

The criteria for termination is checked. If satisfied, apply the fuzzy set theory to identify the best possible compromise solution out of the Pareto external set. Figure 6.10 illustrates the evolutionary steps involved in the SPEA2 evolutionary process.

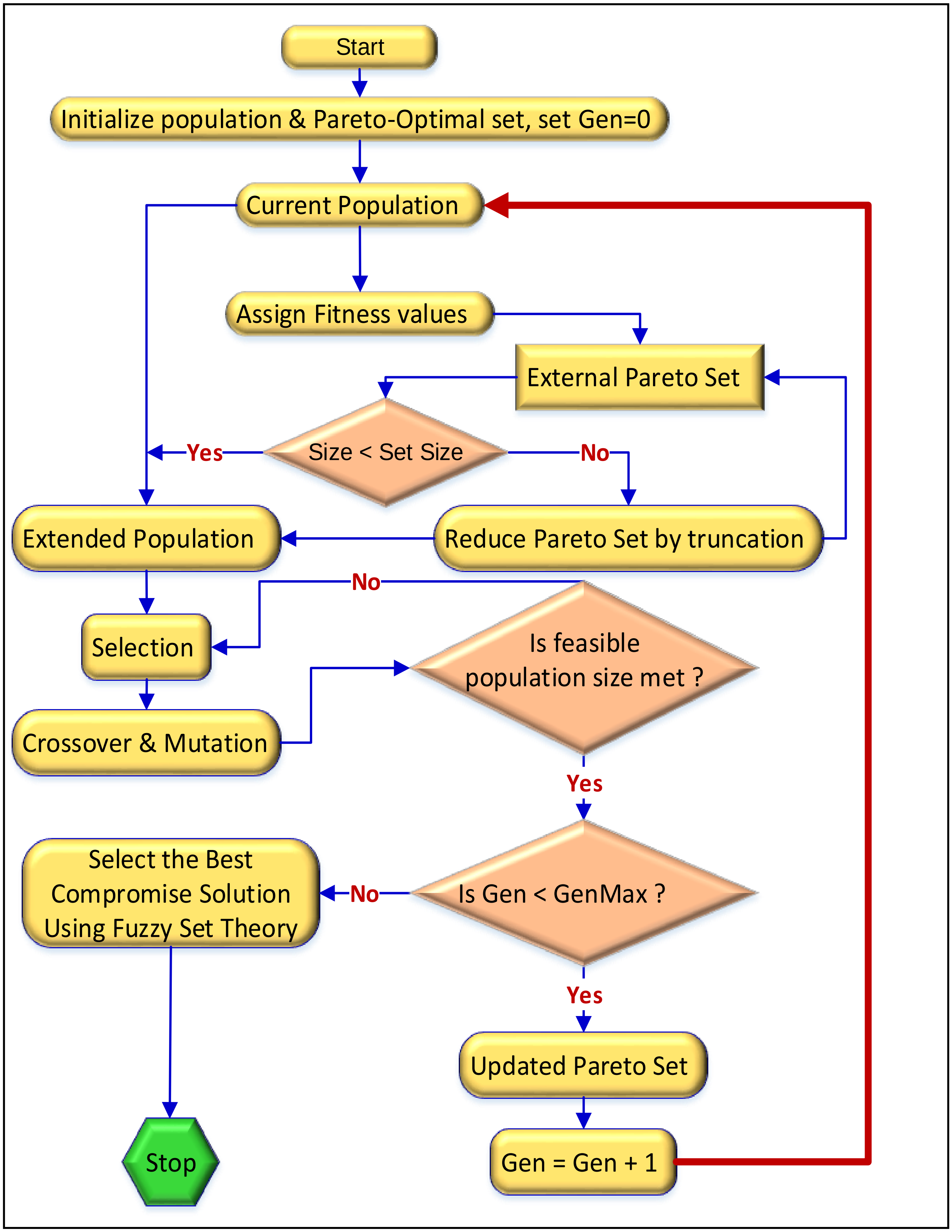

Fig. 6.10: *SPEA2 evolutionary process chart*

6.5 Fuzzy Set Theory for Extracting the Best Compromise Solution

The decision makers need a trusted selection technique to identify the best compromise solution out of the Pareto-optimal set of non-dominated individuals [12, 37, 42]. The fuzzy set theory will help in presenting one solution to the decision makers as the best compromise solution. Due to the decision makers' inexact conclusion, the i^{th} objective function J_i is represented by a membership function μ_i defined as:

$$\mu_i = \begin{cases} 1 & J_i \le J_i^{\min} \\ \dfrac{J_i^{\max} - J_i}{J_i^{\max} - J_i^{\min}} & J_i^{\min} < J_i < J_i^{\max} \\ 0 & J_i \ge J_i^{\max} \end{cases} \tag{6.12}$$

where $J_i^{\min}$ and $J_i^{\max}$ are the minimum and maximum value of the i^{th} objective function among all non-dominated solutions, respectively. For each non-dominated solution k, the normalized membership function μ^k is calculated as follows:

$$\mu^k = \frac{\sum_{i=1}^{N_{obj}} \mu_i^k}{\sum_{k=1}^{M} \sum_{i=1}^{N_{obj}} \mu_i^k} \tag{6.13}$$

where M is the number of non-dominated solutions. The best compromise solution is that having the maximum value of μ^k. The MATLAB sub-program codes of the fuzzy set theory (Extra_Best_Chr.m) is posted in Appendix XIV.

6.6 Front Pareto Set Reduction by Truncation

In some cases, the front Pareto optimal set size can be very large. In such cases, reducing the set of nondominated solutions without destroying the characteristics of the trade-off front is necessary. One method to accomplish this goal is the truncation technique. In the truncation technique, a minimum distance based algorithm [12] is utilized to reduce the size of the Pareto set to a manageable size, i.e. the specified external Pareto set size. At each iteration an individual *i* is chosen for removal from the external pareto set P_{t+1} in case its size gets larger than the predefined size. The MATLAB sub-program codes of the truncation technique (TruncationR1.m) is posted in Appendix XV.

6.7 Differential Evolutionary Algorithm (DEA)

DEA utilizes special differential operators in creating the offspring individuals from the parent individuals' population in place of the classical crossover and mutation operators used in the GA and SPEA2. In DEA, there are two control parameters, which are the mutation constant F and the crossover constant C. Also in DEA, the mutations are performed before crossover. DEA's in its single objective format first three evolutionary process steps are similar to the GA ones. The same applicable in its multi-objective format as the first three evolutionary process are similar to the SPEA2 [32, 37, 38, 65, 71, 72]. The remaining steps are as described below:

Step 4: Mutation

Different from the SPEA2, in which the individuals subjected to crossover and mutation are selected from the Pareto optimal set, in the DEA the individuals are selected from the population. In DEA, mutations are performed using the DE/rand/1 mutation technique [65]. V_i (t) - the mutated vector, is created for each population member X_i (t) set by randomly selecting three individuals' x_{r1}, x_{r2} and x_{r3} and not corresponding to the current individual x_i. Then, a scalar number F is used to scale the difference between any two of the selected individuals. The resultant difference is added to the third selected individual. The mutation process can be written as:

$$V_{i,j}(t) = x_{r1,j}(t) + F * [\, x_{r2,j}(t) - x_{r3,j}(t) \,] \quad (6.15)$$

The value of F is usually selected between 0.4 and 1.0. In this study, F was set to be 0.5 (50%). In [73], scaling mutation based on the frequency of successful mutations is applied.

Step 5: Crossover

Perform the binomial crossover, which can be expressed as follows:

$$u_{i,j}(t) = \begin{cases} v_{i,j}(t) \; if \; rand\ (0,1) < CR \\ x_{i,j}\ (t) \qquad\qquad else \end{cases} \tag{6.16}$$

CR is the crossover control parameter, and it is usually set within the range [0, 1]. The child $u_{ij}(t)$ will compete with its parent $x_{ij}(t)$. CR is set equal to 0.9 (90%) in this study.

Step 6: Selection

The procedure for the selection is described as follows:

$$x_i(t+1) = u_i(t) \quad \text{condition} \quad f(u_i(t)) \leq f(x_i(t)) \tag{6.17}$$

$$x_i(t+1) = x_i(t) \quad \text{condition} \quad f(x_i(t)) \leq f(u_i(t)) \tag{6.18}$$

where $f(\,)$ is the objective function to be minimized.

Step 7: Looping back

Look for the terminating criteria. If the criteria are not fulfilled, then generate new offspring population and begin again.

Step 8: Termination

If the termination criteria are met, apply the fuzzy set theory for the identification of the best compromise pareto set. Figure 6.11 demonstrates DEA evolutionary steps in its multi-objective format. The DEA in single objective mode is shown in Figure 6.12. DEA was selected as a senod algorithm for this thesis application as it is relatievly new with very promising performance [23, 36, 37, 38].

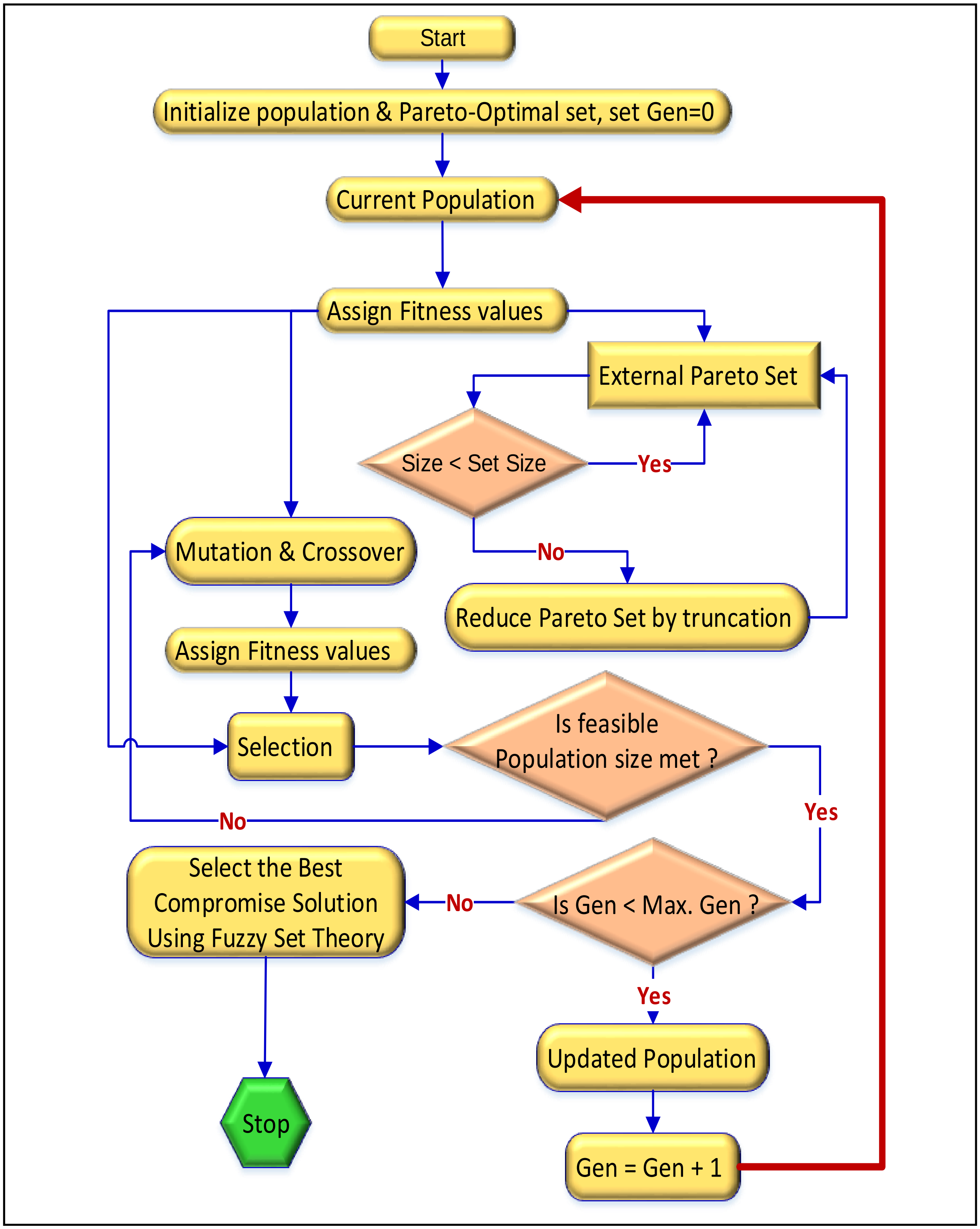

***Fig. 6.11:** DEA in multi-objective mode evolutionary process chart*

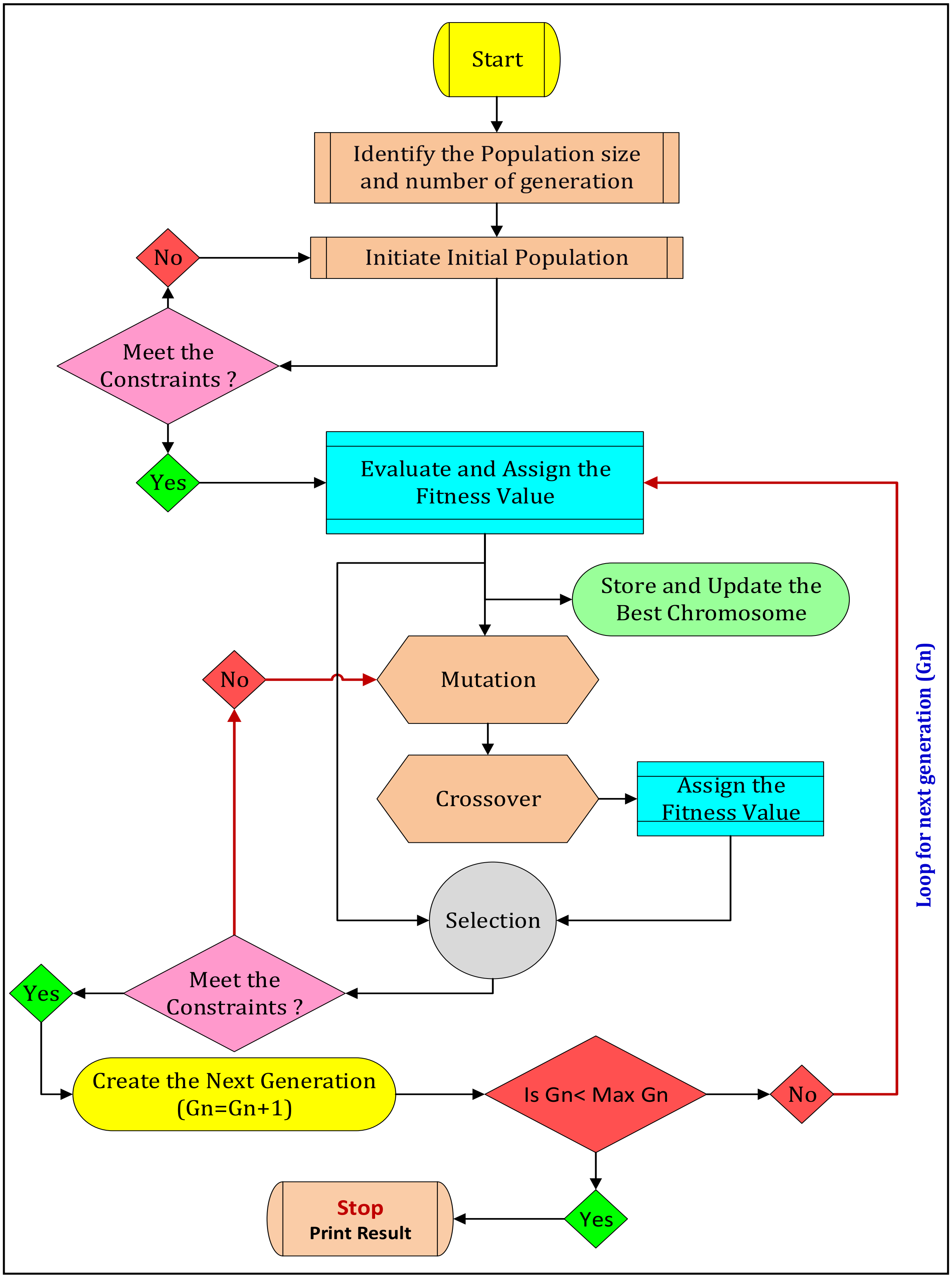

Fig. 6.12: *DEA in single-objective mode evolutionary process chart*

6.8 SPEA2 and DEA MATLAB Programming Codes Structure

The structure of MATLAB programming codes for the multiobjective SPEA2 and DEA algorithms is very similar to the GA MATLAB coding structure. The multiobjective structure includes identifying the non-dominated chromosomes within the population. This is performed by the MATLAB sub-program named Non_Dominat_Chr_Pop.m. The MATLAB programming codes of this sub-program are posted in Appendix XVI. The identified non-dominated chromosomes within the population is moved to the external optimal Pareto set (EOPS). The EOPS is then searched to identify the non-doimnated chromosomes via the sub-program MATLAB codes named Non_Dominat_Chr_EOPS.m. This sub-programs MATLAB codes are illustrated in Appnedix XVII.

The non-dominated EOPS chromosomes are selected, crossovered and mutated to generate the offspring next generation in the case of SPEA2. In case the pre-defined EOPS size is not met, then more chromosomes will be selected and go through the crossover and mutation procedures until the predefined EOPS size is met.

The new offspring (generation) will go through the same procedure until the maximum number of generation is met. In case that the number of non dominated chromosomes within the EOPS exceeds the predefined size of the EOPS, then, the truncation technique is implemented to reduce it to the pre-defined size. In case, the number of chromosmes within the EOPS is less than the predefined size of the EOPS, then, the dominated chromosmes with best fitness values are moved to the EOPS until the size is met. Refer to Figure 6.13.

In the case of DEA, the offspring is generated from the population and not the EOPS. This gives the DEA an advantage over the SPEA2 to wildly explore the solution space. As in the case of the SPEA2, the offpring is generated from the EPOS which makes the evolution process focus on the most promising solutions and not the whole solution space. Exploring other solutions outside the EOPS chromosomes space is archived via the mutation. Refer to Figure 6.14. Also, selecting the best individuals out of the parents and off-springs is another advantage of DEA over SPEA2. The DEA

MATLAB structure in its single objective mode is shown in Figure 6.15.

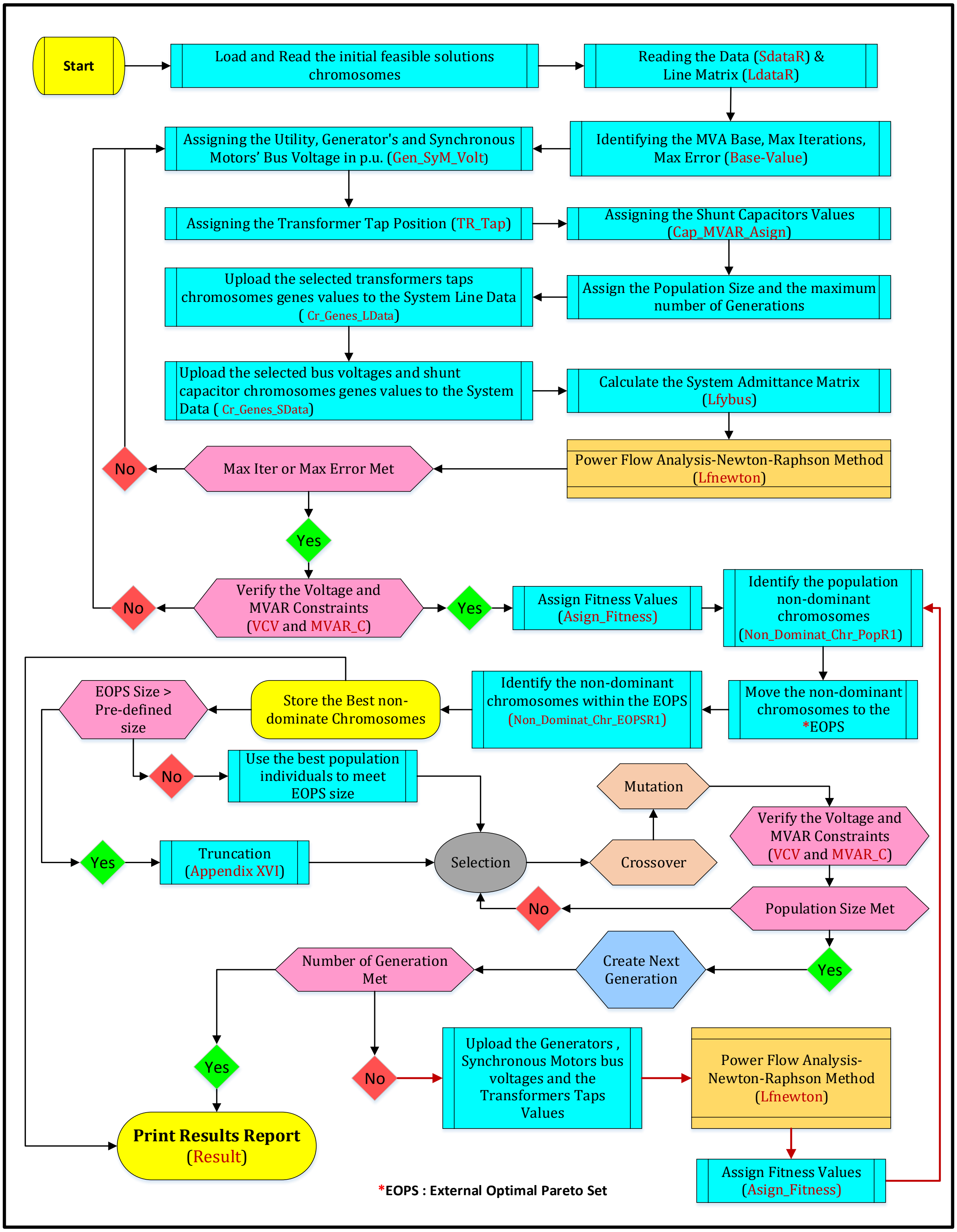

Fig. 6.13: The SPEA2 MATLAB programming structure

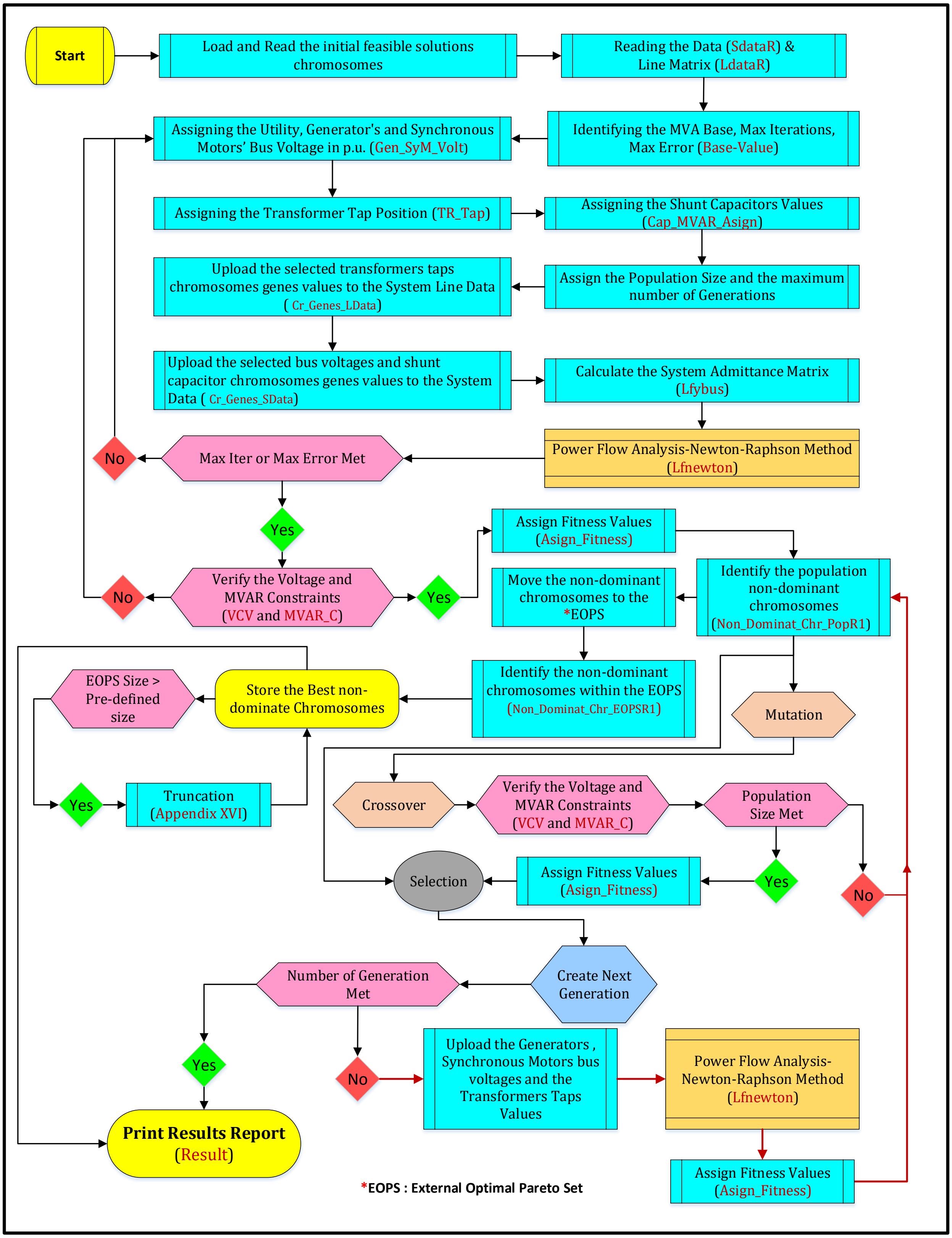

Fig. 6.14: *The DEA in multi-objective mode MATLAB programming structure*

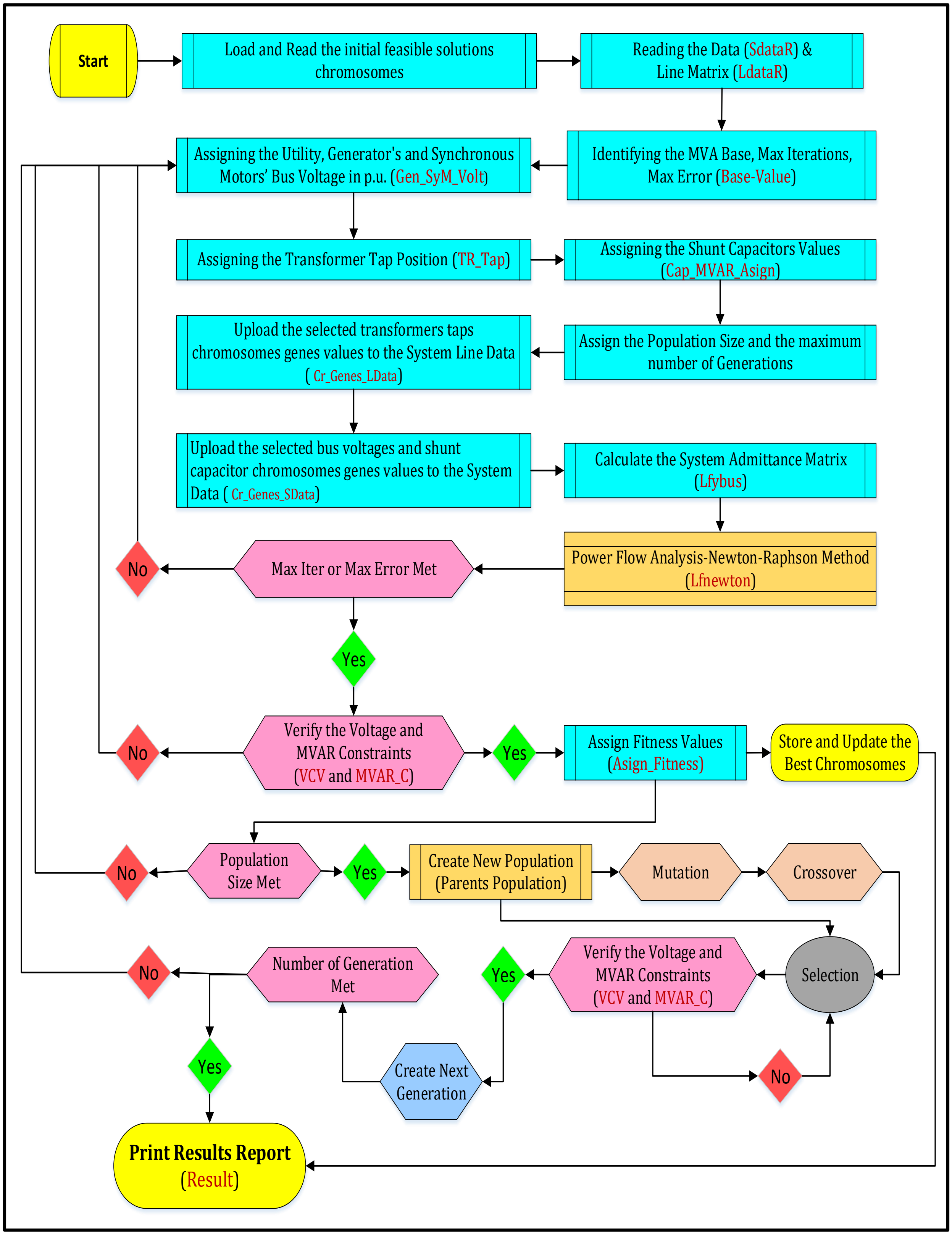

Fig. 6.15: The DEA in single-objective mode MATLAB programming structure

6.9 Summary

In this chapter, the concepts of evolutionary and GA process have been explained. The GA two selection operators (tournament and random), sample and differential crossover operators and the two mutation operators (random and non-uniform) have been developed and addressed. The multi-objective evolution process has been introduced. The GA multi-objective version SPEA2 evolution process and the fuzzy set theory method for extracting the best compromise solution out of the front pareto set have been developed and illustrated. The truncation method for reducing the front pareto set size to the pre-defined size has been addressed. The DEA in its multi-objective and single objective mode process has been developed. The GA, SPEA2 and DEA MATLAB program structure in both its single objective and multi-objective format have been developed and demonstrated. The applied intelligent algorithm crossover and mutation values were based on common tuned values for the application of power system optimization problem.

CHAPTER 7: RESEARCH STUDY CASES AND RESULTS ANALYSIS

7.1 Business as Usual (BAU) Case

In the BAU case, the real life hydrocarbon plant electrical system operators adjust the system parameters to satisfy the following conditions:

(a) Minimize the imported real power from the grid.

(b) The utility bus and all the generators buses to be kept at 1.0 p.u. voltage.

(c) All the synchronous motors are set to operate very close to the unity power factor.

(d) All downstream distribution transformers and the captive synchronous motors transformers offload tap changers are put on the neutral tap.

(e) The main transformer taps of the causeway substations are raised to meet the very conservative voltage constraints at these substations' downstream buses (≥ 0.95 p.u.). Yet, the other substations bus voltages are set ≥ 0.90 p.u.

Table 7.1 lists all the BAU independent variable values in light of the aforementioned conditions. The BAU case - normal system operation mode - was simulated to be benchmarked with the other optimal cases.

Table 7.1: The selected feasible bus voltages' and transformer taps' values

Substation Number	Transformer Tap	Bus Voltage p.u.
Utility Bus Voltage	N/A	1.0
GTGs Terminal Bus Voltage	N/A	1.0
STGs Terminal Bus voltage	N/A	1.0
Captive Synchronous Motors Bus Voltage	N/A	0.98
Causeway Substation #1	+3 (1.019 p.u.)	N/A
Causeway Substation #2	Neutral (1.0 p.u.)	N/A
Causeway Substation #3	+3 (1.019 p.u.)	N/A
Main Substation Transformers	+1 (1.006 p.u.)	N/A

7.1.1 BAU Generation and Load Values

The BAU load, generation and power loss values are shown in Figure 7.1. All these values are as stated before based on actual system parameters of the real hydrocarbon facility considered.

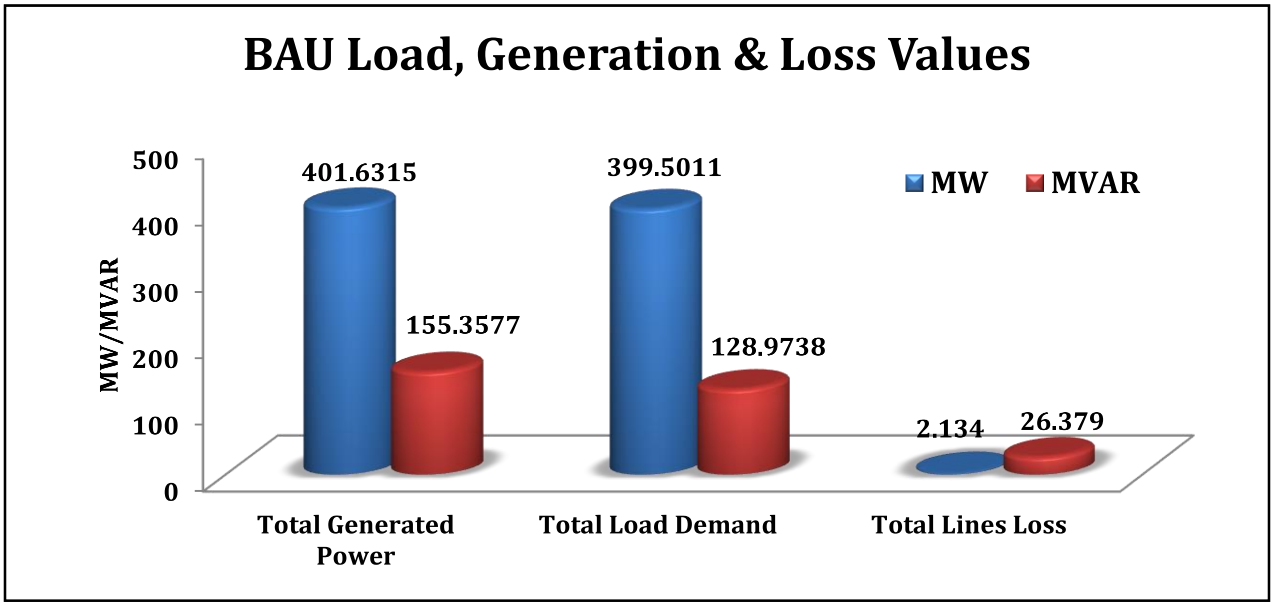

Fig. 7.1: *The hydrocarbon facility system BAU load, generation and loss values*

The real (MW) and reactive power (MVAR) flow between the hydrocarbon plant and the grid are posted in Figure 7.2. The negative value, means that the flow is from the grid to the hydrocarbon facility system.

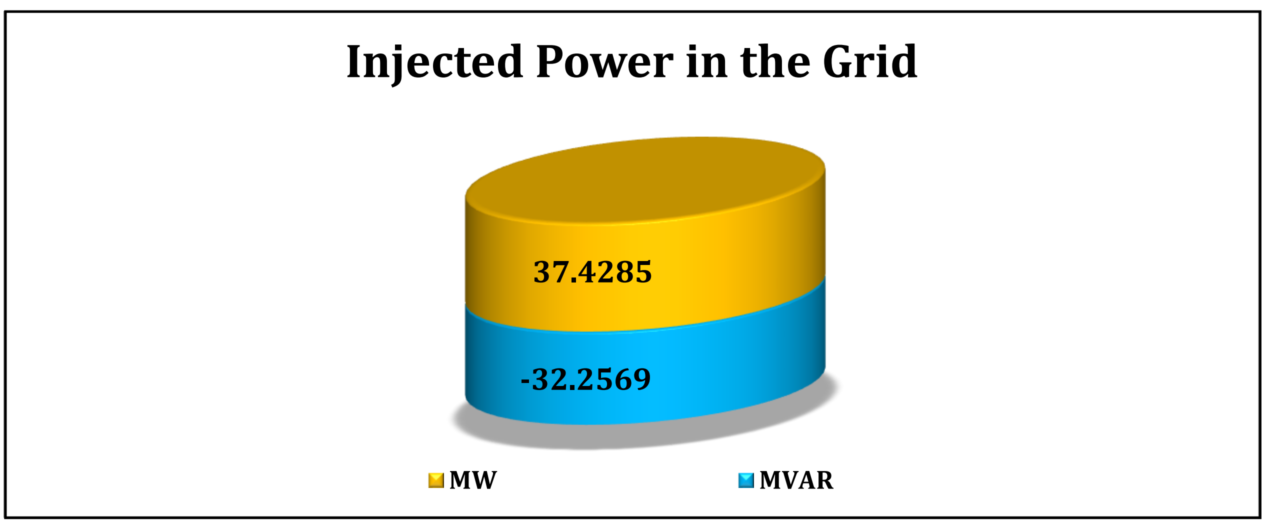

Fig. 7.2: *Power flow between the hydrocarbon facility and the grid*

7.1.2 BAU Generation and Real Power Loss Cost

Based on the Kingdom of Saudi Arabia electrical tariff, the cost of the hydrocarbon plant electrical generation and power loss are posted in Figure 7.3 and 7.4. annually

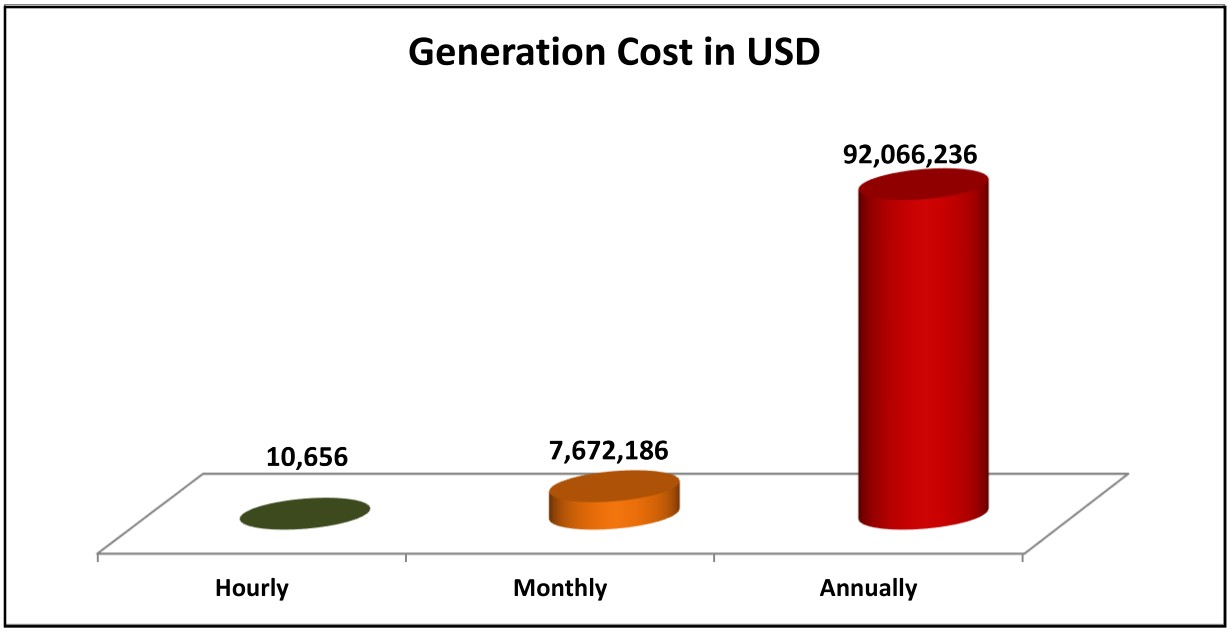

Fig. 7.3: *The hydrocarbon facility generation cost*

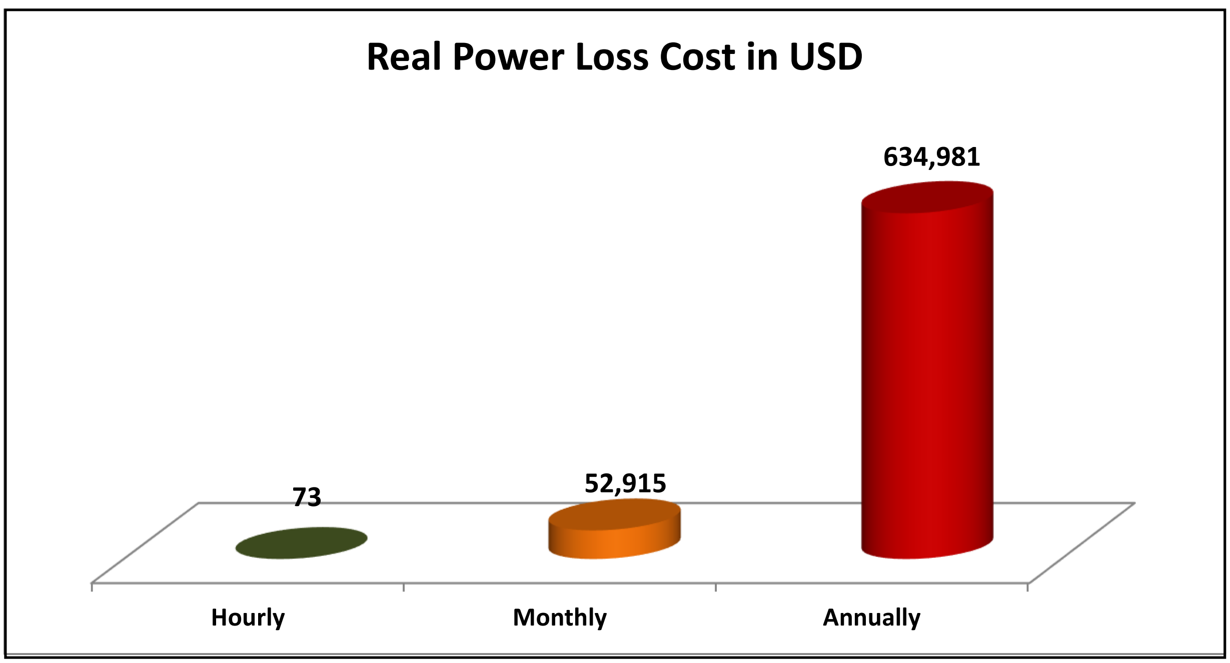

Fig. 7.4: *The hydrocarbon facility real power loss cost*

7.1.3 BAU Revenue Due to Real Power Injection

The revenue due to the real power injected to the grid – sold to the utility company – for the BAU case is posted in Figure 7.5 in hourly, monthly and annually base. This revenue was calculated based on the kingdom of Saudi Arabia electrical tariff.

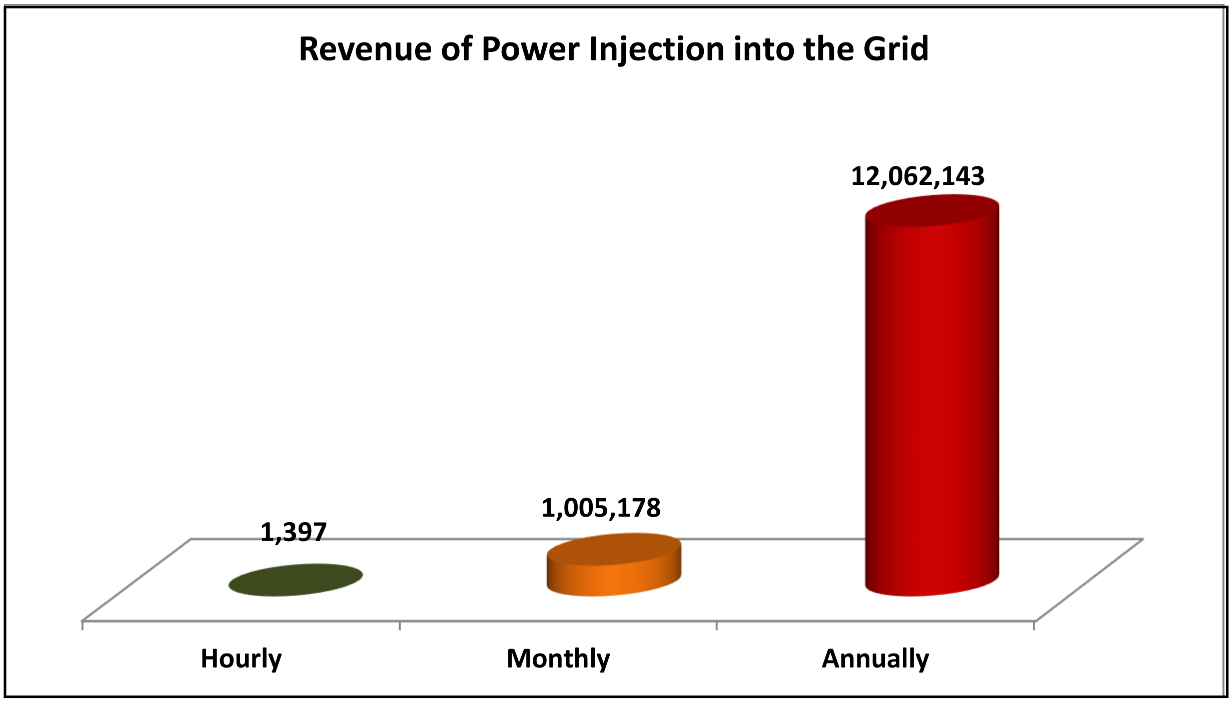

Fig. 7.5: *Revenue of the real power injected into the grid*

7.1.4 BAU Avoided Oil Burning

The real power injected into the grid from the hydrocarbon plant will replace some of the grid MW demand that will come from less efficient oil burning electrical power generation plant if this real power was not injected into the grid. BAU avoided number of oil barrels burning due to real power injection is shown in Figure 7.6.

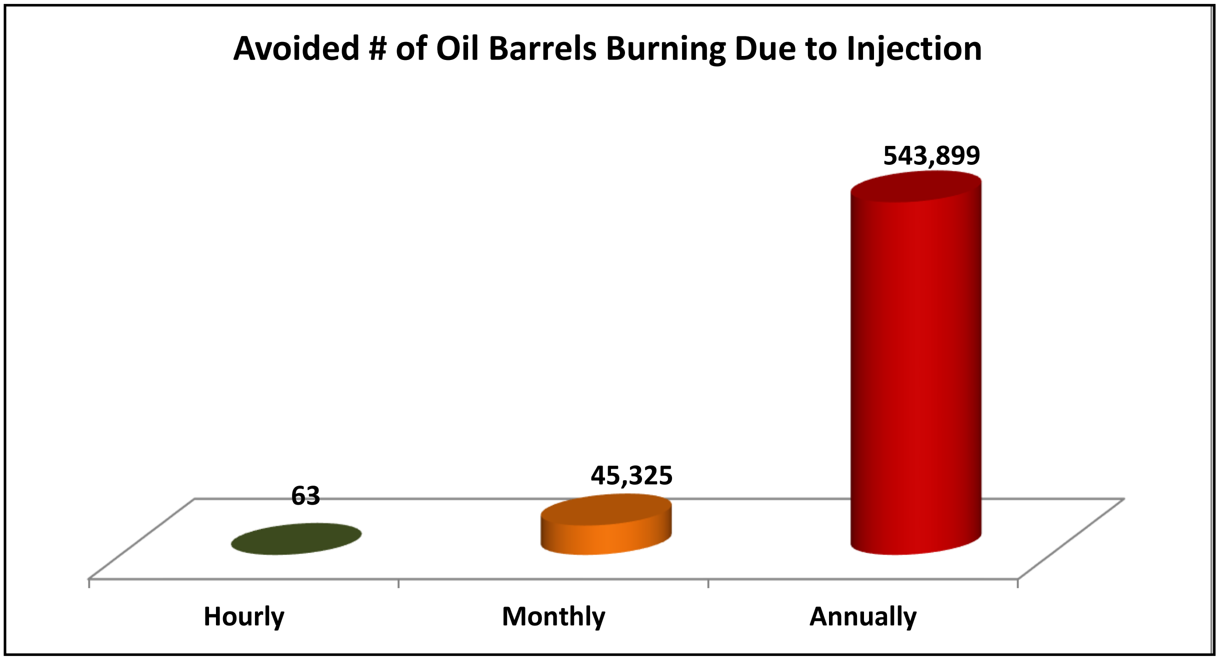

Fig. 7.6: *Avoided number of oil barrels burning due to power injection into the grid*

The cost of the avoided number of oil burning assuming the price of $60/barrels as stated earlier is posted in Figure 7.7 in hourly, mothy and annually base.

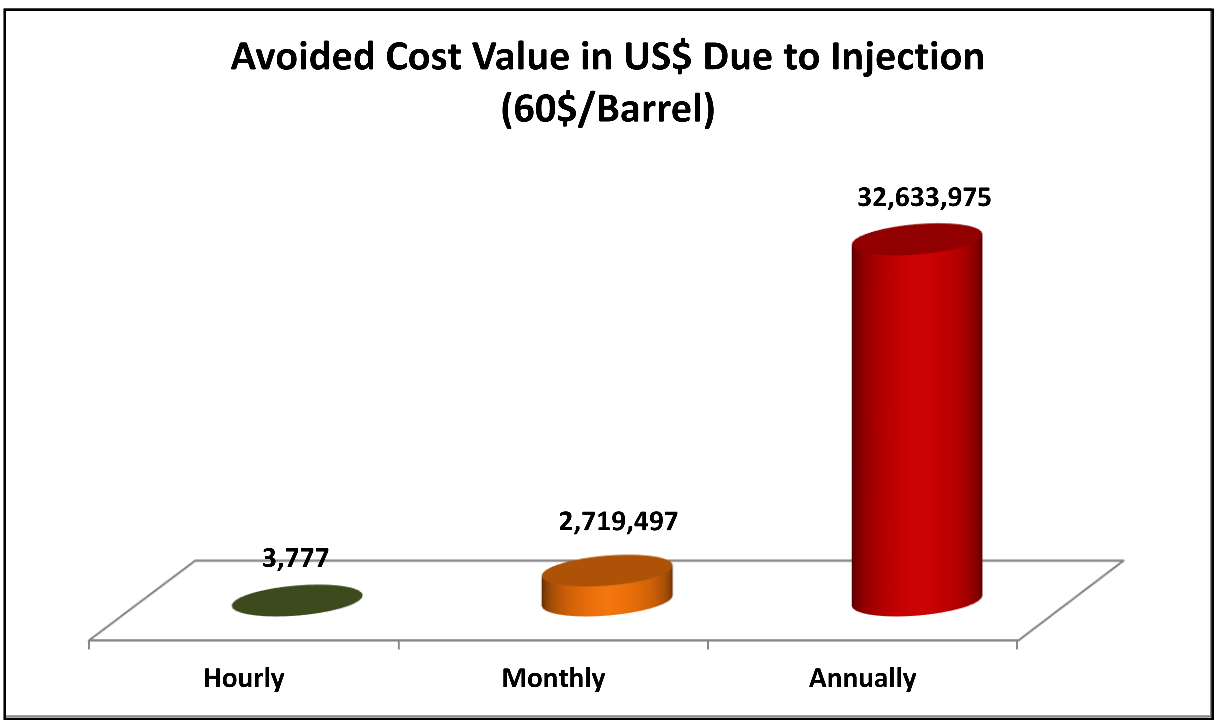

Fig. 7.7: *Cost of the avoided oil barrels burning due to power injection into the grid*

7.1.5 BAU Main Bus Voltages Profile

The BAU main bus voltage profile of the hydrocarbon facility electrical system is shown in Figure 7.8.

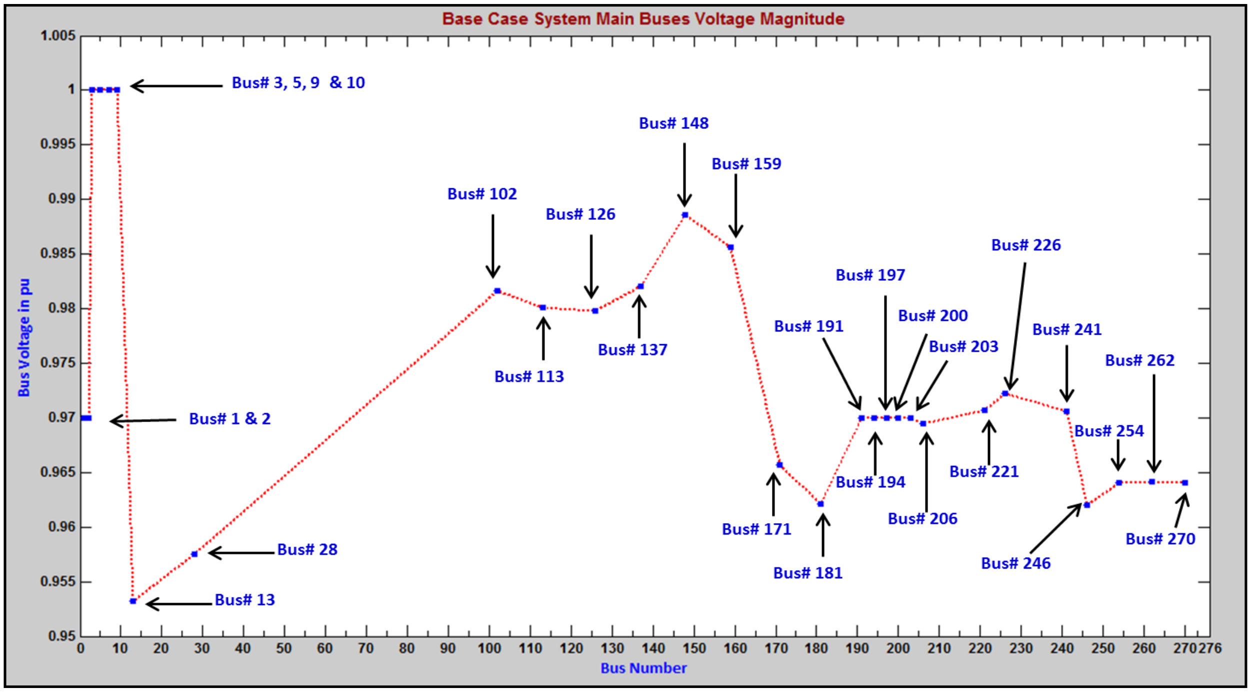

Fig. 7.8: *Hydrocarbon facility BAU case main buses voltage profile*

7.2 Single Objective Real Power Loss (RPL) Optimization Case (Case-1)

In this case, the genetic algorithm (GA) is implemented to optimize the electrical real power loss for the real hydrocarbon industrial plant as single objective problem. The minimization of the real power losses (RPL) objective (J_1) is used to guide the optimization process, consequently the injected real power into the grid (PR_{Inject}) is monitored. The results obtained were benchmarked with the BAU case results.

An initial population of 250 feasible chromosomes which satisfy the system constraints out of 1000 randomly created chromosomes were identified. This initial population was subjected to GA evolution process to identify the best system parameters – chromosomes-that meet the J_1 objective function. The PR_{Inject} was monitored as J_1 evolves. The GA process was set with 90% simple crossover probability and 10% random mutation probability.

In order to optimize the evolution process time the unfeasible transformer tap values-genes- were not selected. In other words, the gene values were limited to certain taps around the neutral taps out of the all taps full range (±16 taps). Table 3 below posts the selected range of the transformer tap values and the percentage of the voltage change for each tap.

Table 7.2: The selected transformer taps feasible gene values for case-1

Description	Upper Tap	Lower Tap
Main Transformers	+8 (0.625%)	-4 (0.625%)
Causeway Main Transformers	+8 (0.625%)	-4 (0.625%)
Captive Motors/Distribution Transformers	+1 (2.5%)	-1 (2.5%)
Generator Step-Up Transformers	+8 (1.25%)	-4 (1.25%)

7.2.1 Evolution of the RPL Objective (J_1)

The evolution of J_1 objective function and PR_{Inject} value over the GA process is captured in Figure 7.9. The RPL objective (J_1) evolved over the 10 generations to 1.964 MW which is 92.04% of the BAU case RPL value of 2.134 MW. The number of generations was selected to be ten (10) after runing many itertaions with different number of generations. This revealed that the ten (10) generations was the optimal number to be used for this case – always converge to the same optimal value when compared with higher geneation number. The PR_{Inject} reached 37.5948 MW over the evolution process which is 0.1663 MW more than the injected power (37.4285MW) in the BAU case.

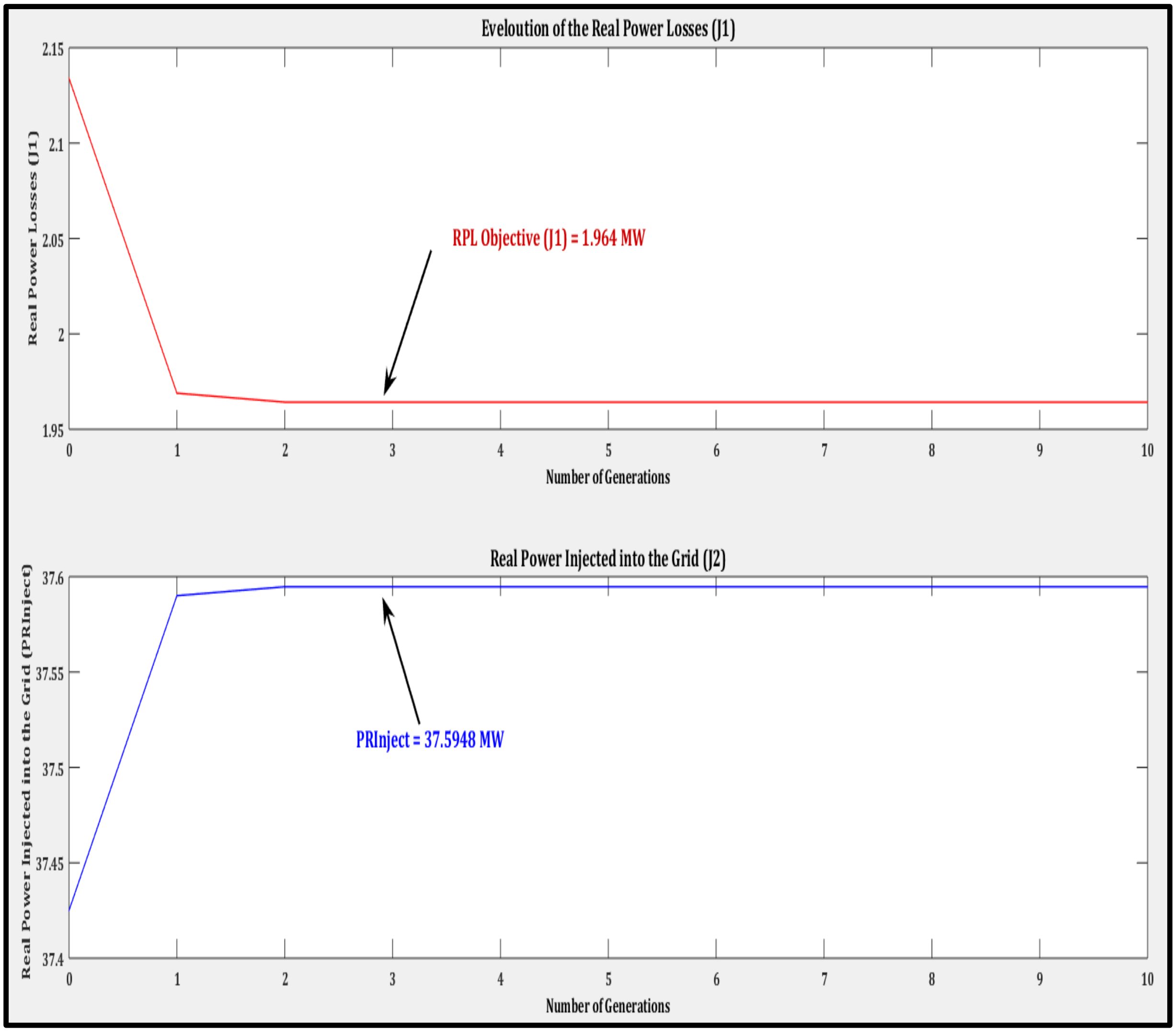

Fig. 7.9: *J_1 and PR_{Inject} Value Convergent for 10 Generations*

7.2.2 Power Loss and Injected Power Vs. BAU Results

The benchmark of the system real power loss and the injected power into the grid with the BAU results is demonstrated in Figure 7.10. There is 0.17 MW reduction in the system loss and 0.166 MW increase in the power injection to the grid when comparing this case with the BAU case.

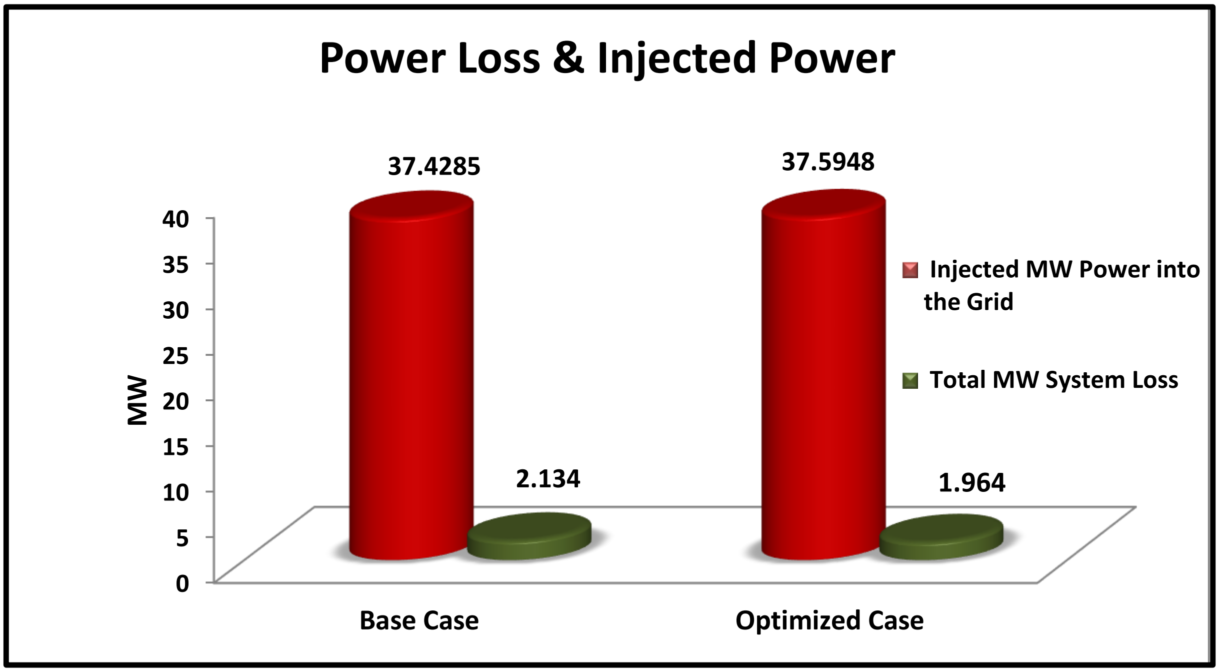

Fig. 7.10: J_1 and PR_{Inject} values - case 1- benchmarked with BAU case values

7.2.3 Comparison of Voltage Profile of Main Buses

The system at the optimal case demonstrates an improvement in the system buses p.u. voltage profile which increases the robustness of the system. Refer to Figure 7.11.

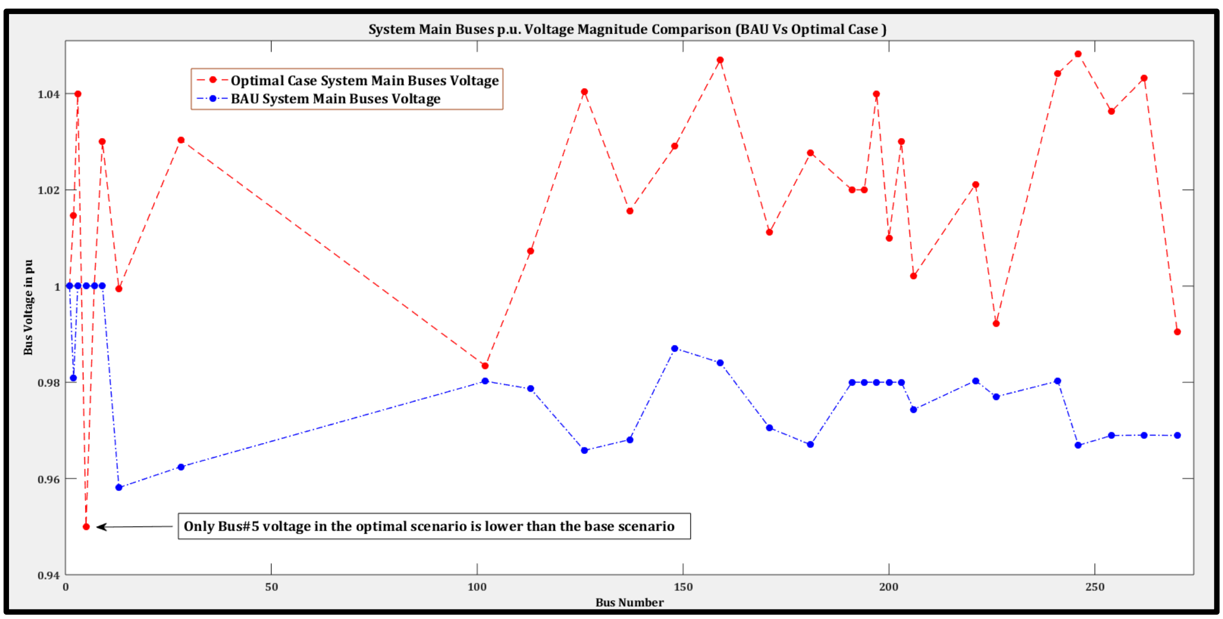

Fig. 7.11: The main system buses voltage profile - case 1 - versus BAU

7.2.4 Economic Benchmark Vs. BAU Results

The avoided cost due to the optimization of the system real power loss (RPL)- case 1- is demonstrated in Figure 7.12 at daily, monthly and annual bases. The annual cost avoidance is around $49,554/year.

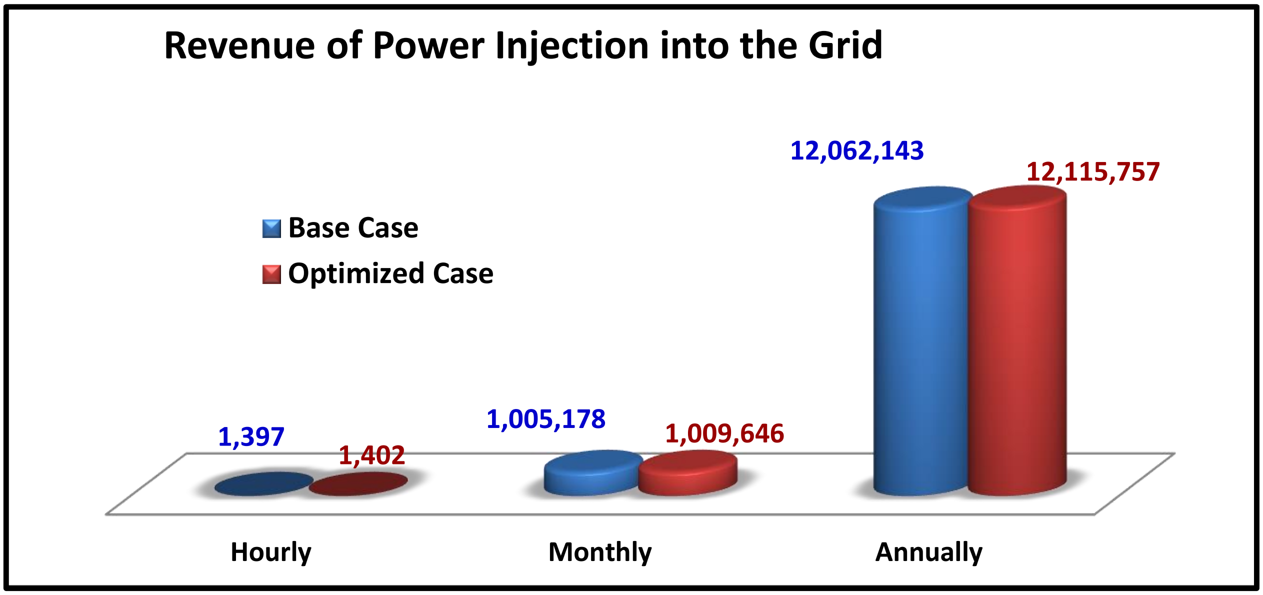

Fig. 7.12: *The RPL cost - case 1 - versus BAU RPL cost*

The revenue due to the power injection into the grid at both BAU case and case-1 is shown in Figure 7.13. The figure illustrates the potential of case-1 – RPL (J_1) optimization -in increasing the revenue by $53,614 per year.

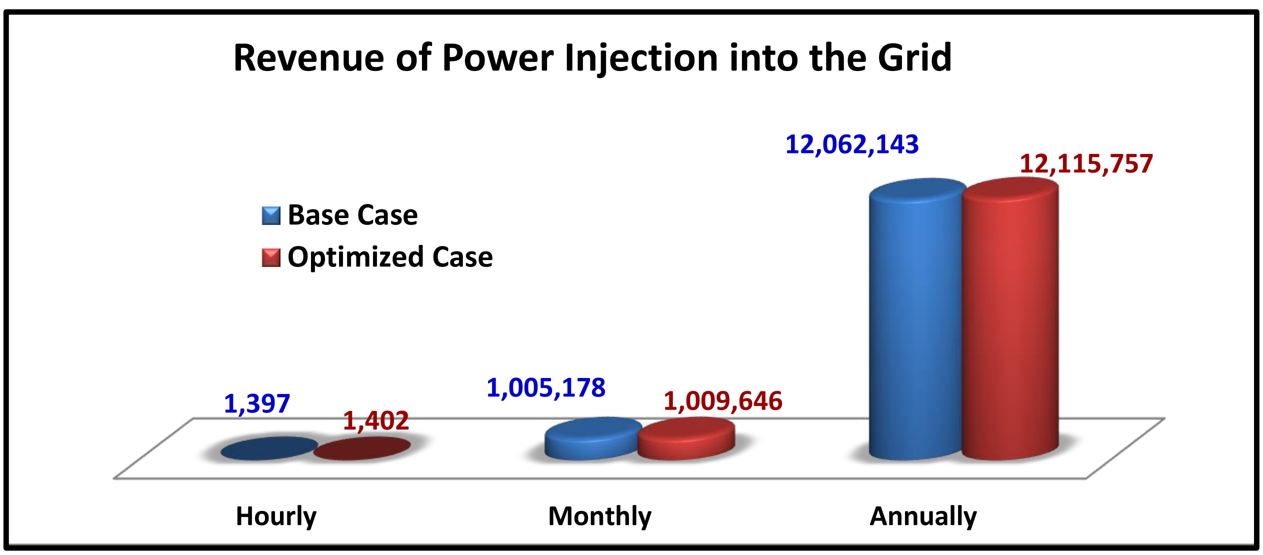

Fig. 7.13: *The RP_{Inject} revenue - case 1- versus BAU values*

The cost of the avoided oil barrels burning due to power injection into the grid when case 1 is benchmarked with the BAU values is posted in Figure 7.14. The annual cost avoidance increase is $145,053/year.

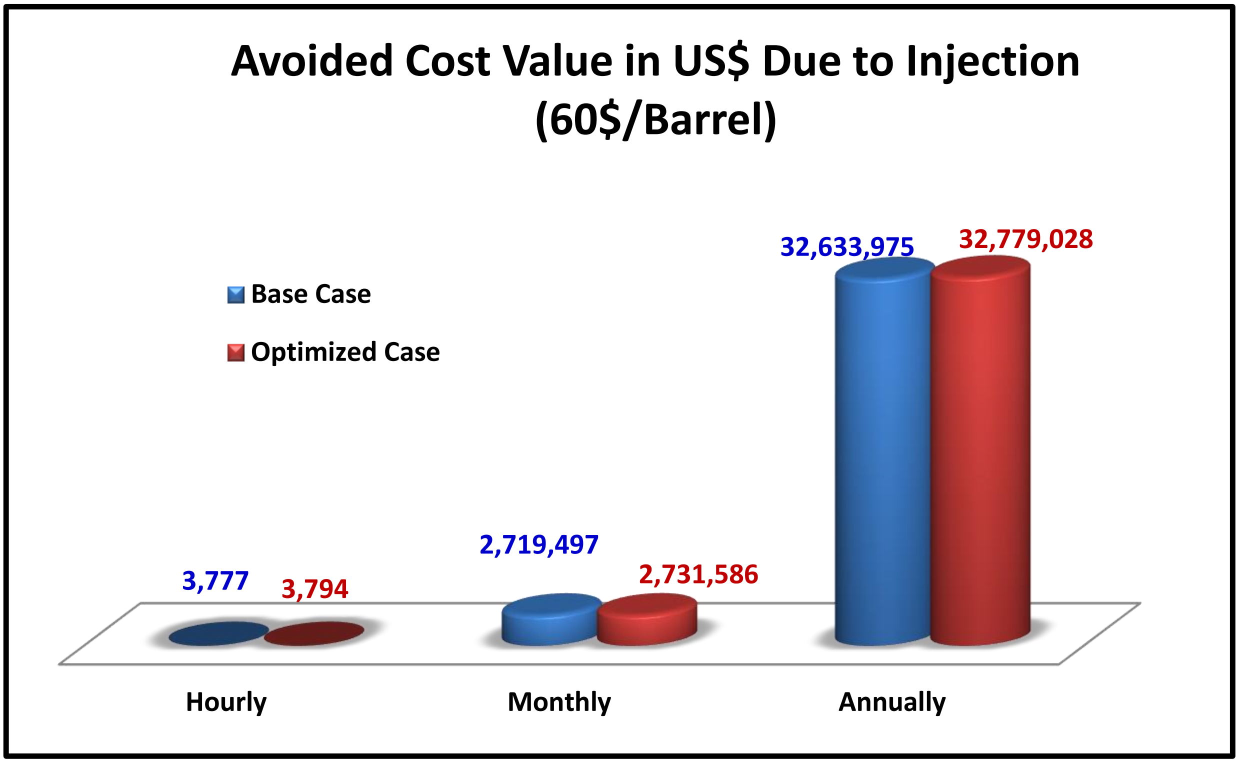

Fig. 7.14: *The avoided oil barrels cost - case 1- versus BAU values*

7.3 Power Loss with VAR Injection Optimization Case (Case-2)

In this case, the GA is considered for assessing the effect of MVAR power injection via shunt capacitors in optimizing the electrical power loss for the real hydrocarbon industrial plant. The minimization of real power losses (RPL) objective (J_1) is used to guide the optimization process, and, consequently, the injected power in the grid is monitored, refer to equation (4.10). First, the optimal locations of MVAR injection will be identified using a voltage stability index, refer to equation (4.24). The potential of power loss optimization with and without MVAR injection versus the BAU case will be discussed in the results. The results obtained demonstrate the potential and effectiveness of the proposed approach to optimize the power consumption in both scenarios (with and without MVAR injection). Also, in this case study a cost appraisal for the potential daily, monthly and annual cost saving in both scenarios is addressed.

To optimize the evolution process time, the unfeasible transformer tap values (genes) were not selected- similar to case 1. In other words, the genes values were limited to certain taps around the neutral taps out of the all taps full range (±16 taps). Table 7.3 below posts the selected range of the transformer tap values and the percentage of the voltage change for each tap.

Table 7.3: The selected transformer taps feasible genes values for case-2

Description	Upper Tap	Lower Tap
Main Transformers	+8 (0.625%)	-4 (0.625%)
Causeway Main Transformers	+8 (0.625%)	-3 (0.625%)
Captive Motors/Distribution Transformers	+1 (2.5%)	-1 (2.5%)
Generator Step-up Transformers	+5 (1.25%)	-4 (1.25%)

7.3.1 Power Loss without VAR Injection Optimization Scenario

An initial populations with 200 feasible chromosomes (individuals) which meet both the bus voltage and synchronous machine reactive power constraints was created. These feasible solutions are associated with case 2 first optimal scenario - without MVAR injection. The feasible solutions with their associated chromosomes were subjected to the GA evolutionary process of 20 generations guided by the objective function RPL (J_1). The PR_{Inject} was monitored as J_1 evolved. The GA process was set with 90% simple crossover probability and 10% random mutation probability. The system parameters and the objective function value associated with the optimal solution of this scenario were identified.

7.3.2 Power Loss with VAR Injection Optimization Scenario

The evolutionary process was optimized via the same method employed in the first scenario. Another initial 300 feasible individuals were identified for the second optimal scenario- with MVAR injection. In this scenario, MVAR shunt capacitors are connected to the preselected buses as posted in Table 7.4. The preselected buses for shunt capacitor connection was identified using voltage stability index (J_3). In this case, the MVAR chromosomes are extended to include MVAR injection considered as control variables. The feasible populations with their associated chromosomes were subjected to 20 generation of GA evolutionary process. The crossover and mutation probability were set equal to those in the scenario without MVAR injection- 90% and 10% - respectively.

Table 7.4: The selected buses for MVAR injection based on L-Index (J_3) value

Substation Number	Bus Number	Potential MVAR
Substation#2	13 and 28	[8 8.5 9 9.5 10]
Causeway Substation#1	102 and 113	[1.5 2 2.5 3 3.5]
Causeway Substation#2	124 and 137	[2 2.5 3]
Causeway Substation#3	148 and159	[2 2.5 3]

7.3.3 Evolution of the RPL Objective (J_1)

The evolution of the objective function (J_1) and PR_{Inject} values over the GA process is captured in Figure 7.15. As shown, the MVAR injection senario produces better power loss reduction value over the evolution process.

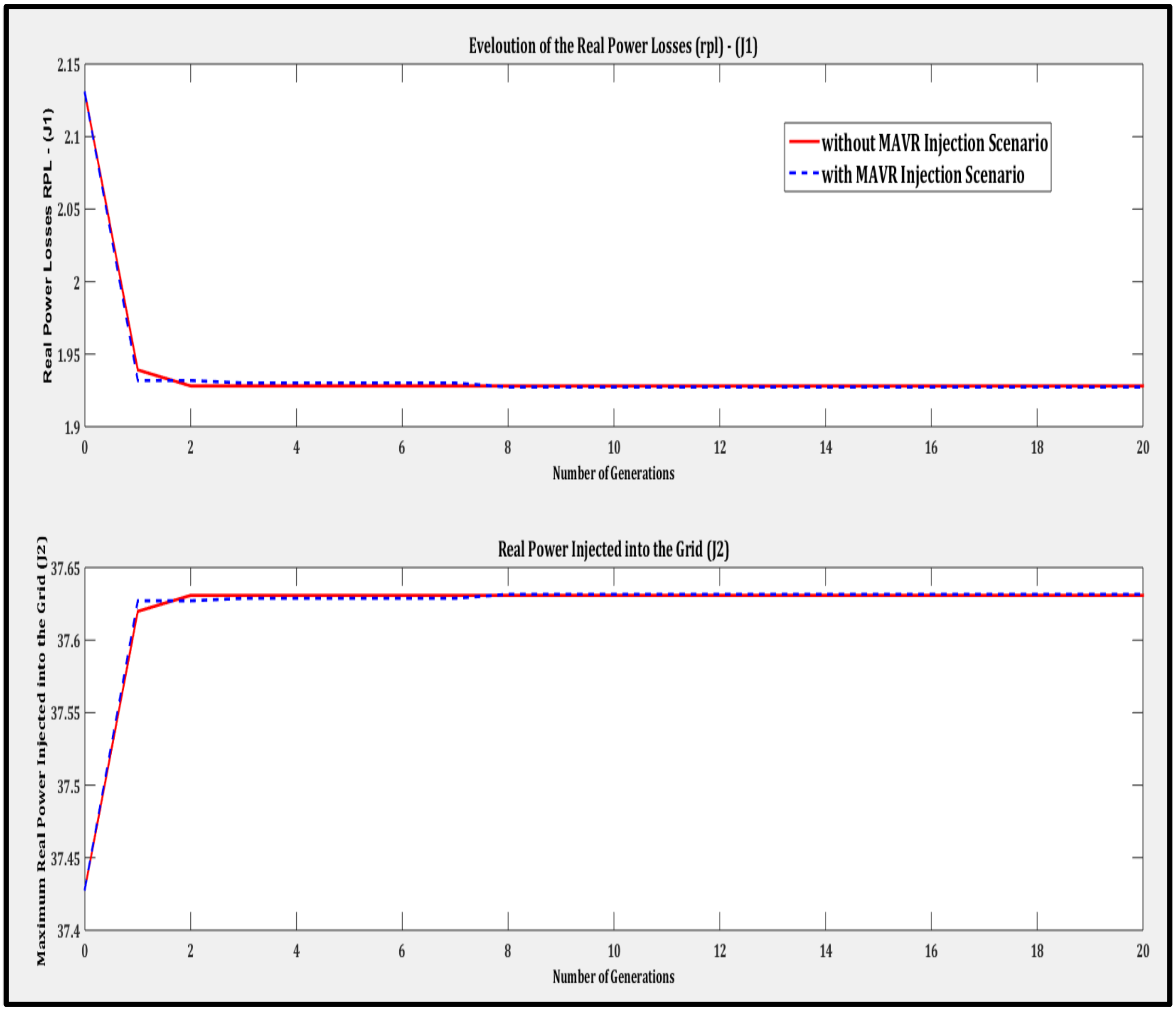

Fig. 7.15: *J_1 and PR_{Inject} value evolution over 20 generations.*

7.3.4 Power Loss and Injected Power Vs. BAU Results

The benchmark of case 2 optimal system real power loss and the injected power in the grid with the BAU case is demonstrated in Figure 7.16. There is 0.202 MW reduction in the system loss between the BAU case and the no MVAR injection scenario. A 0.203 MW power loss reduction is achieved when comparing the BAU case with the MVAR injection optimal scenario. The same amount of MW were injected in the grid for both scenarios.

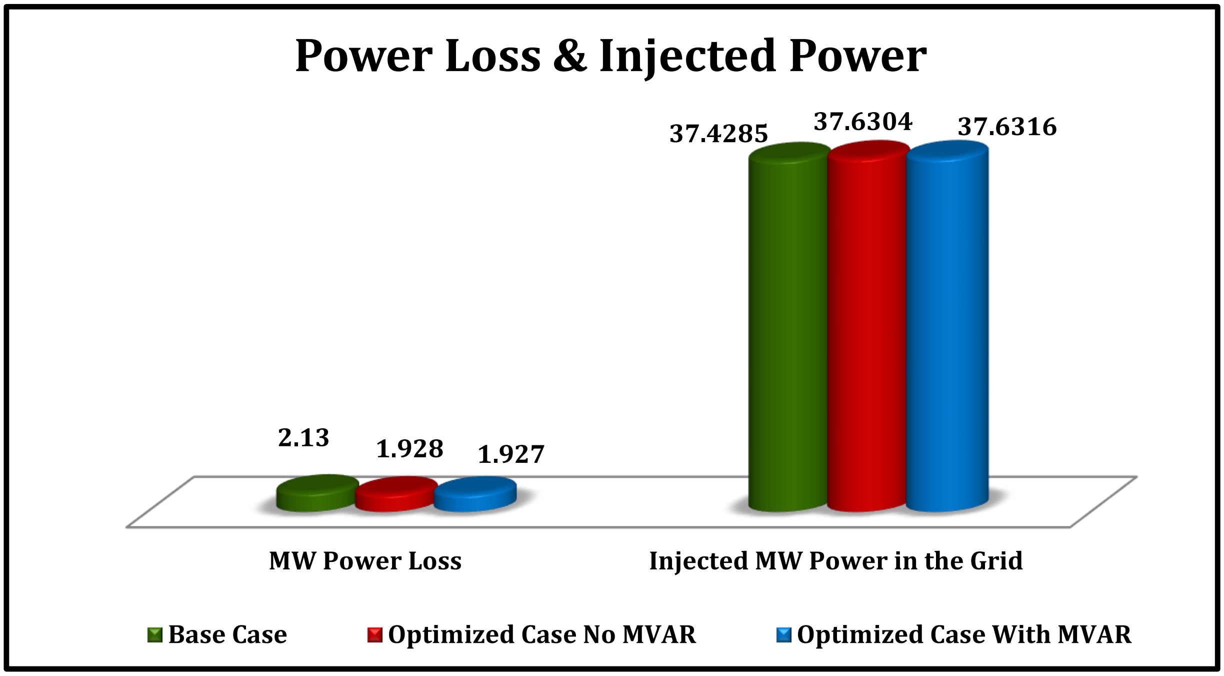

Fig. 7.16: *J_1 and PR_{Inject} values benchmarked with BAU case values*

Figure 7.17 is capturing the system main buses voltage profile for the BAU case and the optimal case-2, two scenarios. The MVAR injection scenario shows better improvement of the buses voltage profile when compared with BAU.

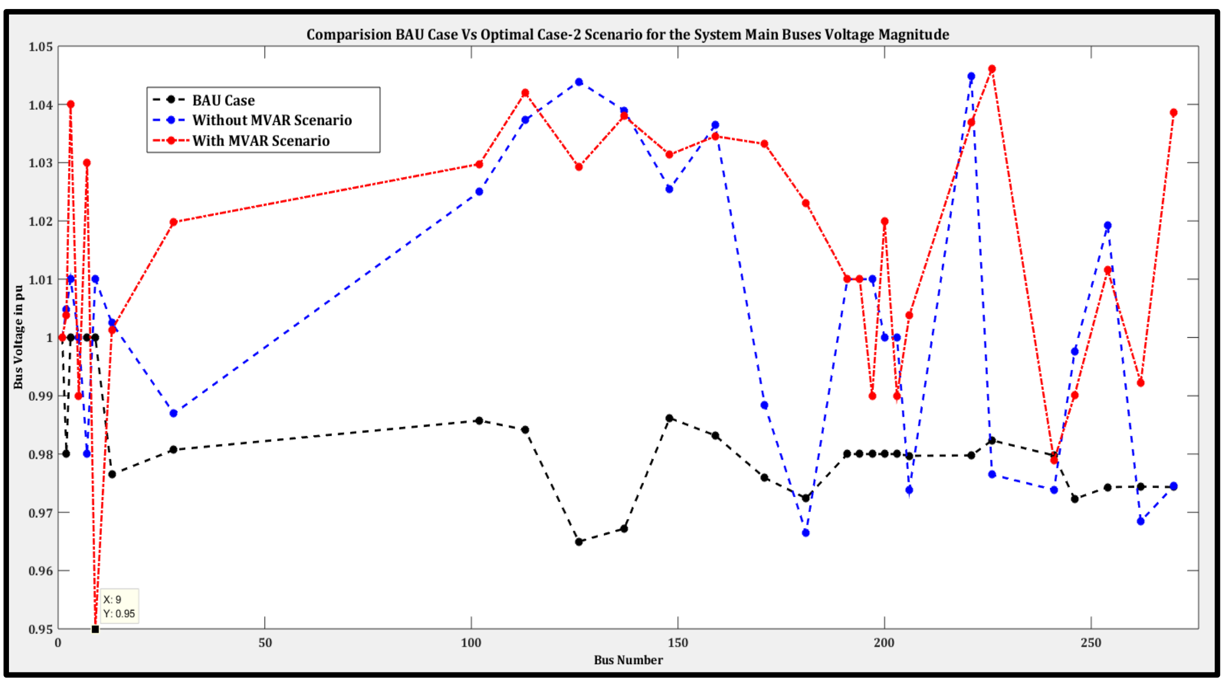

Fig. 7.17: *Case-2 two scenarios main system buses voltage profile versus BAU benchmark*

7.3.5 Economic Benchmark Vs. BAU Results

The avoided cost due to the optimization of the system real power loss – RPL (J_1) - for case 2 two scenarios when compared with the BAU case is demonstrated in Figure 7.18 at daily, monthly and annual bases. The annual cost avoidance based on natural gas cost of $3.5 per MMscf is around $60,300/year and $60,500/year for the no MVAR and with MVAR optimal scenarios respectively when compared to the base case (BAU).

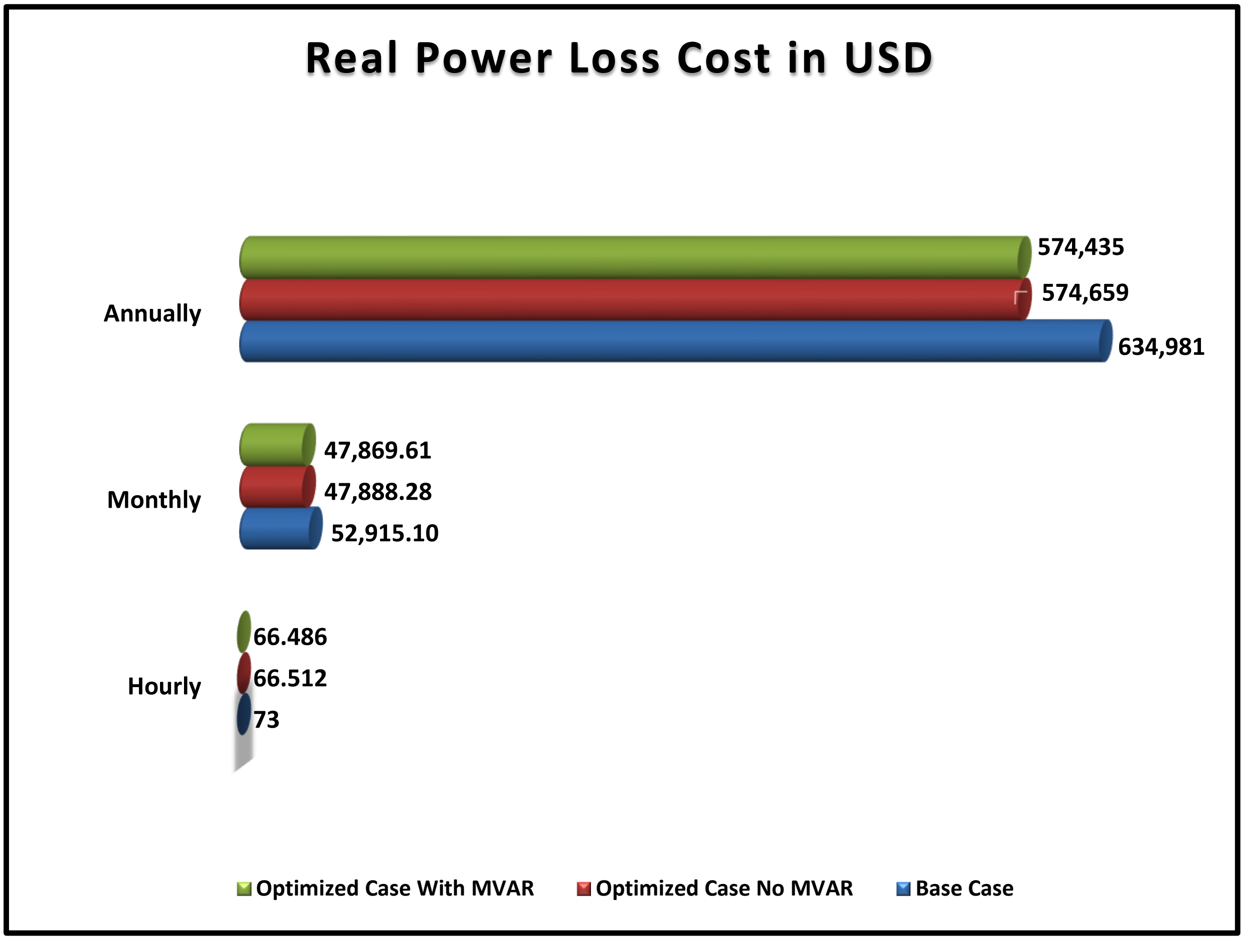

Fig. 7.18: *Case – 2 RPL cost versus BAU RPL cost*

The revenue due to the power injection in the grid for case 2 scenarios is shown in Figure 7.19. The figure illustrates the potential of the optimal scenarios in increasing the revenue using $37.3 MWh tariff rate- $65,200/year for the no MVAR injection optimal scenario and $65,500/year for the with MVAR injection optimal scenario benchmarked to the BAU case.

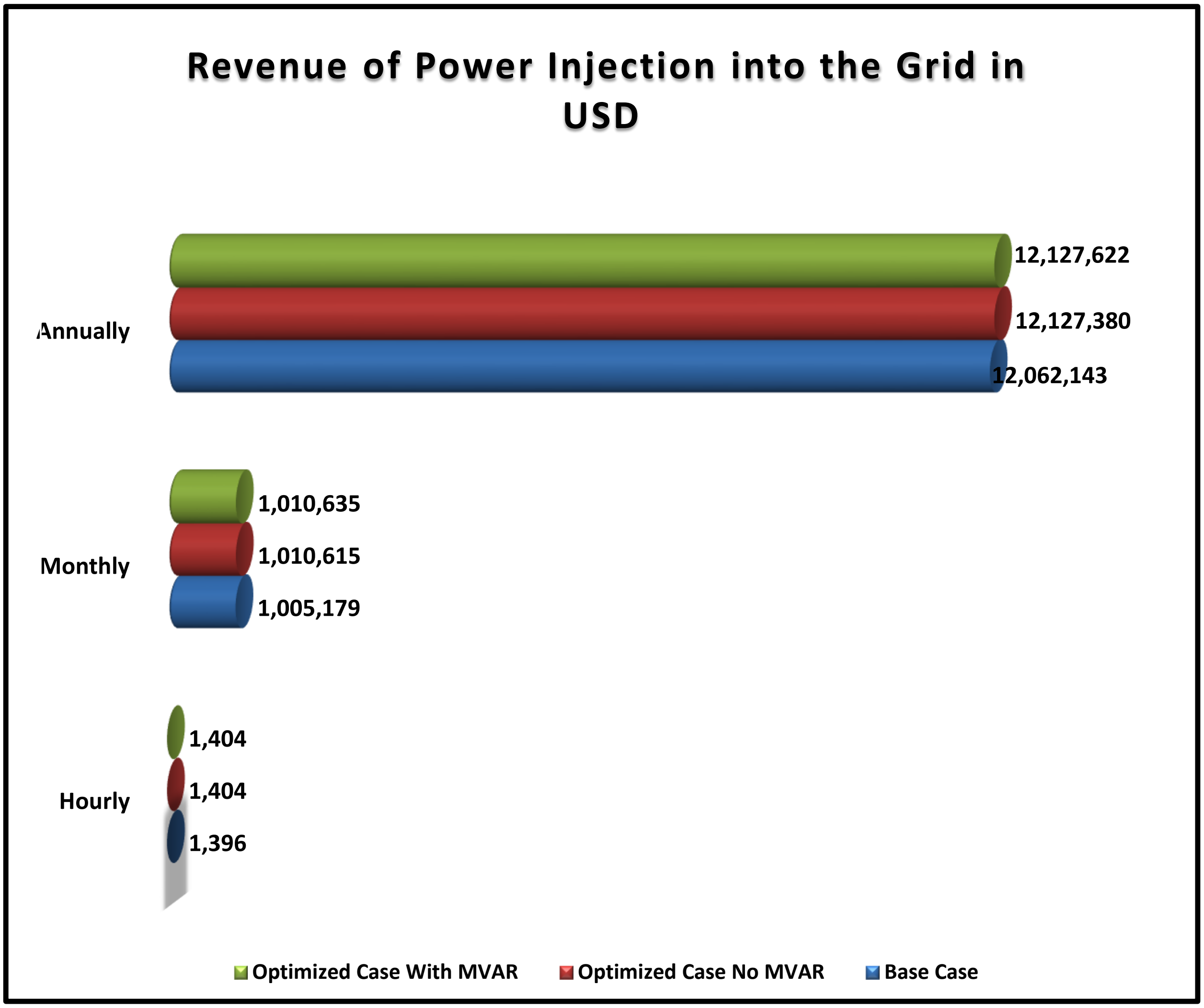

Fig. 7.19: *Revenue due to power injection in the grid*

The cost of the avoided oil barrels burning due to power injection in the grid when benchmarked with the BAU values is posted in Figure 7.20. The annual cost avoidance increase is $176,496/year for the no MVAR injection scenario and $178,062/year for the MVAR injection scenario.

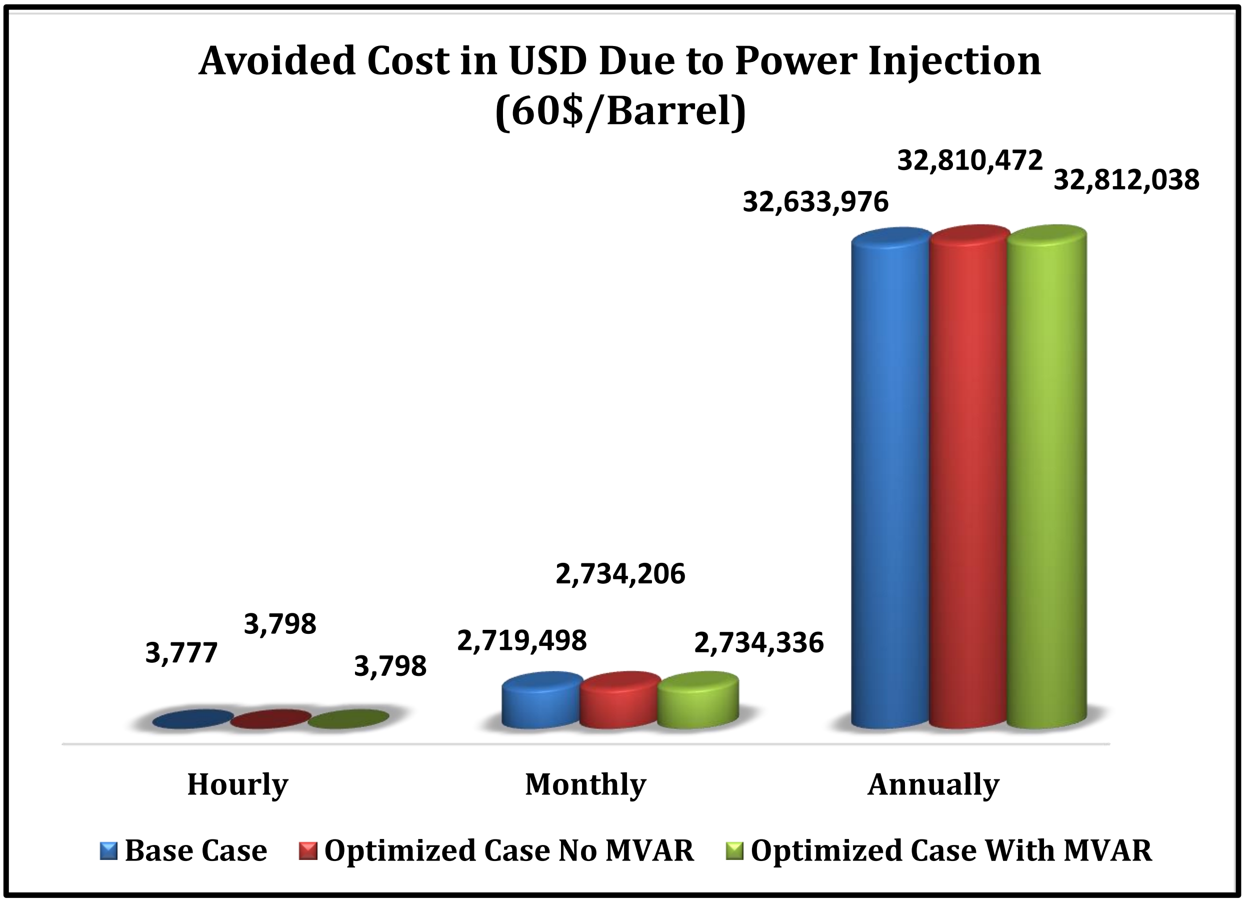

Fig. 7.20: *The avoided oil barrels cost versus BAU value*

7.4 Impact of Genetic Algorithm Operators on Optimization Case (Case-3)

In this case study, different selection, crossover and mutation techniques are considered for optimizing the electrical power loss in real hydrocarbon industrial plant using genetic algorithm (GA). The minimization of real power losses (RPL) objective (J_1) is used to guide the optimization process. Eight GA selection, crossover and mutation techniques combination cases are simulated for optimizing the system real power loss. The potential of power loss optimization for each case versus the BAU case is discussed in the results. The results obtained demonstrate the potential and effectiveness of the proposed techniques combination cases in optimizing the power consumption. Also, in this case study, a cost assessment for the potential daily, monthly and annual cost saving associated with the power loss optimization for each case is addressed.

The effectiveness of different GA selection, crossover and mutation techniques combination in optimizing the power system loss was addressed in literature when applied to standard virtual IEEE electrical system. None of the previous studies addressed the effectiveness of these different techniques combination in optimizing the power loss of real hydrocarbon facility- small system footprint, shorter lines, large machines, combined cycle generation configuration with large load [23, 63].

As in the previous cases, to optimize the elevation process time, the unfeasible transformers tap values (genes) were not selected. Refer to Table 7.3 which posts the selected range of the transformers tap values and the percentage of the voltage change for each tap.

Initial 200 feasible chromosomes (individuals) which meet both the equity and inequality constraints were identified. The feasible solutions with their associated chromosomes were subjected to the GA evolutionary process for 20 generations guided by the objective function J_1. The system parameters and the objective function value associated with the optimal solution were identified subject to eight GA operator combination cases as shown in Table 7.5.

Table 7.5: The GA operators' combination cases

Selection	Crossover	Mutation	Abbreviation
Random	Differential	Random	R-D-R
Random	Differential	Non-Uniform	R-D-UN
Random	Simple	Random	R-S-R
Random	Simple	Non-Uniform	R-S-UN
Tournament	Differential	Random	T- D -R
Tournament	Differential	Non-Uniform	T-D-UN
Tournament	Simple	Random	T-S-R
Tournament	Simple	Non-Uniform	T-S-UN

7.4.1 Evolution of the RPL Objective (J_1)

The evolution of the objective function (J_1) value over the GA process for the different GA operator combinations is captured in Figure 7.21. In reference to Table 7.5, the tournament selection, simple crossover and random mutation GA operators' combination (T-S-R) produced the best optimized value of J_1 (1.905 MW), this is 0.225 MW reduction in real power loss when compare with BAU value. The Random selection, differential crossover and random mutation (R-D-R) GA operator mix produced the second best J_1 value (1.923 MW).

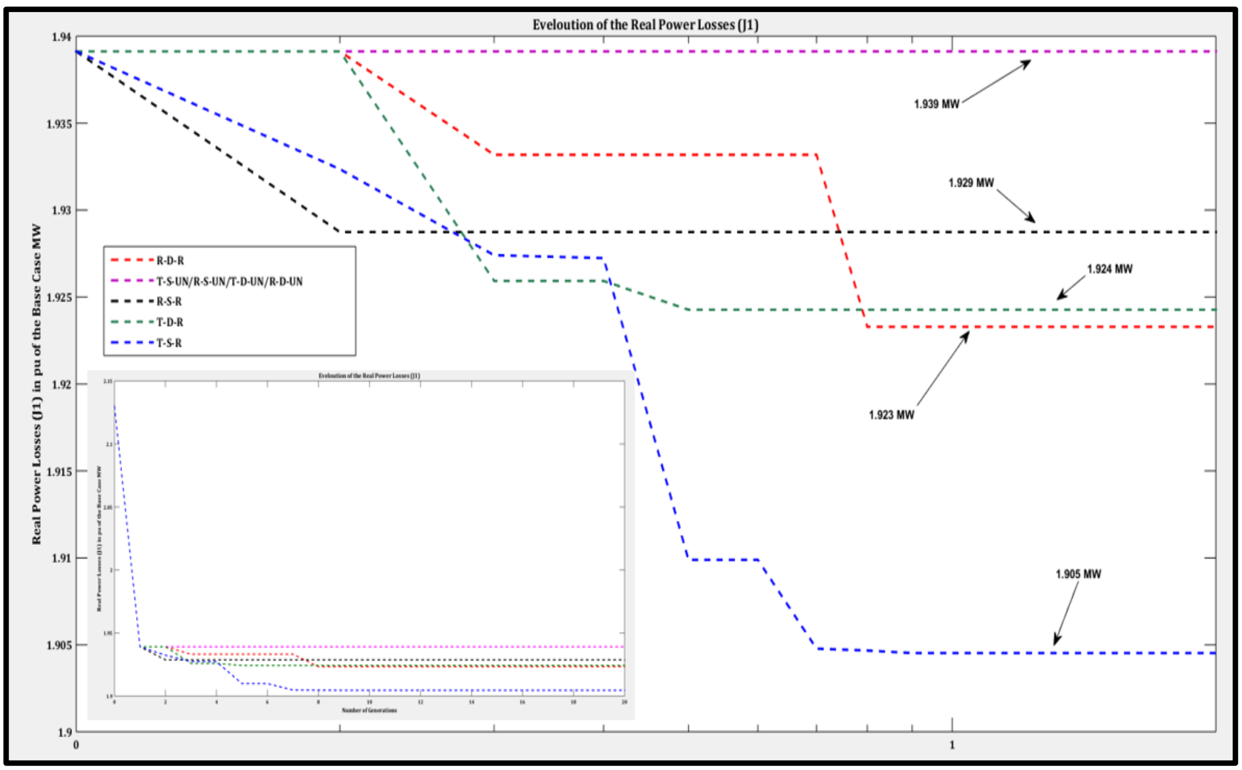

Fig. 7.21: *J_1 and PR_{Inject} value evolution over 20 generations.*

7.4.2 Economic Benchmark Vs. BAU Results

The annual avoided cost due to the optimization of the system power loss is demonstrated in Figure 7.22. The annual cost avoidance based on natural gas cost of \$3.5 per MMscf is around $67,346/year for the T-S-R combination scenario when compared with the BAU case. The R-DE-R combination scenario comes second with $61,755/year.

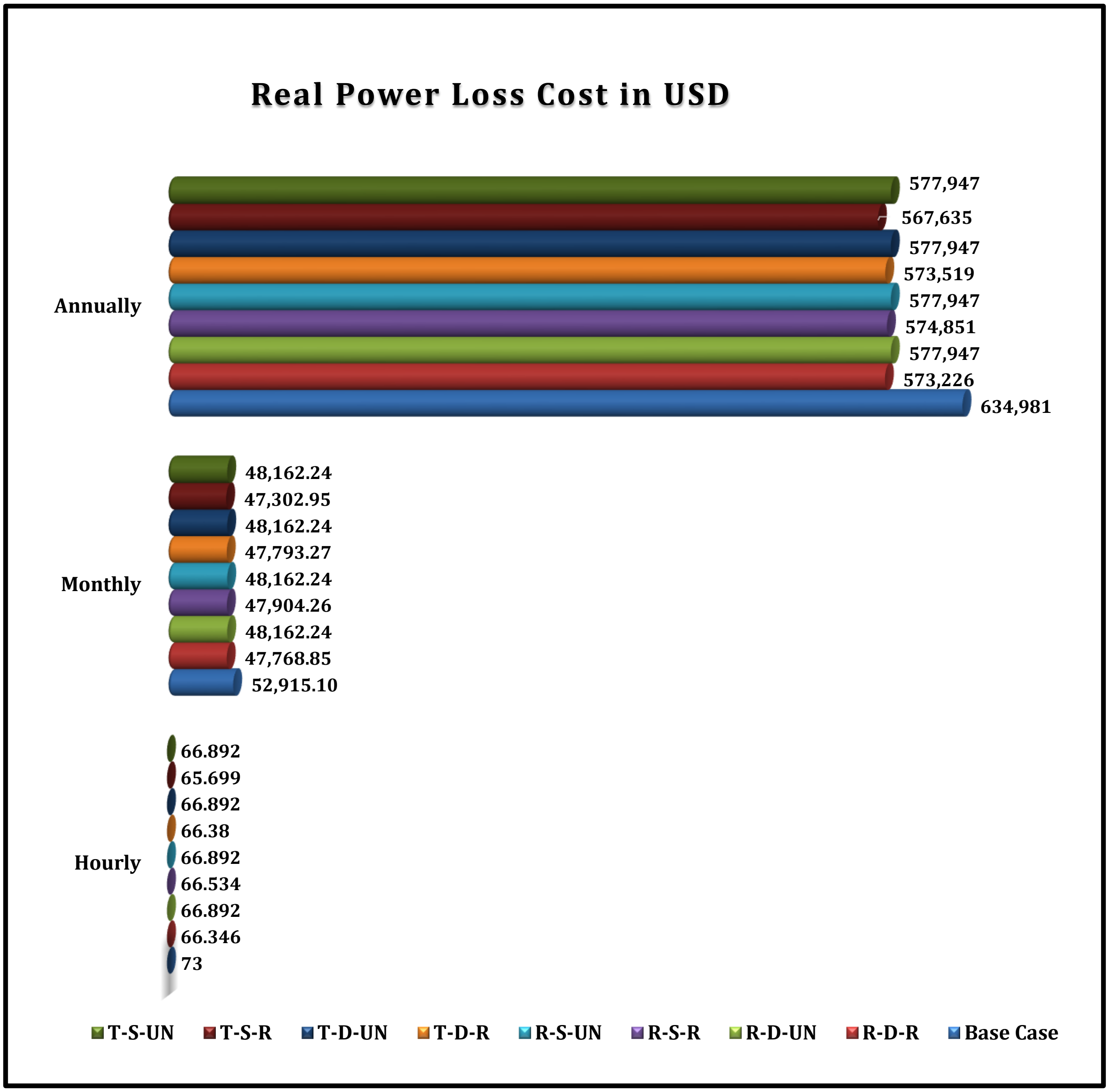

Fig. 7.22: *The RPL cost versus BAU RPL cost*

As shown in Figure 7.23, the annual revenue for real power injection is $12,134,986 annually for the T-S-R combination scenario when compared with the BAU case - $72,843 increase. The R-D-R combination scenario comes second with annual revenue of $12,128,966.

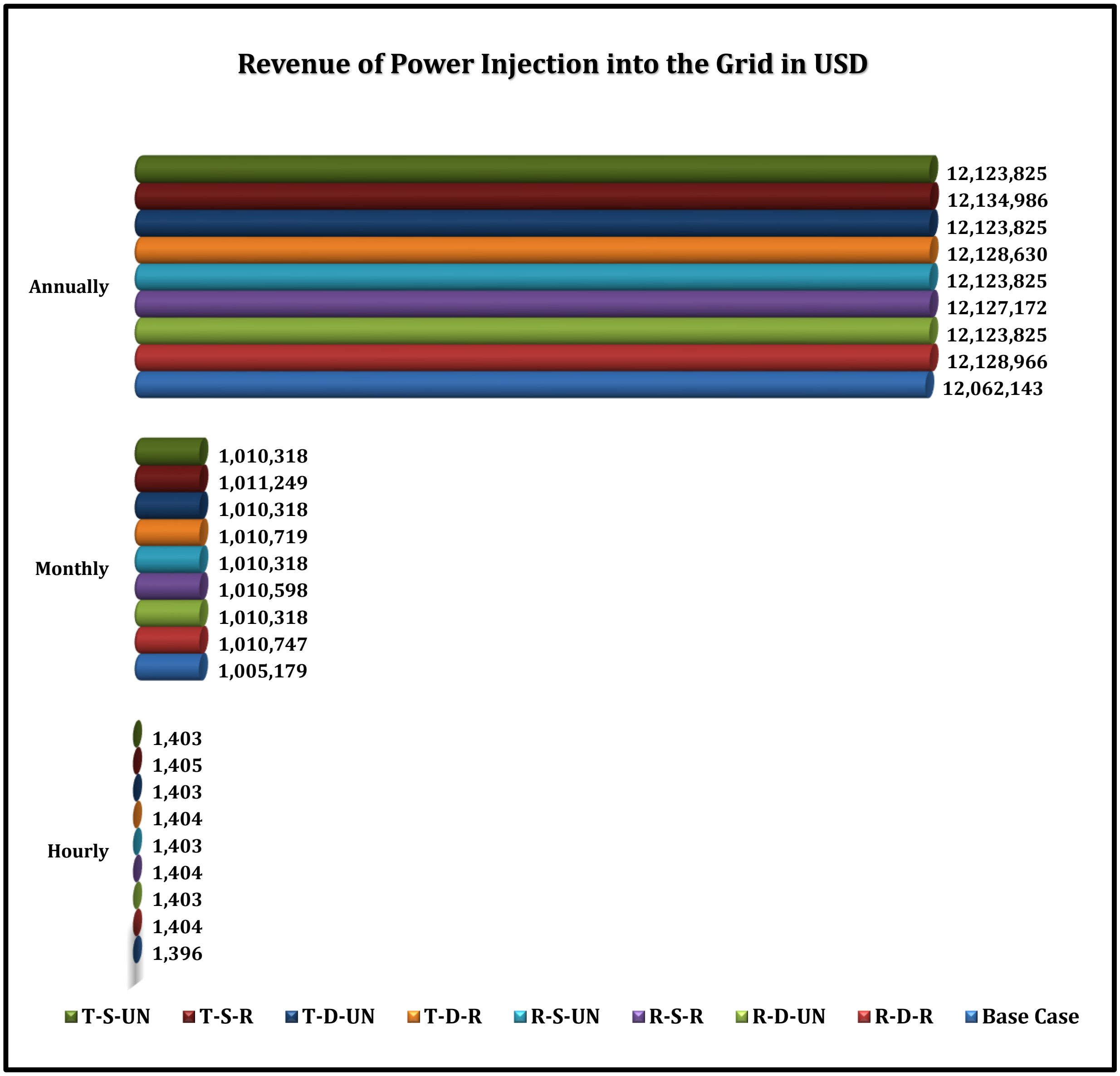

Fig. 7.23: *Revenue due to power injection in the grid*

7.5 Single and Multi-Objective Optimization Case (Case-4)

This case study presents a techno-economic assessment of the real life hydrocarbon facility electrical system real power loss (RPL) objective (J_1) and grid connection power factor (GCPF) objective (J_2) optimization. The problem was articulated as a single and multiobjective problem. This optimization was attained by using the Genetic Algorithm (GA), the Improved Strength Pareto

Evolutionary Algorithm (SPEA2) and the Differential Evolutionary Algorithm (DEA). The fuzzy set theory was implemented to extract the best compromise value out of the optimal objective values in the Pareto optimal set. The performance of GA, SPEA2 and the DEA were benchmarked in the course of optimizing the single objective of real power loss (J_1) reduction and the two competing objectives - real power loss (J_1) and grid connection power factor (J_2). The consequences obtained from this case study show the efficiency and prospects of the proposed algorithms in solving the described multiple objectives of the study case. In this case, the by-product, in terms of annual cost avoidance, due to the study objectives' optimization, based on real fuel value is presented.

In case-4, two generation modes are applied- all generators are online and two generators are offline. The two generators that are offline are one gas turbine generator (GTG#2), and one steam turbine generator (STG#2). Aligned with the generation modes, three different study subcases have been considered. The results obtained for each subcase is benchmarked against the BAU case and each other cases. The benchmarking illustrates the effect of the different optimization algorithms in addressing the objective functions.

7.5.1 The Study Sub-Cases

In this study case (case-4), there are three subcases, the BAU (subcase-1), single objective subcase (subcase-2) and two objectives subcase (subcase-3). In the single objective subcase, GA and DEA are implemented to optimize the system real power loss only (J_1) giving the two generation modes. In the two multiobjective subcase, SPEA2 and DEA are implemented to optimize the system real power loss (J_1) and the grid connection power factor (J_2) simultaneously for the two aforementioned generation modes.

In the single objective subcase (subcase-2), the crossover rate was fixed at 90%, the mutation rate was fixed at 10%, the population size was fixed to be 100 individuals and the number of generations was fixed to be 100. In the multiobjective subcase (subcase-3), the crossover rate, mutation rate, population size and the number of generations when considering the SPEA2 are maintained at the same rate of the single objective subcase. The mutation rate only was changed to 50% when considering DEA. The front Pareto optimal set was fixed to be 50 individuals.

7.5.2 The Evolution of the RPL objective (J_1)-subcase-2

The evolution of the RPL objective (J_1) in the single objective subcase (subcase-2) is posted in Figure 7.24. DEA evolves to better J_1 value when compared with the GA over the generation. Yet, for all the generation online mode DEA took longer time – generation number 78 – prior to converge to the optimal J_1 value when compared with GA.

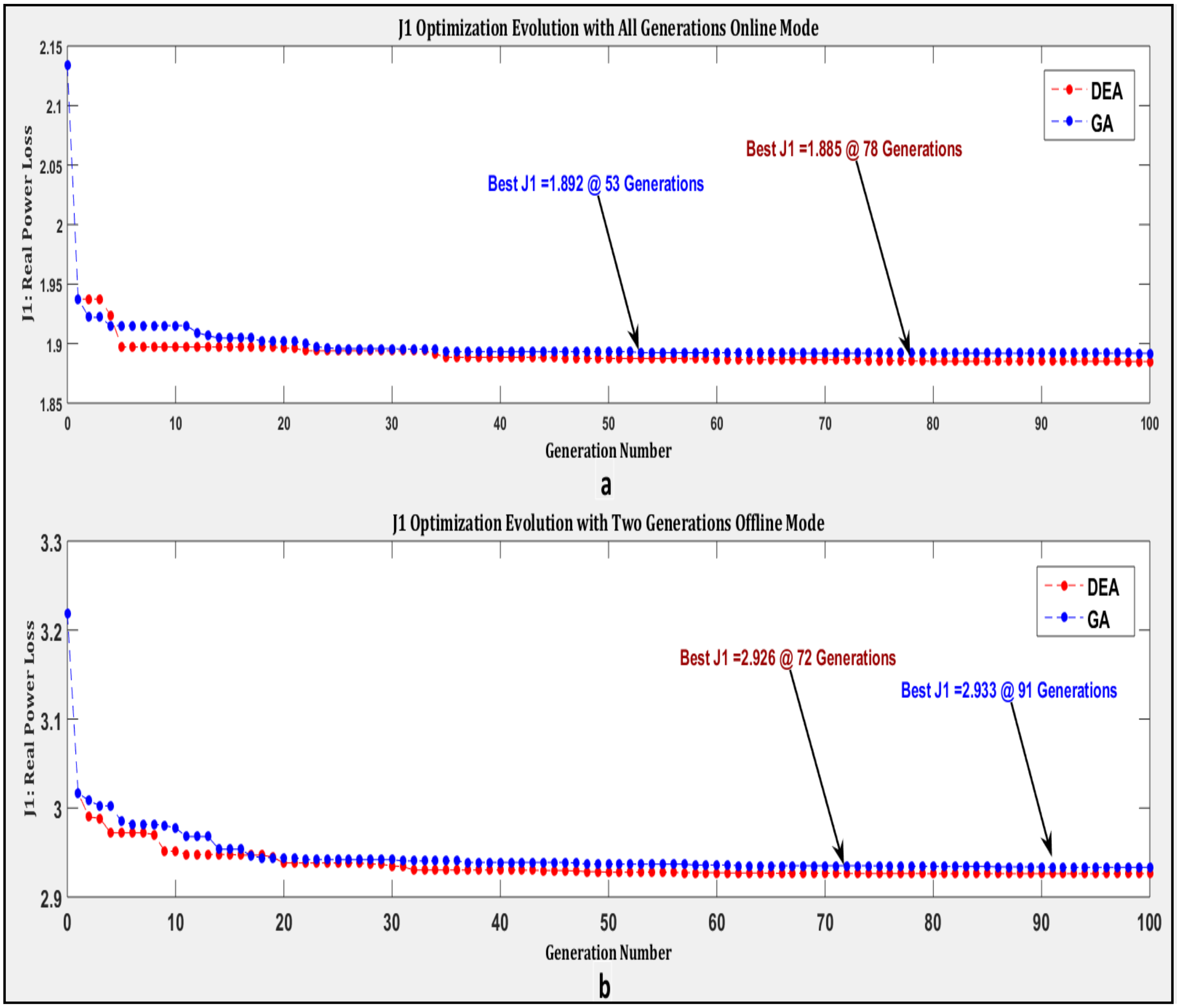

Fig. 7.24: *J_1 evolution when applying GA and DEA for the two generation modes*
a Subfigure: *J_1 evolution when applying GA and DEA for all generation online modes*
b Subfigure: *J_1 evolution when applying GA and DEA for two generation offline modes*

In Table 7.6, comparison of the objective function values are posted for the three studied subcases. The DEA performs better than the GA for subcase-2, when all generation online mode is applied. J_1 was reduced by 11.67% and J_2 was increased by 25.22% when compared with the BAU values. Also, DEA produces better objective values than GA for the two generation offline modes.

Table 7.6: The objective functions values for the studied three subcases

Generation Mode	Case #	J_1 (RPL)	Δ P_{loss} %	J_2 (GCPF)	Δ GCPF %
All online	BAU (case #1)	2.134	0.00	0.7704	0.00
Two offline	BAU (case #1)	3.219	0.00	0.9746	0.00
All online	Single-GA (case #2)	1.892	-11.34%	0.9637	25.10%
All online	Single-DEA (case #2)	**1.885**	**-11.67%**	**0.9647**	**25.22%**
Two offline	Single-GA (case #2)	2.933	-8.89%6	0.994	1.99%
Two offline	Single-DEA (case #2)	**2.926**	**-9.10%**	**1.00**	**2.61%**
All online	SPEA2 (case #3)	1.918	-10.12%	**1.00**	**29.80%**
All online	DEA (case #3)	**1.898**	**-11.06%**	0.9974	29.47%
Two offline	SPEA2 (case #3)	2.987	-7.21%	1.00	2.61%
Two offline	DEA (case #3)	**2.943**	**-8.67%**	**1.00**	**2.61%**

7.5.3 The Evolution of the RPL objective (J_1) and GCPF objective (J_2)- subcase-3

Figure 7.25 illustrates the multiobjective subcase (subcase-3) giving the two generation modes, SPEA2 and DEA Pareto-optimal set when addressing the two objective functions - J_1 and J_2. DEA has a more diversified Pareto-optimal set when compared with SPEA2. At generation 100, DEA Pareto-optimal set individuals – 50 individuals - are all non-dominated, while in the case of SPEA2 only a small number out of the 50 individuals are non-dominated.

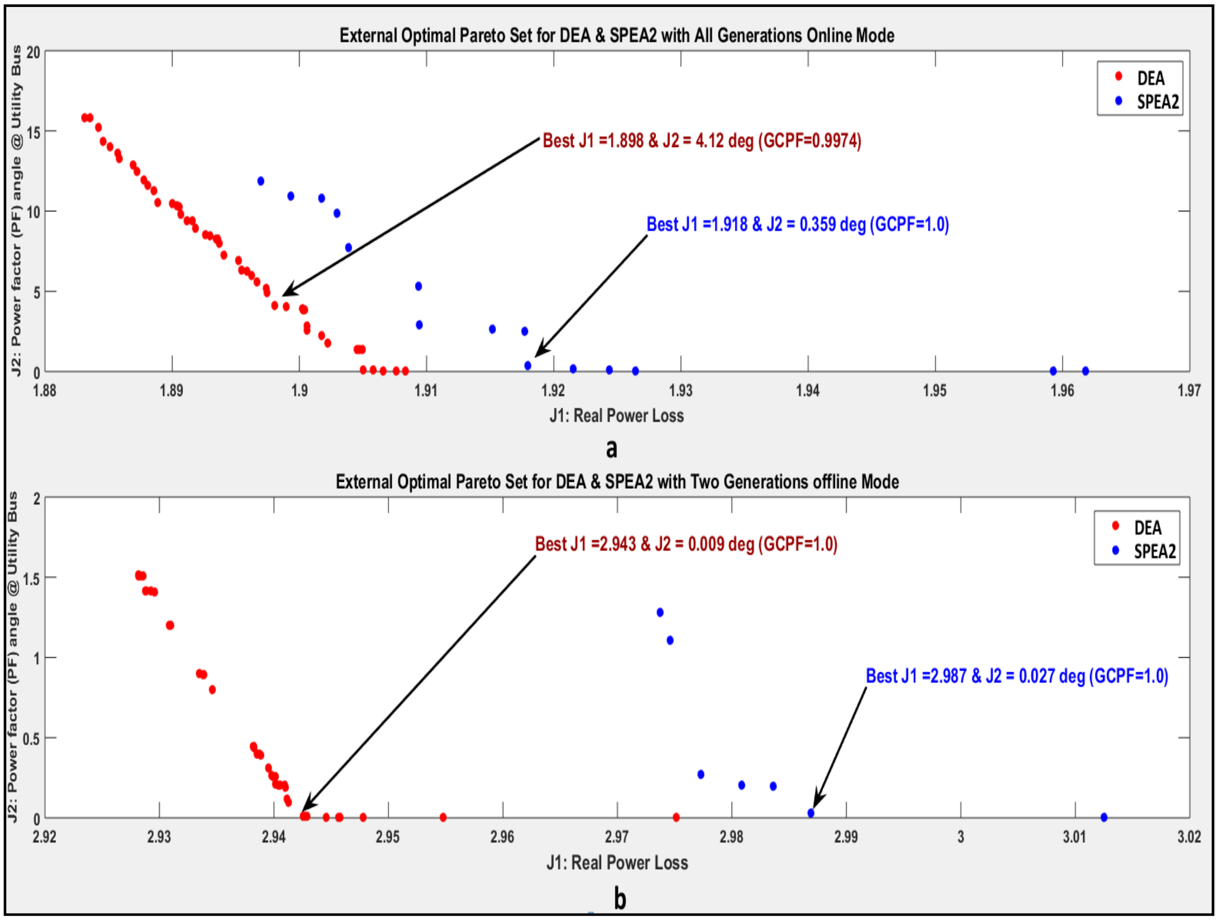

Fig. 7.25: *J_1 and J_2 front Pareto optimal sets for SPEA2 and DEA*
a Subfigure*: J_1 and J_2 evolution when applying GA and DEA for all generation online mode*
b Subfigure*: J_1 and J_2 evolution n when applying GA and DEA for two generation offline mode*

In subcase-3 - all generations are online mode - DEA produces better best compromise values for J_1 than SPEA2, when compared to the BAU values. Yet, SPEA2 produces a better J_2 value. Furthermore, in subcase-3, the two generations offline mode scenario, DEA outperforms SPEA2 for getting better J_1 values when compared to the BAU value - an 8.6% reduction. However, both had almost the same J_2 value. The value of J_1 in subcase-2 (1.885 MW) for all the generation online scenario is lower than subcase-3's DEA (1.898MW) and SPEA2 (1.918 MW) values. On the other hand, the J_2 values are very close. The same was applicable for the two generation offline scenarios. This implies that technically, limiting the optimization to single objective, subcase-2, produces better objective values. In the next section, evidences will be shown regarding whether this conclusion is valid economically.

7.5.4 Economic Benchmark Subcase-2 Vs. BAU (subcase-1) Results

Table 7.7 summarizes the cost avoidance associated with all three sub-cases, giving the two generation modes. The calculated costs are the system real power loss expenses, the real power injected to (revenue) or absorbed from (expense) the grid, the reactive power injected to or absorbed from the grid penalty expense, if |PF| < 0.85 limit and the cost balance of adding these three expenses or revenues.

In subcase-2 when applying all generation online modes, DEA had a higher cost balance when compared to GA. It had $1,028,352 cost balance revenue over the BAU $10,551,157cost balance value. The same was applicable for the two generation offline modes. DEA had a lower expense cost balance than SPEA2 with $181,823 reduction in the expense cost balance when compared with BAU $67,018,903 expense cost balance.

7.5.5 Economic Benchmark Subcase-3 Vs. BAU (subcase-1) Results

In subcase-3, the all generation online mode, DEA had a better cost balance over SPEA2 $11,571,336, with potential revenue of $1,010,178 over the BAU $10,551,157 value. On the other hand, when applying the two generation offline mode, DEA had less expense cost balance - $66,847,245 - compared to SPEA2, with $171,658 less expense cost balance when compared to BAU $67,018,903 value. In order to financially verify the aforementioned technical conclusion – section 7.5.3 - that limiting the optimization to single objective (subcase-2), produces better objective values than the multiobjective optimization (subcase-3), the cost balances were benchmarked against each other. In the all generation online mode, DEA in subcase-2 produces a better revenue cost balance of $11,579,510 - when compared to DEA in subcase-3, $11,571,336. Also, for the two generation offline mode, DEA in subcase-2, had less expense cost balance of $66,837,080, compared to DEA in subcase-3, $66,847,245 value. This implies that single objective optimization (subcase – 2) produces better technical and economic objective values for the studied system.

Table 7.7: Economical analysis for the studied three subcases

Generation Mode	Case#	Power Loss Cost	Real Power Injected into the Grid Cost	Reactive Power Injected into the Grid Cost	Cost Balance
All online	BAU (subcase -1)	(635,982.30)	12,061,076.96	(873,937.37)	10,551,157.29
Two offline	BAU (subcase-1)	(959,485.76)	(66,059,417.66)	0.00	(67,018,903.42)
All online	Single-GA (subcase -2)	(563,858.10)	12,139,047.53	0.00	11,575,189.42
All online	Single-DEA (subcase-2)	(561,782.11)	12,141,292.27	0.00	**11,579,510.16**
Two offline	Single-GA (subcase-2)	(874,285.01)	(65,967,291.40)	0.00	(66,841,576.41)
Two offline	Single-DEA (subcase-2)	(872,124.64)	(65,964,955.43)	0.00	**(66,837,080.06)**
All online	SPEA2 (subcase-3)	(571,637.10)	12,130,636.22	0.00	11,558,999.12
All online	DEA (subcase-3)	(565,709.31)	12,137,045.85	0.00	**11,571,336.54**
Two offline	SPEA2 (subcase-3)	(890,231.57)	(65,984,534.18)	0.00	(66,874,765.76)
Two offline	DEA (subcase-3)	(877,008.86)	(65,970,236.66)	0.00	**(66,847,245.52)**

7.6 Oil Price Sensitivity Analysis

All the study cases economic analysis for calculating the avoided oil burning cost was based on an assumed oil price of $60 per barrel. In this section, a sensitivity analysis for an assumed set of oil prices per barrel will be conducted. This analysis will give the decision maker a clear idea about the effect of oil barrel different prices on the avoided oil burning cost over the BAU avoided cost. Case-1 (single objective real power loss optimization case) was selected to apply the oil barrel price sensitivity analysis. Applying the sensitivity analysis to case-1 will cover other cases as the impact of the oil barrel price on the avoided oil burning cots is similar. As shown in Figure 7.26, the relation between the avoided oil barrel burning cost and the oil barrel price is linear. The recognized avoided oil burning barrel cost increase in linearly with the oil barrel price.

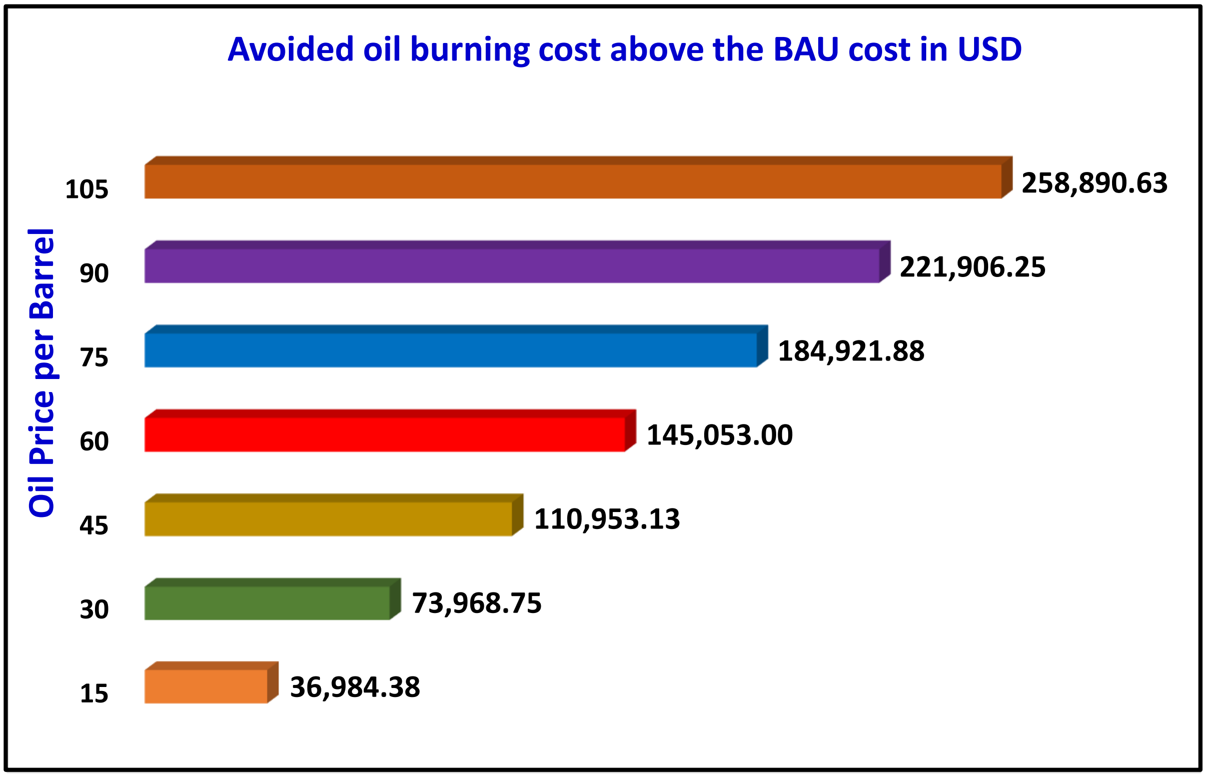

Fig. 7.26: *Oil price senesitivety analysis for case-1*

7.7 Summary

In this chapter four study cases were addressed. Case 4 consists of four subcases addressing two generation modes. Table 7.8 summarizes these cases with their objectives values and the applied intelligent algorithm. In the all online generation mode the DEA in its single objective mode produces the best minimized power loss of 1.885 MW without violating the grid connection power factor limitation as mandated by the regulator. The DEA in its single objective mode for all online generation scenario produces the best revenue balance ($ 11,579,510.16) considering the avoided cost due to real power loss minimization, revenue of real power injection to the grid and cost avoidance for meeting the GCPF limitation. Also, it produces the less net cost ($ -66,837,080.06) on the facility for the two offline generation scenario. In the other hand, the BAU case for all online generation senario is violating the grid connection power factor limitation and will result on additional cost on the plant.

Table 7.8: Summary of the study cases

Generation Mode	Case#	J_1 (Power Loss in MW)	J_2 (GCPF)	Applied IA
All online	BAU	2.134	0.7704	NA
Two offline	BAU	3.219	0.9746	NA
All online	Case-1	1.964	0.9976	GA
All online	Case-2 without MVAR	1.928	0.9900	GA
All online	Case-2 with MVAR	1.927	0.9207	GA
All online	Case-3 (T-S-R operators mix)	1.905	0.9638	GA
All online	Case-4 (GA subcase-2)	1.892	0.9637	GA
All online	Case-4 (DEA subcase-2)	**1.885**	**0.9647**	DEA
Two offline	Case-4 (GA subcase-2)	2.933	0.994	GA
Two offline	Case-4 (DEA subcase-2)	2.926	1.00	DEA
All online	Case-4 (SPEA2 subcase-3)	1.918	1.00	SPEA2
All online	Case-4 (DEA subcase-3)	1.898	0.9974	DEA
Two offline	Case-4 (SPEA2 subcase-3)	2.987	1.00	SPEA2
Two offline	Case-4 (DEA subcase-3)	2.943	1.00	DEA

CHAPTER 8: CONCLUSIONS AND FUTURE WORK

8.1 Recommendations

There is a very promising potential of real power loss minimization in real hydrocarbon facility which need to be captured. Although, the recognized real power loss is making small percentage compare to the facility load, it has very rewarding economic benefits. It is no recommended to operate the plant in BAU scenario with all generation online mode as it will result in high cost on the facility due to the violation of GCPF limitation mandated by the regulator. The MVAR injection will not add much value to the real power loss minimization due to the fact that the hydrocarbon facilities are designed to deal with very conservative system constraints. Therefore, there is no need for any shunt capacitor installation. The GA different operators' will have small impact on the real power loss minimization, this to be addressed further in future work. The DEA in its single objective mode produces the best real power loss value over the other applied algorithms. Yet, the GCPF need to be monitored to make sure that it does not violate the limitations. The potential of the real power minimization to be integrated to the scale of all facilities. An economic feasibility study considering the cost of applying the proposed methodology to be conducted considering all the hydrocarbon facility operation modes.

8.2 Summary

In this thesis the potential of real power loss (RPL) optimization at a real hydrocarbon facility power system when applying GA, SPEA2 and DEA intelligent algorithms is assessed. All the system parameters and constraints are real values as per the manufacturers' data sheets or the system real constraints. Chapter one of this work reviewed the electrical power generations challenges with regard to low efficiency and exponential increase in the electrical demand. It also sheds the lights on the thesis motivation driven for the fact that the hydrocarbon facilities at the kingdom of Saudi Arabia are large hubs of electrical demand and generation. This unleashed the idea of assessing the real power loss optimization in these facilities. The fact that none of the previous studies in the literature used real hydrocarbon facility electrical model as the model of their studies considering real system parameters and constraints is the main contribution of the thesis.

Chapter two was a literature survey of many papers tackling the application of intelligent algorithms in power system studies. The application of these algorithms in addressing single and multi-objective problem were addressed in this chapter. The issues of power system real power loss were reviewed. Many of the reviewed papers used virtual IEEE models as their study models. Some of the paper used utilities' transmission system as their research models but without emphasising on the reality of the used independent parameters. There are few papers of the reviewed ones which used real system independent parameters such as the transformers' tap positions but did not extend this to other parameters where approximate values were utilized. None of the previous papers studied hydrocarbon facility electrical system where unique mix of generation, large synchronous motors, transmission lines and distribution system do exist. This makes such system different from the IEEE virtual system or the utility transmission and distribution system.

In Chapter three, the real life hydrocarbon facility electrical system model parameters were addressed. An overall simplified one line diagram of the system configuration was posted. The facility generation mode and the utility connection parameters were identified. The specifications and capacity curves of the gas turbine generators, the steam turbine generators and the synchronous motors were addressed together with the large medium voltage induction motors. In this chapter, the model substations' category and parameters were identified. All the power and distribution transformers impedances and tap position values were addressed. The transmission and distribution lines real parameters were posted together with the loads' real values.

The thesis problem formulation was posted in Chapter four. All the system real equality and inequality constraints were identified. The three objective functions of the thesis were addressed in this chapter. The problem economic formulations were posted also in this chapter. This includes identifying the gas turbine generators MBTU value and its mathematical relation with the MMSCF of gas. The gas subsidiary cost for the kingdom of Saudi Arabia as well as electricity tariff were identified. Calculating the number of avoided oil burning barrels and its cost base on assumed oil barrel price were addressed.

The problem programming using the MATLAB software were addressed in Chapter five. This includes the MATLAB programming codes for reading the system parameters, line data, bus data and load data. It also, includes the MATLAB programming codes for assigning the independent parameters' such as the generator and synchronous motor terminal bus voltages and the transformers' tap position values. The MATLAB programming codes of calculating the system admittance matrix and performing power flow were posted in this chapter. The programming for the verification of the constraints compliance was developed part of this chapter.

In Chapter six, an overview of the thesis three - GA, SPEA2 and DEA - evolution algorithms was conducted. The GA evolution process, selection, crossover and mutation methods were described in this chapter. The concept of multiobjective evolutionary algorithm (MOEA) was described together with the fuzzy set theory for extracting the best compromise solution out of the Pareto optimal solutions. The SPEA2 evolution algorithm and the DEA algorithm mechanisms were addressed. The mechanism of reducing the front Pareto optimal solution size by truncation technique was illustrated in this chapter.

The thesis five study cases and their results were discussed in Chapter seven. The first case is business as usual (BUA) case to be benchmarked with the other optimized cases results. The second case is addressing the real power loss optimization problem as single objective constrained problem using the genetic algorithm. The third case is similar to case two but considering the effect of MVAR injection in predefined buses base on the system voltage stability index ranking. The fourth case address the effects of GA different selection, crossover and mutation methods on the real power system loss optimization. The fifth case address the problem as single objective similar to case two and multiobjective considering the grid connection power factor optimization together with the system real power loss. This case was addressed with two generation modes. In all the five cases, economic analysis of the results were illustrated.

All the thesis cases results demonstrated very promising potential in optimizing the objective functions considering all the system constraints and real parameters. The economic advantages of optimizing the objective functions were very rewarding for all the identified cost avoidance due to the real power loss optimization or the revenue due to power injection in the grid.

8.3 Future Works

A proposed future works may consider the followings:

- Addressing the effect of the independent parameters changes as recommended by the optimal solution.
- Developing smaller – less details - model of the hydrocarbon facility representing this work detail model with high credibility.
- Applying this work practically in the field.

8.3.1 Effect of Independent Parameters Changes

The thesis optimal solution always couple with recommended values of the system independent parameters such as the generation and synchronous motors terminal bus voltages and the transformer tap position values. In real life, implementing these independent parameters values cannot be done at the same time. They will be changed one after another, this raises challenges as the system condition changes with each of its independent parameter change which may push the system to the instability region or violation of the system constraints. Also, a weight factor to be linked to each prameters depends on the cost and operation complexity associated with applying it. So, future works may consider ranking the order of applying the independent parameters optimal values to avoid any additional operation cost or disturbance to the hydrocarbon facility electrical system and operation.

8.3.2 Development of Smaller Electrical Model

Considering smaller - optimized - electrical model of the hydrocarbon facility will support reducing the time of the evolution process for identifying the optimal system condition parameters. In some of this thesis studied cases, the evolution process took more than 24 hours which may not be acceptable for the decision makers. Yet, the credibility of the reduced model need to be verified by benchmarking the optimal results with the results obtained from the full version electrical model before being adapted.

8.3.3 Assessing the Application of the Recommended Algorithms

Any future work may consider an assessment of implementing this thesis optimization tools application in real life. This requires identifying the infrastructures and the tools needed to support the application. Also, the net present value (NPV) of the implementation project of this thesis optimization tools need to be calculated to assess the economic feasibility of the same. As shown in Figure 8.1, the implementation required real time reading of the hydrocarbon facilities electrical system parameters, control mechanism to adjust the system parameters and an economic advantages of implementing the recommended optimal system parameters.

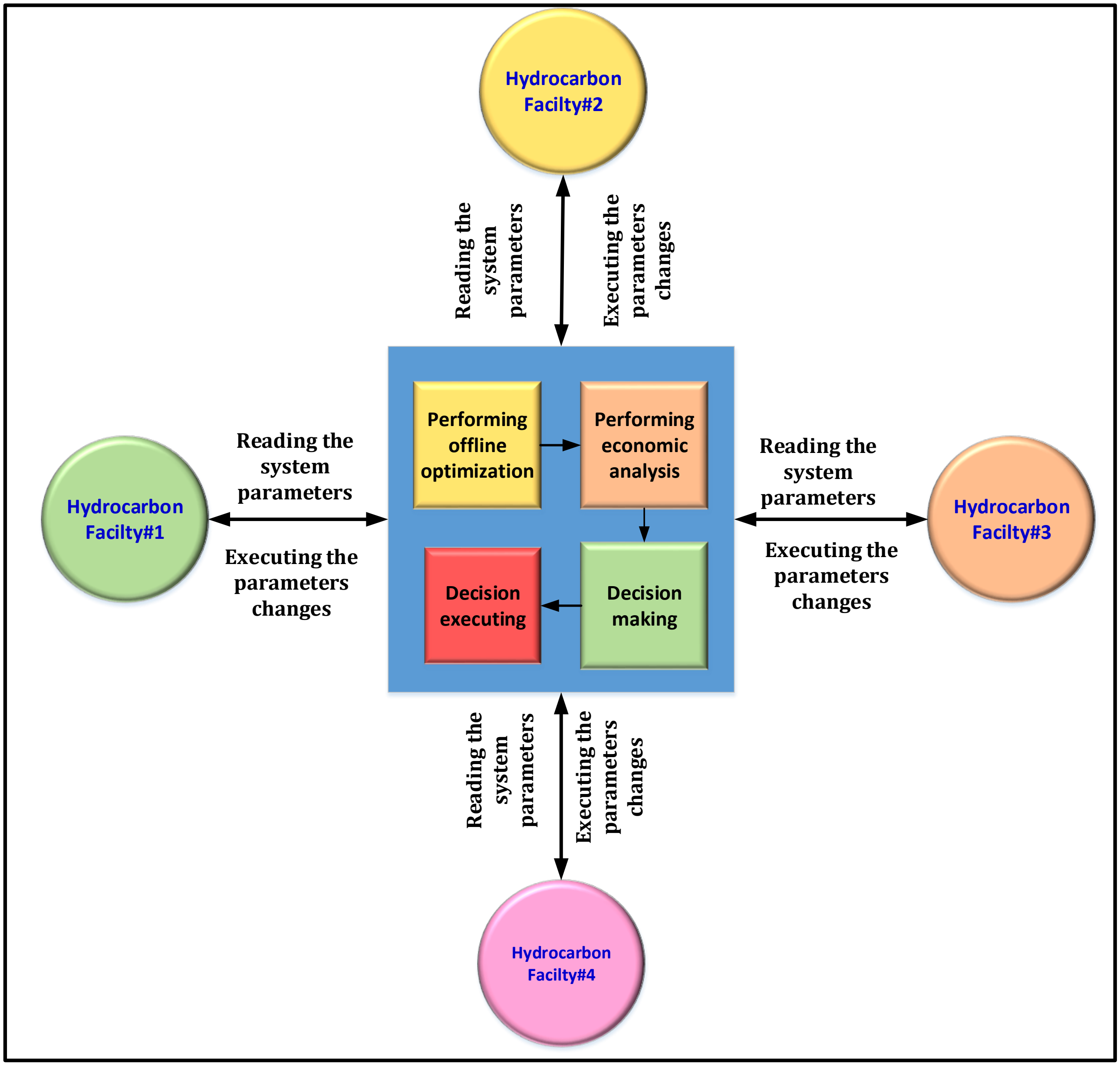

Fig. 8.1: *Overall lay out of the thesis optimization tool implementation*

APPENDIX I

The System Transformer Parameters. [Refer to section 3.8]

Substation #	Transformer Tag #	MVA Rating	Voltage in kV		Z %	Voltage Variance %	# of Taps	Each Tap Step %
			Primary	Secondary				
Sub#2	2T-1	60/80	115	13.8	9	±10%	±16	0.625
Sub#2	2T-2	60/80	115	13.8	9	±10%	±16	0.625
Sub#2	2T-3	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#2	2T-4	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2	2T-5	2.5/3.125	13.8	0.9	5.75	±5%	±2	2.5
Sub#2	2T-6	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2	2T-7	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2	2T-8	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2	2T-9	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#2	2T-10	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2	2T-11	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2	2T-12	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2	2T-13	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2	2T-14	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/3	3T-1	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#2/3	3T-2	2.0	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/3	3T-3	2.0	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/3	3T-4	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/3	3T-5	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#2/3	3T-6	2.0	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/3	3T-7	2.0	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/3	3T-8	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/4	4T-1	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#2/4	4T-2	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/4	4T-3	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/4	4T-4	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#2/4	4T-5	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/4	4T-6	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/5	5T-1	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#2/5	5T-1	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/5	5T-1	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/5	5T-1	10/12.5	13.8	4.16	6.5	±5%	±2	2.5

Substation #	Transformer Tag #	MVA Rating	Voltage in kV		Z %	Voltage Variance %	# of Taps	Each Tap Step %
			Primary	Secondary				
Sub#2/5	5T-1	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/5	5T-1	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/6	6T-1	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#2/6	6T-2	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/6	6T-3	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/6	6T-4	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#2/6	6T-5	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/6	6T-6	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/11	11T-1	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#2/11	11T-2	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#7	C7-T1	50/66.6	115	13.8	9	±10%	±16	0.625
Sub#7	7T-6	2	13.8	0.48	5.75	±5%	±2	2.5
Sub#7	7T-7	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#7	7T-8	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#7	7T-2	50/66.6	115	13.8	9	±10%	±16	0.625
Sub#7	7T-3	2	13.8	0.48	5.75	±5%	±2	2.5
Sub#7	7T-4	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#7	7T-5	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#7/W1	W1-T	50	115	13.8	8	±10%	±16	0.625
Sub#7/W2	W2-T	50	115	13.8	8	±10%	±16	0.625
Sub#7/W3	W3-T	50	115	13.8	8	±10%	±16	0.625
Sub#7/W4	W4-T	50	115	13.8	8	±10%	±16	0.625
Sub#7/W5	W5-T	50	115	13.8	8	±10%	±16	0.625
Sub#8	8T-1	50/66.6	115	13.8	9	±10%	±16	0.625
Sub#8	8T-2	50/66.6	115	13.8	9	±10%	±16	0.625
Sub#8	8T-6	2	13.8	0.48	5.75	±5%	±2	2.5
Sub#8	8T-3	50/66.6	115	13.8	9	±10%	±16	0.625
Sub#8	8T-2	50/66.6	115	13.8	9	±10%	±16	0.625
Sub#8	8T-6	2	13.8	0.48	5.75	±5%	±2	2.5

Substation #	Transformer Tag #	MVA Rating	Voltage in kV		Z %	Voltage Variance %	# of Taps	Each Tap Step %
			Primary	Secondary				
Sub#8	8T-3	50/66.6	115	13.8	9	±10%	±16	0.625
Sub#8	8T-7	2	13.8	0.48	5.75	±5%	±2	2.5
Sub#8	8T-4	50/66.6	115	13.8	9	±10%	±16	0.625
Sub#8	8T-8	2	13.8	0.48	5.75	±5%	±2	2.5
Sub#9	9T-1	50/66	115	13.8	9	±10%	±16	0.625
Sub#9	9T-5	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#9	9T-6	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#9	9T-7	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#9	9T-8	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#9	9T-2	50/66	115	13.8	9	±10%	±16	0.625
Sub#9	9T-3	50/66	115	13.8	9	±10%	±16	0.625
Sub#9	9T-4	50/66	115	13.8	9	±10%	±16	0.625
Sub#9	9T-9	10/12.5	13.8	4.16	6.5	±5%	±2	2.5
Sub#9	9T-10	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#9	9T-11	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5
Sub#9	9T-12	2.5/3.125	13.8	0.48	5.75	±5%	±2	2.5

APPENDIX II

The System Load Data. [Refer to section 3.11]

Substation #	Bus #	Load Tag#	Voltage	Load	
				MW	MVAR
Sub#2	15	2-L1	4.16kV	2.464	1.33
Sub#2	17	2-L2	480 V	1.044	0.592
Sub#2	19	2-L3	480 V	0.875	0.484
Sub#2	21	2-L4	480 V	1.225	0.678
Sub#2	23	2-L5	480 V	0.555	0.228
Sub#2	25	2-L6	480 V	0.592	0.236
Sub#2	30	2-L7	4.16 kV	1.672	0.898
Sub#2	32	2-L8	480 V	0.6944	0.3973
Sub#2	34	2-L9	480 V	0.5412	0.2590
Sub#2	36	2-L10	480 V	0.8055	0.4015
Sub#2	38	2-L11	480 V	0.7368	0.3117
Sub#2	40	2-L12	480 V	0.674	0.265
Sub#2/3	42	3-L1	4.16 kV	1.2613	0.5275
Sub#2/3	44	3-L2	480 V	0.6084	0.3234
Sub#2/3	46	3-L3	480 V	0.6056	0.326
Sub#2/3	48	3-L4	480 V	1.1236	0.6467
Sub#2/3	50	3-L5	4.16 kV	1.2613	0.5275
Sub#2/3	52	3-L6	480 V	0.5346	0.2956
Sub#2/3	54	3-L7	480 V	0.5779	0.2566
Sub#2/3	56	3-L8	480 V	1.051	0.5899
Sub#2/4	58	4-L1	4.16 kV	2.3488	0.9481
Sub#2/4	60	4-L2	480 V	1.1094	0.5639
Sub#2/4	62	4-L3	480 V	0.4863	0.39399
Sub#2/4	64	4-L4	4.16 kV	2.723	0.9481
Sub#2/4	66	4-L5	480 V	0.5846	0.2939
Sub#2/4	68	4-L6	480 V	0.4485	0.3578
Sub#2/5	70	5-L1	4.16 kV	2.3488	0.94806
Sub#2/5	72	5-L2	480 V	0.6618	0.32059
Sub#2/5	74	5-L3	480 V	0.5644	0.45732
Sub#2/5	76	5-L4	4.16 kV	2.7902	1.12576
Sub#2/5	78	5-L5	480 V	0.4633	0.1869
Sub#2/5	80	5-L5	480 V	0.42504	0.31897
Sub#2/6	82	6-L1	4.16 kV	3.267	1.28664
Sub#2/6	84	6-L2	480 V	0.60558	0.30901
Sub#2/6	86	6-L3	480 V	0.57158	0.46292
Sub#2/6	88	6-L4	4.16 kV	1.9396	0.77536

Substation #	Bus#	Load Tag#	Voltage	Load	
				MW	Mvar
Sub#2/6	90	6-L5	480 V	0.26785	0.11271
Sub#2/6	92	6-L6	480 V	0.42435	0.31732
Sub#2/I	93	I-L1	13.8 kV	2.3204	1.972
Sub#2/I	94	I-L2	13.8 kV	2.3204	1.972
Sub#2/11	96	11-L1	480 V	1.0532	0.60592
Sub#2/11	98	11-L2	480 V	1.0901	0.72306
Sub#C1/MB1	104	C1-L1	480 V	0.1326	0.09945
Sub#C1/MB1	105	C1-P1	13.8 kV	3.7415	0.69368
Sub#C1/MB1	106	C1-P2	13.8 kV	0.0767	0.0475
Sub#C1/MB1	107	C1-F440	13.8 kV	4.7342	0.9099
Sub#C1/MB1	108	C1-P3	13.8 kV	4.8212	0.9854
Sub#C1/MB1	109	C1-P4	13.8 kV	4.8212	0.9831
Sub#C1/MB1	110	C1-L2	13.8 kV	4.8142	0.9825
Sub#C1/MB2	115	C1-L3	480 V	0.2336	0.1752
Sub#C1/MB2	116	C1-P5	13.8 kV	3.7389	0.67528
Sub#C1/MB2	117	C1-P6	13.8 kV	0.1904	0.118
Sub#C1/MB2	118	C1-F28	13.8 kV	2.849	0.53796
Sub#C1/MB2	119	C1-F430	13.8 kV	4.8606	1.002
Sub#C1/MB2	120	C1-L4	13.8 kV	4.8124	0.9813
Sub#C1/MB2	121	C1-P8	13.8 kV	4.8176	0.9837
Sub#C1/MB2	122	C1-P7	13.8 kV	4.8264	0.9873
Sub#C2/MB1	128	C2-L2	480 V	0.141	0.10575
Sub#C2/MB1	129	C2-P11	13.8 kV	0.1063	0.06585
Sub#C2/MB1	130	C2-P9	13.8 kV	4.814	0.9825
Sub#C2/MB1	131	C2-P10	13.8 kV	4.8106	0.9807
Sub#C2/MB1	132	C2-P12	13.8 kV	4.829	0.9885
Sub#C2/MB1	133	C2-P13	13.8 kV	4.8106	0.9807
Sub#C2/MB1	134	C2-L1	13.8 kV	4.814	0.9825
Sub#C2/MB2	139	C2-L3	480 V	0.17338	0.130
Sub#C2/MB2	140	C2-P14	13.8 kV	0.05219	0.03234
Sub#C2/MB2	141	C2-P16	13.8 kV	4.8132	0.9819
Sub#C2/MB2	142	C2-P17	13.8 kV	4.8106	0.9807
Sub#C2/MB2	143	C2-P15	13.8 kV	4.8106	0.9807
Sub#C2/MB2	144	C2-L4	13.8 kV	4.8572	0.9999
Sub#C3/MB1	150	C3-L2	480 V	0.1652	0.1239
Sub#C3/MB1	151	C3-P18	13.8 kV	0.1114	0.06901
Sub#C3/MB1	152	C3-P19	13.8 kV	4.8114	0.9813
Sub#C3/MB1	153	C3-P20	13.8 kV	4.8274	0.9873
Sub#C3/MB1	154	C3-F420	13.8 kV	3.739	0.69248

Substation #	Bus#	Load Tag#	Voltage	Load	
				MW	Mvar
Sub#C3/MB1	155	C3-F34	13.8 kV	2.8498	0.53796
Sub#C3/MB1	156	C3-L1	13.8 kV	5.0953	0.7
Sub#C3/MB2	161	C3-L4	480 V	0.22383	0.16787
Sub#C3/MB2	162	C3-P21	13.8 kV	4.8732	1.0065
Sub#C3/MB2	163	C3-F410	13.8 kV	3.739	0.69248
Sub#C3/MB2	164	C3-P23	13.8 kV	4.8116	0.9813
Sub#C3/MB2	165	C3-P24	13.8 kV	4.8116	0.9813
Sub#C3/MB2	166	C3-L3	13.8 kV	0.03995	0.02476
Sub#C3/MB2	167	C3-P22	13.8 kV	4.8636	1.002
Sub#C3/MB2	168	C3-P25	13.8 kV	5.0927	0.6988
Sub#7/MB1	173	7-L4	480 V	1.089	0.574
Sub#7/MB1	175	7-L5	4.16 kV	1.554	0.973
Sub#7/MB1	177	7-L6	480 V	0.544	0.361
Sub#7/MB1	178	7-L8	13.8 kV	8.476	3.320
Sub#7/MB2	183	7-L1	480 V	0.773	0.549
Sub#7/MB2	185	7-L2	4.16 kV	1.692	1.3
Sub#7/MB2	187	7-L3	480 V	1.231	0.845
Sub#7/MB2	188	7-L7	13.8 kV	8.476	4.32
Sub#9/MB1	209	9-M1	4.16 KV	2.806	1.3
Sub#9/MB1	212	9-L1	480 V	1.259	0.505
Sub#9/MB1	215	9-L2	480 V	1.266	0.639
Sub#9/MB1	218	9-L3	480 V	1.2174	0.74419
Sub#9/MB2	229	9-L4	480 V	0.874	0.3159
Sub#9/MB2	232	9-L5	480 V	0.845	0.4415
Sub#9/MB2	235	9-M4	4.16k V	1.3174	0.7893
Sub#9/MB2	238	9-L6	480 V	0.7955	0.41424
Sub#9/MB2	238	9-L6	480 V	0.7955	0.41424
Sub#9/MB2	238	9-L6	480 V	0.7955	0.41424
Sub#8/MB1	247	8-M1	13.8 kV	4.801	2.202
Sub#8/MB1	248	8-M2	13.8 kV	4.801	2.202
Sub#8/MB1	249	8-M3	13.8 kV	4.801	2.202
Sub#8/MB1	251	8-L1	480V	0.8406	0.48274
Sub#8/MB2	255	8-M4	13.8 kV	4.801	1.872
Sub#8/MB2	256	8-M5	13.8 kV	4.801	1.872
Sub#8/MB2	257	8-M6	13.8 kV	4.801	1.872
Sub#8/MB2	259	8-L2	480V	0.71715	0.42552
Sub#8/MB3	263	8-M7	13.8 kV	4.801	1.872
Sub#8/MB3	264	8-M8	13.8 kV	4.801	1.872
Sub#8/MB3	265	8-M9	13.8 kV	4.801	1.872

Substation #	Bus#	Load Tag#	Voltage	Load	
				MW	Mvar
Sub#8/MB3	267	8-L3	480V	0.57645	0.38549
Sub#8/MB4	271	8-M10	13.8 kV	5.534	1.872
Sub#8/MB4	272	8-M11	13.8 kV	5.534	1.872
Sub#8/MB4	274	8-L4	480V	0.54861	0.35709
Sub#8/MB4	275	8-M12	13.8 kV	5.534	1.872

APPENDIX III

The MATLAB Program Codes for Reading The System Bus Data (SdataR.m). [Refer to section 5.2.1]

```
%*********************************************************************
% This Sub-Program is for Reading the System Data from the System Data
% excel file. This is needed to create the busdata matrix as part of the
% MATLAB code for performing the NEWTON- RAPHSON power flow solution.
%************************************************************************

% Reading the 1st System Data Sheet SD1_1
SD1T = xlsread('System Data.xlsx', 'Data-1','a5:n31');

% Save the Read Excel Sheet to a Variable SD1; In case the original read
% Sheet need to be accessed.
SD1=SD1T;

% Delete the Buses Name Column from the SD1 Matrix
SD1T(:,3)=[];

% Delete the Generator Maximum MW Column from the SD1 Matrix
SD1T(:,8)=[];

% Delete the Voltage (KV) Column from the SD1 Matrix
SD1T(:,3)=[];

% The Matrix Name after Deleting ALL NOT Needed Columns
SD1_1=SD1T;

% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)

% Saving the Generation Four Columns Temporary; by assigning it
% to a matrix A1T
A1T=SD1_1(:,5:8);

% Saving the Load Two Columns; assigning it to matrix A2T
A2T=SD1_1(:,9:10);

% Posting the Load Two Columns in their rights columns order
%(columns 5 & 6)
SD1_1(:,5:6)=A2T;

% Posting the Generation Four Columns in their rights columns order
%(columns 7, 8, 9 & 10)
SD1_1(:,7:10)=A1T;

% Final System Data-1 (sheet 1) Matrix
SD1_1;

% Clear all Used Variables
```

```
clear A1T A2T SDIT


%**************************************************************************
% Reading the 2nd System Data Sheet SD2_2
SD2T = xlsread('System Data.xlsx', 'Data-2','a5:n33');
% T mean temporary variable
SD2=SD2T;
% Delete the Buses Name Column from the SD1 Matrix
SD2T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD2T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD2T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD2_2=SD2T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD2_2(:,5:8);
% Saving the Load Two Columns
A2T=SD2_2(:,9:10);
% Posting the Load Two Columns
SD2_2(:,5:6)=A2T;
% Posting the Generation Four Columns
SD2_2(:,7:10)=A1T;
% Final System Data-2 (sheet 2) Matrix
SD2_2;
% Clear all Used Variables
clear A1T A2T SD2T
%**********************************************************************

% Reading the 3rd System Data Sheet SD3_3
SD3T = xlsread('System Data.xlsx', 'Data-3','a5:n33');
% T mean temporary variable
SD3=SD3T;
% Delete the Buses Name Column from the SD1 Matrix
SD3T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD3T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD3T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD3_3=SD3T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD3_3(:,5:8);
% Saving the Load Two Columns
A2T=SD3_3(:,9:10);
% Posting the Load Two Columns
SD3_3(:,5:6)=A2T;
% Posting the Generation Four Columns
SD3_3(:,7:10)=A1T;
% Final
System Data-3 (sheet 3) Matrix
```

```
SD3_3;
% Clear all Used Variables
clear A1T A2T SD3T

%**********************************************************************

% Reading the 4th System Data Sheet SD4_4
SD4T = xlsread('System Data.xlsx', 'Data-4','a5:n33');
% T mean temporary variable
SD4=SD4T;
% Delete the Buses Name Column from the SD1 Matrix
SD4T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD4T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD4T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD4_4=SD4T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD4_4(:,5:8);
% Saving the Load Two Columns
A2T=SD4_4(:,9:10);
% Posting the Load Two Columns
SD4_4(:,5:6)=A2T;
% Posting the Generation Four Columns
SD4_4(:,7:10)=A1T;
% Final System Data-4 (sheet 4) Matrix
SD4_4;
% Clear all Used Variables
clear A1T A2T SD4T

%**********************************************************************

% Reading the 5th System Data Sheet SD5_5
SD5T = xlsread('System Data.xlsx', 'Data-5','a5:n33');
% T mean temporary variable
SD5=SD5T;
% Delete the Buses Name Column from the SD1 Matrix
SD5T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD5T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD5T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD5_5=SD5T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD5_5(:,5:8);
% Saving the Load Two Columns
A2T=SD5_5(:,9:10);
% Posting the Load Two Columns
SD5_5(:,5:6)=A2T;
% Posting the Generation Four Columns
```

```
SD5_5(:,7:10)=A1T;
% Final System Data-5 (sheet 5) Matrix
SD5_5;
% Clear all Used Variables
clear A1T A2T SD5T

%**********************************************************************

% Reading the 6th System Data Sheet SD6_6
SD6T = xlsread('System Data.xlsx', 'Data-6','a5:n33');
% T mean temporary variable
SD6=SD6T;
% Delete the Buses Name Column from the SD1 Matrix
SD6T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD6T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD6T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD6_6=SD6T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD6_6(:,5:8);
% Saving the Load Two Columns
A2T=SD6_6(:,9:10);
% Posting the Load Two Columns
SD6_6(:,5:6)=A2T;
% Posting the Generation Four Columns
SD6_6(:,7:10)=A1T;
% Final System Data-6 (sheet 6) Matrix
SD6_6;
% Clear all Used Variables
clear A1T A2T SD6T
%**********************************************************************

% Reading the 7th System Data Sheet SD7_7
SD7T = xlsread('System Data.xlsx', 'Data-7','a5:n33');
% T mean temporary variable
SD7=SD7T;
% Delete the Buses Name Column from the SD1 Matrix
SD7T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD7T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD7T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD7_7=SD7T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD7_7(:,5:8);
% Saving the Load Two Columns
A2T=SD7_7(:,9:10);
% Posting the Load Two Columns
SD7_7(:,5:6)=A2T;
```

```
% Posting the Generation Four Columns
SD7_7(:,7:10)=A1T;
% Final System Data-7 (sheet 7) Matrix
SD7_7;
% Clear all Used Variables
clear A1T A2T SD7T

%**********************************************************************

% Reading the 8th System Data Sheet SD8_8
SD8T = xlsread('System Data.xlsx', 'Data-8','a5:n32');
% T mean temporary variable
SD8=SD8T;
% Delete the Buses Name Column from the SD1 Matrix
SD8T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD8T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD8T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD8_8=SD8T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD8_8(:,5:8);
% Saving the Load Two Columns
A2T=SD8_8(:,9:10);
% Posting the Load Two Columns
SD8_8(:,5:6)=A2T;
% Posting the Generation Four Columns
SD8_8(:,7:10)=A1T;
% Final System Data-8 (sheet 8) Matrix
SD8_8;
% Clear all Used Variables
clear A1T A2T SD8T
%**********************************************************************

% Reading the 9th System Data Sheet SD9_9
SD9T = xlsread('System Data.xlsx', 'Data-9','a5:n33');
% T mean temporary variable
SD9=SD9T;
% Delete the Buses Name Column from the SD1 Matrix
SD9T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD9T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD9T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD9_9=SD9T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD9_9(:,5:8);
% Saving the Load Two Columns
A2T=SD9_9(:,9:10);
% Posting the Load Two Columns
```

```
SD9_9(:,5:6)=A2T;
% Posting the Generation Four Columns
SD9_9(:,7:10)=A1T;
% Final System Data-9 (sheet 9) Matrix
SD9_9;
% Clear all Used Variables
clear A1T A2T SD9T

%**********************************************************************

% Reading the 10th System Data Sheet SD10_10
SD10T = xlsread('System Data.xlsx', 'Data-10','a5:n34');
% T mean temporary variable
SD10=SD10T;
% Delete the Buses Name Column from the SD1 Matrix
SD10T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD10T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD10T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD10_10=SD10T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD10_10(:,5:8);
% Saving the Load Two Columns
A2T=SD10_10(:,9:10);
% Posting the Load Two Columns
SD10_10(:,5:6)=A2T;
% Posting the Generation Four Columns
SD10_10(:,7:10)=A1T;
% Final System Data-10 (sheet 10) Matrix
SD10_10;
% Clear all Used Variables
clear A1T A2T SD10T
%**********************************************************************

% Reading the 11th System Data Sheet SD11_11
SD11T = xlsread('System Data.xlsx', 'Data-11','a5:n33');
% T mean temporary variable
SD11=SD11T;
% Delete the Buses Name Column from the SD1 Matrix
SD11T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD11T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD11T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD11_11=SD11T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD11_11(:,5:8);
% Saving the Generation Four Columns
A2T=SD11_11(:,9:10);
```

```
% Posting the Load Two Columns
SD11_11(:,5:6)=A2T;
% Posting the Load Two Columns
SD11_11(:,7:10)=A1T;
% Final System Data-11 (sheet 11) Matrix
SD11_11;
% Clear all Used Variables
clear A1T A2T SD11T

%**********************************************************************

% Reading the 12th System Data Sheet SD12_12
SD12T = xlsread('System Data.xlsx', 'Data-12','a5:n34');
% T mean temporary variable
SD12=SD12T;
% Delete the Buses Name Column from the SD1 Matrix
SD12T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD12T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD12T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD12_12=SD12T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD12_12(:,5:8);
% Saving the Load Two Columns
A2T=SD12_12(:,9:10);
% Posting the Load Two Columns
SD12_12(:,5:6)=A2T;
% Posting the Generation Four Columns
SD12_12(:,7:10)=A1T;
% Final System Data-12 (sheet 12) Matrix
SD12_12;
% Clear all Used Variables
clear A1T A2T SD12T
%**********************************************************************

% Reading the 13th System Data Sheet SD13_13
SD13T = xlsread('System Data.xlsx', 'Data-13','a5:n34');
% T mean temporary variable
SD13=SD13T;
% Delete the Buses Name Column from the SD1 Matrix
SD13T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD13T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD13T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD13_13=SD13T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD13_13(:,5:8);
% Saving the Load Two Columns
```

```
A2T=SD13_13(:,9:10);
% Posting the Generation Four Columns
SD13_13(:,5:6)=A2T;
% Posting the Load Two Columns
SD13_13(:,7:10)=A1T;
% Final System Data-13 (sheet 13) Matrix
SD13_13;
% Clear all Used Variables
clear A1T A2T SD13T

%**********************************************************************

% Reading the 14th System Data Sheet SD14_14
SD14T = xlsread('System Data.xlsx', 'Data-14','a5:n31');
% T mean temporary variable
SD14=SD14T;
% Delete the Buses Name Column from the SD1 Matrix
SD14T(:,3)=[];
% Delete the MW Range Max Column from the SD1 Matrix
SD14T(:,8)=[];
% Delete the Voltage (KV) Column from the SD1 Matrix
SD14T(:,3)=[];
% The Matrix Name after Deleting ALL not needed Columns
SD14_14=SD14T;
% Arrange The Matrix (switching the Load Two Columns with the Generation
% Four Columns)
% Saving the Generation Four Columns
A1T=SD14_14(:,5:8);
% Saving the Load Two Columns
A2T=SD14_14(:,9:10);
% Posting the Load Two Columns
SD14_14(:,5:6)=A2T;
% Posting the Generation Four Columns
SD14_14(:,7:10)=A1T;
% Final System Data-14 (sheet 14) Matrix
SD14_14;
% Clear all Used Variables
clear A1T A2T SD14T
%**************************************************************************

% After Reading all the Excel file sheets, they are assigned to Temporary
% Matrix (SDataT) as showing below
SDataT=[SD1_1;SD2_2;SD3_3;SD4_4;SD5_5;SD6_6;SD7_7;SD8_8;SD9_9;SD10_10;
         SD11_11;SD12_12;SD13_13;SD14_14];

% The length (number of rows) of the SDataT is read
L_SDataT = length(SDataT(:,1));

% The following loop will go through the SDataT and identify all the
% repeating Buses and assign it to an array called T_Delete (duplicated
% buses to be removed)
for x=1:L_SDataT-1;
    for y=x+1:L_SDataT;
        if SDataT(x,1)==SDataT(y,1)
           h=y;
           T_Delete(h,1)=h;
```

```
        else,end
    clear h
    end
end

% Clear the Variable
clear h x y

%**************************************************************************
% The following subroutine will clean the Repeating Bus Number from SDataT.
% A counter is establish to be used in the Subroutine (Cont). The counter
% is set equal to Zero (0)
Cont=0;
% Measure the Length of T_Delete
LT_Delete=length(T_Delete(:,1));

% This loop will go through the SDataT matrix and delete the repeated buses
for x=1:LT_Delete;
    if T_Delete(x,1)>0;
       SDataT(x-Cont,:)=[];
       Cont=Cont+1;
    else,end
end

% Clear the Variable
clear h x y Cont LT_Delete T_Delete L_SDataT

%**************************************************************************

% The Temporary System Data Matrix (SDataT) is assigned to the permanent
% Matrix SData
SData=SDataT;
% Clear the Temporary System Data Matrix (SDataT)
clear SDataT
%**************************************************************************
```

APPENDIX IV
The System Transformer and Line Parameters. [Refer to section 5.2.2]

Bus nl: Source Bus Number
Bus nr: Receiving Bus Number
Y/KM: Line Capacitance Per KM

R p.u. : Line Resistance in Per Unit

R/KM: Line Resistance Per KM
X/KM: Line Reactance Per KM
½ B p.u. : ½ the Line Capacitance Susceptancee in Per Unit
X p.u. : Line Resistance in Per Unit

Bus nl	Bus nr	R/KM	R p.u.	X/KM	X p.u.	Y/KM	½ B p.u.
1	2	0	0.002609	0	0.0660068	0	0
3	4	0	0	0	0.09333	0	0
4	2	0.0404	0.00003055	0.150857	0.000114	0.000067886	0.00404
5	6	0	0	0	0.26	0	0
6	2	0.0404	0.000128	0.150857	0.000479	0.000067886	0.001885
7	8	0	0	0	0.09333	0	0
8	2	0.0404	0.00002189	0.150857	0.00008175	0.000067886	0.002895
9	10	0	0	0	0.1566	0	0
10	2	0.0404	0.00005804	0.150857	0.0002167	0.000067886	0.0034116
2	11	0.0404	0.0001054	0.150857	0.000393	0.000067886	0.001548
11	12	0	0	0	0.15	0	0
12	13	0.047	0.000148	0.123316	0.00038855	0.00016217	0.00001482
13	14	0.124	0.0022789	0.14218	0.002613	0.00010937	0.00001458
14	15	0	0	0	0.65	0	0
13	16	0.124	0.0026045	0.14218	0.002986	0.00010937	0.00001666
16	17	0	0	0	2.3	0	0
13	18	0.124	0.0026045	0.14218	0.002986	0.00010937	0.00001666
18	19	0	0	0	2.3	0	0
13	20	0.124	0.0026045	0.14218	0.002986	0.00010937	0.00001666
20	21	0	0	0	2.3	0	0
13	22	0.124	0.0026045	0.14218	0.002986	0.00010937	0.00001666
22	23	0	0	0	2.3	0	0
13	24	0.124	0.0026045	0.14218	0.002986	0.00010937	0.00001666
24	25	0	0	0	2.3	0	0
2	26	0.0404	0.0000669	0.150857	0.0002498	6.78857E-05	0.000983
26	27	0	0	0	0.15	0	0
27	28	0.047	0.000148	0.123316	0.00038855	0.00016217	0.00001482
28	29	0.124	0.0022789	0.14218	0.002613	0.00010937	0.00001458

Bus nl	Bus nr	R/KM	R p.u.	X/KM	X p.u.	Y/KM	½ B p.u.
29	30	0	0	0	0.65	0	0
28	31	0.124	0.002605	0.14218	0.002986	0.00010937	0.00001666
31	32	0	0	0	2.3	0	0
28	33	0.124	0.002605	0.14218	0.002986	0.00010937	0.00001666
33	34	0	0	0	2.3	0	0
28	35	0.124	0.002605	0.14218	0.002986	0.00010937	0.00001666
35	36	0	0	0	2.3	0	0
28	37	0.124	0.002605	0.14218	0.002986	0.00010937	0.00001666
37	38	0	0	0	2.3	0	0
28	39	0.124	0.002605	0.14218	0.002986	0.00010937	0.00001666
39	40	0	0	0	2.3	0	0
13	41	0.0754	0.011224	0.132	0.01976	6.7886E-05	0.0001425
41	42	0	0	0	0.65	0	0
13	43	0.0754	0.021855	0.13275	0.03848	0.000132	0.00006938
43	44	0	0	0	2.875	0	0
13	45	0.0754	0.020628	0.13275	0.036319	0.000132	0.00006548
45	46	0	0	0	2.875	0	0
13	47	0.0754	0.021301	0.13275	0.037504	0.000132	0.000067621
47	48	0	0	0	2.875	0	0
28	49	0.0754	0.011205	0.13275	0.019728	0.000132	0.0001422
49	50	0	0	0	0.65	0	0
28	51	0.0754	0.021776	0.13275	0.03834	0.000132	0.000069129
51	52	0	0	0	2.875	0	0
28	53	0.0754	0.020628	0.13275	0.036319	0.000132	0.000065485
53	54	0	0	0	2.875	0	0
28	55	0.0754	0.021301	0.13275	0.037504	0.000132	0.00006762
55	56	0	0	0	2.3	0	0
13	57	0.0601	0.020119	0.128229	0.04292	0.000147	0.000357
57	58	0	0	0	0.65	0	0
13	59	0.124	0.0843	0.142	0.09668	0.0001094	0.00013487
59	60	0	0	0	2.3	0	0
13	61	0.124	0.085623	0.142	0.098178	0.0001094	0.00013695
61	62	0	0	0	2.3	0	0
28	63	0.0601	0.02027	0.1282	0.0432	0.000147	0.0003599
63	64	0	0	0	0.65	0	0
28	65	0.124	0.08497	0.142	0.0974	0.0001093	0.0001359
65	66	0	0	0	2.3	0	0

Bus nl	Bus nr	R/KM	R p.u.	X/KM	X p.u.	Y/KM	½ B p.u.
28	67	0.124	0.08594	0.142	0.09855	0.0001093	0.0001375
67	68	0	0	0	2.3	0	0
13	69	0.0601	0.016489	0.1282	0.03518	0.000147	0.0001927
69	70	0	0	0	0.65	0	0
13	71	0.124	0.069344	0.142	0.079513	0.0001093	0.0001109
71	72	0	0	0	2.3	0	0
13	73	0.124	0.07032	0.142	0.080633	0.0001093	0.00011247
73	74	0	0	0	2.3	0	0
28	75	0.0601	0.016647	0.1282	0.035518	0.000147	0.0002955
75	76	0	0	0	0.65	0	0
28	77	0.124	0.06967	0.142	0.079886	0.0001093	0.0001114
77	78	0	0	0	2.3	0	0
28	79	0.124	0.0709	0.142	0.08138	0.000109	0.0001135
79	80	0	0	0	2.3	0	0
13	81	0.0601	0.01286	0.1282	0.027438	0.000147	0.0002283
81	82	0	0	0	0.65	0	0
13	83	0.0991	0.043451	0.13803	0.0605223	0.000120686	0.0000959
83	84	0	0	0	2.3	0	0
13	85	0.0991	0.04423	0.13803	0.0616095	0.000120686	0.00009768
85	86	0	0	0	2.3	0	0
28	87	0.0601	0.013017	0.1282	0.0277748	0.000147	0.000231
87	88	0	0	0	0.65	0	0
28	89	0.0991	0.043711	0.13803	0.060885	0.000120686	0.0000965
89	90	0	0	0	2.3	0	0
28	91	0.0991	0.044752	0.13803	0.062334	0.000120686	0.000098828
91	92	0	0	0	2.3	0	0
13	93	0.0991	0.046313	0.13803	0.064509	0.000120686	0.000102
28	94	0.0991	0.046313	0.13803	0.064509	0.000120686	0.000102
13	95	0.047	0.015449	0.123325	0.040538	0.00016217	0.000096667
95	96	0	0	0	2.3	0	0
28	97	0.047	0.015449	0.123325	0.040538	0.00016217	0.000096667
97	98	0	0	0	2.3	0	0
2	99	0.0335	0.003116	0.1433	0.013329	0.00007165	0.05828
99	100	0.0625	4.02E-05	0.30171	0.000194	0.0000565	0.00032
100	101	0	0	0	0.18	0	0

Bus nl	Bus nr	R/KM	R p.u.	X/KM	X p.u.	Y/KM	½ B p.u.
101	102	0.047	0.000197	0.1233	0.000518	0.000162	0.00003089
102	103	0.0991	0.00208	0.13803	0.002899	0.0001206	0.000004597
103	104	0	0	0	5.75	0	0
102	105	0.0991	0.43711	0.13803	0.60885	0.0001206	0.000965
102	106	0.387	1.2802	0.16405	0.54272	0.0000792	0.000475
102	107	0.0991	0.1301	0.13803	0.1812	0.0001206	0.000287
102	108	0.0991	0.1457	0.13803	0.20295	0.0001206	0.0003217
102	109	0.0991	0.1613	0.13803	0.22469	0.0001206	0.000356
102	110	0.0991	0.07285	0.13803	0.10147	0.0001206	0.00016088
99	111	0.0625	2.84E-05	0.30171	0.000137	0.0000565	0.0002244
111	112	0	0	0	0.18	0	0
112	113	0.047	0.000148	0.12333	0.000389	0.000162	0.000023163
113	114	0.0991	0.00234	0.13803	0.00326	0.0001207	0.0000517
114	115	0	0	0	5.75	0	0
113	116	0.0991	0.5412	0.13803	0.7538	0.0001207	0.001195
113	117	0.387	0.3048	0.16406	0.1292	0.0000792	0.000113
113	118	0.0991	0.325338	0.13803	0.4532	0.00012	0.00071846
113	119	0.0991	0.1405	0.13803	0.1957	0.00012	0.00031027
113	120	0.0991	0.16132	0.13803	0.2247	0.00012	0.000356
113	121	0.0991	0.16132	0.13803	0.2247	0.00012	0.000356
113	122	0.0991	0.13009	0.13803	0.1812	0.00012	0.000287
99	123	0.0335	0.00147	0.14331	0.00632	0.00007165	0.02767
123	124	0.0625	4.02E-05	0.30171	0.000194	0.00005657	0.00031796
124	125	0	0	0	0.18	0	0
125	126	0.047	0.000197	0.1233	0.000518	0.000162	0.00003088
126	127	0.0991	0.00298	0.13803	0.0029	0.0001206	4.5966E-06
127	128	0	0	0	7.67	0	0
126	129	0.387	0.46739	0.16405	0.19813	0.0000792	0.00017345
126	130	0.0991	0.1405	0.13803	0.1957	0.0001206	0.00031027
126	131	0.0991	0.150908	0.13803	0.210197	0.0001206	0.000333
126	132	0.0991	0.17693	0.13803	0.246438	0.0001206	0.0003907
126	133	0.0991	0.078056	0.13803	0.10872	0.0001206	0.0001723
126	134	0.0991	0.08325	0.13803	0.11597	0.0001206	0.0001838

Bus nl	Bus nr	R/KM	R p.u.	X/KM	X p.u.	Y/KM	½ B p.u.
123	135	0.0625	2.83E-05	0.30171	0.000137	0.0000565	0.00022444
135	136	0	0	0	0.18	0	0
136	137	0.047	0.000148	0.12333	0.000389	0.000162	0.000023163
137	138	0.0991	0.00234	0.13803	0.00326	0.00012	0.00000517
138	139	0	0	0	7.67	0	0
137	140	0.387	0.2235	0.164057	0.09476	0.0000792	0.000082955
137	141	0.0991	0.15091	0.13803	0.210197	0.0001206	0.000333
137	142	0.0991	0.07806	0.13803	0.108722	0.0001206	0.000172
137	143	0.0991	0.1769	0.13803	0.2464	0.0001206	0.0003907
137	144	0.0991	0.1301	0.13803	0.1812	0.0001206	0.000287
123	145	0.0335	0.00119	0.14331	0.00509	0.00007165	0.0222701
2	145	0.0335	0.005699	0.14331	0.02438	0.00007165	0.106612
145	146	0.0625	4.01E-05	0.30171	0.000194	0.0000565	0.000317
146	147	0	0	0	0.18	0	0
147	148	0.047	0.000197	0.12332	0.000518	0.000162	0.00003088
148	149	0.0991	0.00208	0.13803	0.002899	0.0001207	4.5966E-06
149	150	0	0	0	5.75	0	0
148	151	0.387	0.20321	0.164057	0.086146	0.0000792	0.0000754
148	152	0.0991	0.07806	0.13803	0.108722	0.0001207	0.000172
148	153	0.0991	0.18733	0.13803	0.26093	0.0001207	0.0004137
148	154	0.0991	0.59047	0.13803	0.82245	0.0001207	0.0001304
148	155	0.0991	0.37467	0.13803	0.523318	0.0001207	0.0008297
148	156	0.0991	0.19254	0.13803	0.26818	0.0001207	0.000425
145	157	0.0625	0.0000284	0.30171	0.000136	0.0000567	0.0002244
157	158	0	0	0	0.18	0	0
158	159	0.047	0.000148	0.12332	0.000386	0.000162	0.00002316
159	160	0.0991	0.00234	0.13803	0.00326	0.0001207	0.00000517
160	161	0	0	0	5.75	0	0
159	162	0.387	0.54867	0.164057	0.232595	0.0001207	0.0002036
159	163	0.0991	0.47895	0.13803	0.66712	0.0001207	0.001057
159	164	0.0991	0.10407	0.13803	0.14496	0.0001207	0.000229
159	165	0.0991	0.156112	0.13803	0.21744	0.0001207	0.000344
159	166	0.0991	0.281001	0.13803	0.391402	0.0001207	0.00062055
159	167	0.0991	0.22896	0.13803	0.318919	0.0001207	0.0005056
159	168	0.0991	0.156112	0.13803	0.21744	0.0001207	0.0003447
2	169	0.0625	0.000535	0.30171	0.002583	0.00005657	0.0042345
169	170	0	0	0	0.18	0	0
170	171	0.047	0.000197	0.123326	0.000518	0.000162	0.00003088
171	172	0.047	0.00148	0.123326	0.003885	0.000162	0.00000926
172	173	0	0	0	2.875	0	0

Bus nl	Bus nr	R/KM	R p.u.	X/KM	X p.u.	Y/KM	½ B p.u.
171	174	0.047	0.00148	0.123326	0.003885	0.000162	0.00000926
174	175	0	0	0	0.65	0	0
171	176	0.047	0.00148	0.123326	0.003885	0.000162	0.00000926
176	177	0	0	0	2.3	0	0
171	178	0.0601	0.008079	0.12822	0.017237	0.000147	0.00003585
2	179	0.0625	0.000502	0.30171	0.0025506	5.66E-05	0.004182
179	180	0	0	0	0.18	0	0
180	181	0.047	0.000197	0.12332	0.000518	0.000162	0.00003088
181	182	0.047	0.00148	0.12332	0.003885	0.000162	9.265E-06
182	183	0	0	0	2.875	0	0
181	184	0.047	0.00148	0.12332	0.003885	0.000162	9.265E-06
184	185	0	0	0	0.65	0	0
181	186	0.047	0.00148	0.12332	0.003885	0.000162	9.265E-06
186	187	0	0	0	2.3	0	0
181	188	0.0601	0.008836	0.12822	0.01885	0.000147	3.9215E-05
2	189	0.0625	0.000488	0.30171	0.0023589	5.66E-05	0.00386
189	190	0	0	0	0.18	0	0
190	191	0.047	0.002813	0.123325	0.00738	0.000162	0.000281
2	192	0.0625	0.000481	0.30171	0.0023202	5.66E-05	0.003804
192	193	0	0	0	0.16	0	0
193	194	0.047	0.00269	0.123325	0.00705	0.000162	0.000269
2	195	0.0625	0.0005	0.30171	0.002416	5.66E-05	0.00396
195	196	0	0	0	0.16	0	0
196	197	0.047	0.002591	0.123325	0.0067996	0.000162	0.0002594
2	198	0.0625	0.000493	0.30171	0.0023818	5.66E-05	0.003905
198	199	0	0	0	0.16	0	0
199	200	0.047	0.003073	0.123325	0.00806	0.000162	0.0003076
2	201	0.0625	0.000514	0.30171	0.0024798	5.66E-05	0.004066
201	202	0	0	0	0.16	0	0
202	203	0.047	0.003702	0.123325	0.00971	0.000162	0.0003706
2	204	0.0625	0.000614	0.30171	0.0029658	5.66E-05	0.004863
204	205	0	0	0	0.18	0	0
205	206	0.047	0.000204	0.123325	0.000534	0.000162	0.00002038
206	207	0.0601	0.002209	0.128229	0.004713	0.000147	0.000009803
207	208	0	0	0	0.65	0	0
208	209	0.0754	0.00174	0.23194	0.00536	0.0001697	0.000001468

Bus nl	Bus nr	R/KM	R p.u.	X/KM	X p.u.	Y/KM	½ B p.u.
206	210	0.0991	0.002601	0.13803	0.00362	0.0001206	5.7458E-06
210	211	0	0	0	2.3	0	0
211	212	0.0626	0.058868	0.089	0.083695	0	0
206	213	0.0991	0.002601	0.13803	0.00362	0.0001206	5.7458E-06
213	214	0	0	0	2.3	0	0
214	215	0.0626	0.086038	0.089	0.122323	0	0
206	216	0.0991	0.0044232	0.13803	0.006161	0.0001206	9.7679E-06
216	217	0	0	0	2.3	0	0
217	218	0.0626	0.090567	0.089	0.12876	0	0
2	219	0.0625	0.000646	0.30171	0.003118	0.00005657	0.0051137
219	220	0	0	0	0.18	0	0
220	221	0.047	0.000148	0.123326	0.000389	0.000162	0.000023163
221	222	0.0601	0.011677	0.128229	0.024913	0.000147	0.00005182
221	223	0.0754	0.006928	0.13275	0.012199	0.000132	8.79833E-05
2	224	0.0625	0.000614	0.30171	0.002965	0.00005657	0.004863
224	225	0	0	0	0.18	0	0
225	226	0.047	0.000162	0.123326	0.000427	0.000162	0.000025479
226	227	0.0991	0.0046833	0.13803	0.00652	0.0001206	0.00001034
227	228	0	0	0	2.3	0	0
228	229	0.0626	0.10868	0.089	0.154514	0	0
226	230	0.0991	0.003122	0.13803	0.004349	0.000120069	0.000006895
230	231	0	0	0	2.3	0	0
231	232	0.0626	0.058868	0.089	0.083695	0	0
226	233	0.047	0.00185	0.123326	0.00485	0.000162	0.00000319
233	234	0	0	0	0.65	0	0
234	235	0.0754	0.0019171	0.23194	0.005897	0.0001697	0.000001615
226	236	0.0991	0.0026019	0.13803	0.003624	0.0001207	0.00000574
236	237	0	0	0	2.3	0	0
237	238	0.0626	0.0815104	0.089	0.115885	0	0
2	239	0.0625	0.000646	0.301714	0.0031187	0.00005657	0.0051136
239	240	0	0	0	0.18	0	0
240	241	0.047	0.000197	0.123326	0.000518	0.00016217	0.00003088
241	242	0.0601	0.014359	0.128228	0.030636	0.00014709	0.0000637
241	243	0.0601	0.006943	0.128228	0.014813	0.00014709	0.000123
2	244	0.0625	0.000518	0.301714	0.0025004	0.00005657	0.0040999
244	245	0	0	0	0.18	0	0
245	246	0.047	0.000197	0.123326	0.000518	0.00016217	0.00003088
246	247	0.047	0.005923	0.123326	0.01554	0.00016217	0.00003706
246	248	0.047	0.005503	0.123326	0.014441	0.00016217	0.000034435
246	249	0.047	0.0045904	0.123326	0.012045	0.00016217	0.000028722
246	250	0.047	0.0014807	0.123326	0.003885	0.00016217	0.00000926

Bus nl	Bus nr	R/KM	R p.u.	X/KM	X p.u.	Y/KM	½ B p.u.
250	251	0	0	0	2.875	0	0
2	252	0.0625	0.000494	0.301714	0.002386	0.00005657	0.0039128
252	253	0	0	0	0.18	0	0
253	254	0.047	0.000197	0.123326	0.000518	0.00016217	0.00003088
254	255	0.047	0.005923	0.123326	0.01554	0.00016217	0.00003706
254	256	0.047	0.005947	0.123326	0.015606	0.00016217	0.000037215
254	257	0.047	0.005034	0.123326	0.013216	0.00016217	0.0000315
254	258	0.047	0.0014807	0.123326	0.003885	0.00016217	0.00000926
258	259	0	0	0	2.875	0	0
2	260	0.0625	0.0005302	0.301714	0.0025597	0.00005657	0.004197
260	261	0	0	0	0.18	0	0
261	262	0.047	0.000197	0.123326	0.0005181	0.00016217	0.00003088
262	263	0.047	0.005923	0.123326	0.01554	0.00016217	0.00003706
262	264	0.047	0.005923	0.123326	0.01554	0.0001622	0.00003706
262	265	0.047	0.005923	0.123326	0.01554	0.0001622	0.00003706
262	266	0.047	0.0014807	0.123326	0.003885	0.0001622	0.00000926
266	267	0	0	0	2.875	0	0
2	268	0.0625	0.0004806	0.301714	0.00232	5.657E-05	0.003804
268	269	0	0	0	0.18	0	0
269	270	0.047	0.000197	0.123326	0.000518	0.0001622	0.00003088
270	271	0.047	0.005923	0.123326	0.01554	0.0001622	0.00003706
270	272	0.047	0.005923	0.123326	0.01554	0.0001622	0.00003706
270	273	0.047	0.0014807	0.123326	0.003885	0.0001622	0.00000926
273	274	0	0	0	2.875	0	0
270	275	0.047	0.0014807	0.123326	0.003885	0.0001622	0.00000926

APPENDIX V

The MATLAB Program Codes for Reading The System Line and Transformer Impedance Data (LdataR.m). [Refer to section 5.2.2]

```
%**********************************************************************
% This Sub-Program is for Reading the System Line Data Excel file sheets
% and developing them to be ready for the NEWTON- RAPHSON Power Flow using
% MATLAB
%**********************************************************************

% Reading the 1st System Line Data Sheet Line-1
LD1 = xlsread('Line Data.xlsx', 'Line-1','a4:i31');

% Reading the 2nd System Line Data Sheet Line-2
LD2 = xlsread('Line Data.xlsx', 'Line-2','a4:i32');

% Reading the 3rd System Line Data Sheet Line-3
LD3 = xlsread('Line Data.xlsx', 'Line-3','a4:i32');

% Reading the 4th System Line Data Sheet Line-4
LD4 = xlsread('Line Data.xlsx', 'Line-4','a4:i33');

% Reading the 5th System Line Data Sheet Line-5
LD5 = xlsread('Line Data.xlsx', 'Line-5','a4:i34');

% Reading the 6th System Line Data Sheet Line-6
LD6 = xlsread('Line Data.xlsx', 'Line-6','a4:i33');

% Reading the 7th System Line Data Sheet Line-7
LD7 = xlsread('Line Data.xlsx', 'Line-7','a4:i32');

% Reading the 8th System Line Data Sheet Line-8
LD8 = xlsread('Line Data.xlsx', 'Line-8','a4:i32');

% Reading the 9th System Line Data Sheet Line-9
LD9 = xlsread('Line Data.xlsx', 'Line-9','a4:i31');

% Reading the 10th System Line Data Sheet Line-10
LD10 = xlsread('Line Data.xlsx', 'Line-10','a4:i15');

%**********************************************************************
% After Reading all sheets, the read data is posted in a Matrix called
% Line Data Matrix (LData). The LData contains the None Needed Columns
LData=[LD1;LD2;LD3;LD4;LD5;LD6;LD7;LD8;LD9;LD10];
% The following commands will Delete the None-Needed Column from the LData
% Matrix (Column 3, 5 & 7)
LData(:,3)=[];
LData(:,4)=[];
LData(:,5)=[];
% The Final version of the Line Data Matrix (LData) is now ready
LData;
```

APPENDIX VI

The MATLAB Program Codes for Reading The Base Values (Base_Value.m). [Refer to section 5.2.3]

```
%*************************************************************************
% This Sub-program will assign all the base values and the power flow
% accepted accuracy and maximum number of iterations
%*************************************************************************

%-------------------------------------------------------------------------
% Identifying the MVA base value; the MVA base value was assigned to the
% variable basmva. The MVA value was chosen to be 100MVA
basemva=100;
%-------------------------------------------------------------------------
% Identifying the Maximum mismatching error between the bus voltages
% resulted from two consecutive iterations; this value was assigned
% to the variable accuracy. The accuracy value was chosen to be 0.0001.
accuracy=0.0001;
%-------------------------------------------------------------------------
% Identifying the Maximum number of iterations for the Newtown Raphson
% power flow method. This value was assigned to the variable maxiter.
% The number of iteration was chosen to be 200
maxiter=100;
```

APPENDIX VII

The MATLAB Program Codes For Assigning The Generator, Utlity and Synchronous Motor Bus Voltages (Gen_SyM_Volt.m). [Refer to section 5.2.4]

```
%**************************************************************************
% Assigning the Utility, Generators and Synchronous Motors Buses Voltage
%**************************************************************************

%------------------Assigning the Utility Bus Voltage (UBV)------------------
% Assigning The Utility Bus Voltage (UBV) (1.0-0.03 p.u.)
UBV=1;
% Insert the UBV value in the SData Matrix
SData(1,3)= UBV;
%--------------------------------------------------------------------------

%----- Assigning the Gas Turbine Generators Terminal Bus Voltage(GTG_VT) ---
% The GTG terminal bus has ±5% Voltage Regulation Bandwidth. The terminal
% voltage will have ± 0.01 Steps in p.u.; ± 5 steps.
% A variable GTG_TV will be assigned to represents the GTG terminal bus
% voltage

% Assigning the GTG_TV variable
GTG_TV=[(1-0.05); (1-0.04);  (1-0.03);(1-0.02);(1-0.01);(1.0); (1+0.01);
        (1+0.02); (1+0.03); (1+0.04); (1+0.05)];
% Flip the GTG_TV variable column array to make it One Row Array
GTG_TV=GTG_TV';
%--------------------------------------------------------------------------

%-----Assigning the Steam Turbine Generators Terminal Bus Voltage(STG_VT)---
% Although the STG terminal bus has ±10 Voltage Regulation Bandwidth, it was
% limited to ±5% Voltage Regulation Bandwidth to reduce the system voltage
% stress. The Voltage will have ± 0.01 Steps in p.u.; ± 5 steps
% A variable STG_TV will be assigned to represents the STG Terminal Voltage

% Assigning the STG_TV variable
STG_TV=[(1-0.05); (1-0.04);  (1-0.03);(1-0.02);(1-0.01);(1.0); (1+0.01);
        (1+0.02); (1+0.03);  (1+0.04); (1+0.05)];
% Flip the STG_TV variable column array to make it One Row Array
STG_TV=STG_TV';
%-------------------------------------------------------------------------

%--------------Synchronous Motors Terminal Bus Voltage (SyM_TV)-------------
% The Synchronous Motors Terminal bus has ±10% Voltage Regulation Bandwidth.
% Yet, it was set at -10% and +5% Voltage Regulation Bandwidth. This reduce
% the system stress.
% The Voltage will have ± 0.01 Steps in p.u.; -10 steps and +5 steps
% A variable SyM_TV will be assigned to represents the SyM Terminal Voltage

% Assigning the SyM_TV variable
```

```
SyM_TV=[(1-0.10); (1-0.09);(1-0.08); (1-0.07);(1-0.06);(1-0.05); (1-0.04);
        (1-0.03);(1-0.02);(1-0.01);(1.0); (1+0.01);(1+0.02); (1+0.03);
        (1+0.04);(1+0.05)];
% Flip the SyM_TV variable column array to make it One Row Matrix
SyM_TV=SyM_TV';
%------------------------------------------------------------------------

%************************************************************************
% Inserting the Generators and Synchronous Motor Terminal Buses Voltage in
% the System Data Matrix
%*************************************************************************

%-----------------------------------------------------------------------
% Assigning the GTG terminal bus voltage. V_GTG variable is used to identify
% the GTG_VT p.u. value within the GTG_TV array
V_GTG=6;

%-----------------------------------------------------------------------
% Assigning the STG terminal bus voltage. V_STG variable is used to identify
% the STG_VT p.u. value within the STG_TV array
V_STG=6;

%-----------------------------------------------------------------------
% Assigning the Captive Synchronous Motor SyCM Terminal Voltage. V_SyCM
% variable is used to identify the SyM_VT p.u. value within the SyM_TV array
V_SyCM=9;

%-----------------------------------------------------------------------
% Assigning Substation#9 Synchronous Motor SyM Terminal Voltage. V_Sy9M
% variable is used to identify the SyM_VT p.u. value within the SyM_TV array
V_Sy9M=8;

%-----------------------------------------------------------------------
%-Inserting the GTG, STG and SyM Terminal Bus Voltage in the SData Matrix--
% Inserting The GTG Terminal Voltage in p.u.
SData(3,3)= GTG_TV(V_GTG);
SData(7,3)= GTG_TV(V_GTG);

% Inserting The STG Terminal Voltage in p.u.
SData(5,3)= STG_TV(V_STG);
SData(9,3)= STG_TV(V_STG);

% Inserting The Captive Synchronous Motors Terminal Voltage in p.u.
SData(191,3)= SyM_TV(V_SyCM);
SData(194,3)= SyM_TV(V_SyCM);
SData(197,3)= SyM_TV(V_SyCM);
SData(200,3)= SyM_TV(V_SyCM);
SData(203,3)= SyM_TV(V_SyCM);

% Inserting Substation#9 Synchronous Motors Terminal Voltage in p.u.
SData(223,3)= SyM_TV(V_Sy9M);
SData(243,3)= SyM_TV(V_Sy9M);

%----- Clear all none needed Variables-----------------------------------
clear V_SyM V_STG V_GTG STG_TV GTG_TV SyM_TV
```

APPENDIX VIII

The MATLAB Program Codes for Assigning The Transformer Taps' Values (TR_Tap.m). [Refer to section 5.2.5]

```
%************************************************************************
% Reading and Assigning the Transformer Tap Value
%************************************************************************

% ------------------------- Large Power Transformer-------------------------
% The Large Power Transformer with 115kV primary Voltage has ±10% Voltage
% Regulation Bandwidth. Their tap changer has ±16 Tap Steps each step can
% adjust the bus voltage by 0.625% (of the nominal voltage).
% The code 16 was used to identify all large Substations Transformers
% The code 22 was used to identify Substation#2 Transformers
% Two arrays were assigned to these Transformer Taps
% Tap33_Up is an array assigned for the upper taps (+16 Taps)
% Tap33_Down is an array assigned for the lower taps (-16 Taps)
% Tap33 is an array assigned for all up taps, down taps and neutral tap

% Assigning the Upper Taps to Tap33_Up array
Tap33_Up=(0.00625*[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16])+1;
% Assigning the Lower Taps to Tap33_Down array
Tap33_Down=(1-(0.00625*[16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1]));
% Assigning all the Taps to Tap33 array
Tap33=[Tap33_Down 1 Tap33_Up];
%--------------------------------------------------------------------------

% -------------------Causeway Substations Main Transformers----------------
% It was decided to segregate the Causeway Main Substation Transformers
% from the remaining large power transformers. This will give us flexibility
% to assign these transformers taps independently.
% The code 161 was used to identify Causeway Substation#1 Transformers
% The code 162 was used to identify Causeway Substation#2 Transformers
% The code 163 was used to identify Causeway Substation#3 Transformers
% Two arrays were assigned to these Transformer taps
% Tap33_CUp is the array assigned for the upper taps (+16 Taps)
% Tap33_CDown is the array assigned for the lower taps (-16 Taps)
% Tap33_C is the array assign for all up taps, down taps and neutral tap

% Assigning the Upper Taps to Tap33_Up array
Tap33_CUp=(0.00625*[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16])+1;
% Assigning the Lower Taps to Tap33_Down array
Tap33_CDown=(1-(0.00625*[16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1]));
% Assigning all the Taps to Tap33 array
Tap33C=[Tap33_Down 1 Tap33_Up];

%--------------------------------------------------------------------------

% ---------Captive Synchronous Motors and Distribution Transformers--------
% The Distribution Transformers with < 115kV primary Voltage has ±5 Voltage
% The code 51 was used to identify captive synchronous motors Transformers
% The code 5 was used to identify Distribution Transformers
% Regulation Bandwidth. They have ±2 Tap Steps each step with (2.5%) Voltage
```

```
% regulation
% Two arrays were assigned to these Transformers taps
% Tap5_Up is the array for the upper taps (+2 Taps)
% Tap5_Down is the array for the lower taps (-2 Taps)
% Tap5 is the array assign for all up taps, down taps and neutral tap

% Assigning the Upper Taps to Tap5_Up array
Tap5_Up=(0.025*[1 2])+1;
% Assigning the lower Taps to Tap5_Down array
Tap5_Down=(1-(0.025*[2 1]));
% Assigning all the Taps to Tap33 array
Tap5=[Tap5_Down 1 Tap5_Up];
%---------------------------------------------------------------------------

% ----------------------Generator Transformer------------------------------
% The code 8 was used to identify Generators step-up Transformers
% The Generator Step-Up Transformers have ±10% Voltage Regulation Bandwidth.
% They have ±8 Tap Steps each step with (1.25%) Voltage Regulation
% Two arrays were assigned to these Transformers taps
% TapG_Up is the array assigned for the upper taps (+8 Taps)
% TapG_Down is the array assigned for the lower taps (-8 Taps)
% TapG is the array assigned for all up taps, down taps and neutral tap

% Assigning the Upper Taps to Tap33_Up array
TapG_Up=(0.0125*[1 2 3 4 5 6 7 8])+1;
% Assigning the Lower Taps to Tap33_Down array
TapG_Down=(1-(0.0125*[8 7 6 5 4 3 2 1]));
% Assigning all the Taps to Tap33 array
TapG=[TapG_Down 1 TapG_Up];

%---------------------------------------------------------------------------
% Clear all Non used variables
clear Tap33_Up Tap33_Down Tap5_Up Tap5_Down TapG_Up TapG_Down
clear Tap33_CUp Tap33_CDown Tap33_C

%************************************************************************

%*************************************************************************
% Inserting the Transformer Tap Value in the Line Data Matrix
% The taps values can be changed
%*************************************************************************

L_LData=length(LData(:,1));
for x=1:L_LData
    if      LData(x,6)== 22
            LData(x,6)= Tap33(14);
            % Sub#2 Transformers Tap Code 16
            % Tap assign to Tap 14 (Post Voltage Up)
    elseif  LData(x,6)== 16
            LData(x,6)= Tap33(16);
            % Large OLTC Transformers Tap Code 16
            % Tap assign to Tap 16 (Post Voltage Up)
    elseif  LData(x,6)== 161
            LData(x,6)= Tap33C(14);
            % Causeway Sub#1 Up Stream Transformer Tap (Code 161)
```

```
            % Tap assign to Tap 14 (Post Voltage Up)
    elseif  LData(x,6)== 162
            LData(x,6)= Tap33C(17);
            % Causeway Sub#2 Up Stream Transformer Tap Code 162
            % Tap assign to Tap position 17 (Neutral Tap)
    elseif  LData(x,6)== 163
            LData(x,6)= Tap33C(14);
            % Causeway Sub#3 Up Stream Transformer Tap Code 163
            % Tap assign to Tap position 14 (Post Voltage Up)
    elseif  LData(x,6)== 51
            LData(x,6)= Tap5(3);
            % Captive Synchronous Motors Transformers Tap Code 51
            % Tap assign to Neutral Tap
    elseif  LData(x,6)== 5
            LData(x,6)= Tap5(3);
            % Distribution Transformers Tap Code 5
            % Tap assign to Neutral Tap
    elseif  LData(x,6)== 8
            LData(x,6)= TapG(9);
            % Generators Step-Up Transformers Tap Code 8
            % Tap assign to Neutral Tap
    end
end
%------------------ Clear none used variables----------------------------
clear L_LData TapG Tap33 Tap33C Tap5

%************************************************************************
```

APPENDIX IX

The MATLAB Program Codes for Verifying The Bus Voltages Constraints (VCV.m). [Refer to section 5.4.1]

```
%*************************************************************************
% This Sub-Program will verify if the Bus Voltage Magnitude are within
% the acceptable limit as per below Table
%************************************************************************
% ___________________________________________________________________________
%  Bus Type                  | Lower Limit % in p.u | Upper Limit % in p.u.
% ___________________________________________________________________________
% Load Bus                   |       90%            |          105%
% --------------------------------------------------------------------------
% Causeway Substation        |                      |
% Load Bus                   |       95%            |          105%
% --------------------------------------------------------------------------
% Gas Turbine Generators     |                      |
% (GTG) Terminal Bus         |       95%            |          105%
% --------------------------------------------------------------------------
% Steam Turbine Generators |                        |
% (STG) Terminal Bus         |        90% (*95%)    |          110%(*105%)
%---------------------------------------------------------------------------
% * The STG voltage was limited to 95% and 105% although it can tolerates
%   90% and 110%. This is to reduce the system voltage stress
%**************************************************************************

%**************************************************************************
% Assign the above Bus Voltage Magnitude limitations to Variables

% Assigning the GTG Bus Voltage Limitation
  Vmin_GTG = 0.95;
  Vmax_GTG = 1.05;
% GTG Buses are Bus# 3 and Bus# 7
% Assigning the GTG Buses to Variable BV_GTG
  BV_GTG=[3;7];
% --------------------------------------------------------------------------
% Assigning the STG Bus Voltage Limitation
  Vmin_STG = 0.95;
  Vmax_STG = 1.05;
% STG Buses are Bus# 5 and Bus# 9
% Assigning the STG Buses to Variable BV_STG
  BV_STG=[5;9];
% --------------------------------------------------------------------------
% Assigning the Causeway Substations Bus Voltage Limitation
  Vmin_CL = 0.95;
  Vmax_CL = 1.05;

% Causeway Substations Buses are as follow:
% Causeway Substation# 1
% Upstream Buses (115kV and 13.8kV)----------------------------------------
% Bus#99/Bus#100/Bus#101/Bus#102
% Downstream Buses (13.8kV and 480V)---------------------------------------
```

```
% Bus#103/Bus#104/Bus#105/Bus#106/Bus#107/Bus#108/Bus#109/Bus#110

% Upstream Buses (115kV and 13.8kV)-----------------------------------------
% Bus#111/Bus#112/Bus#113
% Downstream Buses (13.8kV and 480V)----------------------------------------
% Bus#114/Bus#115/Bus#116/Bus#117/Bus#118/Bus#119
% Bus#120/Bus#121/Bus#122

% Assigning the Causeway Substation#1 Upstream Buses to Variable BV_CL1U and
% the Downstream Buses to Variable BV_CL1D
BV_CL1U=[99; 100; 101; 102; 111; 112; 113];
BV_CL1D=[105; 106; 107; 108; 109; 110; 116; 117; 118;
         119; 120; 121; 122];

% Causeway Substation# 2
% Upstream Buses (115kV and 13.8kV)-----------------------------------------
% Bus#123/Bus#124/Bus#125/Bus#126
% Downstream Buses (13.8kV and 480V)----------------------------------------
% Bus#127/Bus#128/Bus#129/Bus#130/Bus#131/Bus#132/Bus#133/Bus#134
% Upstream Buses (115kV and 13.8kV)-----------------------------------------
% Bus#135/Bus#136/Bus#137
% Downstream Buses (13.8kV and 480V)----------------------------------------
% Bus#138/Bus139/Bus#140/Bus#141/Bus#142/Bus#143/Bus#144

% Assigning the Causeway Substation#2 Upstream Buses to Variable BV_CL2U and
% the Downstream Buses to Variable BV_CL2D
BV_CL2U=[123; 124; 125; 126; 135; 136; 137];
BV_CL2D=[129; 130; 131; 132; 133; 134;140; 141; 142;
         143; 144];

% Causeway Substation# 3
% Upstream Buses (115kV and 13.8kV)-----------------------------------------
% Bus#145/Bus#146/Bus#147/Bus#148
% Downstream Buses (13.8kV and 480V)----------------------------------------
% Bus#149/Bus#150/Bus#151/Bus#152/Bus#153/Bus#154/Bus#155/Bus#156
% Upstream Buses (115kV and 13.8kV)-----------------------------------------
% Bus#157/Bus#158/Bus#159
% Downstream Buses (13.8kV and 480V)----------------------------------------
% Bus#160/Bus#161/Bus#162/Bus#163/Bus#164/Bus#165/Bus#166/Bus#167/Bus#168

% Assigning the Causeway Substation#3 Upstream Buses to Variable BV_CL3U and
% the Downstream Buses to Variable BV_CL3D
BV_CL3U=[145; 146; 147; 148; 157; 158; 159];
BV_CL3D=[151; 152; 153; 154; 155; 156;162; 163; 164;
         165; 166; 167; 168];
% -----------------------------------------------------------------------
% Assigning the Others Load Buses Voltage Limitation
  Vmin_L = 0.9;
  Vmax_L = 1.05;
%**************************************************************************

%**************************************************************************
% Read the Buses Voltage Magnitude (Vm)from the Newton Raphson power flow
% program Lfnewton.m). The Lfnewton.m Sub-Program is already available in
% reference [38]
  Vltbus=Vm';
```

```
% Measure the length of the Vltbus; Number of the System Buses
  n=length(Vltbus(:,1));

%*************************************************************************
% The below commands measure the length of the buses matrixes for each
% cluster of the system buses with common constraints
%*************************************************************************

% Assign the other Loads buses to Matrix BV_OL
% Assign all GTG, STG , Causeway Substations Buses to A Matrix N_OL
 N_OL=[ BV_STG; BV_GTG; BV_CL1U; BV_CL1D; BV_CL2U; BV_CL2D; BV_CL3U;
        BV_CL3D];
% Measure the length of N_OL
m=length(N_OL(:,1));
% Assign variable BV_OL to be Zero Matrix
BV_OL(1:n,1)=0;
% Identify none other load buses; GTG, STG , Causeway Substations Buses
for x=1:m
    BV_OL(N_OL(x,1),1)=N_OL(x,1);
end
% The other Load Matrix
BV_OL=find(BV_OL(:,1)==0);
clear x m

%----------------------------------------------------------------------
% Measure the length of each system parameter matrix, this is needed for
% the voltage verification loop as posted below
% Measure the length of GTG Buses Matrix; Number of the GTG Buses
  n_GTG=length(BV_GTG(:,1));
%-------------------------------------------------------------------------
% Measure the length of STG Buses Matrix; Number of the STG Buses
  n_STG=length(BV_STG(:,1));
%-------------------------------------------------------------------------
% Measure the length of BV_CL1U Buses Matrix; Number of the Causeway
% Substation#1 Upstream Buses
  n_BV_CL1U=length(BV_CL1U(:,1));
% Measure the length of BV_CL1D Buses Matrix; Number of the Causeway
% Substation#1 Downstream Buses
  n_BV_CL1D=length(BV_CL1D(:,1));
%-------------------------------------------------------------------------
% Measure the length of BV_CL1U Buses Matrix; Number of the Causeway
% Substation#2 Upstream Buses
  n_BV_CL2U=length(BV_CL2U(:,1));
% Measure the length of BV_CL1D Buses Matrix; Number of the Causeway
% Substation#2 Downstream Buses
  n_BV_CL2D=length(BV_CL2D(:,1));
%-------------------------------------------------------------------------
% Measure the length of BV_CL1U Buses Matrix; Number of the Causeway
% Substation#3 Upstream Buses
  n_BV_CL3U=length(BV_CL3U(:,1));
% Measure the length of BV_CL1D Buses Matrix; Number of the Causeway
% Substation#3 Downstream Buses
  n_BV_CL3D=length(BV_CL3D(:,1));
%-------------------------------------------------------------------------
% Measure the length of BV_OL Buses Matrix; Number of the Other Load Buses
  n_BV_OL=length(BV_OL(:,1));
```

```
%*************************************************************************


%-------- Verify the above Buses Voltage Constraints are Met------------

% The following Subroutine (loop) will check the buses voltage constraints
% and post a message in case the buses voltage constrained was not met

%*************************************************************************
%========================================================================
% Set the Logic Default value one (1) for not meeting the voltage
% constraints and Zero (0) for meeting the voltage constraints
%-----------------------------------------------------------------------
% Default value for meeting the voltage constraints
L_MV_GTG=0; L_MV_STG=0;L_MV_CL1U=0;L_MV_CL2U=0;L_MV_CL3U=0;L_MV_OL=0;
L_MV_CL1D=0;L_MV_CL2D=0;L_MV_CL3D=0;
%-----------------------------------------------------------------------


% The main Loop for verifying the Buses Voltage Constrines
for k=1:n
   % Check the GTG Buses Voltage Limitations
   for k_GTG=1:n_GTG
   if  k==BV_GTG(k_GTG,1)
   if Vltbus(k,1)> 1.05 || Vltbus(k,1)<0.95
   %=====================================================================
   % In case the bus voltage constraints are not met, the generation number,
   % the bus number and the bus voltage will be captured
    NMV_GTG(k,:)=[k Vltbus(k,1)];
   %=====================================================================
   % Develop a logic to capture the Chromosomes that Does Not meet the
   % Voltage constraints
   L_NMV_GTG=0;
   else
   %=====================================================================
   % In case the bus voltage constraints are met, the generation number, the
   % bus number and the bus voltage will be captured

    MV_GTG(k,:)=[k Vltbus(k,1)];
   % Develop a logic to capture the Chromosomes that Does meet the
   % Voltage constraints
   L_MV_GTG=1;
   %======================================================================
   end
   end
   end
   %----------------------------------------------------------------------
   % Check the STG Buses Voltage Limitations
   for k_STG=1:n_STG
   if  k==BV_STG(k_STG,1)
   if Vltbus(k,1)> 1.05 || Vltbus(k,1)<0.95
   %=====================================================================
   % In case the bus voltage constraints are not met, the generation number,
   % the bus number and the bus voltage will be captured
    NMV_STG(k,:)=[ k Vltbus(k,1)];
```

```
%==================================================================
% Develop a logic to capture the Chromosomes that Does Not meet the
% Voltage constraints
L_NMV_STG=0;
else
%==================================================================
% In case the bus voltage constraints are met, the generation number,the
% bus number and the bus voltage will be captured
 MV_STG(k,:)=[k Vltbus(k,1)];
% Develop a logic to capture the Chromosomes that Does meet the
% Voltage constraints
L_MV_STG=1;
%===================================================================
end
end
end

%-------------------------------------------------------------------
% Check the Causeway Sub#1 Upstream Buses Voltage Limitations
for k_CL1U=1: n_BV_CL1U
if  k==BV_CL1U(k_CL1U,1)%||Vltbus(k,1)> 1.05 || Vltbus(k,1)<0.9
if Vltbus(k,1)> 1.05 || Vltbus(k,1)<0.9
%==================================================================
% In case the bus voltage constraints are not met, the generation number,
% the bus number and the bus voltage will be captured
 NMV_CL1U(k,:)=[k Vltbus(k,1)];
%==================================================================
%==================================================================
% Develop a logic to capture the Chromosomes that Does Not meet the
% Voltage constraints
L_NMV_CL1U=0;

else
%==================================================================
% In case the bus voltage constraints are met, the generation number, the
% bus number and the bus voltage will be captured
 MV_CL1U(k,:)=[k Vltbus(k,1)];
% Develop a logic to capture the Chromosomes that Does meet the
% Voltage constraints
L_MV_CL1U=1;
%==================================================================
end
end
end
%-------------------------------------------------------------------
% Check the Causeway Sub#2 Upstream Buses Voltage Limitations
for k_CL2U=1: n_BV_CL2U
if  k==BV_CL2U(k_CL2U,1)
if Vltbus(k,1)> 1.05 || Vltbus(k,1)<0.9
%==================================================================
% In case the bus voltage constraints are not met, the generation number,
% the bus number and the bus voltage will be captured
 NMV_CL2U(k,:)=[k Vltbus(k,1)];
%==================================================================
%==================================================================
% Develop a logic to capture the Chromosomes that Does Not meet the
```

```
% Voltage constraints
L_NMV_CL2U=0;

else
%=====================================================================
% In case the bus voltage constraints are met, the generation number, the
% bus number and the bus voltage will be captured
 MV_CL2U(k,:)=[k Vltbus(k,1)];
%======================================================================
% Develop a logic to capture the Chromosomes that Does meet the
% Voltage constraints
L_MV_CL2U=1;
%=====================================================================
end
end
end
%-----------------------------------------------------------------------
% Check the Causeway Sub#3 Upstream Buses Voltage Limitations
for k_CL3U=1: n_BV_CL3U
if  k==BV_CL3U(k_CL3U,1)
if Vltbus(k,1)> 1.05 || Vltbus(k,1)<0.9
%=====================================================================
% In case the bus voltage constraints are not met, the generation number,
% the bus number and the bus voltage will be captured
 NMV_CL3U(k,:)=[k Vltbus(k,1)];
%=====================================================================
% Develop a logic to capture the Chromosomes that Does Not meet the
% Voltage constraints
L_NMV_CL3U=0;
else
%=====================================================================
% In case the bus voltage constraints are met, the generation number, the
% bus number and the bus voltage will be captured
 MV_CL3U(k,:)=[k Vltbus(k,1)];
%======================================================================
% Develop a logic to capture the Chromosomes that Does meet the
% Voltage constraints
L_MV_CL3U=1;
%=====================================================================
end
end
end
%----------------------------------------------------------------------
% Check the Causeway Sub#1 Downstream Buses Voltage Limitations
for k_CL1D=1: n_BV_CL1D
if  k==BV_CL1D(k_CL1D,1)
if Vltbus(k,1)> 1.05 || Vltbus(k,1)<0.95
%=====================================================================
% In case the bus voltage constraints are not met, the generation number,
% the bus number and the bus voltage will be captured
 NMV_CL1D(k,:)=[k Vltbus(k,1)];
%=====================================================================
%=====================================================================
% Develop a logic to capture the Chromosomes that Does Not meet the
% Voltage constraints
L_NMV_CL1D=0;
else
```

```
%=====================================================================
% In case the bus voltage constraints are met, the generation number, the
% bus number and the bus voltage will be captured
MV_CL1D(k,:)=[k Vltbus(k,1)];
%=====================================================================
% Develop a logic to capture the Chromosomes that Does meet the
% Voltage constraints
L_MV_CL1D=1;
%=====================================================================
end
end
end

%-----------------------------------------------------------------------
% Check the Causeway Sub#2 Downstream Buses Voltage Limitations
for k_CL2D=1: n_BV_CL2D
if  k==BV_CL2D(k_CL2D,1)
if Vltbus(k,1)> 1.05 || Vltbus(k,1)<0.95
%=====================================================================
% In case the bus voltage constraints are not met, the generation number,
% the bus number and the bus voltage will be captured
 NMV_CL2D(k,:)=[k Vltbus(k,1)];
%=====================================================================
%=====================================================================
% Develop a logic to capture the Chromosomes that Does Not meet the
% Voltage constraints
L_NMV_CL2D=0;
else
%=====================================================================
% In case the bus voltage constraints are met, the generation number, the
% bus number and the bus voltage will be captured

MV_CL2D(k,:)=[k Vltbus(k,1)];
%=====================================================================
% Develop a logic to capture the Chromosomes that Does meet the
% Voltage constraints
L_MV_CL2D=1;
%=====================================================================
end
end
end

%-----------------------------------------------------------------------
% Check the Causeway Sub#3 Downstream Buses Voltage Limitations
for k_CL3D=1: n_BV_CL3D
if  k==BV_CL3D(k_CL3D,1)
if Vltbus(k,1)> 1.05 || Vltbus(k,1)<0.95
%=====================================================================
% In case the bus voltage constraints are not met, the generation number,
% the bus number and the bus voltage will be captured
 NMV_CL3D(k,:)=[k Vltbus(k,1)];
%=====================================================================
%=====================================================================
% Develop a logic to capture the Chromosomes that Does Not meet the
% Voltage constraints
L_NMV_CL3D=0;
else
```

```
%=====================================================================
% In case the bus voltage constraints are met, the generation number, the
% bus number and the bus voltage will be captured
MV_CL3D(k,:)=[k Vltbus(k,1)];
%=====================================================================
% Develop a logic to capture the Chromosomes that Does meet the
% Voltage constraints
L_MV_CL3D=1;
%=====================================================================
end
end
end
%---------------------------------------------------------------------
% Check the Others (not causeway substations buses) System Load Buses
% Voltage Limitations together with violation buses
for k_OL=1: n_BV_OL
if  k==BV_OL(k_OL,1)
if Vltbus(k,1)> 1.05 || Vltbus(k,1)<0.9
 %=====================================================================
% In case the bus voltage constraints are not met, the generation number,
% the bus number and the bus voltage will be captured
 NMV_OL(k,:)=[k Vltbus(k,1)];
%=====================================================================
%=====================================================================
% Develop a logic to capture the Chromosomes that Does Not meet the
% Voltage constraints
L_NMV_OL=0;
else
%=====================================================================
% In case the bus voltage constraints are met, the generation number, the
% bus number and the bus voltage will be captured
MV_OL(k,:)=[k Vltbus(k,1)];
%=====================================================================
% Develop a logic to capture the Chromosomes that Does meet the
% Voltage constraints
L_MV_OL=1;
%=====================================================================
end
end
end
end
%---------------------------------------------------------------------
```

APPENDIX X

The MATLAB Program Codes for Verifying the Reactive Power Constraints (MAVR_C.m). [Refer to section 5.4.2]

```
%***************************************************************************
% This Sub-Program will verify if the Generators and Synchronous Motors
% MVAR limits are within the acceptable as per below Table
%***************************************************************************
% ___________________________________________________________________________
%  Machine Type                     | Lower MVAR Limit | Upper MVAR Limit
% ___________________________________________________________________________
% GTG                               |      -62.123     |      95.72
% ---------------------------------------------------------------------------
% STG (43 MW) - Bus 5               |       -22.4      |      20.919
% ---------------------------------------------------------------------------
% STG (86 MW) - Bus 9               |       -41.9      |      53.837
% ---------------------------------------------------------------------------
% Captive Synchronous Motors        |       -6.96      |     16.42
%----------------------------------------------------------------------------
% None-Captive Synchronous Motors |       -6.104     |     2.4411
%---------------------------------------------------------------------------

%-------- Verify the above Buses Voltage Constraints are Met------------
% The following Subroutine (loop) will check the MVAR constraints
% and post a message in case the MVAR constrained was not met
%----------------------------------------------------------------------------
% The buses number and the reactive power values (Qg) are obtained from
% the Newton Raphson power flow program developed in reference [38]
% Measure the length of the Vltbus; Number of the System Buses
  n=length(Vltbus(:,1));
%****************************************************************************
for k=1:n
   if k==3 || k==7;
   % Check the GTG Buses MVAR Limitations
   if Qg(1,k)<= -62.123 || Qg(1,k)>= 95.72
   % In case the MVAR limit is not met, the below message will be
   % posted together with violation buses
   disp('The Below GTG MVAR is not meeting MVAR Constraints')
   NMVR_GTG=[k Qg(1,k)];
   else,end
   end
   %-----------------------------------------------------------------------
   % Check the STG Buses MVAR Limitations for 43MW STG
   if k==5;
   if Qg(1,k)<= -22.4 || Qg(1,k)>= 20.919
   % In case the MVAR limit is not met, the below message will be
   % posted together with violation buses
   disp('The Below STG (43 MW)MVAR is not meeting MVAR Constraints')
   NMVR_SSTG=[k Qg(1,k)];
   else,end
   end
   %-----------------------------------------------------------------------
   % Check the STG Buses MVAR Limitations for 86MW STG
   if k==9;
```

```
    if Qg(1,k)<= -41.9 || Qg(1,k)>= 53.837
    % In case the MVAR limit is not met, the below message will be
    % posted together with violation buses
    disp('The Below STG (86 MW)MVAR is not meeting MVAR Constraints')
    NMVR_LSTG=[k Qg(1,k)];
    else,end
    end
    %----------------------------------------------------------------------
    % Check the MVAR Limitations for Captive Synchronous Motors terminal
    % buses
    if  k==191 || k==194 || k==197 || k==200 || k==203
    if Qg(1,k)<= -6.96 || Qg(1,k)>=  16.42
    % In case the MVAR limit is not met, the below message will be
    % posted together with violation buses
    disp('The Below Captive Synchronous Motors MVAR is not meeting MVAR
    Constraints')
    NMVR_SyMC=[k Qg(1,k)];
    else,end
    end
    %----------------------------------------------------------------------
    % Check the MVAR Limitations for None Captive Synchronous Motors terminal
    % buses
    if  k==223 || k==243
    if Qg(1,k)<= -6.104 || Qg(1,k)>= 2.4411
    % In case the MVAR limit is not met, the below message will be
    % posted together with violation buses
    disp('The Below None Captive Synchronous Motors MVAR is not meeting MVAR
    Constraints')
    NMVR_SyM=[k Qg(1,k)];
    else,end
    end
end
```

APPENDIX XI

The MATLAB Program Codes for Tournament and Random Selection. [Refer to section 6.2.5]

The Tournament Selection sub-program MATLAB codes

```
%**********************************************************************
% This Sub-Program will Perform Tourmenant Selection of the Chromosomes
% Parents who will go to the Mating pool
%**********************************************************************

%----------------------------------------------------------------------
% Post the individuals number, population size
  IN=[1:N_Pop]';
% Post the Fitness value list
Tour_Selec_Fitness=[IN Tour_Selec_Fitness];
%-----------------------------------------------------------------------
% The Tourmenant Selection loop
  for k=1:3:N_Pop-2
      h=1+k;
      d=2+k;
      %----------Each time select three (3) chromosomes-----------------
      Tour_Selec_Fitness_Sort=sortrows(Tour_Selec_Fitness,2);
      %------------The Best Chromosomes to be copied twice----------------
      % Capture the Main Transformer Tap Value Chromosomes
      Cr_TM_Selec(k,:)=Cr_TM(Tour_Selec_Fitness_Sort(k,1),:);
      Cr_TM_Selec(h,:)=Cr_TM(Tour_Selec_Fitness_Sort(k,1),:);
      % The 2nd Best Chromosomes to be copied once
      Cr_TM_Selec(d,:)=Cr_TM(Tour_Selec_Fitness_Sort(h,1),:);

      % Capture the Causeway Sub#1 Main Transformer Tap Value Chromosomes
      Cr_TC1_Selec(k,:)=Cr_TC1(Tour_Selec_Fitness_Sort(k,1),:);
      Cr_TC1_Selec(h,:)=Cr_TC1(Tour_Selec_Fitness_Sort(k,1),:);
      % The 2nd Best Chromosomes to be copied once
      Cr_TC1_Selec(d,:)=Cr_TC1(Tour_Selec_Fitness_Sort(h,1),:);

      % Capture the Causeway Sub#2 Main Transformer Tap Value Chromosomes
      Cr_TC2_Selec(k,:)=Cr_TC2(Tour_Selec_Fitness_Sort(k,1),:);
      Cr_TC2_Selec(h,:)=Cr_TC2(Tour_Selec_Fitness_Sort(k,1),:);
      % The 2nd Best Chromosomes to be copied once
      Cr_TC2_Selec(d,:)=Cr_TC2(Tour_Selec_Fitness_Sort(h,1),:);

      % Capture the Causeway Sub#3 Main Transformer Tap Value Chromosomes
      Cr_TC3_Selec(k,:)=Cr_TC3(Tour_Selec_Fitness_Sort(k,1),:);
      Cr_TC3_Selec(h,:)=Cr_TC3(Tour_Selec_Fitness_Sort(k,1),:);
      % The 2nd Best Chromosomes to be copied once
      Cr_TC3_Selec(d,:)=Cr_TC3(Tour_Selec_Fitness_Sort(h,1),:);

      % Capture the Captive Synchronous Motor Transformer Tap Value
      % Chromosomes
      Cr_TCM_Selec(k,:)=Cr_TCM(Tour_Selec_Fitness_Sort(k,1),:);
```

```
        Cr_TCM_Selec(h,:)=Cr_TCM(Tour_Selec_Fitness_Sort(k,1),:);
        % The 2nd Best Chromosomes to be copied once
        Cr_TCM_Selec(d,:)=Cr_TCM(Tour_Selec_Fitness_Sort(h,1),:);

        % Capture the Distribution Transformer Tap Value Chromosomes
        Cr_TD_Selec(k,:)=Cr_TD(Tour_Selec_Fitness_Sort(k,1),:);
        Cr_TD_Selec(h,:)=Cr_TD(Tour_Selec_Fitness_Sort(k,1),:);
        % The 2nd Best Chromosomes to be copied once
        Cr_TD_Selec(d,:)=Cr_TD(Tour_Selec_Fitness_Sort(h,1),:);

        % Capture the Generators Step-Up Transformer Tap Value Chromosomes
        Cr_TG_Selec(k,:)=Cr_TG(Tour_Selec_Fitness_Sort(k,1),:);
        Cr_TG_Selec(h,:)=Cr_TG(Tour_Selec_Fitness_Sort(k,1),:);
        % The 2nd Best Chromosomes to be copied once
        Cr_TG_Selec(d,:)=Cr_TG(Tour_Selec_Fitness_Sort(h,1),:);

        % Capture the Gas Turbine Generators Bus Voltage Value Chromosomes
        Cr_VGTG_Selec(k,:)=Cr_VGTG(Tour_Selec_Fitness_Sort(k,1),:);
        Cr_VGTG_Selec(h,:)=Cr_VGTG(Tour_Selec_Fitness_Sort(k,1),:);
        % The 2nd Best Chromosomes to be copied once
        Cr_VGTG_Selec(d,:)=Cr_VGTG(Tour_Selec_Fitness_Sort(h,1),:);

         % Capture the Steam Turbine Generators Bus Voltage Value Chromosomes
        Cr_VSTG_Selec(k,:)=Cr_VSTG(Tour_Selec_Fitness_Sort(k,1),:);
        Cr_VSTG_Selec(h,:)=Cr_VSTG(Tour_Selec_Fitness_Sort(k,1),:);
        % The 2nd Best Chromosomes to be copied once
        Cr_VSTG_Selec(d,:)=Cr_VSTG(Tour_Selec_Fitness_Sort(h,1),:);

        % Capture the Captive Synchronous Motors Bus Voltage Value
        Cr_VSyMC_Selec(k,:)=Cr_VSyMC(Tour_Selec_Fitness_Sort(k,1),:);
        Cr_VSyMC_Selec(h,:)=Cr_VSyMC(Tour_Selec_Fitness_Sort(k,1),:);
        % The 2nd Best Chromosomes to be copied once
         Cr_VSyMC_Selec(d,:)=Cr_VSyMC(Tour_Selec_Fitness_Sort(h,1),:);

        clear A B h d
    end

%------------------------------------------------------------------------
% Assign the last selected chromosomes as 200 pouplation is even number
% Randomly select the last chromosome. This is needed since this selection
% method selects three (3) chromosomes each time
%-------------------------------------------------------------------------
% Assign random number (R_N_Pop)
R_N_Pop=randi(N_Pop);

% Capture the Main Transformer Tap Value Chromosomes
Cr_TM_Selec(N_Pop,:)=Cr_TM(R_N_Pop,:);

 % Capture the Causeway Sub#1 Main Transformer Tap Value Chromosomes
 Cr_TC1_Selec(N_Pop,:)=Cr_TC1(R_N_Pop,:);

 % Capture the Causeway Sub#2 Main Transformer Tap Value Chromosomes
 Cr_TC2_Selec(N_Pop,:)=Cr_TC1(R_N_Pop,:);

 % Capture the Causeway Sub#3 Main Transformer Tap Value Chromosomes
```

```
Cr_TC3_Selec(N_Pop,:)=Cr_TC3(R_N_Pop,:);

% Capture the Captive Synchronous Motor Transformer Tap Value Chromosomes
Cr_TCM_Selec(N_Pop,:)=Cr_TCM(R_N_Pop,:);

% Capture the Distribution Transformer Tap Value Chromosomes
Cr_TD_Selec(N_Pop,:)=Cr_TD(R_N_Pop,:);

% Capture the Generators Step-Up Transformer Tap Value Chromosomes
Cr_TG_Selec(N_Pop,:)=Cr_TG(R_N_Pop,:);

% Capture the Gas Turbine Generators Bus Voltage Value Chromosomes
Cr_VGTG_Selec(N_Pop,:)=Cr_VGTG(R_N_Pop,:);

% Capture the Steam Turbine Generators Bus Voltage Value Chromosomes
Cr_VSTG_Selec(N_Pop,:)=Cr_VSTG(R_N_Pop,:);

% Capture the Captive Synchronous Motors Bus Voltage Value Chromosomes
Cr_VSyMC_Selec(N_Pop,:)=Cr_VSyMC(R_N_Pop,:);

%-------------------------------------------------------------------------
clear Tour_Selec_Fitness Tour_Selec_Fitness_Sort
```

The Random Selection sub-program MATLAB codes

```
%*************************************************************************
% This Subprogram will Perform Random Selection of the Chromosomes
% Parents who will go to the Mating pool
%*************************************************************************
% N_Pop is the population size
for k=1:N_Pop;
   % Generate random number for selection (S_RN)
   S_RN=randi(N_Pop);
   % Capture the Main Transformer Tap Value Chromosomes
   Cr_TM_Selec(k,:)=Cr_TM(S_RN,:);
   % Capture the Causeway Sub#1 Main Transformer Tap Value Chromosomes
   Cr_TC1_Selec(k,:)=Cr_TC1(S_RN,:);
   % Capture the Causeway Sub#2 Main Transformer Tap Value Chromosomes
   Cr_TC2_Selec(k,:)=Cr_TC2(S_RN,:);
   % Capture the Causeway Sub#3 Main Transformer Tap Value Chromosomes
   Cr_TC3_Selec(k,:)=Cr_TC3(S_RN,:);
   % Capture the Captive Synchronus Motor Transformer Tap Value Chromosomes
   Cr_TCM_Selec(k,:)=Cr_TCM(S_RN,:);
   % Capture the Distribution Transformer Tap Value Chromosomes
   Cr_TD_Selec(k,:)=Cr_TD(S_RN,:);
   % Capture the Generators Step-Up Transformer Tap Value Chromosomes
   Cr_TG_Selec(k,:)=Cr_TG(S_RN,:);
   % Capture the Gas Turbine Generators Bus Voltge Value Chromosomes
   Cr_VGTG_Selec(k,:)=Cr_VGTG(S_RN,:);
   % Capture the Steam Turbine Generators Bus Voltge Value Chromosomes
   Cr_VSTG_Selec(k,:)=Cr_VSTG(S_RN,:);
   % Capture the Captive Synchronus Motors Bus Voltge Value Chromosomes
   Cr_VSyMC_Selec(k,:)=Cr_VSyMC(S_RN,:);
   % Capture the Synchronus Motors Bus Voltge Value Chromosomes
   %Cr_VSyM_Selec(k,:)=Cr_VSyM(S_RN,:);
end
clear k S_RN
```

APPENDIX XII

The MATLAB Program Codes for Simple and Differential Crossover/Mutation. [Refer to section 6.2.5]

The Simple Crossover sub-program MATLAB codes

```
%***********************************************************************
% This Sub-Program will perform Simple Crossover on the Chromosomes
%***********************************************************************

%------------------------------------------------------------------------
% Save the Selected Parents Chromosomes in Temporary Matrixes;
% Cr_TM_Selec_T
% This is to perform the Crossover while saving the Selected Parents
% Save the Transformer Taps Value Chromosomes
Cr_TM_Selec_T= Cr_TM_Selec;
Cr_TC1_Selec_T= Cr_TC1_Selec;
Cr_TC2_Selec_T= Cr_TC2_Selec;
Cr_TC3_Selec_T= Cr_TC3_Selec;
Cr_TCM_Selec_T= Cr_TCM_Selec;
Cr_TD_Selec_T= Cr_TD_Selec;
Cr_TG_Selec_T= Cr_TG_Selec;
% Save the PV Buses Voltages Chromosomes
Cr_VGTG_Selec_T= Cr_VGTG_Selec;
Cr_VSTG_Selec_T= Cr_VSTG_Selec;
Cr_VSyMC_Selec_T= Cr_VSyMC_Selec;
% Due to the narrow bandwidth of the non captive synchronous motor MVAR
% limitations, there were treated as induction motors
% Cr_VSyM_Selec_T= Cr_VSyM_Selec;

%------------------------------------------------------------------------
% Save the Temporary Selected Parents Chromosomes in Crossover Matrixes;
% Cr_TM_Cros. In case the Crossover does not happen because Probability is
% < 90% the Crossover genes does not change.
% Save the Transformer Taps Value Chromosomes
Cr_TM_Cros=Cr_TM_Selec_T;
Cr_TC1_Cros=Cr_TC1_Selec_T;
Cr_TC2_Cros=Cr_TC2_Selec_T;
Cr_TC3_Cros=Cr_TC3_Selec_T;
Cr_TCM_Cros=Cr_TCM_Selec_T;
Cr_TD_Cros=Cr_TD_Selec_T;
Cr_TG_Cros=Cr_TG_Selec_T;
% Save the PV Buses Voltages Chromosomes
Cr_VGTG_Cros=Cr_VGTG_Selec_T;
Cr_VSTG_Cros=Cr_VSTG_Selec_T;
Cr_VSyMC_Cros=Cr_VSyMC_Selec_T;
% Due to the narrow bandwidth of the non captive synchronous motor MVAR
% limitations, there were treated as induction motors
%Cr_VSyM_Cros=Cr_VSyM_Selec_T;

%-----------------------------------------------------------------------
```

```
% Perform Simple Crossover for the Large Power Transformer with 115kV
% primary Voltage Tap Chromosome (Cr_TM_Selec_T)
%-----------------------------------------------------------------------
% Perform the Simple Crossover with 90% Probability
% Assign random number to each chromosomes;
  Rmin=0;
  Rmax=1;
% Set the Crossover Probability 90%
  Cros_P=0.9;

% Measure the Chromosomes matrix (Cr_TM_Selec_T) lengths
n=length(Cr_TM_Selec_T(:,1));
d=length(Cr_TM_Selec_T(1,:));
% Identify the Crossover Vertical Lines CVL where the simple crossover will
% take place
  CVL=randi(d-1);

for h=1:2:n-1;
    k=h+1;
     % Obtain random number (0-1)
     ChR(h)=(Rmin+rand*(Rmax-Rmin));
     % Find out if the number is with 90% probability
     if ChR(h)<Cros_P;
            for u=1:CVL
            % Store the Genes in Temporary Chromosome Cr_TMS
            Cr_TMS(h,u)=Cr_TM_Selec_T(k,u);
            % Perform the crossover for the second chromosome
            Cr_TM_Cros(k,u)=Cr_TM_Selec_T(h,u);
            % Perform the crossover for the first chromosome
            Cr_TM_Cros(h,u)=Cr_TMS(h,u);
            end
     end

end
clear ChR a d u k h I d n Cr_TMS CVL

%-----------------------------------------------------------------------
% Perform Simple Crossover for the Causeway Substation#1 Power Transformer
% primary Voltage Tap Chromosomes (Cr_TC1_Selec_T)
%-----------------------------------------------------------------------
% Perform the Simple Crossover with 90% Probability
% Assign random number to each chromosomes;
  Rmin=0;
  Rmax=1;
% Measure Chromosome matrix (Cr_TC1_Selec_T) lengths
n=length(Cr_TC1_Selec_T(:,1));
d=length(Cr_TC1_Selec_T(1,:));

% Identify the Crossover Vertical Lines CVL where the simple crossover will
% take place
  CVL=randi(d-1);

for h=1:2:n-1;
    k=h+1;
     % Obtain random number (0-1)
     ChR(h)=(Rmin+rand*(Rmax-Rmin));
```

```
      % Find out if the number is with 90% probability
      if ChR(h)<Cros_P;
             for u=1:CVL
             % Store the Genes in Temporary Chromosome Cr_TMS
             Cr_TC1S(h,u)=Cr_TC1_Selec_T(k,u);
             % Perform the crossover for the second chromosome
             Cr_TC1_Cros(k,u)=Cr_TC1_Selec_T(h,u);
             % Perform the crossover for the first chromosome
             Cr_TC1_Cros(h,u)=Cr_TC1S(h,u);
             end
      end

end
clear ChR a d u k h I d n Cr_TMS CVL
%--------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Perform Simple Crossover for the Causeway Substation#2 Power Transformer
% primary Voltage Tap Chromosome (Cr_TC2_Selec_T)
%--------------------------------------------------------------------------
% Perform the Simple Crossover with 90% Probability
% Assaign random number to each chromosomes;
  Rmin=0;
  Rmax=1;
% Measure the Chromosomes matrix (Cr_TC2_Selec_T) lengths
n=length(Cr_TC2_Selec_T(:,1));
d=length(Cr_TC2_Selec_T(1,:));
% Identify the Crossover Vertical Lines CVL where the simple crossover will
% take place
  CVL=randi(d-1);

for h=1:2:n-1;
    k=h+1;
     % Obtain random number (0-1)
     ChR(h)=(Rmin+rand*(Rmax-Rmin));
     % Find out if the number is with 90% probability
     if ChR(h)<Cros_P;
            for u=1:CVL
            % Store the Genes in Temporary Chromosome Cr_TMS
            Cr_TC2S(h,u)=Cr_TC2_Selec_T(k,u);
            % Perform the crossover for the second chromosome
            Cr_TC2_Cros(k,u)=Cr_TC2_Selec_T(h,u);
            % Perform the crossover for the first chromosome
            Cr_TC2_Cros(h,u)=Cr_TC2S(h,u);
            end
     end

end
clear ChR a d u k h I d n Cr_TMS CVL
%---------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Perform Simple Crossover for the Causeway Substation#3 Power Transformer
% primary Voltage Tap Chromosome (Cr_TC3_Selec_T)
%--------------------------------------------------------------------------
% Perform the Simple Crossover with 90% Probability
```

```
% Assign random number to each chromosomes;
  Rmin=0;
  Rmax=1;
% Measure Chromosome matrix (Cr_TC3_Selec_T) lengths
n=length(Cr_TC3_Selec_T(:,1));
d=length(Cr_TC3_Selec_T(1,:));

% Identify the Crosover Vertical Lines CVL where the simple crossover will
% take place
  CVL=randi(d-1);

for h=1:2:n-1;
    k=h+1;
     % Obtain random number (0-1)
     ChR(h)=(Rmin+rand*(Rmax-Rmin));
     % Find out if the number is with 90% probability
     if ChR(h)<Cros_P;
            for u=1:CVL
            % Store the Genes in Temporary Chromosome Cr_TMS
            Cr_TC3S(h,u)=Cr_TC3_Selec_T(k,u);
            % Perform the crossover for the second chromosome
            Cr_TC3_Cros(k,u)=Cr_TC3_Selec_T(h,u);
            % Perform the crossover for the first chromosome
            Cr_TC3_Cros(h,u)=Cr_TC3S(h,u);
            end
     end

end

clear ChR a d u k h I d n Cr_TMS CVL
%--------------------------------------------------------------------------

%-------------------------------------------------------------------------
% Perform Simple Crossover for the Captive Synchronous Motors power
% Transformer primary Voltage Tap Chromosome (Cr_TCM_Selec_T)
%-------------------------------------------------------------------------
% Perform the Simple Crossover with 90% Probability
% Assign random number to each chromosomes;
  Rmin=0;
  Rmax=1;
% Measure Chromosome matrix(Cr_TCM_Selec_T) lengths
n=length(Cr_TCM_Selec_T(:,1));
d=length(Cr_TCM_Selec_T(1,:));

% Identify the Crossover Vertical Lines CVL where the simple crossover will
% take place
 CVL=randi(d-1);

for h=1:2:n-1;
    k=h+1;
     % Obtain random number (0-1)
     ChR(h)=(Rmin+rand*(Rmax-Rmin));
     % Find out if the number is with 90% probability
     if ChR(h)<Cros_P;
            for u=1:CVL
```

```
            % Store the Genes in Temporary Chromosome Cr_TCMS
            Cr_TCMS(h,u)=Cr_TCM_Selec_T(k,u);
            % Perform the crossover for the second chromosome
            Cr_TCM_Cros(k,u)=Cr_TCM_Selec_T(h,u);
            % Perform the crossover for the first chromosome
            Cr_TCM_Cros(h,u)=Cr_TCMS(h,u);
            end
     end

end
clear ChR a d u k h I d n Cr_TMS CVL
%---------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Perform Simple Crossover for the Distribution Power Transformer primary
% Voltage Tap Chromosome (Cr_TD_Selec_T)
%--------------------------------------------------------------------------
% Perform the Simple Crossover with 90% Probability
% Assign random number to each chromosomes;
  Rmin=0;
  Rmax=1;
% Measure Chromosome matrix (Cr_TD_Selec_T) lengths
n=length(Cr_TD_Selec_T(:,1));
d=length(Cr_TD_Selec_T(1,:));

% Identify the Crossover Vertical Lines CVL where the simple crossover will
% take place
  CVL=randi(d-1);

for h=1:2:n-1;
    k=h+1;
     % Obtain random number (0-1)
     ChR(h)=(Rmin+rand*(Rmax-Rmin));
     % Find out if the number is with 90% probability
     if ChR(h)<Cros_P;
             for u=1:CVL
            % Store the Genes in Temporary Chromosome Cr_TMS
            Cr_TDS(h,u)=Cr_TD_Selec_T(k,u);
            % Perform the crossover for the second chromosome
            Cr_TD_Cros(k,u)=Cr_TD_Selec_T(h,u);
            % Perform the crossover for the first chromosome
            Cr_TD_Cros(h,u)=Cr_TDS(h,u);
            end

     end

end
clear ChR a d u k h I d n Cr_TMS CVL
%---------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Perform Simple Crossover for the Generator Step-Up Power Transformer
% Voltage Tap Chromosome (Cr_TG_Selec_T)
%--------------------------------------------------------------------------
% Perform the Simple Crossover with 90% Probability
```

```
% Assaign random number to each chromosomes;
  Rmin=0;
  Rmax=1;
% Measure Chromosome matrix (Cr_TG_Selec_T) lengths
n=length(Cr_TG_Selec_T(:,1));
d=length(Cr_TG_Selec_T(1,:));

% Identify the Crossover Vertical Lines CVL where the simple crossover will
% take place
  CVL=randi(d-1);

for h=1:2:n-1;
    k=h+1;
     % Obtain random number (0-1)
     ChR(h)=(Rmin+rand*(Rmax-Rmin));
     % Find out if the number is with 90% probability
     if ChR(h)<Cros_P;
            for u=1:CVL
            % Store the Genes in Temporary Chromosome Cr_TGS
            Cr_TGS(h,u)=Cr_TG_Selec_T(k,u);
            % Perform the crossover for the second chromosome
            Cr_TG_Cros(k,u)=Cr_TG_Selec_T(h,u);
            % Perform the crossover for the first chromosome
            Cr_TG_Cros(h,u)=Cr_TGS(h,u);
            end
     end

end

clear ChR a d u k h I d n Cr_TMS CVL
%--------------------------------------------------------------------------

%-------------------------------------------------------------------------
% Perform Simple Crossover for the Gas Turbine Generator Bus Voltage
% Chromosome (Cr_VGTG_Selec_T)
%-------------------------------------------------------------------------
% Perform the Simple Crossover with 90% Probability
% Assign random number to each chromosomes;
  Rmin=0;
  Rmax=1;
% Measure Chromosome matrix(Cr_VGTG_Selec_T) lengths
n=length(Cr_VGTG_Selec_T(:,1));
d=length(Cr_VGTG_Selec_T(1,:));

% Identify the Crossover Vertical Lines CVL where the simple crossover will
% take place
   CVL=randi(d-1);

for h=1:2:n-1;
    k=h+1;
     % Obtain random number (0-1)
     ChR(h)=(Rmin+rand*(Rmax-Rmin));
     % Find out if the number is with 90% probability
     if ChR(h)<Cros_P;
            for u=1:CVL
```

```
            % Store the Genes in Temporary Chromosome Cr_VGTGS
            Cr_VGTGS(h,u)=Cr_VGTG_Selec_T(k,u);
            % Perform the crossover for the second chromosome
            Cr_VGTG_Cros(k,u)=Cr_VGTG_Selec_T(h,u);
            % Perform the crossover for the first chromosome
            Cr_VGTG_Cros(h,u)=Cr_VGTGS(h,u);
            end
    end

end

clear ChR a d u k h I d n Cr_TMS CVL
%---------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Perform Simple Crossover for the Steam Turbine Generator Bus Voltage
% Chromosome (Cr_VSTG_Selec_T)
%--------------------------------------------------------------------------
% Perform the Simple Crossover with 90% Probability
% Assign random number to each chromosomes;
  Rmin=0;
  Rmax=1;
% Measure Chromosome matrix (Cr_VSTG_Selec_T) lengths
n=length(Cr_VSTG_Selec_T(:,1));
d=length(Cr_VSTG_Selec_T(1,:));

% Identify the Crossover Vertical Lines CVL
% simple crossover will take place
  CVL=randi(d-1);

for h=1:2:n-1;
    k=h+1;
     % Obtain random number (0-1)
     ChR(h)=(Rmin+rand*(Rmax-Rmin));
     % Find out if the number is with 90% probability
     if ChR(h)<Cros_P;
            for u=1:CVL
            % Store the Genes in Temporary Chromosome Cr_VSTGS
            Cr_VSTGS(h,u)=Cr_VSTG_Selec_T(k,u);
            % Perform the crossover for the second chromosome
            Cr_VSTG_Cros(k,u)=Cr_VSTG_Selec_T(h,u);
            % Perform the crossover for the second chromosome
            Cr_VSTG_Cros(h,u)=Cr_VSTGS(h,u);
            end
     end

end
% MTH Capture CVL
A9=CVL;
clear ChR a d u k h I d n Cr_TMS CVL
%---------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Perform Simple Crossover for Captive Synchronous Motors Bus Voltage
% Chromosome (Cr_VSyMC_Selec_T)
```

```
%----------------------------------------------------------------------------
% Perform the Simple Crossover with 90% Probability
% Assign random number to each chromosomes;
  Rmin=0;
  Rmax=1;
% Measure Chromosome matrix(Cr_VSyMC_Selec_T) lengths
n=length(Cr_VSyMC_Selec_T(:,1));
d=length(Cr_VSyMC_Selec_T(1,:));

% Identify the Crossover Vertical Lines CVL where the simple crossover will
% take place
  CVL=randi(d-1);

for h=1:2:n-1;
    k=h+1;
     % Obtain random number (0-1)
     ChR(h)=(Rmin+rand*(Rmax-Rmin));
     % Find out if the number is with 90% probability
     if ChR(h)<Cros_P;
            for u=1:CVL
            % Store the Genes in Temporary Chromosome Cr_VGTGS
            Cr_VSyMCS(h,u)=Cr_VSyMC_Selec_T(k,u);
            % Perform the crossover for the second chromosome
            Cr_VSyMC_Cros(k,u)=Cr_VSyMC_Selec_T(h,u);
            % Perform the crossover for the first chromosome
            Cr_VSyMC_Cros(h,u)=Cr_VSyMCS(h,u);
            end
     end

end
% clear all used variables
clear ChR a d u k h I d n Cr_TMS CVL
%-----------------------------------------------------------------------------
```

The Differential Crossover/Mutation sub-program MATLAB codes

```
%*************************************************************************
% This Sub-Program will perform differential Crossover/Mutation on the
% Chromosomes as per equations 6.2, 6.3 and 6.4
%*************************************************************************

%---------------------------------------------------------------------------
% Save the Selected Parents Chromosomes in Temporary Matrixes; Cr_TM_Selec_T
% This is to perform the Crossover while saving the Selected Parents
% Save the Transformer Taps Value
%---------------------------------------------------------------------------
% Save the Transformer Taps Value Chromosomes
Cr_TM_Selec_T= Cr_TM_Selec;
Cr_TC1_Selec_T= Cr_TC1_Selec;
Cr_TC2_Selec_T= Cr_TC2_Selec;
Cr_TC3_Selec_T= Cr_TC3_Selec;
Cr_TCM_Selec_T= Cr_TCM_Selec;
Cr_TD_Selec_T= Cr_TD_Selec;
Cr_TG_Selec_T= Cr_TG_Selec;
% Save the PV Buses Voltages Chromosomes
Cr_VGTG_Selec_T= Cr_VGTG_Selec;
Cr_VSTG_Selec_T= Cr_VSTG_Selec;
Cr_VSyMC_Selec_T= Cr_VSyMC_Selec;
%Cr_VSyM_Selec_T= Cr_VSyM_Selec;

%--------------------------------------------------------------------------
% Save the Temporary Selected Parents Chromosomes in Crossover Matrixes;
% Cr_TM_Cros. In case the Crossover does not happen because Probability is
% < 90% the Crossover genes does not change.
% Save the Transformer Taps Value Chromosomes
Cr_TM_Cros=Cr_TM_Selec_T;
Cr_TC1_Cros=Cr_TC1_Selec_T;
Cr_TC2_Cros=Cr_TC2_Selec_T;
Cr_TC3_Cros=Cr_TC3_Selec_T;
Cr_TCM_Cros=Cr_TCM_Selec_T;
Cr_TD_Cros=Cr_TD_Selec_T;
Cr_TG_Cros=Cr_TG_Selec_T;
% Save the PV Buses Voltages Chromosomes
Cr_VGTG_Cros=Cr_VGTG_Selec_T;
Cr_VSTG_Cros=Cr_VSTG_Selec_T;
Cr_VSyMC_Cros=Cr_VSyMC_Selec_T;
%Cr_VSyM_Cros=Cr_VSyM_Selec_T;

%-------------------------------------------------------------------------
% Perform Differential Crossover/Mutation for the Large Power Transformer
% with 115kV primary Voltage Tap Chromosome (Cr_TM_Selec_T)
%-------------------------------------------------------------------------
% Measure Chromosome (Cr_TM_Selec_T) lengths
n=length(Cr_TM_Cros(:,1));
d=length(Cr_TM_Cros(1,:));
%==========================================================================
%------------------The DE Crossover Loop-----------------------------------
% Identify β(B)value to be 0.5 (50%), refer to equations 6.2, 6.3 and 6.4
B=0.5;
```

```
for k=1:3:N_Pop-2
    h=1+k;
    d=2+k;
% Capture the Main Transformer Tap Value chromosomes
Cr_TM_Cros_DET(k,:)=Cr_TM_Cros(k,:)+ B*(Cr_TM_Cros(h,:)-Cr_TM_Cros(d,:));
Cr_TM_Cros_DET(h,:)=Cr_TM_Cros(h,:)+ B*(Cr_TM_Cros(d,:)-Cr_TM_Cros(k,:));
Cr_TM_Cros_DET(d,:)=Cr_TM_Cros(d,:)+ B*(Cr_TM_Cros(k,:)-Cr_TM_Cros(h,:));
end
clear k h d n

 %--------------------------------------------------------------------------
 % Assign the last selected chromosomes as 200 population is even number
 % Randomly select the last chromosome R_N_Pop=randi(N_Pop);
 % Capture the Main Transformer Tap Value
 Cr_TM_Cros_DET(N_Pop,:)=Cr_TM_Cros(N_Pop,:);
% Make sure that the crossovered/mutated Genes are matching the real Tap
% Values
% Measure Chromosome (Cr_TM) lengths
 n=length(Cr_TM_Cros_DET(:,1));
 d=length(Cr_TM_Cros_DET(1,:));
% Measure The Tap Values
x=length(Tap17(1,:));

for h=1:n
     for k=1:d
         for z=1:x
         Cr_TM_Cros_DET_T(z,:)=[z abs(Cr_TM_Cros_DET(h,k)-Tap17(1,z))];
         end
         Cr_TM_Cros_DET_TS(:,:)=sortrows(Cr_TM_Cros_DET_T(1:z,:),2);
         Cr_TM_Cros_DET_R(h,k)=Tap17(1,Cr_TM_Cros_DET_TS(1,1));
         Cr_TM_Cros(h,k)=Cr_TM_Cros_DET_R(h,k);
     end
end

clear h k n d x Cr_TM_Mut_TS Cr_TM_Mut_R Cr_TM_Mut_T

%---------------------------------------------------------------------------
% Perform Differential Crossover/Mutation for the Causeway substation#1
% Power Transformer with 115kV primary Voltage Tap Chromosome
%(Cr_TC1_Selec_T)
%---------------------------------------------------------------------------
% Measure Chromosome (Cr_TM_Selec_T) lengths
n=length(Cr_TC1_Cros(:,1));
d=length(Cr_TC1_Cros(1,:));

%-------------------The DE Crossover Loop------------------------------------
for k=1:3:N_Pop-2
    h=1+k;
    d=2+k;
% Capture the Main Transformer Tap Value
Cr_TC1_Cros_DET(k,:)=Cr_TC1_Cros(k,:)+ B*(Cr_TC1_Cros(h,:)-
Cr_TC1_Cros(d,:));
Cr_TC1_Cros_DET(h,:)=Cr_TC1_Cros(h,:)+ B*(Cr_TC1_Cros(d,:)-
Cr_TC1_Cros(k,:));
Cr_TC1_Cros_DET(d,:)=Cr_TC1_Cros(d,:)+ B*(Cr_TC1_Cros(k,:)-
Cr_TC1_Cros(h,:));
```

```
end
clear k h d n

 %-------------------------------------------------------------------------
 % Assign the last selected chromosomes as 200 population is even number
 % Randomly select the last chromosome R_N_Pop=randi(N_Pop);
 % Capture the Main Transformer Tap Value
 Cr_TC1_Cros_DET(N_Pop,:)=Cr_TC1_Cros(N_Pop,:);

% Make sure that the Crossovered/Mutated Genes are matching the real Tap
% Values
% Measure Chromosome (Cr_TM) lengths
 n=length(Cr_TC1_Cros_DET(:,1));
 d=length(Cr_TC1_Cros_DET(1,:));
% Meaure The Tap Values
x=length(Tap12C(1,:));

for h=1:n
     for k=1:d
         for z=1:x
         Cr_TC1_Cros_DET_T(z,:)=[z abs(Cr_TC1_Cros_DET(h,k)-Tap12C(1,z))];
         end
         Cr_TC1_Cros_DET_TS(:,:)=sortrows(Cr_TC1_Cros_DET_T(1:z,:),2);
         Cr_TC1_Cros_DET_R(h,k)=Tap12C(1,Cr_TC1_Cros_DET_TS(1,1));
         Cr_TC1_Cros(h,k)=Cr_TC1_Cros_DET_R(h,k);
     end
end

clear h k n d x Cr_TM_Mut_TS Cr_TM_Mut_R Cr_TM_Mut_T

%-------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Perform Differential Crossover/Mutation for the Causeway substation#2
% Power Transformer with 115kV primary Voltage Tap Chromosome
%(Cr_TC2_Selec_T)
%--------------------------------------------------------------------------
% Measure Chromosome (Cr_TC2_Selec_T) lengths
n=length(Cr_TC2_Cros(:,1));
d=length(Cr_TC2_Cros(1,:));

%------------------The DE Crossover Loop----------------------------------
for k=1:3:N_Pop-2
    h=1+k;
    d=2+k;
% Capture the Main Transformer Tap Value
Cr_TC2_Cros_DET(k,:)=Cr_TC2_Cros(k,:)+ B*(Cr_TC2_Cros(h,:)-
Cr_TC2_Cros(d,:));
Cr_TC2_Cros_DET(h,:)=Cr_TC2_Cros(h,:)+ B*(Cr_TC2_Cros(d,:)-
Cr_TC2_Cros(k,:));
Cr_TC2_Cros_DET(d,:)=Cr_TC2_Cros(d,:)+ B*(Cr_TC2_Cros(k,:)-
Cr_TC2_Cros(h,:));
end
clear k h d n
 %-------------------------------------------------------------------------
 % Assign the last selected chromosomes as 200 population is even number
```

```
 % Randomly select the last chromosome R_N_Pop=randi(N_Pop);
 % Capture the Main Transformer Tap Value
 Cr_TC2_Cros_DET(N_Pop,:)=Cr_TC2_Cros(N_Pop,:);

% Make sure that the Crossovered/Mutated Genes are matching the real Tap
% Values
% Measure Chromosome (Cr_TM) lengths
 n=length(Cr_TC2_Cros_DET(:,1));
 d=length(Cr_TC2_Cros_DET(1,:));
% Measure The Tap Values
x=length(Tap12C(1,:));

for h=1:n
     for k=1:d
         for z=1:x
         Cr_TC2_Cros_DET_T(z,:)=[z abs(Cr_TC2_Cros_DET(h,k)-Tap12C(1,z))];
         end
         Cr_TC2_Cros_DET_TS(:,:)=sortrows(Cr_TC2_Cros_DET_T(1:z,:),2);
         Cr_TC2_Cros_DET_R(h,k)=Tap12C(1,Cr_TC2_Cros_DET_TS(1,1));
         Cr_TC2_Cros(h,k)=Cr_TC2_Cros_DET_R(h,k);
     end
end

clear h k n d x Cr_TM_Mut_TS Cr_TM_Mut_R Cr_TM_Mut_T

%-------------------------------------------------------------------------
% Perform Differential Crossover/Mutation for the Causeway substation#3
% Power Transformer with 115kV primary Voltage Tap Chromosome
%(Cr_TC3_Selec_T)
%-------------------------------------------------------------------------
% Measure Chromosome (Cr_TC3_Selec_T) lengths
n=length(Cr_TC3_Cros(:,1));
d=length(Cr_TC3_Cros(1,:));
% Identify the Crosover Vertical Lines CVL

%------------------The DE Crossover Loop-----------------------------------
for k=1:3:N_Pop-2
    h=1+k;
    d=2+k;
% Capture the Main Transformer Tap Value
Cr_TC3_Cros_DET(k,:)=Cr_TC3_Cros(k,:)+ B*(Cr_TC3_Cros(h,:)-
Cr_TC3_Cros(d,:));
Cr_TC3_Cros_DET(h,:)=Cr_TC3_Cros(h,:)+ B*(Cr_TC3_Cros(d,:)-
Cr_TC3_Cros(k,:));
Cr_TC3_Cros_DET(d,:)=Cr_TC3_Cros(d,:)+ B*(Cr_TC3_Cros(k,:)-
Cr_TC3_Cros(h,:));
end
clear k h d n
 %------------------------------------------------------------------------
 % Assign the last selected chromosomes as 200 population is even number
 % Randomly select the last chromosome R_N_Pop=randi(N_Pop);
 % Capture the Main Transformer Tap Value
 Cr_TC3_Cros_DET(N_Pop,:)=Cr_TC3_Cros(N_Pop,:);

% Make sure that the Corossovered/Mutated Genes are matching the real Tap
% Values
```

```
% Measure Chromosome (Cr_TM) lengths
 n=length(Cr_TC3_Cros_DET(:,1));
 d=length(Cr_TC3_Cros_DET(1,:));
% Meaure The Tap Values
x=length(Tap12C(1,:));

for h=1:n
     for k=1:d
         for z=1:x
         Cr_TC3_Cros_DET_T(z,:)=[z abs(Cr_TC3_Cros_DET(h,k)-Tap12C(1,z))];
         end
         Cr_TC3_Cros_DET_TS(:,:)=sortrows(Cr_TC3_Cros_DET_T(1:z,:),2);
         Cr_TC3_Cros_DET_R(h,k)=Tap12C(1,Cr_TC3_Cros_DET_TS(1,1));
         Cr_TC3_Cros(h,k)=Cr_TC3_Cros_DET_R(h,k);
     end
end

clear h k n d x Cr_TM_Mut_TS Cr_TM_Mut_R Cr_TM_Mut_T

%------------------------------------------------------------------------

%-------------------------------------------------------------------------
% Perform Differential Crossover/Mutation for the Captive Synchronous
% Motors power Transformer primary Voltage Tap Chromosome (Cr_TCM_Selec_T)
%-------------------------------------------------------------------------
% Measure Chromosome (Cr_TC3_Selec_T) lengths
n=length(Cr_TCM_Cros(:,1));
d=length(Cr_TCM_Cros(1,:));
%==========================================================================
%------------------The DE Crossover Loop-----------------------------------
for k=1:3:N_Pop-2
    h=1+k;
    d=2+k;
% Capture the Main Transformer Tap Value
Cr_TCM_Cros_DET(k,:)=Cr_TCM_Cros(k,:)+ B*(Cr_TCM_Cros(h,:)-
Cr_TCM_Cros(d,:));
Cr_TCM_Cros_DET(h,:)=Cr_TCM_Cros(h,:)+ B*(Cr_TCM_Cros(d,:)-
Cr_TCM_Cros(k,:));
Cr_TCM_Cros_DET(d,:)=Cr_TCM_Cros(d,:)+ B*(Cr_TCM_Cros(k,:)-
Cr_TCM_Cros(h,:));

end
clear k h d n

 %------------------------------------------------------------------------
 % Assign the last selected chromosomes as 200 population is even number
 % Capture the Main Transformer Tap Value
 Cr_TCM_Cros_DET(N_Pop,:)=Cr_TCM_Cros(N_Pop,:);

% Make sure that the Crossovered/Mutated Genes are matching the real Tap
% Values
% Measure Chromosome (Cr_TCM) lengths
 n=length(Cr_TCM_Cros_DET(:,1));
 d=length(Cr_TCM_Cros_DET(1,:));
% Measure The Tap Values
```

```
x=length(Tap5(1,:));

for h=1:n
     for k=1:d
         for z=1:x
         Cr_TCM_Cros_DET_T(z,:)=[z abs(Cr_TCM_Cros_DET(h,k)-Tap5(1,z))];
         end
         Cr_TCM_Cros_DET_TS(:,:)=sortrows(Cr_TCM_Cros_DET_T(1:z,:),2);
         Cr_TCM_Cros_DET_R(h,k)=Tap5(1,Cr_TCM_Cros_DET_TS(1,1));
         Cr_TCM_Cros(h,k)=Cr_TCM_Cros_DET_R(h,k);
     end
end

clear h k n d x Cr_TM_Mut_TS Cr_TM_Mut_R Cr_TM_Mut_T

%-----------------------------------------------------------------------

%-------------------------------------------------------------------------
% Perform Differential Crossover/Mutation the Distribution Power
% Transformer primary Voltage Tap Chromosome (Cr_TD_Selec_T)
%-------------------------------------------------------------------------
% Meaure Chromosome (Cr_TD_Selec_T) lengths
n=length(Cr_TD_Cros(:,1));
d=length(Cr_TD_Cros(1,:));

%------------------The DE Crossover Loop-----------------------------------
for k=1:3:N_Pop-2
      h=1+k;
      d=2+k;
% Capture the Main Transformer Tap Value
Cr_TD_Cros_DET(k,:)=Cr_TD_Cros(k,:)+ B*(Cr_TD_Cros(h,:)-Cr_TD_Cros(d,:));
Cr_TD_Cros_DET(h,:)=Cr_TD_Cros(h,:)+ B*(Cr_TD_Cros(d,:)-Cr_TD_Cros(k,:));
Cr_TD_Cros_DET(d,:)=Cr_TD_Cros(d,:)+ B*(Cr_TD_Cros(k,:)-Cr_TD_Cros(h,:));
end
clear k h d n
%-----------------------------------------------------------------------
% Assign the last selected chromosomes as 200 population is even number
% Capture the Main Transformer Tap Value
 Cr_TD_Cros_DET(N_Pop,:)=Cr_TD_Cros(N_Pop,:);
% Make sure that the Crossovered/Mutated Genes are matching the real Tap
% Values
% Measure Chromosome (Cr_TCM) lengths
 n=length(Cr_TD_Cros_DET(:,1));
 d=length(Cr_TD_Cros_DET(1,:));
% Measure The Tap Values
x=length(Tap5(1,:));

for h=1:n
     for k=1:d
         for z=1:x
         Cr_TD_Cros_DET_T(z,:)=[z abs(Cr_TD_Cros_DET(h,k)-Tap5(1,z))];
         end
         Cr_TD_Cros_DET_TS(:,:)=sortrows(Cr_TD_Cros_DET_T(1:z,:),2);
         Cr_TD_Cros_DET_R(h,k)=Tap5(1,Cr_TD_Cros_DET_TS(1,1));
         Cr_TD_Cros(h,k)=Cr_TD_Cros_DET_R(h,k);
     end
```

```
end

clear h k n d x Cr_TM_Mut_TS Cr_TM_Mut_R Cr_TM_Mut_T
%----------------------------------------------------------------------

%------------------------------------------------------------------------
% Perform Differential Crossover/Mutation for the Generator Step-Up Power
% Transformer Voltage Tap Chromosome (Cr_TG_Selec_T)
%------------------------------------------------------------------------
% Measure Chromosome (Cr_TD_Selec_T) lengths
n=length(Cr_TG_Cros(:,1));
d=length(Cr_TG_Cros(1,:));
% Identify the Crosover Vertical Lines CVL
%=========================================================================
%------------------The DE Crossover Loop-----------------------------------
for k=1:3:N_Pop-2
    h=1+k;
    d=2+k;
% Capture the Main Transformer Tap Value
Cr_TG_Cros_DET(k,:)=Cr_TG_Cros(k,:)+ B*(Cr_TG_Cros(h,:)-Cr_TG_Cros(d,:));
Cr_TG_Cros_DET(h,:)=Cr_TG_Cros(h,:)+ B*(Cr_TG_Cros(d,:)-Cr_TG_Cros(k,:));
Cr_TG_Cros_DET(d,:)=Cr_TG_Cros(d,:)+ B*(Cr_TG_Cros(k,:)-Cr_TG_Cros(h,:));
end
clear k h d n

 %----------------------------------------------------------------------
 % Assign the last selected chromosomes as 200 population is even number
 % Capture the Main Transformer Tap Value
 Cr_TG_Cros_DET(N_Pop,:)=Cr_TG_Cros(N_Pop,:);

% Make sure that the Crossovered/Mutated Genes are matching the real Tap
% Values---
% Measure Chromosome (Cr_TCM) lengths
 n=length(Cr_TG_Cros_DET(:,1));
 d=length(Cr_TG_Cros_DET(1,:));
% Measure The Tap Values
x=length(TapG(1,:));

for h=1:n
     for k=1:d
         for z=1:x
         Cr_TG_Cros_DET_T(z,:)=[z abs(Cr_TG_Cros_DET(h,k)-TapG(1,z))];
         end
         Cr_TG_Cros_DET_TS(:,:)=sortrows(Cr_TG_Cros_DET_T(1:z,:),2);
         Cr_TG_Cros_DET_R(h,k)=TapG(1,Cr_TG_Cros_DET_TS(1,1));
         Cr_TG_Cros(h,k)=Cr_TG_Cros_DET_R(h,k);
     end
end

clear h k n d x Cr_TM_Mut_TS Cr_TM_Mut_R Cr_TM_Mut_T
%----------------------------------------------------------------------
 %-----------------------------------------------------------------------
% Perform Differential Crossover/Mutation for the Gas Turbine
% Generator Bus Voltage Chromosome (Cr_VGTG_Selec_T)
%------------------------------------------------------------------------
% Measure Chromosome (Cr_TD_Selec_T) lengths
```

```
n=length(Cr_VGTG_Cros(:,1));
d=length(Cr_VGTG_Cros(1,:));

%------------------The DE Crossover Loop-----------------------------------
for k=1:3:N_Pop-2
    h=1+k;
    d=2+k;
% Capture the Main Transformer Tap Value
Cr_VGTG_Cros_DET(k,:)=Cr_VGTG_Cros(k,:)+ B*(Cr_VGTG_Cros(h,:)-
Cr_VGTG_Cros(d,:));
Cr_VGTG_Cros_DET(h,:)=Cr_VGTG_Cros(h,:)+ B*(Cr_VGTG_Cros(d,:)-
Cr_VGTG_Cros(k,:));
Cr_VGTG_Cros_DET(d,:)=Cr_VGTG_Cros(d,:)+ B*(Cr_VGTG_Cros(k,:)-
Cr_VGTG_Cros(h,:));
end
clear k h d n

 %-------------------------------------------------------------------------
 % Assign the last selected chromosomes as 200 population is even number
 % Capture the Main Transformer Tap Value
 Cr_VGTG_Cros_DET(N_Pop,:)=Cr_VGTG_Cros(N_Pop,:);

%---Make sure that the Crossovered/Mutated Genes are matching the real Tap
% Values---
% Measure Chromosome (Cr_VGTG) lengths
 n=length(Cr_VGTG_Cros_DET(:,1));
 d=length(Cr_VGTG_Cros_DET(1,:));
% Measure The Tap Values
x=length(B_VTG(1,:));

for h=1:n
     for k=1:d
         for z=1:x
         Cr_VGTG_Cros_DET_T(z,:)=[z abs(Cr_VGTG_Cros_DET(h,k)-B_VTG(1,z))];
         end
         Cr_VGTG_Cros_DET_TS(:,:)=sortrows(Cr_VGTG_Cros_DET_T(1:z,:),2);
         Cr_VGTG_Cros_DET_R(h,k)=B_VTG(1,Cr_VGTG_Cros_DET_TS(1,1));
         Cr_VGTG_Cros(h,k)=Cr_VGTG_Cros_DET_R(h,k);
     end
end

clear h k n d x Cr_TM_Mut_TS Cr_TM_Mut_R Cr_TM_Mut_T
%-------------------------------------------------------------------------


%--------------------------------------------------------------------------
% Perform Differential Crossover/Mutation for the Steam Turbine Generator
% Bus Voltage Chromosome (Cr_VSTG_Selec_T)
%--------------------------------------------------------------------------
% Measure Chromosome (Cr_TD_Selec_T) lengths
n=length(Cr_VSTG_Cros(:,1));
d=length(Cr_VSTG_Cros(1,:));
% Identify the Crossover Vertical Lines CVL
%==========================================================================
%------------------The DE Crossover Loop-----------------------------------
for k=1:3:N_Pop-2
    h=1+k;
```

```
    d=2+k;
% Capture the Main Transformer Tap Value
Cr_VSTG_Cros_DET(k,:)=Cr_VSTG_Cros(k,:)+ B*(Cr_VSTG_Cros(h,:)-
Cr_VSTG_Cros(d,:));
Cr_VSTG_Cros_DET(h,:)=Cr_VGTG_Cros(h,:)+ B*(Cr_VSTG_Cros(d,:)-
Cr_VSTG_Cros(k,:));
Cr_VSTG_Cros_DET(d,:)=Cr_VSTG_Cros(d,:)+ B*(Cr_VSTG_Cros(k,:)-
Cr_VSTG_Cros(h,:));
end
clear k h d n
%------------------------------------------------------------------------
 % Assign the last selected chromosomes as 200 population is even number
 % Capture the Main Transformer Tap Value
 Cr_VSTG_Cros_DET(N_Pop,:)=Cr_VSTG_Cros(N_Pop,:);

%--Make sure that the Crossovered/Mutated Genes are matching the real Tap
% Values----
% Measure Chromosome (Cr_VGTG) lengths
 n=length(Cr_VSTG_Cros_DET(:,1));
 d=length(Cr_VSTG_Cros_DET(1,:));
% Meaure The Tap Values
x=length(B_VTG(1,:));

for h=1:n
     for k=1:d
         for z=1:x
         Cr_VSTG_Cros_DET_T(z,:)=[z abs(Cr_VSTG_Cros_DET(h,k)-B_VTG(1,z))];
         end
         Cr_VSTG_Cros_DET_TS(:,:)=sortrows(Cr_VSTG_Cros_DET_T(1:z,:),2);
         Cr_VSTG_Cros_DET_R(h,k)=B_VTG(1,Cr_VSTG_Cros_DET_TS(1,1));
         Cr_VSTG_Cros(h,k)=Cr_VSTG_Cros_DET_R(h,k);
     end
end

clear h k n d x Cr_TM_Mut_TS Cr_TM_Mut_R Cr_TM_Mut_T
%------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Perform Differential Crossover/Mutation for Captive Synchronous Motors Bus
% Voltage Chromosome (Cr_VSyMC_Selec_T)
%--------------------------------------------------------------------------
% Measure Chromosome (Cr_TD_Selec_T) lengths
n=length(Cr_VSyMC_Cros(:,1));
d=length(Cr_VSyMC_Cros(1,:));

%------------------The DE Crossover Loop------------------------------------
for k=1:3:N_Pop-2
    h=1+k;
    d=2+k;
% Capture the Main Transformer Tap Value
Cr_VSyMC_Cros_DET(k,:)=Cr_VSyMC_Cros(k,:)+ B*(Cr_VSyMC_Cros(h,:)-
Cr_VSyMC_Cros(d,:));
Cr_VSyMC_Cros_DET(h,:)=Cr_VSyMC_Cros(h,:)+ B*(Cr_VSyMC_Cros(d,:)-
Cr_VSyMC_Cros(k,:));
Cr_VSyMC_Cros_DET(d,:)=Cr_VSyMC_Cros(d,:)+ B*(Cr_VSyMC_Cros(k,:)-
Cr_VSyMC_Cros(h,:));
```

```
end
clear k h d n

 %-----------------------------------------------------------------------
 % Assign the last selected chromosomes as 200 population is even number
 % Capture the Main Transformer Tap Value
 Cr_VSyMC_Cros_DET(N_Pop,:)=Cr_VSyMC_Cros(N_Pop,:);

%--Make sure that the Crossovered/Mutated Genes are matching the real Tap
% Values----
% Measure Chromosome (Cr_VGTG) lengths
 n=length(Cr_VSyMC_Cros_DET(:,1));
 d=length(Cr_VSyMC_Cros_DET(1,:));
% Measure The Tap Values
x=length(B_VSyMC(1,:));

for h=1:n
     for k=1:d
         for z=1:x
         Cr_VSyMC_Cros_DET_T(z,:)=[z abs(Cr_VSyMC_Cros_DET(h,k)-
         B_VSyMC(1,z))];
         end
         Cr_VSyMC_Cros_DET_TS(:,:)=sortrows(Cr_VSyMC_Cros_DET_T(1:z,:),2);
         Cr_VSyMC_Cros_DET_R(h,k)=B_VSyMC(1,Cr_VSyMC_Cros_DET_TS(1,1));
         Cr_VSyMC_Cros(h,k)=Cr_VSyMC_Cros_DET_R(h,k);
     end
end

clear h k n d x Cr_TM_Mut_TS Cr_TM_Mut_R Cr_TM_Mut_T
%----------------------------------------------------------------------
```

APPENDIX XIII

The MATLAB Program Codes for Random and Non-uniform Mutation. [Refer to section 6.2.5]

The Random Mutation sub-program MATLAB codes

```
%*************************************************************************
% This Sub-Program will perform Random Mutation on the Chromosomes
%*************************************************************************

%--------------------------------------------------------------------------
% Save the Crossovered Parents Chromosomes in Temporary Matrixes;
% Cr_TM_Cros_T. This is to perform the Random Mutation while saving the
% Crossovered Parents saved
% Save the Transformer Taps Value chromosomes
Cr_TM_Cros_T= Cr_TM_Cros;
Cr_TC1_Cros_T= Cr_TC1_Cros;
Cr_TC2_Cros_T= Cr_TC2_Cros;
Cr_TC3_Cros_T= Cr_TC3_Cros;
Cr_TCM_Cros_T= Cr_TCM_Cros;
Cr_TD_Cros_T= Cr_TD_Cros;
Cr_TG_Cros_T= Cr_TG_Cros;
% Save the PV Buses Voltages chromosomes
Cr_VGTG_Cros_T= Cr_VGTG_Cros;
Cr_VSTG_Cros_T= Cr_VSTG_Cros;
Cr_VSyMC_Cros_T= Cr_VSyMC_Cros;
%Cr_VSyM_Cros_T= Cr_VSyM_Cros;

%--------------------------------------------------------------------------
% Save the Temporary Selected Parents Chromosomes in Crossover Matrixes;
% Cr_TM_Cros. In case the Crossover does not happen because Probability is
% < 90% the Crossover genes does not change.
% Save the Transformer Taps Value chromosomes
Cr_TM_Mut=Cr_TM_Cros_T;
Cr_TC1_Mut=Cr_TC1_Cros_T;
Cr_TC2_Mut=Cr_TC2_Cros_T;
Cr_TC3_Mut=Cr_TC3_Cros_T;
Cr_TCM_Mut=Cr_TCM_Cros_T;
Cr_TD_Mut=Cr_TD_Cros_T;
Cr_TG_Mut=Cr_TG_Cros_T;
% Save the PV Buses Voltages
Cr_VGTG_Mut=Cr_VGTG_Cros_T;
Cr_VSTG_Mut=Cr_VSTG_Cros_T;
Cr_VSyMC_Mut=Cr_VSyMC_Cros_T;
%Cr_VSyM_Mut=Cr_VSyM_Cros_T;

%-------------------------------------------------------------------------
% Perform the Random Mutation with 10% Probability
% Assign random number to each chromosomes;
  Rmin=0;
  Rmax=1;
```

```
% Assign the Mutation rate 10%
MutR=0.1;
%------------------------------------------------------------------------
% Select specific transformer tap values out of the full taps values
% Refer to Appendix VIII
%-------------- The selected tap for Large Transformers--------------------
Tap17=[Tap33_Down(1,9:16) 1 Tap33_Up(1,1:4)];
%------------The selected tap for Causeway Substations Main Transformers---
Tap12C=[Tap33_CDown(1,9:16) 1 Tap33_CUp(1,1:3)];
%----Captive Synchronous Motors and Distribution Transformers--------------
Tap5=[Tap5_Down(1,2) 1 Tap5_Up(1,1)];
%----------------------Generator Transformer-------------------------------
TapG=[TapG_Down(1,4:8) 1 TapG_Up(1,1:4)];
%------------------------------------------------------------------------
% Perform Random Mutation for the Large Power Transformer with 115kV
% primary Voltage Tap Chromosome (Cr_TM_Cros_T)
%------------------------------------------------------------------------
% Measure the Chromosomes Matrix(Cr_TM) lengths
 n=length(Cr_TM_Cros_T(:,1));
 d=length(Cr_TM_Cros_T(1,:));

% The Mutation loop that will mutate the genes with 10% Probability
 for k=1:n;
       for h=1:d;
       ChR(k,h)=(Rmin+rand*(Rmax-Rmin));
       if ChR(k,h)<MutR;
           if Cr_TM_Cros_T(k,h)>Tap17(1,2)& Cr_TM_Cros_T(k,h)<Tap17(1,12)
           A1=[-2 -1 1 2];
           B1=randi(4);
           Cr_TM_Mut(k,h)= Cr_TM_Cros_T(k,h)+0.00625*A1(1,B1);
          elseif
          Cr_TM_Cros_T(k,h)==Tap17(1,2)||Cr_TM_Cros_T(k,h)==Tap17(1,12)
           A2=[-1 1];
           B2=randi(2);
           Cr_TM_Mut(k,h)= Cr_TM_Cros_T(k,h)+0.00625*A2(1,B2);
           elseif Cr_TM_Cros_T(k,h)==Tap17(1,1)
           A3=[1 2];
           B3=randi(2);
           Cr_TM_Mut(k,h)= Cr_TM_Cros_T(k,h)+0.00625*A3(1,B3);
           elseif Cr_TM_Cros_T(k,h)==Tap17(1,13)
           A4=[-1 -2];
           B4=randi(2);
           Cr_TM_Mut(k,h)= Cr_TM_Cros_T(k,h)+0.00625*A4(1,B4);
           end
       end
    end
  end
% Clear all not needed variables
clear k n h n d ChR A1 A2 A3 A4 B1 B2 B3 B4

%------------------------------------------------------------------------
% Perform Random Mutation for the Causeway Substation#1 Power Transformer
% primary Voltage Tap Chromosome (Cr_TC1)
%------------------------------------------------------------------------
% Measure the Chromosomes Matrix (Cr_TC1) lengths
n=length(Cr_TC1_Cros_T(:,1));
```

```
d=length(Cr_TC1_Cros_T(1,:));

% The Mutation loop that will mutate the genes with 10% Probability
     for k=1:n;
       for h=1:d;
       ChR(k,h)=(Rmin+rand*(Rmax-Rmin));
       if ChR(k,h)<MutR;
        if Cr_TC1_Cros_T(k,h)>Tap12C(1,2)& Cr_TC1_Cros_T(k,h)<Tap12C(1,11)
           A1=[-2 -1 1 2];
           B1=randi(4);
           Cr_TC1_Mut(k,h)= Cr_TC1_Cros_T(k,h)+0.00625*A1(1,B1);
            elseif
            Cr_TC1_Cros_T(k,h)==Tap12C(1,2)||Cr_TC1_Cros_T(k,h)==Tap12C(1,11)
           A2=[-1 1];
           B2=randi(2);
           Cr_TC1_Mut(k,h)= Cr_TC1_Cros_T(k,h)+0.00625*A2(1,B2);
           elseif Cr_TC1_Cros_T(k,h)==Tap12C(1,1)
           A3=[1 2];
           B3=randi(2);
           Cr_TC1_Mut(k,h)= Cr_TC1_Cros_T(k,h)+0.00625*A3(1,B3);
           elseif Cr_TC1_Cros_T(k,h)==Tap12C(1,12)
           A4=[-1 -2];
           B4=randi(2);
           Cr_TC1_Mut(k,h)= Cr_TC1_Cros_T(k,h)+0.00625*A4(1,B4);
           end
       end
    end
  end

% Clear all not needed variables
clear k n h n d ChR A1 A2 A3 A4 B1 B2 B3 B4

%------------------------------------------------------------------------
% Perform Random Mutation for the Causeway Substation#2 Power Transformer
% primary Voltage Tap Chromosome (Cr_TC2)
%------------------------------------------------------------------------
% Measure the Chromosomes Matrix (Cr_TC2) lengths
n=length(Cr_TC2_Cros_T(:,1));
d=length(Cr_TC2_Cros_T(1,:));

% The Mutation loop that will mutate the genes with 10% Probability
     for k=1:n;
       for h=1:d;
       ChR(k,h)=(Rmin+rand*(Rmax-Rmin));
        if ChR(k,h)<MutR;
        if Cr_TC2_Cros_T(k,h)>Tap12C(1,2)& Cr_TC2_Cros_T(k,h)<Tap12C(1,11)
           A1=[-2 -1 1 2];
           B1=randi(4);
           Cr_TC2_Mut(k,h)= Cr_TC2_Cros_T(k,h)+0.00625*A1(1,B1);
          elseif
          Cr_TC2_Cros_T(k,h)==Tap12C(1,2)||Cr_TC2_Cros_T(k,h)==Tap12C(1,11)
           A2=[-1 1];
           B2=randi(2);
           Cr_TC2_Mut(k,h)= Cr_TC2_Cros_T(k,h)+0.00625*A2(1,B2);
           elseif Cr_TC2_Cros_T(k,h)==Tap12C(1,1)
           A3=[1 2];
```

```
           B3=randi(2);
           Cr_TC2_Mut(k,h)= Cr_TC2_Cros_T(k,h)+0.00625*A3(1,B3);
           elseif Cr_TC2_Cros_T(k,h)==Tap12C(1,12)
           A4=[-1 -2];
           B4=randi(2);
           Cr_TC2_Mut(k,h)= Cr_TC2_Cros_T(k,h)+0.00625*A4(1,B4);
           end
         end
    end
  end

% Clear all not needed variables
clear k n h n d ChR A1 A2 A3 A4 B1 B2 B3 B4

%-------------------------------------------------------------------------
% Perform Random Mutation for the Causeway Substation#3 Power Transformer
% primary Voltage Tap Chromosome (Cr_TC3)
%-------------------------------------------------------------------------
% Measure the Chromosomes Matrix (Cr_TC2) lengths
n=length(Cr_TC3_Cros_T(:,1));
d=length(Cr_TC3_Cros_T(1,:));

% The Mutation loop that will mutate the genes with 10% Probability
     for k=1:n;
       for h=1:d;
       ChR(k,h)=(Rmin+rand*(Rmax-Rmin));
       if ChR(k,h)<MutR;
           if Cr_TC3_Cros_T(k,h)>Tap12C(1,2)&
            Cr_TC3_Cros_T(k,h)<Tap12C(1,11)
           A1=[-2 -1 1 2];
           B1=randi(4);
           Cr_TC3_Mut(k,h)= Cr_TC3_Cros_T(k,h)+0.00625*A1(1,B1);
           elseif
          Cr_TC3_Cros_T(k,h)==Tap12C(1,2)||Cr_TC3_Cros_T(k,h)==Tap12C(1,11)
           A2=[-1 1];
           B2=randi(2);
           Cr_TC3_Mut(k,h)= Cr_TC3_Cros_T(k,h)+0.00625*A2(1,B2);
           elseif Cr_TC3_Cros_T(k,h)==Tap12C(1,1)
           A3=[1 2];
           B3=randi(2);
           Cr_TC3_Mut(k,h)= Cr_TC3_Cros_T(k,h)+0.00625*A3(1,B3);
           elseif Cr_TC3_Cros_T(k,h)==Tap12C(1,12)
           A4=[-1 -2];
           B4=randi(2);
           Cr_TC3_Mut(k,h)= Cr_TC3_Cros_T(k,h)+0.00625*A4(1,B4);
           end
       end
    end
  end

% Clear all not needed variables
clear k n h n d ChR A1 A2 A3 A4 B1 B2 B3 B4
%-------------------------------------------------------------------------
% Perform Random Mutation for the Captive Synchronous Motors power
% Transformer primary Voltage Tap Chromosome (Cr_TCM)
%-------------------------------------------------------------------------
```

```
% Measure the Chromosomes Matrix (Cr_TCM) lengths
n=length(Cr_TCM_Cros_T(:,1));
d=length(Cr_TCM_Cros_T(1,:));

% The Mutation loop that will mutate the genes with 10% Probability
     for k=1:n;
       for h=1:d;
       ChR(k,h)=(Rmin+rand*(Rmax-Rmin));
       if ChR(k,h)<MutR;
           if Cr_TCM_Cros_T(k,h)==Tap5(1,2)
           A1=[-1 1 ];
           B1=randi(2);
           Cr_TCM_Mut(k,h)= Cr_TCM_Cros_T(k,h)+0.025*A1(1,B1);
           elseif Cr_TCM_Cros_T(k,h)==Tap5(1,1)
           A2=[1 2];
           B2=randi(2);
           Cr_TCM_Mut(k,h)= Cr_TCM_Cros_T(k,h)+0.025*A2(1,B2);
           elseif Cr_TCM_Cros_T(k,h)==Tap5(1,3)
           A3=[-1 -2];
           B3=randi(2);
           Cr_TCM_Mut(k,h)= Cr_TCM_Cros_T(k,h)+0.025*A3(1,B3);
           end
       end
    end
  end

% Clear all not needed variables
clear k n h n d ChR A1 A2 A3 A4 B1 B2 B3 B4
%-------------------------------------------------------------------------
% Perform Random Mutation for the Distribution Power Transformer primary
% Voltage Tap Chromosome (Cr_TD)
%-------------------------------------------------------------------------
% Meaure the Chromosomes Matrix (Cr_TD) lengths
n=length(Cr_TD_Cros_T(:,1));
d=length(Cr_TD_Cros_T(1,:));

% The Mutation loop that will mutate the genes with 10% Probability
     for k=1:n;
       for h=1:d;
       ChR(k,h)=(Rmin+rand*(Rmax-Rmin));
       if ChR(k,h)<MutR;
           if Cr_TD_Cros_T(k,h)==Tap5(1,2)
           A1=[-1 1 ];
           B1=randi(2);
           Cr_TD_Mut(k,h)= Cr_TD_Cros_T(k,h)+0.025*A1(1,B1);
           elseif Cr_TD_Cros_T(k,h)==Tap5(1,1)
           A2=[1 2];
           B2=randi(2);
           Cr_TD_Mut(k,h)= Cr_TD_Cros_T(k,h)+0.025*A2(1,B2);
           elseif Cr_TD_Cros_T(k,h)==Tap5(1,3)
           A3=[-1 -2];
           B3=randi(2);
           Cr_TD_Mut(k,h)= Cr_TD_Cros_T(k,h)+0.025*A3(1,B3);
           end
       end
    end
```

```
  end

% Clear all not needed variables
clear k n h n d ChR A1 A2 A3 A4 B1 B2 B3 B4

%-------------------------------------------------------------------------
% Perform Random Mutation for the Generator Step-Up Power Transformer
% Voltage Tap Chromosome (Cr_TG)
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
% Measure the Chromosomes Matrix (Cr_TG) lengths
 n=length(Cr_TG_Cros_T(:,1));
 d=length(Cr_TG_Cros_T(1,:));

% The Mutation loop that will mutate the genes with 10% Probability

  for k=1:n;
       for h=1:d;
       ChR(k,h)=(Rmin+rand*(Rmax-Rmin));
       if ChR(k,h)<MutR;
           if Cr_TG_Cros_T(k,h)>TapG(1,2)& Cr_TG_Cros_T(k,h)<TapG(1,9)
           A1=[-2 -1 1 2];
           B1=randi(4);
           Cr_TG_Mut(k,h)= Cr_TG_Cros_T(k,h)+0.0125*A1(1,B1);
           elseif Cr_TG_Cros_T(k,h)==TapG(1,2)||Cr_TG_Cros_T(k,h)==TapG(1,9)
           A2=[-1 1];
           B2=randi(2);
           Cr_TG_Mut(k,h)= Cr_TG_Cros_T(k,h)+0.0125*A2(1,B2);
           elseif Cr_TG_Cros_T(k,h)==TapG(1,1)
           A3=[1 2];
           B3=randi(2);
           Cr_TG_Mut(k,h)= Cr_TG_Cros_T(k,h)+0.0125*A3(1,B3);
           elseif Cr_TG_Cros_T(k,h)==TapG(1,10)
           A4=[-1 -2];
           B4=randi(2);
           Cr_TG_Mut(k,h)= Cr_TG_Cros_T(k,h)+0.0125*A4(1,B4);
           end
       end
     end
  end
% Clear all not needed variables
clear k n h n d ChR A1 A2 A3 A4 B1 B2 B3 B4
%
clear k n h n d ChR A1 A2 A3 A4 B1 B2 B3 B4

%-------------------------------------------------------------------------
% Perform Random Mutation for the Gas Turbine Generator Bus Volatge
% Chromosome (Cr_VGTG)
%-------------------------------------------------------------------------
% Measure the Chromosomes Matrix (Cr_VGTG) lengths
n=length(Cr_VGTG_Cros_T(:,1));
d=length(Cr_VGTG_Cros_T(1,:));
%-------------------------------------------------------------------------
% Select specific bus voltage values out of the full taps values
% Refer to Appendix VII
%------------The selected tap for Synchronous motors------------------------
```

```
B_VSyMC=SyM_TV(1,3:7);
%--------------------------------------------------------------------------
% The Mutation loop that will mutate the genes with 10% Probability
     for k=1:n;
       for h=1:d;
       ChR(k,h)=(Rmin+rand*(Rmax-Rmin));
       if ChR(k,h)<MutR;
           if Cr_VGTG_Cros_T(k,h)>B_VTG(1,2)& Cr_VGTG_Cros_T(k,h)<B_VTG(1,10)
           A1=[-2 -1 1 2];
           B1=randi(4);
           Cr_VGTG_Mut(k,h)= Cr_VGTG_Cros_T(k,h)+0.01*A1(1,B1);
            elseif
            Cr_VGTG_Cros_T(k,h)==B_VTG(1,2)||Cr_VGTG_Cros_T(k,h)==B_VTG(1,10)
           A2=[-1 1];
           B2=randi(2);
           Cr_VGTG_Mut(k,h)= Cr_VGTG_Cros_T(k,h)+0.01*A2(1,B2);
           elseif Cr_VGTG_Cros_T(k,h)==B_VTG(1,1)
           A3=[1 2];
           B3=randi(2);
           Cr_VGTG_Mut(k,h)= Cr_VGTG_Cros_T(k,h)+0.01*A3(1,B3);
           elseif Cr_VGTG_Cros_T(k,h)==B_VTG(1,11)
           A4=[-1 -2];
           B4=randi(2);
           Cr_VGTG_Mut(k,h)= Cr_VGTG_Cros_T(k,h)+0.01*A4(1,B4);
           end
       end
    end
  end
% Clear all not needed variables
clear k n h n d ChR A1 A2 A3 A4 B1 B2 B3 B4

%--------------------------------------------------------------------------
% Perform Random Mutation for the Steam Turbine Generator Bus Volatge
% Chromosome (Cr_VSTG)
%--------------------------------------------------------------------------
% Measure the Chromosomes matrix (Cr_VSTG) lengths
n=length(Cr_VSTG_Cros_T(:,1));
d=length(Cr_VSTG_Cros_T(1,:));
% The Mutation loop that will mutate the gene with 10% Probability
    for k=1:n;
       for h=1:d;
       ChR(k,h)=(Rmin+rand*(Rmax-Rmin));
       if ChR(k,h)<MutR;
          if Cr_VSTG_Cros_T(k,h)>B_VTG(1,2)& Cr_VSTG_Cros_T(k,h)<B_VTG(1,10)
           A1=[-2 -1 1 2];
           B1=randi(4);
           Cr_VSTG_Mut(k,h)= Cr_VSTG_Cros_T(k,h)+0.01*A1(1,B1);
            elseif
            Cr_VSTG_Cros_T(k,h)==B_VTG(1,2)||Cr_VSTG_Cros_T(k,h)==B_VTG(1,10)
           A2=[-1 1];
           B2=randi(2);
           Cr_VSTG_Mut(k,h)= Cr_VSTG_Cros_T(k,h)+0.01*A2(1,B2);
           elseif Cr_VSTG_Cros_T(k,h)==B_VTG(1,1)
           A3=[1 2];
           B3=randi(2);
           Cr_VSTG_Mut(k,h)= Cr_VSTG_Cros_T(k,h)+0.01*A3(1,B3);
           elseif Cr_VSTG_Cros_T(k,h)==B_VTG(1,11)
```

```
            A4=[-1 -2];
            B4=randi(2);
            Cr_VSTG_Mut(k,h)= Cr_VSTG_Cros_T(k,h)+0.01*A4(1,B4);
            end
        end
     end
   end
% Clear all not needed variables
clear k n h n d ChR A1 A2 A3 A4 B1 B2 B3 B4

%-------------------------------------------------------------------------
% Perform Random Mutation for the Captive Synchronus Motor Bus Volatge
% Chromosome (Cr_VSyMC)
%-------------------------------------------------------------------------
% Measure the Chromosomes matrix (Cr_ VSyMC) lengths
n=length(Cr_VSyMC_Cros_T(:,1));
d=length(Cr_VSyMC_Cros_T(1,:));
% The Mutation loop that will mutate the gene with 10% Probability
     for k=1:n;
       for h=1:d;
       ChR(k,h)=(Rmin+rand*(Rmax-Rmin));
       if ChR(k,h)<MutR;
       if Cr_VSyMC_Cros_T(k,h)==B_VSyMC(1,3)
       A1=[-2 -1 1 2];
       B1=randi(4);
       Cr_VSyMC_Mut(k,h)= Cr_VSyMC_Cros_T(k,h)+0.01*A1(1,B1);
       elseif
       Cr_VSyMC_Cros_T(k,h)==B_VSyMC(1,2)||Cr_VSyMC_Cros_T(k,h)==B_VSyMC(1,4)
       A2=[-1 1];
       B2=randi(2);
       Cr_VSyMC_Mut(k,h)= Cr_VSyMC_Cros_T(k,h)+0.01*A2(1,B2);
       elseif Cr_VSyMC_Cros_T(k,h)==B_VSyMC(1,1)
       A3=[1 2];
       B3=randi(2);
       Cr_VSyMC_Mut(k,h)= Cr_VSyMC_Cros_T(k,h)+0.01*A3(1,B3);
       elseif Cr_VSyMC_Cros_T(k,h)==B_VSyMC(1,5)
       A4=[-1 -2];
       B4=randi(2);
       Cr_VSyMC_Mut(k,h)= Cr_VSyMC_Cros_T(k,h)+0.01*A4(1,B4);
       end
       end
    end
  end
% Clear all not needed variables
clear k n h n d ChR A1 A2 A3 A4 B1 B2 B3 B4
```

The Non-Uniform Mutation sub-program MATLAB codes

```
%***********************************************************************
% This Sub-Program will perform Non-Uniform Mutation on the Chromosomes
%***********************************************************************

%------------------------------------------------------------------------
% Save the Crossovered Parents Chromosomes in Temporary Matrixes;
% Cr_TM_Cros_T. This is to perform the Random Mutation while saving the
% Crossovered Parents saved
% Save the Transformer Taps Value Chromosomes
Cr_TM_Cros_T= Cr_TM_Cros;
Cr_TC1_Cros_T= Cr_TC1_Cros;
Cr_TC2_Cros_T= Cr_TC2_Cros;
Cr_TC3_Cros_T= Cr_TC3_Cros;
Cr_TCM_Cros_T= Cr_TCM_Cros;
Cr_TD_Cros_T= Cr_TD_Cros;
Cr_TG_Cros_T= Cr_TG_Cros;
% Save the PV Buses Voltages Chromosomes
Cr_VGTG_Cros_T= Cr_VGTG_Cros;
Cr_VSTG_Cros_T= Cr_VSTG_Cros;
Cr_VSyMC_Cros_T= Cr_VSyMC_Cros;
%Cr_VSyM_Cros_T= Cr_VSyM_Cros;

%------------------------------------------------------------------------
% Save the Temporary Selected Parents Chromosomes in Crossover Variable;
% Cr_TM_Cros. In case the Crossover does not happen because Probability is
% < 90% the Crossover genes does not change.
% Save the Transformer Taps Value Chromosomes
Cr_TM_Mut=Cr_TM_Cros_T;
Cr_TC1_Mut=Cr_TC1_Cros_T;
Cr_TC2_Mut=Cr_TC2_Cros_T;
Cr_TC3_Mut=Cr_TC3_Cros_T;
Cr_TCM_Mut=Cr_TCM_Cros_T;
Cr_TD_Mut=Cr_TD_Cros_T;
Cr_TG_Mut=Cr_TG_Cros_T;
% Save the PV Buses Voltages
Cr_VGTG_Mut=Cr_VGTG_Cros_T;
Cr_VSTG_Mut=Cr_VSTG_Cros_T;
Cr_VSyMC_Mut=Cr_VSyMC_Cros_T;
%Cr_VSyM_Mut=Cr_VSyM_Cros_T;

%-----------------------------------------------------------------------
% Perform the Random Mutation with 10% Probability
% Assign random number to each chromosome;
  Rmin=0;
  Rmax=1;
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
% Select specific transformer tap values out of the full taps values
% Refer to Appendix VIII
%-------------- The selected tap for Large Transformers--------------------
Tap17=[Tap33_Down(1,9:16) 1 Tap33_Up(1,1:4)];
%------------The selected tap for Causeway Substations Main Transformers---
```

```
Tap12C=[Tap33_CDown(1,9:16) 1 Tap33_CUp(1,1:3)];
%----Captive Synchronous Motors and Distribution Transformers--------------
Tap5=[Tap5_Down(1,2) 1 Tap5_Up(1,1)];
%----------------------Generator Transformer-------------------------------
TapG=[TapG_Down(1,4:8) 1 TapG_Up(1,1:4)];
%--------------------------------------------------------------------------
% ------------------------- Large Main Transformer-------------------------
% The Limit for the Main Substations Transformers Taps Position
% Genes Values refer to Appendix VIII
  Tap17_min=0.95;
  Tap17_max=1.025;

% -------------------Causeway Substations Main Transformers----------------
% The Limit for the Causeway Substations Main Transformers Taps Position
% Gense Values
  Tap12C_min=0.95;
  Tap12C_max=1.0188;

% ---------Captive Synchronous Motors and Distribution Transformers--------
% The Limit for the Captive Synchronous Motors and Distribution
% Transformers Taps Position Genes Values
  Tap5_min=0.975;
  Tap5_max=1.025;

% --------------------Generator Step-Up Transformers-----------------------
% The Limit for the Generator Step-Up Transformers Taps Position Genes
% Values
  TapG_min=0.9375;
  TapG_max=1.05;
%-------------------------------------------------------------------------
% Select specific bus voltage values out of the full taps values
% Refer to Appendix VII

%------------The selected tap for Synchronous motors------------------------
B_VSyMC=SyM_TV(1,3:7);
%-------------------------------------------------------------------------

% -------------Gas & Steam Turbine Generators Buses voltages---------------
% The Limit for the Gas & Steam Turbine Generators Bus voltage Gene Values
  Vmin_TG = 0.95;
  Vmax_TG = 1.05;

% ----------------------Synchronous Motors Buses voltages-------------------
% The Limit for the Synchronous Motors Bus voltage Gene Values
  Vmin_SyMC = 0.97;
  Vmax_SyMC = 1.01;

%------------------------------------------------------------------------
% Assign random number to each chromosome');
  Mmin=0;
  Mmax=1;


%-------------------------------------------------------------------------
% Assign random number for τ (t) and check if t =<0.5 or t>0.5
```

```
tmin=0;
tmax=1;

% Assign random number for r variable
rmin=0;
rmax=1;
%------------------------------------------------------------------------
% Assign the value b;
b=0.05;
%------------------------------------------------------------------------

%------------------------------------------------------------------------
% Perform Random Mutation for the Large Power Transformer with 115kV
% primary Voltage Tap Chromosomes (Cr_TM_Cros_T)
%------------------------------------------------------------------------
% Measure Chromosome Matrix (Cr_TM) lengths
 n=length(Cr_TM_Cros_T(:,1));
 d=length(Cr_TM_Cros_T(1,:));
% Measure The Tap Values
x=length(Tap17(1,:));

%-------------------- The Mutation Loop-----------------------------------
for h=1:n
     for k=1:d
     t=tmin+rand*(tmax-tmin);
     r=rmin+rand*(rmax-rmin);
     if t <= 0.5
    Cr_TM_Mut(h,k)= Cr_TM_Cros_T(h,k)+(Tap17_max-Cr_TM_Cros_T(h,k))*(1-r^(1-
    (Ge_N/Gmax)^b));
    else
    Cr_TM_Mut(h,k)= Cr_TM_Cros_T(h,k)-(Cr_TM_Cros_T(h,k)-Tap17_min)*(1-r^(1-
    (Ge_N/Gmax)^b));
    end
     end
end

%-----Make sure that the Mutated Genes are matching the real Tap Values----
for h=1:n
     for k=1:d
         for z=1:x
         Cr_TM_Mut_T(z,:)=[z abs(Cr_TM_Mut(h,k)-Tap17(1,z))];
         end
         Cr_TM_Mut_TS(:,:)=sortrows(Cr_TM_Mut_T(1:z,:),2);
         Cr_TM_Mut_R(h,k)=Tap17(1,Cr_TM_Mut_TS(1,1));
         Cr_TM_Mut(h,k)=Cr_TM_Mut_R(h,k);
     end
end
BB=Cr_TM_Mut;

clear h k n d x Cr_TM_Mut_TS Cr_TM_Mut_R Cr_TM_Mut_T
%------------------------------------------------------------------------
% Perform Random Mutation for the Causeway Substation#1 Power Transformer
% primary Voltage Tap Chromosomes (Cr_TC1)
%------------------------------------------------------------------------
% Measure Chromosome Matrix(Cr_TC1) lengths
n=length(Cr_TC1_Cros_T(:,1));
```

```
d=length(Cr_TC1_Cros_T(1,:));
% Measure the Tap Values
x=length(Tap12C(1,:));
%-------------------- The Mutation Loop------------------------------------
for h=1:n
     for k=1:d
     t=tmin+rand*(tmax-tmin);
     r=rmin+rand*(rmax-rmin);
             if t <= 0.5
             Cr_TC1_Mut(h,k)= Cr_TC1_Cros_T(h,k)+(Tap12C_max-
             Cr_TC1_Cros_T(h,k))*(1-r^(1-(Ge_N/Gmax)^b));
             else
             Cr_TC1_Mut(h,k)= Cr_TC1_Cros_T(h,k)-(Cr_TC1_Cros_T(h,k)-
             Tap12C_min)*(1-r^(1-(Ge_N/Gmax)^b));
             end
     end
end

%-----Make sure that the Mutated Genes are matching the real Tap Values----
for h=1:n
     for k=1:d
         for z=1:x
         Cr_TC1_Mut_T(z,:)=[z abs(Cr_TC1_Mut(h,k)-Tap12C(1,z))];
         end
         Cr_TC1_Mut_TS(:,:)=sortrows(Cr_TC1_Mut_T(1:z,:),2);
         Cr_TC1_Mut_R(h,k)=Tap12C(1,Cr_TC1_Mut_TS(1,1));
         Cr_TC1_Mut(h,k)=Cr_TC1_Mut_R(h,k);
     end
end

clear h k n d x Cr_TC1_Mut_TS Cr_TC1_Mut_R Cr_TC1_Mut_T


%-------------------------------------------------------------------------
% Perform Random Mutation for the Causeway Substation#2 Power Transformer
% primary Voltage Tap Chromosome (Cr_TC2)
%-------------------------------------------------------------------------
% Measure Chromosome Matrix (Cr_TC2) lengths
n=length(Cr_TC2_Cros_T(:,1));
d=length(Cr_TC2_Cros_T(1,:));
% Measure the Tap Values
x=length(Tap12C(1,:));

%-------------------- The Mutation Loop------------------------------------
for h=1:n
     for k=1:d
     t=tmin+rand*(tmax-tmin);
     r=rmin+rand*(rmax-rmin);
             if t <= 0.5
             Cr_TC2_Mut(h,k)= Cr_TC2_Cros_T(h,k)+(Tap12C_max-
             Cr_TC2_Cros_T(h,k))*(1-r^(1-(Ge_N/Gmax)^b));
             else
             Cr_TC2_Mut(h,k)= Cr_TC2_Cros_T(h,k)-(Cr_TC2_Cros_T(h,k)-
             Tap12C_min)*(1-r^(1-(Ge_N/Gmax)^b));
             end
     end
```

```
end

%-----Make sure that the Mutated Genes are matching the real Tap Values----
for h=1:n
     for k=1:d
         for z=1:x
         Cr_TC2_Mut_T(z,:)=[z abs(Cr_TC2_Mut(h,k)-Tap12C(1,z))];
         end
         Cr_TC2_Mut_TS(:,:)=sortrows(Cr_TC2_Mut_T(1:z,:),2);
         Cr_TC2_Mut_R(h,k)=Tap12C(1,Cr_TC2_Mut_TS(1,1));
         Cr_TC2_Mut(h,k)=Cr_TC2_Mut_R(h,k);
     end
end

clear h k n d x Cr_TC2_Mut_TS Cr_TC2_Mut_R Cr_TC2_Mut_T


%--------------------------------------------------------------------------
% Perform Random Mutation for the Causeway Substation#3 Power Transformer
% primary Voltage Tap Chromosome (Cr_TC3)
%--------------------------------------------------------------------------
% Measure Chromosome Matrix (Cr_TC2) lengths
n=length(Cr_TC3_Cros_T(:,1));
d=length(Cr_TC3_Cros_T(1,:));
% Measure The Tap Values
x=length(Tap12C(1,:));

%-------------------- The Mutation Loop-----------------------------------
for h=1:n
     for k=1:d
     t=tmin+rand*(tmax-tmin);
     r=rmin+rand*(rmax-rmin);
               if t <= 0.5
               Cr_TC3_Mut(h,k)= Cr_TC3_Cros_T(h,k)+(Tap12C_max-
               Cr_TC3_Cros_T(h,k))*(1-r^(1-(Ge_N/Gmax)^b));
               else
               Cr_TC3_Mut(h,k)= Cr_TC3_Cros_T(h,k)-(Cr_TC3_Cros_T(h,k)-
               Tap12C_min)*(1-r^(1-(Ge_N/Gmax)^b));
               end
     end
end

%-----Make sure that the Mutated Genes are matching the real Tap Values----
for h=1:n
     for k=1:d
         for z=1:x
         Cr_TC3_Mut_T(z,:)=[z abs(Cr_TC3_Mut(h,k)-Tap12C(1,z))];
         end
         Cr_TC3_Mut_TS(:,:)=sortrows(Cr_TC3_Mut_T(1:z,:),2);
         Cr_TC3_Mut_R(h,k)=Tap12C(1,Cr_TC3_Mut_TS(1,1));
         Cr_TC3_Mut(h,k)=Cr_TC3_Mut_R(h,k);
     end
end

clear h k n d x Cr_TC3_Mut_TS Cr_TC3_Mut_R Cr_TC3_Mut_T
```

```
%-------------------------------------------------------------------------
% Perform Random Mutation for the Captive Synchronous Motors power
% Transformer primary Voltage Tap Chromosome (Cr_TCM)
%-------------------------------------------------------------------------
% Measure Chromosome Matrix (Cr_TCM) lengths
n=length(Cr_TCM_Cros_T(:,1));
d=length(Cr_TCM_Cros_T(1,:));
% Measure The Tap Values
x=length(Tap5(1,:));

%-------------------- The Mutation Loop------------------------------------
for h=1:n
     for k=1:d
     t=tmin+rand*(tmax-tmin);
     r=rmin+rand*(rmax-rmin);
             if t <= 0.5
             Cr_TCM_Mut(h,k)= Cr_TCM_Cros_T(h,k)+(Tap5_max-
             Cr_TCM_Cros_T(h,k))*(1-r^(1-(Ge_N/Gmax)^b));
             else
             Cr_TCM_Mut(h,k)= Cr_TCM_Cros_T(h,k)-(Cr_TCM_Cros_T(h,k)-
             Tap5_min)*(1-r^(1-(Ge_N/Gmax)^b));
             end
     end
end

%-----Make sure that the Mutated Genes are matching the real Tap Values----
for h=1:n
     for k=1:d
         for z=1:x
         Cr_TCM_Mut_T(z,:)=[z abs(Cr_TCM_Mut(h,k)-Tap5(1,z))];
         end
         Cr_TCM_Mut_TS(:,:)=sortrows(Cr_TCM_Mut_T(1:z,:),2);
         Cr_TCM_Mut_R(h,k)=Tap5(1,Cr_TCM_Mut_TS(1,1));
         Cr_TCM_Mut(h,k)=Cr_TCM_Mut_R(h,k);
     end
end

clear h k n d x Cr_TCM_Mut_TS Cr_TCM_Mut_R Cr_TCM_Mut_T

%-------------------------------------------------------------------------
% Perform Random Mutation for the Distribution Power Transformer primary
% Voltage Tap Chromosome (Cr_TD)
%-------------------------------------------------------------------------
% Measure Chromosome Matrix (Cr_TD) lengths
n=length(Cr_TD_Cros_T(:,1));
d=length(Cr_TD_Cros_T(1,:));
% Measure The Tap Values
x=length(Tap5(1,:));

%-------------------- The Mutation Loop------------------------------------
for h=1:n
     for k=1:d
     t=tmin+rand*(tmax-tmin);
     r=rmin+rand*(rmax-rmin);
             if t <= 0.5
```

```
             Cr_TD_Mut(h,k)= Cr_TD_Cros_T(h,k)+(Tap5_max-
             Cr_TD_Cros_T(h,k))*(1-r^(1-(Ge_N/Gmax)^b));
             else
             Cr_TD_Mut(h,k)= Cr_TD_Cros_T(h,k)-(Cr_TD_Cros_T(h,k)-
             Tap5_min)*(1-r^(1-(Ge_N/Gmax)^b));
             end
     end
end

%-----Make sure that the Mutated Genes are matching the real Tap Values----
for h=1:n
     for k=1:d
         for z=1:x
         Cr_TD_Mut_T(z,:)=[z abs(Cr_TD_Mut(h,k)-Tap5(1,z))];
         end
         Cr_TD_Mut_TS(:,:)=sortrows(Cr_TD_Mut_T(1:z,:),2);
         Cr_TD_Mut_R(h,k)=Tap5(1,Cr_TD_Mut_TS(1,1));
         Cr_TD_Mut(h,k)=Cr_TD_Mut_R(h,k);
     end
end
clear h k n d x Cr_TD_Mut_TS Cr_TD_Mut_R Cr_TD_Mut_T

%--------------------------------------------------------------------------
% Perform Random Mutation for the Generator Step-Up Power Transformer
% Voltage Tap Chromosome (Cr_TG)
%--------------------------------------------------------------------------
%--------------------------------------------------------------------------
% Measure Chromosome Matrix (Cr_TG) lengths
 n=length(Cr_TG_Cros_T(:,1));
 d=length(Cr_TG_Cros_T(1,:));
% Measure The Tap Values
x=length(TapG(1,:));

%-------------------- The Mutation Loop------------------------------------
for h=1:n
     for k=1:d
     t=tmin+rand*(tmax-tmin);
     r=rmin+rand*(rmax-rmin);
             if t <= 0.5
             Cr_TG_Mut(h,k)= Cr_TG_Cros_T(h,k)+(TapG_max-
             Cr_TG_Cros_T(h,k))*(1-r^(1-(Ge_N/Gmax)^b));
             else
             Cr_TG_Mut(h,k)= Cr_TG_Cros_T(h,k)-(Cr_TG_Cros_T(h,k)-
             TapG_min)*(1-r^(1-(Ge_N/Gmax)^b));
             end
     end
end

%-----Make sure that the Mutated Genes are matching the real Tap Values----
for h=1:n
     for k=1:d
         for z=1:x
         Cr_TG_Mut_T(z,:)=[z abs(Cr_TG_Mut(h,k)-TapG(1,z))];
         end
         Cr_TG_Mut_TS(:,:)=sortrows(Cr_TG_Mut_T(1:z,:),2);
         Cr_TG_Mut_R(h,k)=TapG(1,Cr_TG_Mut_TS(1,1));
```

```
            Cr_TG_Mut(h,k)=Cr_TG_Mut_R(h,k);
        end
end

clear h k n d x Cr_TG_Mut_TS Cr_TG_Mut_R Cr_TG_Mut_T

%------------------------------------------------------------------------
% Perform Random Mutation for the Gas Turbine Generator Bus Voltage
% Chromosome (Cr_VGTG)
%------------------------------------------------------------------------
% Measure Chromosome Matrix (Cr_VGTG) lengths
n=length(Cr_VGTG_Cros_T(:,1));
d=length(Cr_VGTG_Cros_T(1,:));
% Measure The Generators Terminal Voltage Values
x=length(B_VTG(1,:));


%-------------------- The Mutation Loop-----------------------------------
for h=1:n
     for k=1:d
     t=tmin+rand*(tmax-tmin);
     r=rmin+rand*(rmax-rmin);
             if t <= 0.5
             Cr_VGTG_Mut(h,k)= Cr_VGTG_Cros_T(h,k)+(Vmax_TG-
             Cr_VGTG_Cros_T(h,k))*(1-r^(1-(Ge_N/Gmax)^b));
             else
             Cr_VGTG_Mut(h,k)= Cr_VGTG_Cros_T(h,k)-(Cr_VGTG_Cros_T(h,k)-
             Vmin_TG)*(1-r^(1-(Ge_N/Gmax)^b));
             end
     end
end

%-----Make sure that the Mutated Genes are matching the real Tap Values----
for h=1:n
     for k=1:d
         for z=1:x
         Cr_VGTG_Mut_T(z,:)=[z abs(Cr_VGTG_Mut(h,k)-B_VTG(1,z))];
         end
         Cr_VGTG_Mut_TS(:,:)=sortrows(Cr_VGTG_Mut_T(1:z,:),2);
         Cr_VGTG_Mut_R(h,k)=B_VTG(1,Cr_VGTG_Mut_TS(1,1));
         Cr_VGTG_Mut(h,k)=Cr_VGTG_Mut_R(h,k);
     end
end

clear h k n d x Cr_VGTG_Mut_TS Cr_VGTG_Mut_R Cr_VGTG_Mut_T

%------------------------------------------------------------------------
% Perform Random Mutation for the Steam Turbine Generator Bus Voltage
% Chromosome (Cr_VSTG)
%------------------------------------------------------------------------
% Measure Chromosome (Cr_VSTG) lengths
n=length(Cr_VSTG_Cros_T(:,1));
d=length(Cr_VSTG_Cros_T(1,:));
% Measure the Generators Terminal Voltage Values
x=length(B_VTG(1,:));
```

```
%-------------------- The Mutation Loop------------------------------------
for h=1:n
     for k=1:d
     t=tmin+rand*(tmax-tmin);
     r=rmin+rand*(rmax-rmin);
             if t <= 0.5
             Cr_VSTG_Mut(h,k)= Cr_VSTG_Cros_T(h,k)+(Vmax_TG-
             Cr_VSTG_Cros_T(h,k))*(1-r^(1-(Ge_N/Gmax)^b));
             else
             Cr_VSTG_Mut(h,k)= Cr_VSTG_Cros_T(h,k)-(Cr_VSTG_Cros_T(h,k)-
             Vmin_TG)*(1-r^(1-(Ge_N/Gmax)^b));
             end
     end
end

%-----Make sure that the Mutated Genes are matching the real Tap Values----
for h=1:n
     for k=1:d
         for z=1:x
         Cr_VSTG_Mut_T(z,:)=[z abs(Cr_VSTG_Mut(h,k)-B_VTG(1,z))];
         end
         Cr_VSTG_Mut_TS(:,:)=sortrows(Cr_VSTG_Mut_T(1:z,:),2);
         Cr_VSTG_Mut_R(h,k)=B_VTG(1,Cr_VSTG_Mut_TS(1,1));
         Cr_VSTG_Mut(h,k)=Cr_VSTG_Mut_R(h,k);
     end
end

clear h k n d x Cr_VSTG_Mut_TS Cr_VSTG_Mut_R Cr_VSTG_Mut_T

%-------------------------------------------------------------------------
% Perform Random Mutation for the Captive Synchronous Motor Bus Voltage
% Chromosome (Cr_VSyMC)
%-------------------------------------------------------------------------
% Measure Chromosome Matrix (Cr_VSyMC) lengths
n=length(Cr_VSyMC_Cros_T(:,1));
d=length(Cr_VSyMC_Cros_T(1,:));
% Measure The Generators Terminal Voltage Values
x=length(B_VSyMC(1,:));

%-------------------- The Mutation Loop------------------------------------
for h=1:n
     for k=1:d
     t=tmin+rand*(tmax-tmin);
     r=rmin+rand*(rmax-rmin);
             if t <= 0.5
             Cr_VSyMC_Mut(h,k)= Cr_VSyMC_Cros_T(h,k)+(Vmax_SyMC-
             Cr_VSyMC_Cros_T(h,k))*(1-r^(1-(Ge_N/Gmax)^b));
             else
             Cr_VSyMC_Mut(h,k)= Cr_VSyMC_Cros_T(h,k)-(Cr_VSyMC_Cros_T(h,k)-
             Vmin_SyMC)*(1-r^(1-(Ge_N/Gmax)^b));
             end
     end
end

%-----Make sure that the Mutated Genes are matching the real Tap Values----
```

```
for h=1:n
     for k=1:d
         for z=1:x
         Cr_VSyMC_Mut_T(z,:)=[z abs(Cr_VSyMC_Mut(h,k)-B_VSyMC(1,z))];
         end
         Cr_VSyMC_Mut_TS(:,:)=sortrows(Cr_VSyMC_Mut_T(1:z,:),2);
         Cr_VSyMC_Mut_R(h,k)=B_VTG(1,Cr_VSyMC_Mut_TS(1,1));
         Cr_VSyMC_Mut(h,k)=Cr_VSyMC_Mut_R(h,k);
     end
end

clear h k n d x Cr_VSyMC_Mut_TS Cr_VSyMC_Mut_R Cr_VSyMC_Mut_T
```

APPENDIX XIV

The MATLAB Program Codes for Fuzzy Theory Set (Extra_Best_Chr.m). [Refer to section 6.5]

```
%**************************************************************************
% Extraction of the Best Chromosomes Solution from the Front Pareto Optimal
% Set by applying Fuzzy set theory
%*************************************************************************
% Load the External Optimal Pareto Set (EOPS) size (Size_EOPS), this is
% needed to identify the EOPS size
load Size_EOPS_Sav
%--------------------------------------------------------------------------
% Identify Maximum & Minimum values of Objectives best values (Obj1 & Obj2)
% for all generations. The maximum number of generations is Ge_N
  Obj1_Max=max(J12_J1_Best(:,Ge_N));
  Obj1_Min=min(J12_J1_Best(:,Ge_N));
  Obj1_Dif=Obj1_Max-Obj1_Min;
  Obj2_Max=max(J12_J2_Best(:,Ge_N));
  Obj2_Min=min(J12_J2_Best(:,Ge_N));
  Obj2_Dif=Obj2_Max-Obj2_Min;
%--------------------------------------------------------------------------
% Assign the Mio values by apply equation 6.12
K=Size_EOPS_P;
  for X=1:K;
      if       J12_J1_Best(X,Gmax)<=Obj1_Min;
               Mio_1(X,1)=1;
      end
      if      J12_J2_Best(X,Gmax)<=Obj2_Min;
               Mio_2(X,1)=1;
      end
      if        J12_J1_Best(X,Gmax)>=Obj1_Max;
               Mio_1(X,1)=0;
      end
      if       J12_J2_Best(X,Gmax)>=Obj2_Max;
               Mio_2(X,1)=0;
      end
      if        J12_J1_Best(X,Gmax)<Obj1_Max &  J12_J1_Best(X,Gmax)>Obj1_Min
               Mio_1(X,1)=(Obj1_Max- J12_J1_Best(X,Gmax))/Obj1_Dif;
      end
      if      J12_J2_Best(X,Gmax)<Obj2_Max & J12_J2_Best(X,Gmax)>Obj2_Min
              Mio_2(X,1)=(Obj2_Max-J12_J2_Best(X,Gmax))/Obj2_Dif;
      end
  end
clear X
% Calculation of Normalized membership function as per equation 6.13
for X=1:K;
Mio_1_2(X,1)=Mio_1(X,1)+Mio_2(X,1);
M1=Mio_1_2;
end
clear X
% Set Mio_1_2_Sum to Zero(0)
Mio_1_2_Sum=0;
```

```
for X=1:K;
Mio_1_2_Sum=Mio_1_2(X,1)+Mio_1_2_Sum;
end
M2=Mio_1_2_Sum;
clear X
for X=1:K;
Mio(X,1)=Mio_1_2(X,1)/Mio_1_2_Sum;
end
M3=Mio;
% Identify the Compromise Individual
Max_Mio=max(Mio);
R=find(Mio==Max_Mio);
M4=R;
% Best Solution
J1_Best=J12_J1_Best(R,Gmax);
J2_Best=J12_J2_Best(R,Gmax);
% Best Solution Chromosomes order
Chr_Best_Number=R;
clear R

% Save Best Compromise Chromosomes objectives and number
save Chr_Best_Comp J1_Best J2_Best Chr_Best_Number Gmax

clear K Mio_1 Mio_2 Mio_1_2 Mio_1_2_Sum R X Obj1_Max Obj1_Min Obj2_Max
Obj2_Min Max_Mio
```

APPENDIX XV

The MATLAB Program Codes for Truncation Technique (TruncationR1.M). [Refer to section 6.6]

```
%***************************************************************************
% This sub-program apply the truncation method to reduce the number of
% individuals when they exceed the pre-identified External Pareto Set
% EOPS_P_P
%***************************************************************************
%---------------------------------------------------------------------------
% Measure the Population of the External Optimal Pareto Set (EOPS)
% number of individuals
 Nrow=length(EOPS_P_P(:,1));
%---------------------------------------------------------------------------
% Measure the number of objective values, There are two objectives
 Nobj=2;
 Ncol=Nobj;
% Identify A1 Matrix (Objective matrix)
 A1=EOPS_J12(:,1:2);
%---------------------------------------------------------------------------
% Create Zero Distance Matrix
% Normalize the Objective Values in matrix A1
dist=zeros(Nrow);
for i=1:Ncol
    dumvec(1:Nrow)=A1(1:Nrow,i);
    xminval=min(dumvec);
    xmaxval=max(dumvec);
    distmax=xmaxval-xminval;
    %disp(distmax);
    for j=1:Nrow
        A(j,i)=(A1(j,i)-xminval)/distmax;
    end
end
%---------------------------------------------------------------------------

%---------------------------------------------------------------------------
% Mesaure the distance between each individual and fill the Distance
% Matrix dist
for i=1:Nrow-1
    for j=i+1:Nrow
        Total=0.0;
        for k=1:Ncol
            Total=Total+((A(i,k)-A(j,k))^2);
        end
        dist(i,j)=sqrt(Total);
        dist(j,i)=dist(i,j);
    end
end

%---------------------------------------------------------------------------
% Clear Variable n & k
clear n k i j Total A A1 xmaxval xminval
clear xminval distmax xmaxval dumvec
```

```
%-----------------------------------------------------------------------------
%-----------------------------------------------------------------------------
% Replace the Zero in the dist Matrix with large dome number 1000
  for i=1:Nrow;
      dist(i,i)=100;
  end
clear i
%-----------------------------------------------------------------------------
% Ientify the minimum distace
distmin=100000.00;
    for i=1:Nrow-1
        for j=i+1:Nrow
            if dist(i,j)<distmin
                distmin=dist(i,j);
                imin=i;
                jmin=j;
            end
        end
    end
%-----------------------------------------------------------------------------
%Replace the Minimum distance with Dome number (100)
dist(imin,jmin)=100+dist(imin,jmin);
dist(jmin,imin)=100+dist(jmin,imin);

%-----------------------------------------------------------------------------
% Identify the distance to the next closet individuals nc_i
% to i & j individual
nc_i=min(dist(imin,:));
nc_j=min(dist(jmin,:));

%-----------------------------------------------------------------------------
% Compare nc_i & nc_j
if nc_i<nc_j;
   EOPS_P_P(imin,:)=[];
else
   EOPS_P_P(jmin,:)=[];
end
%-----------------------------------------------------------------------------
% Update the EOPS_P_P length
  LT=length(EOPS_P_P(:,1));
% If the required EOPS_P_P size is not met, repeat the above steps untial
% the required size is met
%-----------------------------------------------------------------------------

clear nc_i nc_j dist Nrow imin jmin distmin i j
```

APPENDIX XVI

The MATLAB Program Codes for Population Non-Dominate Chromosomes Identification. [Refer to section 6.8]

```
%**********************************************************************
% This Sub-Program will identify the None Dominant Chromosomes within
% the population
%**********************************************************************

%----------------------------------------------------------------------
% Set the matrix capturing the fitness of each chromosome (number of
% chromosomes dominate or dominated by the subject chromosome)
% equal to zero (0)
Pop_Df(N_Pop,1:5)=zeros;
% Set the matrix capturing the tag of each chromosome with regard to its
% relation with current chromosome (dominate or dominated by)
% IndxP(:,1:Size_Pop)=zeros;
Indx(:,1:N_Pop)=zeros;
%----------------------------------------------------------------------

%-----------------------------------------------------------------------
% Identify the None Dominate Chromosomes within the Population Matrix
% Loop to assign column 1,2 and 3; the St(i) refer to equation 6.8
% N_Pop is the population size
% J12 matrix contains the objectives J1 and J2 fitness values

for n=1:N_Pop-1
    for k=n+1:N_Pop
       if J12(n,1)<J12(k,1)& J12(n,2)<J12(k,2)
       % Assign the numbers of Chromosome n dominate Chromosome k
       Pop_Df(n,1)=Pop_Df(n,1)+1;
        % Assign the numbers of Chromosome k Chromosome Dominated by n
       Pop_Df(k,2)=Pop_Df(k,2)+1;
       % Captured the index of the dominated chromosomes
       Indx(n,k)=k;
       Indx(k,n)=0;
       elseif J12(n,1)<J12(k,1)& J12(n,2)==J12(k,2)
       % Assign the numbers of Chromosome n Dominate Chromosome k
       Pop_Df(n,1)=Pop_Df(n,1)+1;
        % Assign the numbers of Chromosome k Dominated by Chromosome n
       Pop_Df(k,2)=Pop_Df(k,2)+1;
       % Captured the index of the dominated chromosomes
       Indx(n,k)=k;
       Indx(k,n)=0;
       elseif J12(n,1)==J12(k,1)& J12(n,2)<J12(k,2)
       % Assign the numbers of Chromosome n Dominate Chromosome k
       Pop_Df(n,1)=Pop_Df(n,1)+1;
        % Assign the numbers of Chromosome k Dominated by Chromosome n
       Pop_Df(k,2)=Pop_Df(k,2)+1;
       % Captured the index of the dominated chromosomes
```

```
        Indx(n,k)=k;
        Indx(k,n)=0;
        elseif J12(k,1)<J12(n,1)& J12(k,2)<J12(n,2)
        % Assign the numbers of Chromosome k Dominate Chromosome n
        Pop_Df(k,1)=Pop_Df(n,1)+1;
         % Assign the numbers of Chromosome n Dominated by Chromosome k
        Pop_Df(n,2)=Pop_Df(k,2)+1;
        % Captured the index of the dominated chromosomes
        Indx(k,n)=n;
        Indx(n,k)=0;
        elseif J12(k,1)<J12(n,1)& J12(k,2)==J12(n,2)
        % Assign the numbers of Chromosomes k Dominate Chromosome n
        Pop_Df(k,1)=Pop_Df(n,1)+1;
         % Assign the numbers of Chromosome n Dominated by Chromosome k
        Pop_Df(n,2)=Pop_Df(k,2)+1;
        % Captured the index of the dominated chromosomes
        Indx(k,n)=n;
        Indx(n,k)=0;

        elseif J12(k,1)==J12(n,1)& J12(k,2)<=J12(n,2)
        % Assign the numbers of Chromosome k Dominate Chromosome n
        Pop_Df(k,1)=Pop_Df(n,1)+1;
         % Assign the numbers of Chromosomes n Dominated by Chromosome k
        Pop_Df(n,2)=Pop_Df(k,2)+1;
        % Captured the index of the dominated chromosomes
        Indx(k,n)=n;
        Indx(n,k)=0;
        else
        Pop_Df(n,3)=Pop_Df(n,3)+1;
        Pop_Df(k,3)=Pop_Df(k,3)+1;
        % Dome 0.1
        Indx(k,n)=0.1;
        Indx(n,k)=0.1;
        end
    end

end

% Clear Variable n & k
clear n k

%-------------------------------------------------------------------------
% Loop to assign column [4], the Rw (i) for Xj which Dominate Xi
% Refer to equation 6.9. Rw (i) is assigned to Pop_Df(i,4)

for n=1:N_Pop-1
    for k=n+1:N_Pop
       if (Indx(n,k)==0 & n~=k) & Indx(n,k)~=0.1 ;
       Pop_Df(n,4)=Pop_Df(k,1)+Pop_Df(n,4);
       elseif (Indx(k,n)==0 & n~=k) & Indx(k,n)~=0.1;
       Pop_Df(k,4)=Pop_Df(n,1)+Pop_Df(k,4);
       else
       end
    end

end
```

```
% Clear Variable n & k
clear n k
%-----------------------------------------------------------------------------

%-----------------------------------------------------------------------------
% Calculation of the Density D(i) refer to equation 6.10
% Measure the Population of the Pareto Set (number of individuals)
Nrow=length(Pop_Df(:,1));
% measure the number of objective values
Nobj=2;
Ncol=Nobj;
% Identify A1 Matrix (Objective matrix)
A1=J12(:,1:2);
%-----------------------------------------------------------------------------
% Create Zero Distance Matrix
% Normalize the Objective Values in matrix A
dist=zeros(Nrow);
for i=1:Ncol
    dumvec(1:Nrow)=A1(1:Nrow,i);
    xminval=min(dumvec);
    xmaxval=max(dumvec);
    distmax=xmaxval-xminval;
    %disp(distmax);
    for j=1:Nrow
        A(j,i)=(A1(j,i)-xminval)/distmax;
    end
end
%-----------------------------------------------------------------------------

%-----------------------------------------------------------------------------
% Measure the distance between each individual and fill the Distance
% Matrix dist
for i=1:Nrow-1
    for j=i+1:Nrow
        Total=0.0;
        for k=1:Ncol
            Total=Total+((A(i,k)-A(j,k))^2);
        end
        dist(i,j)=sqrt(Total);
        dist(j,i)=dist(i,j);
    end
end
% Clear Variable n & k
clear n k i j Total xminval distmax xmaxval dumvec Ncol A A1
%-----------------------------------------------------------------------------
% Sort the dist matrix in an increasing order
for i=1:Nrow;
    dist_sort(i,:)=sort(dist(i,:));
end
clear i
% Identify the Kth distance
  Kth=(N_Pop+Size_EOPS_P)^0.5;

% Round Kth value to the nearest integer
  Kth=round(Kth);
% Calculate the Density D(i) refer to equation 6.10
```

```
for i=1:Nrow;
    D(i)=1/(dist_sort(i,Kth)+2);
    % Calculate F(i)=R(i)+D(i) Fitness Values at column 5
    Pop_Df(i,5)=Pop_Df(i,4)+D(i);
end
clear dist dist_sort D Kth
%---------------------------------------------------------------------------
%---------------------------------------------------------------------------
% Sort the dist matrix in an increasing order
Pop_Df_Temp=Pop_Df;
%---------------------------------------------------------------------------

%---------------------------------------------------------------------------
% Insert the Chromosomes number
Pop_N=[1:N_Pop]';
Pop_Df_Temp=[Pop_Df_Temp Pop_N];
%---------------------------------------------------------------------------
% Select the Most None-Dominate Solution
% Sort the Population in ascending
Pop_Sort=sortrows(Pop_Df_Temp, [5]);
% Find Chromosome who is dominated (Cloume 5 <1)
for x=1:Nrow;
    B=Pop_Sort(x,5);
    if B<1;
    EOPS_T(x,:)=Pop_Sort(x,:);
    % Establish logic for the existence of EOPS_P_T
    EOPS_P_P_Log=0;
    else
    Pop_Sort_T(x,:)= Pop_Sort(x,:);
    % Establish logic for the existence of EOPS_P_T
    EOPS_P_P_Log=1;
    end
clear B
end
clear x B Nrow

%--------------------------------------------------------------------------
% Check if EOPS_P_T logic exist "EOPS_P_T_Log=1" at line 228
if EOPS_P_P_Log==1;
% Clear Pop_Sort_T from zero rows
  [row,clom]=find(Pop_Sort_T(:,5)~=0);
  % Clear Pop_Sort_T from zeros
  Pop_Sort_T_C=Pop_Sort_T(row,:);
end
% EOPS_T contains the non-dominated population chromosomes
%---------------------------------------------------------------------------

% Clear the Temporary Variables B &  BB
clear  BB k n B row colm Pop_Sort_T Pop_Temp Pop_Df EOPS_P_P_Log
clear Pop_Df_Temp Indx Pop_Sort
```

APPENDIX XVII

The MATLAB Program Codes for External Optimal Pareto Set (EOPS) Non-Dominate Chromosomes Identification. [Refer to section 6.8]

```
%**************************************************************************
% This Program is will Establish the Permanent EOPS_P. Then, it will
% identify the Dominated Chromosomes within the External Optimal Pareto Set
% (EOPS) and then Remove them
%**************************************************************************

%--------------------------------------------------------------------------
%------------------------ Establish EOPS_P---------------------------------

% In the first generation the external optimal pareto set population
% (EOPS_P) is = EOPS_T
if          Ge_N==1 & length(EOPS_T(:,1))>=1;
            EOPS_P=EOPS_T;
            clear n
% In all the generation except the first generation insert the Chromosomes
% number and add 400 to colum 6 to differentiate Temporary EOPS_P from
% Permanent one EOPS_T
elseif      Ge_N > 1
            L=length(EOPS_T(:,1));
            LL=length(EOPS_P(:,1));
            EOPS_P(:,6)=EOPS_P(:,6)+400;
            clear n
            % Insert the Chromosomes number;
            EOPS_P(LL+1:L+LL,:)=EOPS_T;
else
disp ('EOPS_P_P at the first loop iteration is Empty (Check
Non_Dominat_Chr_EOPS)')
% This means that EOPS_P is empty which required starting again with new
% population
end
%--------------------------------------------------------------------------

%--------------------------------------------------------------------------
% If EOPS_P is < N (archive) fill with dominated individual from Population
% with best fitness value
% Measure EOPS_P length
  m=length(EOPS_P(:,1));
  if m<Size_EOPS_P;
     EOPS_P(m+1:Size_EOPS_P,:)=Pop_Sort_T_C(1:Size_EOPS_P-m,:);
  else
  end
clear Pop_Sort_T_C m EOPS_T EOPS_P_Sort
%--------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Capture the chromosomes @ Generation#1
```

```
 if Ge_N==1
          % measure the length of EOPS_P @1st Generation
          k=length(EOPS_P(:,1));
          for n=1:k
          Cr_TM_EP_P(n,:)=Cr_TM_In_V(EOPS_P(n,6),:);
          Cr_TC1_EP_P(n,:)=Cr_TC1_In_V(EOPS_P(n,6),:);
          Cr_TC2_EP_P(n,:)=Cr_TC2_In_V(EOPS_P(n,6),:);
          Cr_TC3_EP_P(n,:)=Cr_TC3_In_V(EOPS_P(n,6),:);
          Cr_TCM_EP_P(n,:)=Cr_TCM_In_V(EOPS_P(n,6),:);
          Cr_TG_EP_P(n,:)=Cr_TG_In_V(EOPS_P(n,6),:);
          Cr_TD_EP_P(n,:)=Cr_TD_In_V(EOPS_P(n,6),:);
          Cr_VGTG_EP_P(n,:)=Cr_VGTG_In_V(EOPS_P(n,6),:);
          Cr_VSTG_EP_P(n,:)=Cr_VSTG_In_V(EOPS_P(n,6),:);
          Cr_VSyMC_EP_P(n,:)=Cr_VSyMC_In_V(EOPS_P(n,6),:);
          % Capture Objectives
          EOPS_J12_P(n,:)=J12(EOPS_P(n,6),:);
          end
 end
 clear n k

%--------------------------------------------------------------------------
% Capture the Previous Chromosomes @ Generation > #1
 if Ge_N>1

% Measure the length of EOPS_P > 1st Generation
k=length(EOPS_P(:,1));
 for n=1:k
          if EOPS_P(n,6)>400
          Cr_TM_EP_P(n,:)=Cr_TM_EP_P(EOPS_P(n,6)-400,:);
          Cr_TC1_EP_P(n,:)=Cr_TC1_EP_P(EOPS_P(n,6)-400,:);
          Cr_TC2_EP_P(n,:)=Cr_TC2_EP_P(EOPS_P(n,6)-400,:);
          Cr_TC3_EP_P(n,:)=Cr_TC3_EP_P(EOPS_P(n,6)-400,:);
          Cr_TCM_EP_P(n,:)=Cr_TCM_EP_P(EOPS_P(n,6)-400,:);
          Cr_TG_EP_P(n,:)=Cr_TG_EP_P(EOPS_P(n,6)-400,:);
          Cr_TD_EP_P(n,:)=Cr_TD_EP_P(EOPS_P(n,6)-400,:);
          Cr_VGTG_EP_P(n,:)=Cr_VGTG_EP_P(EOPS_P(n,6)-400,:);
          Cr_VSTG_EP_P(n,:)=Cr_VSTG_EP_P(EOPS_P(n,6)-400,:);
          Cr_VSyMC_EP_P(n,:)=Cr_VSyMC_EP_P(EOPS_P(n,6)-400,:);
          else
          % Capture the Current Chromosomes
          Cr_TM_EP_C(n,:)=Cr_TM(EOPS_P(n,6),:);
          Cr_TC1_EP_C(n,:)=Cr_TC1(EOPS_P(n,6),:);
          Cr_TC2_EP_C(n,:)=Cr_TC2(EOPS_P(n,6),:);
          Cr_TC3_EP_C(n,:)=Cr_TC3(EOPS_P(n,6),:);
          Cr_TCM_EP_C(n,:)=Cr_TCM(EOPS_P(n,6),:);
          Cr_TG_EP_C(n,:)=Cr_TG(EOPS_P(n,6),:);
          Cr_TD_EP_C(n,:)=Cr_TD(EOPS_P(n,6),:);
          Cr_VGTG_EP_C(n,:)=Cr_VGTG(EOPS_P(n,6),:);
          Cr_VSTG_EP_C(n,:)=Cr_VSTG(EOPS_P(n,6),:);
          Cr_VSyMC_EP_C(n,:)=Cr_VSyMC(EOPS_P(n,6),:);
          end
 end
 end

clear n k
clear Cr_TM Cr_TC1 Cr_TC2 Cr_TC3 Cr_TD Cr_TG
```

```
clear Cr_VGTG Cr_VSTG Cr_VSyMC
%---------------------------------------------------------------------------
% Clear the Chromosomes from the zero rows @ Generation > #1
if Ge_N>1
% The below sub-porgram will clear the generation chromosomes matrix from
% the zero rows assigned due to the above loop
Clear_Chr_Zeros
end
%---------------------------------------------------------------------------

%---------------------------------------------------------------------------
% Sort the EOPS chromosomes (Dominated & Non Dominated)
% Set Colume 1 & 5 of the External Pareto Matrix (EOPS_P)= 0
  EOPS_P(:,1:5)=zeros;

% Measures EOPS_P length Temporary
  Row_EOPS_P_T=length(EOPS_P(:,1));
  Col_EOPS_P_T=length(EOPS_P(1,:));

% Identify index matrix Indx_EOPS
  Indx_EOPS(Row_EOPS_P_T,Col_EOPS_P_T)=zeros;

%---------------------------------------------------------------------------
% Assign the EOPS objective EOPS_J12
if Ge_N==1
   EOPS_J12=EOPS_J12_P;
end
%---------------------------------------------------------------------------
if Ge_N>1
%Re-Set Counter K_Count
K_Count_1=0;
K_Count_2=0;
for n=1:Row_EOPS_P_T
       if EOPS_P(n,6)>400
       K_Count_1=1+K_Count_1 ;
       EOPS_P_PT(K_Count_1,:)=EOPS_P(n,:);
       else
       K_Count_2=1+K_Count_2 ;
       EOPS_P_CT(K_Count_2,:)=EOPS_P(n,:);
       end
end
%--------------------------------------------------------------------------

%---------------------------------------------------------------------------
% Establish Logic Log_EOPS_J12_TC for assigning the objective fitness value
% of the EOPS_P (EOPS_J12)
Log_EOPS_J12_TP=0;
% Measure the length
k=length(EOPS_J12_P(:,1));
h=length(EOPS_P_PT(:,1));
  for n=1:h
      for q=1:k
      if EOPS_P_PT(n,6)==EOPS_J12_P(q,35)
         EOPS_J12_TP(n,:)=EOPS_J12_P(q,:);
         Log_EOPS_J12_TP=1;
      end
```

```
      end
  end
clear k n h
%---------------------------------------------------------------------------

%---------------------------------------------------------------------------
% Establish Logic Log_EOPS_J12_TC for assigning the objective fitness value
% of the EOPS_P (EOPS_J12)
Log_EOPS_J12_TC=0;
% Measure the length
k=length(J12(:,1));
h=length(EOPS_P_CT(:,1));
  for n=1:h
      for q=1:k
      if EOPS_P_CT(n,6)==J12(q,35)
         EOPS_J12_TC(n,:)=J12(q,:);
         Log_EOPS_J12_TC=1;
      end
      end
  end
clear k n h
%---------------------------------------------------------------------------

%---------------------------------------------------------------------------
% Establish the objectives fitness values matrix (EOPS_J12) in light of the
% Log_EOPS_J12_TP & Log_EOPS_J12_TC logic
if Log_EOPS_J12_TP==1 & Log_EOPS_J12_TC==1
EOPS_J12=[EOPS_J12_TP;EOPS_J12_TC];
elseif Log_EOPS_J12_TP==1
EOPS_J12=[EOPS_J12_TP];
else
EOPS_J12=[EOPS_J12_TC];
end
end
clear EOPS_J12_P
clear EOPS_J12_TP EOPS_J12_TC EOPS_P_PT EOPS_P_CT
clear Log_EOPS_J12_TP Log_EOPS_J12_TC
%---------------------------------------------------------------------------

%---------------------------------------------------------------------------
% Identify the None Dominate Chromosomes within the EOPS_P St(i) refer
% to equation 6.8
% Row_EOPS_P_T is the EOPS_P size
% EOPS_J12 matrix contain the objective J1 and J2 fitness values of the
% EOPS_P chromosomes
Count=0;

for n=1:Row_EOPS_P_T-1;
    for k=n+1:Row_EOPS_P_T;
       if EOPS_J12(n,1)< EOPS_J12(k,1)& EOPS_J12(n,2)< EOPS_J12(k,2)
       EOPS_P(n,1)=EOPS_P(n,1)+1;
       EOPS_P(k,2)=EOPS_P(k,2)+1;
       Indx_EOPS(n,k)=k;
       Indx_EOPS(k,n)=0;
       elseif EOPS_J12(n,1)< EOPS_J12(k,1)& EOPS_J12(n,2)== EOPS_J12(k,2)
       EOPS_P(n,1)=EOPS_P(n,1)+1;
```

```
        EOPS_P(k,2)=EOPS_P(k,2)+1;
        Indx_EOPS(n,k)=k;
        Indx_EOPS(k,n)=0;
        elseif EOPS_J12(n,1)== EOPS_J12(k,1)& EOPS_J12(n,2)== EOPS_J12(k,2)
        EOPS_P(n,1)=EOPS_P(n,1)+1;
        EOPS_P(k,2)=EOPS_P(k,2)+1;
        Indx_EOPS(n,k)=k;
        Indx_EOPS(k,n)=0;
        elseif EOPS_J12(k,1)< EOPS_J12(n,1)& EOPS_J12(k,2)< EOPS_J12(n,2)
        EOPS_P(k,1)=EOPS_P(k,1)+1;
        EOPS_P(n,2)=EOPS_P(n,2)+1;
        Indx_EOPS(k,n)=n;
        Indx_EOPS(n,k)=0;
        elseif EOPS_J12(k,1)< EOPS_J12(n,1)& EOPS_J12(k,2)== EOPS_J12(n,2)
        EOPS_P(k,1)=EOPS_P(k,1)+1;
        EOPS_P(n,2)=EOPS_P(n,2)+1;
        Indx_EOPS(k,n)=n;
        Indx_EOPS(n,k)=0;
        elseif EOPS_J12(k,1)== EOPS_J12(n,1)& EOPS_J12(k,2)< EOPS_J12(n,2)
        EOPS_P(k,1)=EOPS_P(k,1)+1;
        EOPS_P(n,2)=EOPS_P(n,2)+1;
        Indx_EOPS(k,n)=n;
        Indx_EOPS(n,k)=0;
        else
        EOPS_P(n,3)=EOPS_P(n,3)+1;
        EOPS_P(k,3)=EOPS_P(k,3)+1;
        % Dome 0.1
        Indx_EOPS(k,n)=0.1;
        Indx_EOPS(n,k)=0.1;
        end
    end

end
%--------------------------------------------------------------------------
% Loop to assign column [4], the Row (i)  for Xj which Dominate Xi
% Refer to equation 6.9. Rw (i)  is assigned to EOPS_P(i,4)
for n=1:Row_EOPS_P_T-1;
    for k=n+1:Row_EOPS_P_T;
       if (Indx_EOPS(n,k)==0 & n~=k) & Indx_EOPS(n,k)~=0.1 ;
       EOPS_P(n,4)=EOPS_P(k,1)+EOPS_P(n,4);
       elseif (Indx_EOPS(k,n)==0 & n~=k) & Indx_EOPS(k,n)~=0.1;
       EOPS_P(k,4)=EOPS_P(n,1)+EOPS_P(k,4);
       else
       end
   end
end
% Clear Variable n & k
clear n k Indx_EOPS Row_EOPS_P_T
%--------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Measure the Population of the Pareto Set (number of individuals)
 Nrow=length(EOPS_P(:,1));
% Measure the number of objective values
 Nobj=2;
 Ncol=Nobj;
```

```
% Identify A1 Matrix (Objective matrix)
 A1=EOPS_J12(:,1:2);
%--------------------------------------------------------------------------
% Create Zero Distance Matrix
% Normalize the Objective Values in matrix A
dist=zeros(Nrow);
for i=1:Ncol
    dumvec(1:Nrow)=A1(1:Nrow,i);
    xminval=min(dumvec);
    xmaxval=max(dumvec);
    distmax=xmaxval-xminval;
    %disp(distmax);
    for j=1:Nrow
        A(j,i)=(A1(j,i)-xminval)/distmax;
    end
end
%--------------------------------------------------------------------------

%--------------------------------------------------------------------------
% Measure the distance between each individual and fill the Distance
% Matrix dist
for i=1:Nrow-1
    for j=i+1:Nrow
        Total=0.0;
        for k=1:Ncol
            Total=Total+((A(i,k)-A(j,k))^2);
        end
        dist(i,j)=sqrt(Total);
        dist(j,i)=dist(i,j);
    end
end

%--------------------------------------------------------------------------
% Clear Variables n & k
clear n k i j Total
clear xminval distmax xmaxval dumvec Ncol A A1

% Sort the dist matrix in an increasing order
for i=1:Nrow;
    dist_sort(i,:)=sort(dist(i,:));
end
clear i
% Identify the Kth distance
 Kth=(N_Pop+Size_EOPS_P)^0.5;
% Round Kth value to the nearest integer
  Kth=round(Kth);
% Calculate the Density D(i)
for i=1:Nrow;
    D(i)=1/(dist_sort(i,Kth)+2);
    % Calculate F(i)=R(i)+D(i) Fitness Values
    EOPS_P(i,5)=EOPS_P(i,4)+D(i);
end
clear i
%--------------------------------------------------------------------------

% Sort the dist matrix in an increasing order
```

```
EOPS_P_Temp=EOPS_P;
% Sort the Population in ascending
  EOPS_P_Sort=sortrows(EOPS_P_Temp, [5]);

% Find Chromosome who is dominated (Column 5 <1)
for x=1:Nrow;
    if EOPS_P_Sort(x,5)<1;
    % Establish Perminant External Pareto Set (best of best)
    EOPS_P_P(x,:)=EOPS_P_Sort(x,:);
    % Establish logic for the existence of EOPS_P_T
    EOPS_P_T_Log=0;
    else
    % Establish Temporary External Pareto Set (best of best)
    EOPS_P_T(x,:)=EOPS_P_Sort(x,:);
    % Establish logic for the existence of EOPS_P_T
    EOPS_P_T_Log=1;
    end
end
clear x B
%---------------------------------------------------------------------------
% Check if EOPS_P_T logic exist "EOPS_P_T_Log=1" at line 228
if EOPS_P_T_Log==1;
% Clean EOPS_P_P_T from zero rows
  m=length(EOPS_P_T(:,1));
  if m>0;
  [row,clom]=find(EOPS_P_T(:,5)>0);
  EOPS_P_T=EOPS_P_T(row,:);
  end
end
clear m x k  row clom EOPS_P_T_Log

%---------------------------------------------------------------------------
% Establish EOPS_P_P for next Program
% Measuer EOPS_P_P length
  m=length(EOPS_P_P(:,1));

% if the size of Nondominated Permanent EOPS_P_P_B is < Archive size
  if m<Size_EOPS_P;
     EOPS_P_P(m+1:Size_EOPS_P,:)=EOPS_P_T(1:Size_EOPS_P-m,:);
     % Assign the EOPS_P_P length value < Size_EOPS_P
     LT=Size_EOPS_P-1;
  else
  % Measure the EOPS_P_P length for Truncation Sub-Program
    LT=length (EOPS_P_P(:,1));
  end
clear m
clear  EOPS_P_T EOPS_P
clear  BB k n B row colm L LL Row_EOPS_P_T Col_EOPS_P_T EOPS_P_T
clear  A A1 dist dist_sort D Kth Nrow
clear  n k i j Total xminval distmax xmaxval dumvec Ncol
```

REFERENCES

[1] "Saudi Arabia historical peak demand", http://ecra.gov.sa/peak_load.aspx#.VOg-5I05BKA, accessed March 2016.

[2] "Saudi Electrical Company 2014 annual report", *https://www.se.com.sa/en-us/Lists/AnnualReports/Attachments/14/AnnualReport2014En.pdf* , accessed March 2016.

[3] " Electricity and Co-Generation Requlatory Authority 2014 annual report", *http://www.ecra.gov.sa/documents/Annual%20reports/ecra_ebook2014.pdf , accessed March 2016.*

[4] *H. Habibollahzadeh, G.X. Luo and A. Semlyen,* "Hydrothermal Optimal Power Flow Based On A Combined Linear and Nonlinear Programming Methodology", *IEEE Transactions on Power Systems, Vol. 4, No. 2, May 1989, pp.530-537.*

[5] *D. Pudjianto, S. Ahmed and G. Strbac,* "Allocation of VAr support using LP and NLP based optimal power flows", *IEEE Proceedings – Generation, Transmission and Distribution, Vol. 149, Issue. 4, 2002, pp. 377-383.*

[6] *James A. Momoh, M. E. El-Hawary and Ramababu Adapa,* "A Review of Selected Optimal Power Ftow Literature to 1993 Part I: NonLinear and Quadratic Programming Approaches", *IEEE Transactions on Power Systems, Vol. 14, No. 1, February 1999, pp-96-104.*

[7] *Anthony M. Giacomoni and Bruce F. Wollenberg,*"Linear Programming Optimal Power Flow Utilizing a Trust Region Method", *North American Power Symposium (NAPS), 2010, pp. 1-6.*

[8] *J. Nanda, D.P. Kothari and S.C. Srivastava,* "New optimal power-dispatch algorithm using Fletcher's quadratic programming method", *IEE Proceedings C – Generation, Transmission and Distribution, Vol. 136, Issue. 3, 1989, pp.153-161.*

[9] *J. A. Momoh and J.Z. Zhu,* " Improved Interior Point Method for OPF Problems", *IEEE Transactions on Power Systems, Vol. 14, No. 3, August 1999, pp.1114-1120.*

[10] *B. Venkateswara Rao, G. V. Nagesh Kumar, R. V. S. LakshmiKumari and N. G. S. Raju,* "Optimization of A Power System With Interior Point Method", *2011 International Conference on Power and Energy Systems (ICPS), 2011, pp.1-6.*

[11] *Juan M. Ramirez, Xiomara Gonzalez and Miguel Nedina*, "Reactive power handling by a multi-objective formulation", *IEEE, North American Power Symposium (NAPS), August 4-6, 2011, pp. 1–5.*

[12] *Muhammad Tami Al-Hajri and M. A. Abido,* "Multiobjective Optimal Power Flow Using Improved Strength Pareto Evolutionary Algorithm (SPEA2)", *IEEE, 11th International Conference of Intelligent System Design and Application (ISDA), Cordoba, Spain, November 22-24, 2011, pp. 1–7.*

[13] *Tejaswini Sharma and Laxmi Srivastava*, "Evolutionary computing techniques for optimal reactive power dispatch: an overview and key issue", *IEEE, 4th International Conference on Communication and Network Technologies, 2014, pp. 507–511.*

[14] *A.Y. Abdelaziz, R.A. Osama, S.M. El-Khodary,* "Reconfiguration of Distribution Systems for Loss Reduction using the Hyper-Cube Ant Colony Optimisation Algorithm", *Generation, Transmission & Distribution, IET (Volume:6 , Issue:2), February 2012, pp. 176 – 187.*

[15] *Yingjun Ma and Xueyuan Cui,* "Solving the Fuel Transportation Problem Based on the Improved Genetic Algorithm", *2014 10th International Conference on Natural Computation, 2014, pp.584-588.*

[16] *T. R. Alencar, J. Gramulia Jr., R. F. Otobe Jr. and P. T. L. Asano,* "Decision Support System based on genetic algorithms for optimizing the Operation Planning of Hydrothermal Power Systems", *2015 5th International Youth Conference on Energy (IYCE), 2015, pp.1-6.*

[17] *Ramakanta Mohanthy, Venkatshwarlu Naik and Azmath Mubeen,* "Software Reliability Prediction Using Ant Colony Optimization Technique", *2014 Fourth International Conference on Communication Systems and Network Technologies, 204, pp.496-500.*

[18] *Sanchay Dewan, Shreyansh Bajaj and Shantanu Prakash,* "Using Ants Colony Algorithm for Improved Segmentation for number plate recognition", *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIA), 2015, pp.1117-1119.*

[19] *Amr Rekaby, A. A. Youssif and A. Sharaf Eldin,* "Introducing Adaptive Artificial Bee Colony Algorithm and using it in Solving Traveling Salesman Problem", *Science and Information Conference (SAI), 2013, pp.502-506.*

[20] *Jing Xue, Liutao Li, SaiSai Zhao and Litao Jiao,* "A Study of Task Scheduling Based On Differential Evolution Algorithm in Cloud Computing", *2014 Sixth International Conference on Computational Intelligence and Communication Networks, 2014, pp.637-640.*

[21] *J. A. Malagoli, J. R. Camacho and M. V. Ferreira da Luz,* "Design of Three-Phase Induction Machine Using Differential Evolution Algorithm", *IEEE Latin America Transactions, Vol. 13, No. 7, July 2015, pp.2202-2208.*

[22] *Sandra Scott-Hayward and Emiliano Garcia-Palacios,* "Channel Time Allocation PSO for Gigabit Multimedia Wireless Networks", *IEEE Transactions on Multimedia, Vol. 16, No. 3, April 2014, pp. 828-836.*

[23] *M. A. Abido, "Intelligent Control Course Notes", King Fahad University of Petroleum & Minerals, 2007.*

[24] *H.J. Jayatheertha, Yadagiri J,* "Reactive Power Optimization Using Repeated Genetic Algorithm and Data Prediction", *IEEE Advanced Computing Technologies (ICACT) 5th international Conference, 2013, pp. 1-4.*

[25] *S.N. Deepa1 and J. Rizwana,* "Power System Stability by Reducing Power Losses using Optimization Techniques", *IEEE Computational Intelligence and Computing Research (ICCIC) Conference, 2013, pp. 1-4.*

[26] *Zhao-Yulin, Yu-Qian and Zhao-Chunguang,* "Distribution Network Reactive Power Optimization Based on Ant Colony Optimization and Differential Evolution Algorithm", *IEEE Conference, 2010, pp. 472–476.*

[27] *Attia A. El-Fergany and Almoataz Y. Abdelaziz*, "Efficient heuristic-based approach for multi-objective capacitor allocation in radial distribution networks", *IET Generation, Transmission & Distribution, 2014, vol. 8, Issue.1, pp. 70–80.*

[28] *Gian Luca Storti, Francesca Possemato, Maurizio Paschero, Antonello Rizzi, Fabio Massimo Frattale Mascioli*, "Optimal Distribution Feeders configuration for active power losses minimization by genetic algorithms", *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS) Conference, 2013, pp. 407–412.*

[29] *Ziari, G. Ledwich and A. Ghosh*, "Optimal voltage support mechanism in distribution networks", *IET Generation, Transmission & Distribution, 2011, vol. 5, Issue.1, pp. 127–135.*

[30] *M. A. N. Guimaraes, C. A. Castro and R. Romero*, "Distribution systems operation optimization through reconfiguration and capacitor allocation by a dedicated genetic algorithm", *IET Generation, Transmission & Distribution, 2010, vol. 4, Iss.11, pp. 1213–1222.*

[31] *D. Godwin Immanuel and C. Chritober Asir Rajan*, "An genetic algorithm approach for reactive power control problem", *IEEE International Conference on Circuits, Power and Computing Technologies (ICCPCT), Nagercoil, 2013, pp. 74–78.*

[32] *Haken K. Jevne, Pauline C. Haddow and Alexei A. Gaivoronski*, "Evolving constrained mean-VaR efficient Frontiers", WCCI IEEE World Congress on Computational Intelligence, Brisbane, Australia, June 10-15, 2012.

[33] *Attia A. El-Fergany*, "Optimal capacitor allocations using evolutionary algorithms", *IET Generation, Transmission & Distribution, 2013, vol. 7, Iss.6, pp. 593–601.*

[34] *C. Y. Chung, C. H. Liang, K. P. Wong and X. Z. Duan,* "Hybrid algorithm of differential evolution and evolutionary programming for optimal reactive power flow", *IET Generation, Transmission & Distribution, 2010, vol. 4, Iss.1, pp. 84–93.*

[35] *Y. S. Brar, J. S. Dhillon and D. P. Kothari*, "Multiobjective load dispatch based on genetic-fuzzy technique", *IEEE Power System Conference and Exposition (PSCE), Atlanta, GA, 2006, pp. 931–937.*

[36] *Zhang Xuexia, Chen Weirong and P. N. Suganthan, "*Optimal multi-objective reactive power dispatch considering static voltage stability based on dynamic multi-group self-adaptive differential evolution algorithm", *IEEE International Conference on Intelligent system design and engineering application, 2012, , pp. 1448–1456.*

[37] *S. R. Spea, A. A. Abou El Ela and M. A. Abido*, "Multi-objective differential evolution algorithm for environmental-economic power dispatch", *IEEE International Energy Conference, Manama, 2010,pp. 841–846.*

[38] *M. Varadarajan and K. S. Swarup*, "Solving multi-Objective optimal power flow using differential evolution", *IET Generation, Transmission & Distribution, 2008, vol. 2, No. 5, pp. 720–730.*

[39] *Yujiao Zeng and Yanguang Sun, "* Comparison of multiobjective particle swarm optimization and evolutionary algorithm for optimal reactive power dispatch problem", *IEEE Congress on Evolutionary Computation (CEC), Beijing, China, July 6-11, 2014, pp. 258–265.*

[40] *S. Surender Reddy, A. R. Abhyankar and P. R. Bijwe* , "Market cleaning of joint energy and reactive power using multi objective optimization considering voltage dependent load methods", *IEEE Power and Energy Society General Meeting, San Diego, CA, 2011, pp. 1–8.*

[41] *Chefai Dhifaoui, Yosra Welhazi, Tawfik Guesmi and Hasan H. Abdallah*, "Application of multi-objective PSO algorithm for static reactive power dispatch", *IEEE, 5th International Renewable Energy Congress (IREC), Hammamet, Tunisia, March 25-27, 2014.*

[42] *T. Niknam, M. R. Narimani, J. Aghaei and R. Azizipanah-Abarghooee*, "Improved particle swarm optimization for multi-objective optimal power flow considering the cost, loss, emission and voltage stability index", *IET Generation, Transmission & Distribution, 2012, vol. 6, Issue.6, pp. 515–527.*

[43] *N. Nimpitiwan and C. Chaiyabut,* "Centralized Control of System Voltage/Reactive Power Using Genetic Algorithm", *Intelligent Systems Applications to Power Systems Conference, 2007. ISAP 2007. International Conference on Toki Messe, Niigata, 5-8 Nov. 2007, pp. 1 – 6.*

[44] *C. M. Huang, S. J. Chen, Y. C. Huang and H. T. Yang*, "Comparative study of evolutionary computation methods for active-reactive power dispatch", *IET Generation, Transmission & Distribution, 2012, vol. 6, Issue.7, pp. 636–645.*

[45] *Attia A. El-Fergany and Mahdi El-Arini*, "Meta-heuristic algorithms-based real power loss minimization including line thermal overloading constraints", *IET Generation, Transmission & Distribution, 2013, vol. 7, Issue.6, pp. 613–619.*

[46] *Benemar Alencar de Souza and Angelo Marcio Formiga de Almeida*, "Multiobjective optimization and fuzzy logic applied to planning of the volt/var problem in distribution systems", *IEEE Transactions on Power Systems, 2010, vol. 25, No.3, pp. 1274–1281.*

[47] *Furong Li, James D. Pilgrim, Chavansingh Dabeedin, Ahmad Chebbo, R. K. Aggarwal,* "Genetic Algorithms for Optimal Reactive Power Compensation on the National Grid System" *, IEEE Transactions on Power Systems, Volume:20,Issue:1, Feb. 2005, pp. 493 – 500.*

[48] *N. Gupta, A. Swarnkar, K.R. Niazi, R.C. Bansal, "*Multi-Objective Reconfiguration of Distribution Systems using Adaptive Genetic Algorithm in Fuzzy Framework", *Generation, Transmission & Distribution, IET (Volume:4, Issue: 12), 07 October 2010, pp. 1288 – 1298.*

[49] *Kahtan Mustafa, Kamal Garg and Rajkumar Swaminathan,* "Saudi Aramco Experience – Islanding Scheme for Manifa Central Processing Facility", *Petroleum and Chemical Industry Conference Europe, Amsterdam, 2014, pp. 1-8.*

[50] *Musaab M. Al-Mulla, Mohammed Akhil Fazil and Nicholas C. Seeley,* "Energy Management Systems for Islanded Industrial Facilities", *IEEE GCC Conference and Exhibition, Kuwait, 2009, pp. 1-6.*

[51] *S. Anantasate, C. Chokpanyasuwan, W. Pattaraprakorn, P. Bhasaputra* "Electrical Major Upgrade with Power Management System Implementation for Old Operating Oil & Gas Production Plant", *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on (Volume:01), Pattaya, Chonburi, 6-9 May 2009, pp. 182 – 185.*

[52] G*lenn W. Stagg and Ahmed H. El-Abiad*, "Computer Methods in Power System Analysis", *McGraw-Hill, 1968.*

[53] *Hadi Saadat*, "Power System Analysis", *McGraw-Hill, 2004, 2nd edition*.

[54] *Lynn Powell*, " Power System Load Flow Analysis", *McGraw-Hill, 2005.*

[55] *Y. Hongwen and T. Junna, "*Reactive Power Optimization Research of Power System Considered the Generation Transmission and Distribution*", Industrial Technology Conference, Chengdu, 2008.*

[56] *P. Kessel, H. Glavitsch, "*Estimating the Voltage Stability of a Power System*", IEEE Power Industry Computer Application Conference, San Francisco,USA, May 6-10, 1985, pp. 346-354.*

[57] *C. Belhadj and, M. A. Abido, "*An optimized fast voltage stability indicator*", IEEE Budapest Power Tech '99 Conference, Budapest, Hungary, 1999.*

[58] *K. Vaisakh and P. Kanta Rao,* "Optimum Reactive Power Dispatch Using Differential Evolution for Improvement of Voltage Stability", *Power System Technology and IEEE Power India Conference, 2008. POWERCON 2008. Joint International Conference on New Delhi, 12-15 Oct. 2008, pp. 1 – 9.*

[59] *John R. Koza*, "Genetic Programming", *1992, Massachusetts Institute of Technology*.

[60] *S. Sumathi, T. Hamsapriya and P. surekha, "*Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab*", Springer, 2008.*

[61] *S. N. Sivanandam, S. N. Deepa, "*Introduction to Genetic Algorithms*", Springer, 2008.*

[62] *Joshua Samuel Raj and Riya Mary Thomas* "Genetic Based Scheduling in Grid Systems: A Survey", *Computer Communication and Informatics (ICCCI), 2013 International Conference on Coimbatore, 4-6 Jan. 2013, pp. 1-4.*

[63] *M. T. Al-Hajri and M. Abido* "Assessment of Genetic Algorithm Selection, Crossover and Mutation Techniques in Reactive Power Optimization", *IEEE, CEC 2009, May 8-21, 2009, pp.1005-1011.*

[64] *F. Herrera, M. Lozano and J. L. Verdegay*, "Tackling Real-Coded Genetic Algorithm: Operators and Tools for Behavioural Analysis", *August 1998, Volume 12, Issue 4, pp 265-319*.

[65] *Swagatam Das and Ponnuthurai Nagaratnam Suganthan* "Differential Evolution: A Survey of the State-of-the-Art", *IEEE Transactions on Evolutionary Computation, Volume 15, Issue.1, 22 February 2011, pp. 4 – 31.*

[66] *C. H. Shin-Ju, Y. C. Huang and S. P. Yang*, "Optimal Active-Reactive Power Dispatch Using an Enhanced Differential Evolution Algorithm", *2011, 6th IEEE Conference on Industrial Electronic and Applications, pp.1869-1874.*

[67] *M. A. Abido,* " Multiobjective Evolutionary Algorithm for Electric Power Dispatch Problem*", IEEE Transaction on evolutionary computation, Vol. 10, No.3, June 2006, pp 315-329*.

[68] *E. Zitzler and L. Thiele*, "An Evolutionary Algorithm for Multiobjective optimization: The Strength Pareto Approach", *Swiss Federal Institute of Technology, TIK-Report, No. 43, 1998.*

[69] *M. A. Abido*, " Multiobjective Optimal VAR Dispatch Using Strength Pareto Evolutionary Algorithm", *2006 IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, July 16-21, 2006.*

[70] *E. Zitzler, M. Laumanns, and L. Thiele,* "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", *Swiss Federal Institute of Technology(ETH), Zrich, Switzerland, Technical report TIK-Report, No. 103, May 2001.*

[71] *Nateri K. Madavan,* "Multiobjective Optimization Using a Pareto Differential Evolution Approach", *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress, Volume 2, Honolulu, HI, 2002, pp. 1145 – 1150.*

[72] Kit Po Wong and ZhaoYang Dong, "Differential Evolution, an Alternative Approach to Evolutionary Algorithm" , *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems,Arlington, VA, Nov. 6-10, 2005, pp. 73 – 83.*

[73] *Chao-Ming Huang, Shin-Ju Chen and Sung-Pei Yang,* "Optimal Active-Reactive Power Dispatch Using an Enhanced Differential Evolution Algorithm", *2011 6th IEEE Conference on Industrial Electronics and Applications (ICIEA), Beijing, China, 21-23 June 2011, pp. 1869 – 1874.*

PUBLICATIONS

[1] *Muhammad T. Al-Hajri, M. A. Abido and M. K. Darwish,* "Power Optimization for a Hydrocarbon Industrial Plant Using Genetic Algorithm", *IEEE University Power Engineering Conference (UPEC), Cluj-Napoca, Romania, 2-5 September 2014.*

[2] *Muhammad T. Al-Hajri, M. A. Abido and M. K. Darwish,* "Assessment of MVAR Injection in Power Optimization for a Hydrocarbon Industrial Plant Using Genetic Algorithm", *International Academy, Research and Industry Association (IARIA), Computation World Conference, March 22-27, 2015, Nice, France.*

[3] *Muhammad T. Al-Hajri, M. A. Abido and M. K. Darwish,* "Assessment of Genetic Algorithm Selection, Crossover and Mutation Techniques in Power Loss Optimization for a Hydrocarbon Facility", *IEEE University Power Engineering Conference (UPEC), Stoke-on-Trent, UK, 1-4 September 2015.*

[4] *Muhammad T. Al-Hajri, M. A. Abido and M. K. Darwish,* "Single and Multi-Objective Techno-Economic Assessment of Real Life Hydrocarbon Facility Electrical System Loss and Power Factor Using Improved Strength Pareto and Differential Evolution", *IET Generation, Transmission & Distribution Journal, Novamber 2015, under review.*

[5] *Muhammad T. Al-Hajri, M. A. Abido and M. K. Darwish,* "Multiobjective Power Loss Optimization Versus System Stability Assessment for Hydrocarbon Industrial Plant Using Differential Evolution Algorithm", *International Academy, Research and Industry Association (IARIA), Future Computing conference, March 20-24, 2016, Rome, Italy.*