A NOVEL MODEL FOR OPTIMIZING REACTIVE POWER IN POWER **SYSTEMS USING GENETIC ALGORITHM**

J.A.D.C.A.Jayakody
Department of Computer Systems Engineering,
Sri Lanka Institute of Information Technology,
Malabe, Sri Lanka.
Anuradha.j@sliit.lk

Edirisinghe E.A.G.A.
Department of Computer Systems Engineering,
Sri Lanka Institute of Information Technology,
Malabe, Sri Lanka.
Anu.edirisinghe77@gmail.com

G.A.D.D.Ganepola
Department of Electrical and Electronic Engineering,
Sri Lanka Institute of Information Technology,
Malabe, Sri Lanka.
Dinushiganepola93@gmail.com

***Abstract*— Integration of distributed generators into power systems is enhanced immensely due to the high demand for energy consumption which causes huge problems in power system operations. One of the adverse effects is reactive power variation which leads to insecure and unbalanced operation of the power system. Shunt capacitors and Static VAR compensators are some existing reactive power compensation equipment to regulate reactive power in power systems. Instead of using traditional approaches, a novel reactive power optimizing method will be introduced to ensure the safe and reliable operation of the system. The key objectives are to reduce the active power loss and enhance the power quality of the system while maintaining the voltage profile within the standard limit. To achieve the goals, a Genetic Algorithm based solution will be proposed for the IEEE-39 bus test system. In this paper, a novel reactive power optimization model is simulated using a standard genetic algorithm-based test system. The final simulated outcomes will illustrate that the proposed algorithm is attainable and effective by enhancing the quality of power systems.**

Keywords- Reactive Power, Genetic Algorithm, Power Systems, IEEE 39 Bus System

I. INTRODUCTION

Over the last few decades, power industry-related issues were addressed specifically with the flagitious increment of power crisis in some countries in the world. Among the diverse set of considerations, reactive power optimization is one of the most frequently considered major problems in power systems [1]. The key intention is to keep the voltage profile within the acceptable limits by reducing the active power loss of the transmission line under distinct operating conditions [2]. Multiple approaches were available in the literature regarding reactive power optimization like gradient-based search algorithms and mathematical programming schemes. Heuristics methods were utilized for optimizing reactive power but due to some unavoidable issues such as the complexity of the algorithm and the apprehensive convergence properties, they cannot be able to recognize the global optimum [5]. Due to the ability to mitigate the issues of the conventional method, an evolutionary algorithm was deployed to optimize the reactive power in power systems [3]. Linear programming, non-linear programming, and the interior point method are some techniques used in previous decades. Due to some drawbacks, those methods were also eliminated, and currently, most of the research focused on the genetic algorithm-related solutions generation method like multi-objective optimization which has been used in optimizing reactive power using genetic algorithm [4]. The benefits of using a genetic algorithm are it can give the nearest optimal solution regardless of the initial values and the convergence is fastest when comparing other available methods. So, in this paper, we are focusing on optimizing reactive power on any power system with the approach of a genetic algorithm. Furthermore, we have used the Multi-objective Genetic Algorithm to optimize the Reactive Power.

II. REACTIVE POWER OPTIMIZATION PROBLEM

A. Introduction

The key aims of reactive power optimization are to reduce the active power loss of the whole system and to maintain the voltage profile with minimum fluctuations by balancing the value in between the standard limit in order to intensify the power quality. The core reasons for active power loss are the current flowing through transformers and power lines. With the increment of the active power loss, system power generation cost will be increased and at the same time it will decrement the power factor of the system. Due to all these valid reasons, addressing the reactive p optimization is one of the most crucial functions in betafunctions power system issues.

The formula for active power loss can be expressed as below.

$$P_{loss} = \sum_{x=m,n}^{x\in N_E} G_{mn} (U_m^2 + U_n^2 - 2\, U_m\, U_n\, \cos\beta_{mn}) \quad (1)$$

P_{loss} – Active Power Loss
N_E – Set of numbers of network branches
G_{mn} – Conductance of the line mn
U_m – Voltage of bus m
U_n – Voltage of bus n
β_{mn} – Voltage angle between bus m and n

Equality Constraints

Equality constraints are for balancing the active power and reactive power in the power system. These constraints are satisfied by running the power flow diagram of IEEE39 bus system.

$$P_{Gm} - P_{Dm} = U_m \sum_{n=1} U_n\ (G_{mn}\ \cos\beta_{mn} + B_{mn}\ \sin\beta_{mn}) \qquad m = 1,2 \dots . N_B - 1 \quad (2)$$

$$Q_{Gm} - Q_{Dm} = U_m \sum_{n=1} U_n\ (G_{mn}\ \sin\beta_{mn} - B_{mn}\ \cos\beta_{mn})$$

$m = 1,2 \dots N_{PQ}$ (3)

Where,
P_{Gm} – The injected active power at bus m
P_{Dm} – The demanded active power at bus m
Q_{Gm} – The injected active power at bus m
Q_{Dm} – The demanded active power at bus m
B_{mn} – The transfer susceptance between bus m and n
N_B – Set of numbers of total buses
N_{PQ} – Set of number of PQ buses

Inequality Constraints

Inequality Constraints address the system operating constraints. The main control variables which to formulate the limits are generator terminal bus voltages, transformers tap, setting, and reactive power generated by the capacitor bank. The state variables are the active power generation at the slack bus, load bus voltages, reactive power generation, and line, flow. Below shows the formulas of inequality constraints.

1. **Voltage constraints**

$$U_m^{min} \leq U_m \leq U_m^{max} \qquad ;\ m \in N_B \quad (4)$$

U_m = Generator voltage at bus m
N_B = Set of numbers of total buses

2. **Transformer tap-setting limit**

$$T_K^{min} \leq T_K \leq T_K^{max} \qquad ;\ K \in N_T \quad (5)$$

T_K = Transformer Tap Setting
N_T = Set of numbers of transformer branches

3. **Generator reactive power capability limit**

$$Q_{Gm}^{min} \leq Q_{Gm} \leq Q_{Gm}^{max} \qquad ;\ m \in N_G \quad (6)$$

Q_G = Generator Reactive Power
N_G = Set of numbers of buses combined to bus m

4. **Capacitive reactive power capability limit**

$$Q_{Cm}^{min} \leq Q_{Cm} \leq Q_{Cm}^{max} \qquad ;\ m \in N_C \quad (7)$$

Q_C = Shunt capacitor / inductor
N_C = Set of numbers of possible reactive power source installation buses.

III. Implementation Of Geneteic Algorithm

A. *Overview of Genetic Algorithm*

Genetic Algorithm is a form of optimization algorithm, which means it is used to identify the best solution to a computer issue that maximized or minimized a certain function. Because they mimic the biological processes of reproduction and natural selection to solve for the "fittest" answers, genetic algorithms constitute one area of evolutionary computing research [6]. There is some randomness in genetic algorithms, as there is in evolution, but the amount of randomness and control may be adjusted using this optimization approach [7]. They are more powerful and efficient than random search and exhaustive search algorithms, but they need no additional knowledge about the presented issue. This property enables them to identify solutions to problems that conventional optimization techniques are unable to address because of a lack of continuity, derivatives, linearity, or other properties. A search heuristic based on Darwin's idea of natural selection isknown as a genetic algorithm. It is possible to predict known solutions and mimic evolutionary behavior in complicated systems using genetic algorithms.

B. *Structure of The genetic Algorithm*

It was motivated by Darwin's idea of evolution that Goldberg devised the genetic algorithm. "The strongest species that survives" is what it states. Reproduction, crossover, and mutation all worked together to keep this algorithm intact. The genetic algorithm is broken down into the following eight (8) phases.

Figure 1 illustrates the flow chart of genetic algorithm.

The all steps which are shown in the following flow chart are described in the following section.

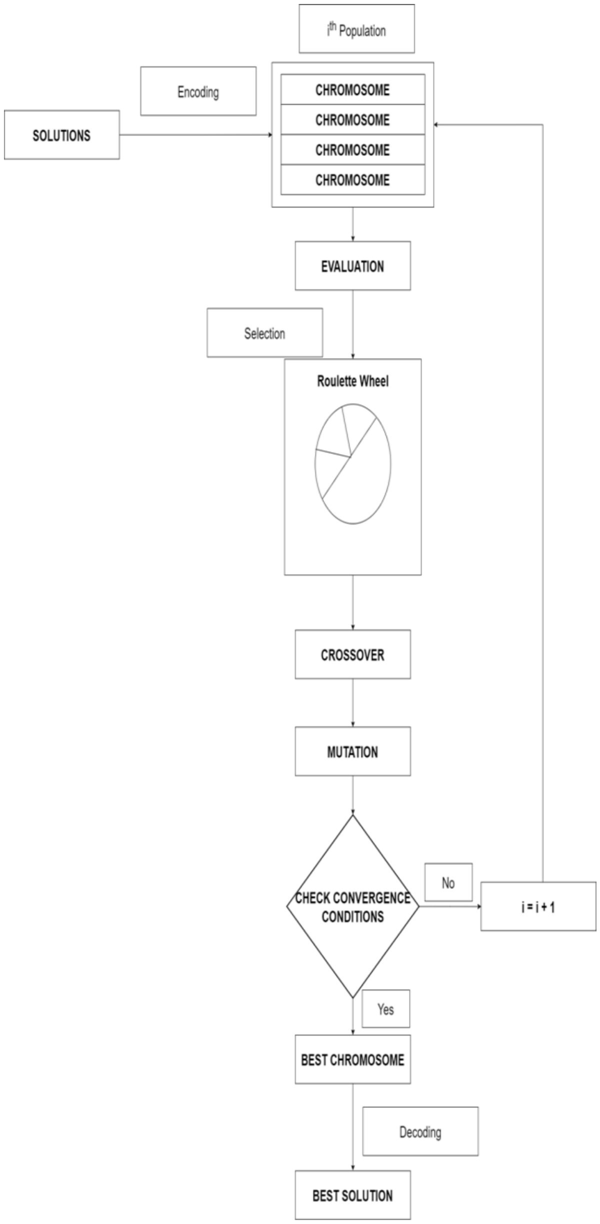

Figure 1. Flow chart of Genetic Algorithm

1. The number of chromosomes, generation, mutation rate, and crossover rate must be determined before the crossover rate can be calculated.
2. Randomly generate the population's chromosome-chromosome number and the gene's chromosome-chromosome initialization value.
3. Process steps 4-7 until the number of generations is met.
4. Evaluation of chromosomal fitness via objective function calculation.
5. Choosing the right set of chromosomes.
6. Crossover.
7. Mutation
8. Solution (Best Chromosomes).

Notion of Natural Selection

The process of natural selection begins with the selection of fittest individuals from a population. They create kids who inherit the qualities of the parents and will be added to the following generation. If parents have superior fitness, their kids will be better than parents and have a better chance of surviving. This process carries on iterating and towards the end, a generation with the fittest people will be discovered.

This idea may be employed for a search issue. We evaluate a collection of solutions for a problem and choose the set of best ones out of them.

A genetic algorithm goes through main five stages.

1. Initial Population
2. Evaluation
3. Selection
4. Crossover
5. Mutation

1. Initial Population

A Population is a collection of individuals from which the process is initiated. Genes are a collection of factors (variables) that define an individual. A chromosome is made up of a string of chromosomes (solution). A genetic algorithm uses a string to describe a gene pool as if it were an alphabet. In most cases, binary numbers are utilized (string of 1s and 0s). Genes are encoded in a chromosome, according to our terminology.

Figure 2 shows the overview of genes, chromosome and population.

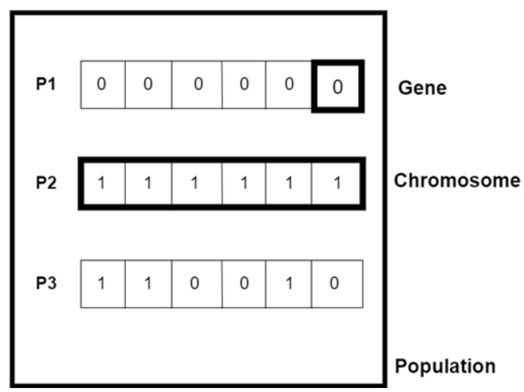

Figure 2. Gene, Chromosome and Population

2. Evaluation

Under Evaluation stage fitness function calculation and sorting are done.

2.1 Fitness Function calculation

The fitness level is determined by the fitness function. Each user receives a fitness rating. An individual's fitness score determines how likely it is that they will be picked for reproduction.

2.2 Sorting

This is one of the crucial component of the algorithm, as it offers search direction to the program by sorting fitness in decreasing order and according to that order it sorts starting population, error and function values. Here error is sorted just to stop the program besides this there is no purpose of it, and functions values are sorted to present optimize function value. Initial population is sorted and from this population some reproduction is maintained, the kept number is determined by population multiplied by selection probability, and from this kept population the highly fit population is chosen for reproduction using roulette wheel selection technique. Then, the reproduction is implied crossover and mutation operator on this current population and make new population.

3. Selection

It ensures that only the fit individuals will be able to pass on their genes. The fitness ratings of the parents are used to choose two pairs of people (parents). In order to reproduce, an individual must have a high level of fitness.

Roulette wheel is used for selection process. Individuals are selected according to their fitness, and the likelihood of selection is inversely proportional to their fitness. There are substantial differences between the approach and real-world roulettes. In a roulette wheel selection, the circular wheel is split. A fixed point is picked on the wheel circumference as depicted and the wheel is revolved. The area of the wheel which comes in front of the fixed point is picked as the parent. For the second parent, the identical procedure isperformed.

4. Crossover

A genetic algorithm's most critical stage is crossover. A crossover point is selected at random for each pair of parents to be mated.

Example:

Crossover point is shown in figure 3.

Crossover Point = 3

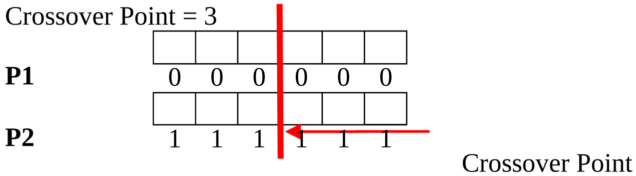

Figure 3. Crossover Point

Until the crossover point is achieved, offspring are produced by transferring the genes of their parents between themselves.

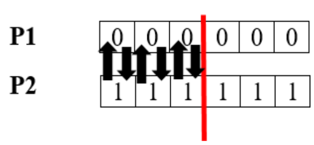

Figure 4. Exchanging genes among parents

The population grows as a result of the birth of new offspring as shown in figure 5.

P4	1	1	1	0	0	0
P5	0	0	0	1	1	1

Figure 5. New Offspring

5. Mutation

Low random mutations are possible in some new children, although this is not always the case. This means that it is possible to flip some of the bits inside the bit string.

Before and after mutation of chromosome P4 is shown in figure 6.

Before Mutation

P4	1	1	1	0	0	0

After Mutation

P4	1	1	**0**	**1**	**1**	1

Figure 6. Before and After Mutation

The purpose of mutation is to keep a population's genetic diversity diverse and to keep it from convergent too soon.

6. Termination

If the population has reached a stable equilibrium, the process is terminated (does not produce offspring which are significantly different from the previous generation). The issue may now be described as having been solved via a genetic algorithm.

IV. Reactive Power Optimization using Genetic Algorithm

A. Introduction

There are different genetic Algorithms like Vector evaluated genetic algorithm (VEGA), non-dominated sorting GA (NSGA), and niched Pareto GA (NPGA) to solve different kind of problems. In this paper, the Multi-Objective Genetic Algorithm is used to optimize the Reactive Power.

In many real-world situations, several objective functions must be optimized simultaneously. These goal functions are often incommensurable and, in some cases, at odds with one another [8]. A collection of optimum solutions, rather than a single optimal solution, is the result of multi-objective optimization with competing goal functions. There is no one solution that is better than any other in terms of all objective functions, and this is why multiple solutions are deemed optimum.

The multi-objective optimization approach is provided in this article as an implementation. A schematic of the Multi-Objective Genetic Algorithm approach which is used to optimize the reactive power is illustrated in Figure 7.

The Multi-Objective Genetic Algorithm differs from the regular Genetic Algorithm in the method in which each solution in the population is awarded a measure of fitness. The rest of the algorithm mimics the original Genetic Algorithm in many aspects. Multi-Objective Genetic Algorithm is a well-known multi-objective optimization technique that excels at handling optimization challenges that have a lot of limitations. The Multi-Objective Genetic Algorithm is essentially a basic genetic algorithm that is customized to solve multi-objective optimization issues. In the evolutionary algorithms, Multi-Objective Genetic Algorithm is the most efficient and straightforward way of optimization. The more dominating a person is, the more useful they are in terms of fitness.

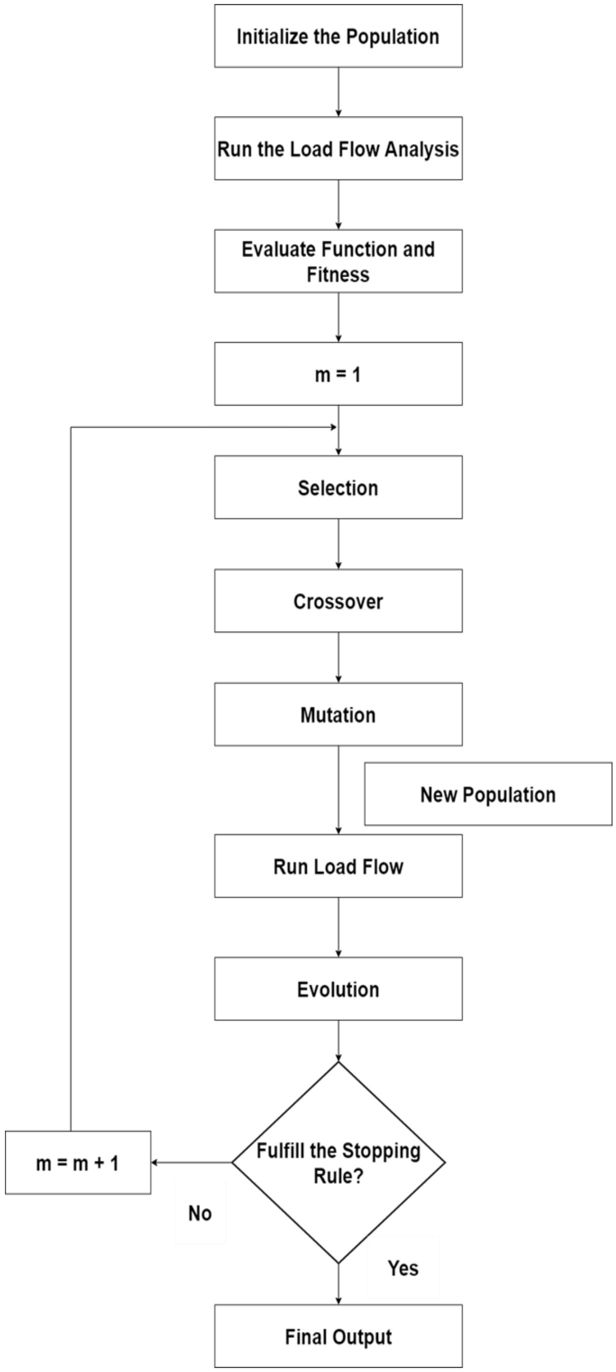

Figure 7. Schematic of the Multi-Objective Genetic Algorithm approach to optimize the reactive power

The dominance of a solution in the population is the first step in this algorithm. Solution m has a rank equal to one plus the number of solutions that predominate over it. Accordingly, a solution that does not dominate the population is considered to be non-dominated and is given arank equal to 1. The raw fitness of a solution is given depending on its rank after the ranking has been completed.

In order to do this, the ranks must first be arranged in order of increasing size. A linear (or any other) mapping function is used to assign a raw fitness to each solution. A typical mapping function assigns fitness between N (the top ranked answer) and 1 to the mapping function (for the worst rank solution). After then, the raw finesses of all of the solutions in a given rank are averaged. The given fitness for each ranked answer replaces the average fitness. This places an emphasis on a diverse range of options among the general public. The introduction of niching among solutions of each rank is necessary to preserve variety among non-dominated solutions.

The following 6 steps describe the Multi-Objective Genetic Algorithm.

Step 1:

Set m = 1. Initialize $\mu(n) = 0$; n = 1, 2…..N.

Step 2:

Figure out how many solutions (a_m) there are for each problem which dominate solution m. Calculate the m^{-th} solution's rank using the formula $r_m = 1 + a_m$. The number of solutions in rank r_m is to be increased by one, $\mu(r_m) = \mu(r_m) + 1$.

Step 3:

If m<N, increase m by one and move to step 1. Otherwise move to step 4.

Step 4:

Check the biggest r_m that has to find the highest rank r*. According to the sorting according to rank and fitness averaging, the average fitness is assigned to any solution that rank has $\mu(r_m)>0$ is the first integer in the series, therefore m=1,...N.

$$Average\ Fitness = N - \sum_{k=1}^{r_m - 1} \mu(k) - 0.5(\mu(r_m) - 1) \tag{8}$$

Step 5:

When calculating the number of solutions in the same rank that have the same niche count as the one we arelooking at.

$$\sum_{n=1}^{\mu(r_m)} Sh(d_{mn}) \tag{9}$$

$\mu(r_m)$ = Number of solutions for rank

Sh = Sharing Function

$$Sh(d_{mn}) = \begin{cases} 1 - \left(\frac{d_{mn}}{\sigma_{share}}\right) & ; d_{mn} \leq \sigma_{share} \\ 0 & ; otherwise \end{cases} \tag{10}$$

σ_{share} = Maximum distance for two values to become a number of niche

α = Scaling factor ; $\alpha \leq 0$

d_{mn} = Normalized distance between m and n

$$d_{mn} = [\sum_{K=1} (\frac{f_K^{m} - f_K^{n}}{f_K^{max} - f_K^{min}})^2]^{1/2} \tag{11}$$

f_K^{max} = Maximum objective function value of the K^{th} objective

f_K^{min} = Minimum objective function value of the K^{th} objective

It is determined by dividing the assigned fitness to a solution by its niche count to get the shared fitness value. In all levels, all solutions have the same fitness, however solutions in less-traveled areas have a higher shared fitness. As a result, there is a lot of pressure on solutions that are underrepresented, regardless of rank.

Then the shared fitness is calculated using the following equation.

$$F_n' \leftarrow \frac{r_n \mu(r)}{\sum_{K=1}^{\mu(r)} F_K'} \tag{12}$$

Step 6:

If r is less than or equal to r*, then go to step 5. If not, the procedure is complete. As a result, the process is carried out until all ranks have been examined. To establish a new population of organisms, the selection, crossover, and mutation operators are used.

Ranking, fitness, and fitness sharing are all calculated using the aforementioned algorithmic stages. The fitness of each solution is reduced when the allocated fitness values are divided by the niche count. The fitness values are scaled in such a way that their average shared fitness value is the same as their average assigned fitness value in order to maintain the average fitness of the solutions in a rank. These computations are followed by the following rank's answer. As a result, the procedure is repeated until all rankings have been processed. After that, a new population is generated using stochastic universal selection, crossover, and mutation operators with shared fitness values.

B. Implementation of Genetic Algorithm to Optimize reactive Power

The following three steps are done to optimize the reactive power.

1. Parameter Encoding
2. Evaluate the Fitness value
3. Apply the Genetic Operators

2.1 Parameter Encoding

The encoding of a problem's parameters is the first step in implementing Genetic Algorithm (i.e. the representation of the problem). In the genetic population, each person is a potential solution. Every single decision variable in the system is a part of the answer. The components of the solution are shown in Figure 8. The magnitude of the generator bus voltage, the transformer tap settings, and the reactive power sources are among the decision factors in the reactive power optimization issue. Floating point numbers are used to represent the solution variables. Keeping the population in memory is easier because to the use of direct representations of the solution variables.

u1	u2	….	ut	T1	T2	….	Tt	λ1	λ2	….	λt

Figure 8.Components of the Solution

ut = Voltage magnitude
Tt = Transformer tap settings
λt = Reactive power sources

- Length of the elements = Total number of control variables.

Generator terminal voltage and transformer tap setting are utilized as control variables in this example. The natural form of these variables is used to represent them. The reactive power optimization problem's typical chromosome looks like in the figure 9.

0.975	0.996	…	0.17	5	2	…	3	1	3	…	2
u_{G1}	u_{G2}		u_{Ga}	T_1	T_2		T_K	λ_1	λ_2		λ_K

Figure 9. The reactive power optimization problem's typical chromosome

2.2 Evaluate the Fitness Value

Only a few factors are controlled by the power system operator. In the population, the solution variables for some of these control factors are represented as independent variables. Candidate solution controls must be produced, and those control variables that violate the system's upper or lower bounds must be punished to dissuade the infeasible options from being implemented. By maximizing a fitness function, Genetic Algorithm seeks for the best possible solution, and hence it is necessary to supply an evaluation function that indicates how good a solution is. Voltage stability margin and loss minimization are the primary goals of this reactive power optimization issue under examination. Running the power flow software ensures that the equality conditions are met. It is the optimization method that controls the generator terminal bus voltage, transformer tap setting, and the reactive power provided by the capacitor bank. Static factors include: the slack bus's active productionof electricity, as well as that of the load bus, reactive power generation, and the line flow limit. A penalty term is added to the objective function to ensure that the state variables aremet.

2.3 Apply the Genetic Operators

Selection, crossover, and mutation are done as Genetic Operators to the algorithm.

2.3.1 Selection

Selecting the right search terms is critical to the success of a search engine optimization (SEO) campaign. Good solutions are prioritized, while poor ideas are eliminated, and the population size is kept at a stable level. The idea is to make it easier for the "fittest" individuals to reproduce. The term "tournament selection" is used in this paper to describe the process we've used. "a" individuals are chosen at random from the population for inclusion in a new population for additional genetic processing in tournament selection. Matting pools must be filled before this technique may be repeated. Even though it is possible to have bigger tournaments, tournaments are most often conducted in pairs (tournamentsize-2).

2.3.2 Crossover

Sharing information across chromosomes is done via the crossover operator. One or both parents' traits are combined to create two new chromosomes, and it's possible that excellent chromosomes may produce even better ones.

2.3.3 Mutation

In order to introduce new genetic material into the population, a mutation operator is utilized. The new progeny are subist to random mutation. Mixed variables are subjected to several variations of the "Non UniNon-Uniformon" operator in this study. The first step is to randomly choose a variable from an individual.

After mutation, the next generation is complete, and the algorithm starts again with the fitness assessment of the population.

V. RESULTS

The simulation of IEEE 39 bus system is done as in figure 10.

Transmission lines which are connected in between buses. Upper and lower Voltage limits of all buses except slack bus are defined as 1.10PU and 0.95PU. Slack bus voltage is set to 1.05PU which is the pre-defined value for the bus. Three optimization variables which have been takenfor the analysis are generator terminal voltages, transformer tap settings and reactive power sources. Nine locations were selected to initiate reactive power sources respectively 17, 18, 21, 24, 27, 28, 35, 36, and 38. First of all initial population should be generated and that was done by randomselection method and ranged between the lower and upper limits of variables. When selecting the members of the new population, tournament selection was applied, and to select individuals, uniform and crossover mutation was deployed. Specifically, the Multi-Objective Genetic Algorithm performance depends on the parameters which were used when generating the probabilities of crossover and mutation process. The range for the performance of the Multi-Objective Genetic Algorithm in terms of the probabilities of crossover and mutation is respectively 0.5-0.8 and 0.001-0.01. After that, the evaluation process was begun and it was executed by considering the diverse set of parameters in order to justify its potential to grant tolerable adjustments nearly to the Pareto optimal front.

In order to achieve optimization, the following parameters were set accordingly.

Generations - 60
Population size - 60
Crossover rate - 0.8
Mutation rate - 0.01
Variable - 20

The outcome of the proposed method was giving 20 optimal solutions in a single run which have distinct set of characteristics.

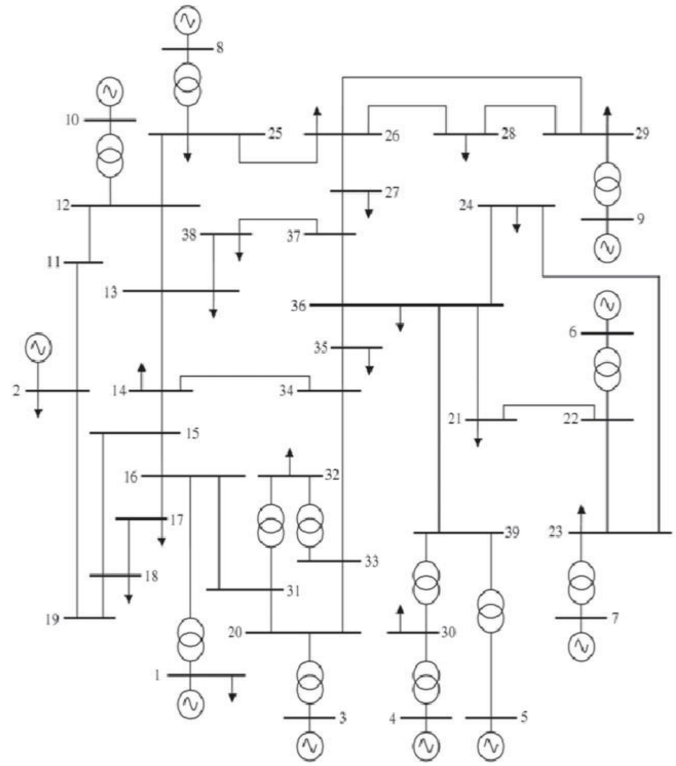

Figure 10. IEEE 39 Bus System

Following table 1 represents the optimal values for the three control variables.

TABLE 1 CONTINGENCY CONSTRAINED REACTIVE POWER OPTIMIZATION FOR IEEE 39-BUS SYSTEM

Variable	**Minimum real power loss solution**	**Maximum voltage stability margin solution**
u1	0.9342	0.9438
u2	0.9721	0.9871
u3	0.9767	0.9927
u4	1.0146	0.9765
u5	1.0273	0.9722
u6	0.9895	0.9943
u7	0.9948	09751

u8	0.9354	0.9649
T1	1.08	1.1
T3	1.06	1.02
T4	1.12	1.76
T5	1.16	1.2
T7	1.1563	1.056
QC21	3	0
QC24	3	0
QC27	5	1
QC28	2	0
QC37	4	1
QC35	1	5
QC18	2	4
Real Power Loss	5.684	4.652
Voltage Stability Margin	0.1762	0.267

As shown in table 2, the proposed method was deployed to decrease the real power loss and enhance the voltage stability margin without considering the contingency constraints. This contingency analysis was done using the control variables which were set before and the results confirm that the lines 20-32, 30-32, 1-16 and 3-20 as the most adverse lines in IEEE-39 bus system. The following table shows the values of maximum voltage stability margin(VSM) which respects to the contingency values generated before. Finally, it can be justified that the minimum Eigen value has escalated remarkably for all the values of contingency in this scenario. Therefore, it is confirmed that the proposed algorithm has enhanced the voltage security of the system significantly.

TABLE II. VOLTAGE STABILITY MARGIN

Line	Voltage Stability Margin	Multi Objective Genetic Algorithm - Voltage Stability Margin
20-32	0.0824	0.0935
30-32	0.0971	0.0987
1-16	0.1356	0.1401
3-20	0.1528	0.1598

VI. CONCLUSION

Multi-Objective Genetic Algorithm is used to gain optimum values of the reactive power variables in this paper. The method is demonstrated using IEEE-39 bus system. The performance of the proposed algorithm is demonstrated through its voltage stability enhancement by modal analysis. The Multi-Objective Genetic Algorithm method produced the highest optimization solutions which lead to a global search with a fast computational rate. The outcome of this method proves that this is more convenient and stable than the conventional genetic algorithms.

REFERENCES

[1] R. K. Kapadia and N. K. Patel, "Reactive Power Optimization Using Genetic Algorithm," in Nirma University International Conference on Engineering (NUiCONE), Visnagar, 2013.

[2] R. ZHU, "The Approach of Genetic Algorithms Application on Reactive Power Optimization of Electric Power Systems," in 4th International Conference on Machinery, Materials and Computer (MACMC 2017), Linzhi , 2017.

[3] H. Bi, "Artificial Intelligence Algorithms for Reactive Power Optimization in a Given Operating Mode," in AIP Conference, Baoding, 2019.

[4] R. Chi, Z. Li , X. Chi and Z. Qu , "Reactive Power Optimization of Power System Based on Improved Differential Evolution Algorithm," Hindawi, Nanchang, 2013.

[5] A. Jeyanthy and D. Devaraj, "MULTI-OBJECTIVE GENETIC ALGORITHM FOR REACTIVE POWER OPTIMIZATION INCLUSING VOLTAGE STABILITY," in International Journal of Engineering Science and Technology, TamilNadu, 2010.

[6] Kinnear, K. E. (1994). A Perspective on the Work in this Book. In K. E. Kinnear (Ed.), Advances in Genetic Programming (pp. 3-17). Cambridge: MIT Press

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

[8] D.B.Fogel, Z.Michalewicz," How to solve it: Modern Heuristics," NewYork, N.Y., Springer, 2004

AUTHORS' BACKGROUND

Your Name	Title*	Research Field	Personal website
J.A.D.C.A.Jayakody	Assistant Professor	Computer Systems Engineering	https://www.linkedin.com/in/dr-anuradha-jayakody-19a42915/
E.A.G.A.Edirisinghe	Research Assistant	Computer Systems Engineering	https://www.linkedin.com/in/gayani-anuradha-edirisinghe-279159121/
G.A.D.D.Ganepola	Instructor	Electrical Engineering	https://www.linkedin.com/in/dinushi-ganepola-839142174/