Scientific Journal of Impact Factor (SJIF): 3.134

ISSN (Print): 2348-6406
ISSN (Online): 2348-4470

International Journal of Advance Engineering and Research Development

GENETIC ALGORITHM BASE REACTIVE POWER DISPATCH AND VOLTAGE CONTROL

Sunil P. Patel [1]

[1]Electrical Department, LDRP-ITR (Kadi sarwa vidhyalaya uni.),Gandhinagar, Gujarat, India,
patelsunil33@yahoo.com

Abstract *- Reactive power control becomes extremely important after deregulation because costumers request high quality power supply with reasonable price. Also the Voltage profile at different nodes in the power system is affected greatly by the variations in load and generation profiles during normal and abnormal operating states. The reactive power control devices such as generators, tap positions of on-load tap changer of transformers, shunt reactors are used to correct voltage limits violations while simultaneously reducing the system real power losses. Genetic algorithms (GAs) are well-known global search techniques anchored on the mechanisms of natural selection and genetics. Hear i present the work on genetic algorithm for reactive power optimization and voltage control. The objective of genetic algorithm base reactive power optimization is minimizing active power losses while maintaining the quality of voltages.*

Keyword: *Genetic Algorithms, global optimization, IEEE Standards, Losses, Optimal Control, Power Systems, Reactive Power dispatch, Voltage profile.*

I. INTRODUCTION

Carpentier first defined the OPF problem in early 1960s and OPF soon many researchers. Optimal power flow (OPF) is one of the main functions of power generation operation and control. Power flow study aims at reaching to the steady state solution of complete power networks. Power flow study is performed during the planning of a new system or the extension of an existing system. It is also necessary to evaluate the effect of different loading conditions of an existing system. Power flow equations represent a set of non-linear simultaneous algebraic equations, for which there has been no general solution until now. OPF is a nonlinear, non-convex, large-scale, static optimization problem with both continuous and discrete control variables. Even in the absence of discrete control variables, the OPF problem is non-convex due to the existence of the nonlinear (AC) power flow equality constraints [12].

II. PROBLEM FORMULATION

The formula for optimal power flow is represented by following standard form,

$$\text{Minimize } f(u,x)$$

Subject to,

$$g(u,x) = 0$$
$$h(u,x) \le 0$$

The objective function f(u,x) represent the system optimization goal. f is usually a scalar function, but in multi objective OPF it may be interpreted as a vector function. Vector function g(u,x) and h(u,x) represent system equality and inequality constraints respectively .

The objective of the reactive power optimization is to minimize the active power losses of the system, which is defined as follows [1].

$$P_{loss} = f(\overrightarrow{x_1}, \overrightarrow{x_2})$$

$$P_{loss} = \sum_{k \in N_E} g_k (V_i^2 + V_j^2 - 2V_i V_j \cos\theta_{ij})$$

In above equation $f(\overrightarrow{x_1}, \overrightarrow{x_2})$ denotes the active power loss function of the transmission network, $\overrightarrow{x_1}$ is the control variable vector [$V_G\ T_K\ Q_C$], $\overrightarrow{x_2}$ is the dependent variable vector [$V_L\ Q_G$] , V_G is the generator voltage (continuous), T_K is the transformer tap (integer), Q_C is the shunt capacitor or reactor (integer), V_L is the load bus voltage, Q_G is the generator reactive power, k = (i, j) , $i \in N_B$ and $j \in N_i$, g_k is the conductance of branch k , θ_{ij} is the voltage angle difference between bus i and j , N_E is the set of number of network branches.

Equality constrain [1],

$$P_{Gi} - P_{Di} - V_i \sum_{j \in N_i} V_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}) \quad Q_{Gi} - Q_{Di} - V_i \sum_{j \in N_i} V_j (G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij})$$

The inequality constraints on control (independent) variable limits are described as follows,

Inequality constrain [1],

$$\begin{array}{ll} V_{i\min} \le V_i \le V_{i\max} & i \in N_B \\ T_{k\min} \le T_k \le T_{k\max} & K \in N_T \\ Q_{Gi\min} \le Q_{Gi} \le Q_{Gi\max} & i \in N_G \\ Q_{Ci\min} \le Q_{Ci} \le Q_{Ci\max} & i \in N_C \\ S_l \le S_{l\max} & l \in N_l \end{array}$$

In these constraints P_{Gi} and Q_{Gi} are the injected active and reactive power at bus i, P_{Di} and Q_{Di} are the demanded active and reactive power at bus i , G_{ij} and B_{ij} are the transfer conductance and transfer susceptance between bus V and J , N_{PQ} is the set of numbers of PQ buses, N_B is the set of total numbers of buses, N_i is the set of numbers of buses adjacent to bus i including bus i , N_o is the set of numbers of buses excluding slack bus, N_c is the set of numbers of possible reactive power sources installation buses, N_G is the set of numbers of generator buses, N_T is the set of numbers of transformer branches, S_l is the power flow in branch l.

III. METHODOLOGY

A. Initialization of Population

Population is a matrix having strings of binary numbers i.e. 0 and 1 which are nothing except potential solution to the problem. We generate a random population to start the solution. A set of real-coded initial populations are generated randomly within the minimum and maximum limits of the control variables and it is chosen as the parent population.

```
chromosome = round(rand(pop_size,nbit));
chromosome = randi([0,1],pop_size,nbit);
```

Where
pop_size = number of strings or simply you can
number of potential solutions
nbit = it is multiplication of bits per variable and
number of variables.

As these binary strings must be converted into real numbers, this is done by using Equation

$$P_i = P_i^{lower} + \frac{P_i^{uper} - P_i^{lower}}{2^n - 1} \times P_i^{decd}$$

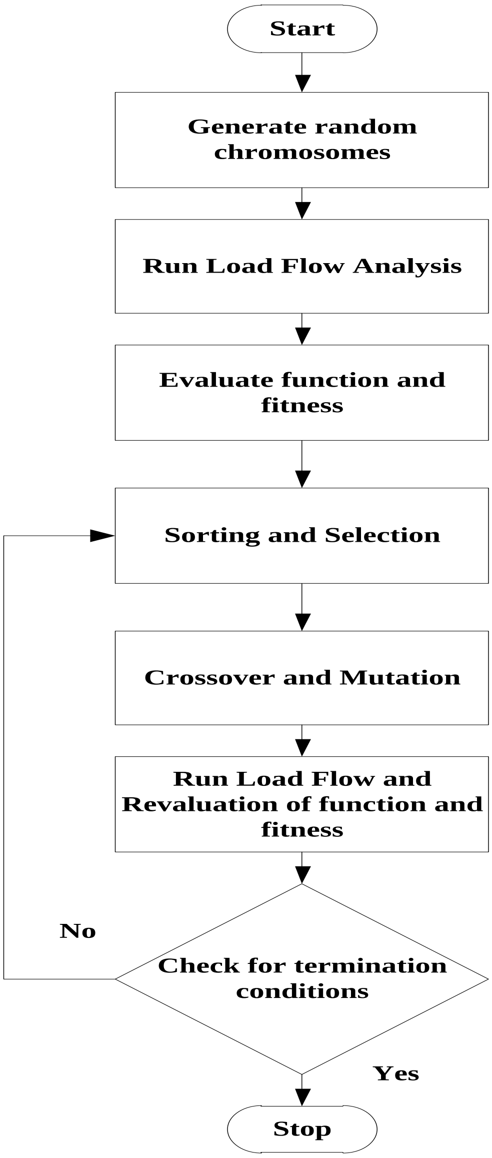
Where

P_i = Real limit of that varia

P_i^{lower} = lower limit of that variable

P_i^{uper} = upper limit of that variable

P_i^{decd} = binary to decimal converted value of that variable



Flow chart for Genetic Algorithm

B. Function Calculation

After decoding all the strings of the initial population and get values of each variable, we put them into our function and get its values.

C. Fitness Calculation

Fitness is the ability of a particular string to solve the problem. It is also a way for satisfying equality constraints. Implementation of a problem in genetic algorithm is realized within the fitness function and in order to emphasize the 'best' chromosomes and speed up convergence of the iteration procedure, fitness is normalized into the range between o and 1 . Fitness calculation is done in following steps.

Step 1. We first calculate error, which is nothing but our equality constraint.

Step 2. Fitness function is $fitn^1 = \frac{1}{function}$.

D. Sorting

it gives search direction to the program by sorting fitness in descending order and according to that order it sorts initial population, error and function values.

E. Selection

Roulette wheel selection procedure uses fitness of that string to make slot sizes .

Step 1. Calculate fitness of each string.

Step 2. Calculate probability of each string i.e. $prob_i = \frac{fitn_i}{\sum_{j=1}^{n} fitn_j}$, where is population size.

Step 3. Calculate cumulative probability for each string by using following equatio $comprob_i = \sum_{j=1}^{n} prob_j$

Step 4. Generate random number 0 < r < 1

Step 5. If r < $comprob_1$ then select the first string, or select the string which has lesservalue of cumulative probability than randomly generated number.

Step 6. Repeat Step 4 and 5 for number of strings is needed.

The whole procedure is explained with an example. Suppose we have pop_size = 5. The total fitness $\sum_{i=1}^{5} fitn_i = 15 + 20 + 27 + 08 + 23 = 93$. The probability of selecting individual string, its cumulative probability and fitness is shown in following Table-1 .

Sting	1	2	3	4	5
Fitness	15	20	27	08	23
Probability	15/93= 0.1612	0.2150	0.2903	0.08602	0.2473
Cumulative Probability	0.1612	0.3762	0.6665	0.7525	1

Table 1 -Roulette Wheel selection method

F. Crossover and Mutation

Selected strings, now onwards we will call them parents, are used to do crossover and mutation to produce children which are responsible for the global search property of the GA. Crossover operator basically combines substructures of two parent chromosome to produce new structure which is child. There are many types of crossover like single point, double point, multipoint, uniform, matrix etc. The final operator in the genetic algorithm is mutation; the mutation operator is used to inject new information into the population. Mutation changes randomly the new offspring. For binary mutation is preferred which switches a few randomly chosen bits from 1 to 0 and vice-a-versa. There are many types of mutation processes like uniform, boundary, non-uniform etc. After mutation, the new generation is completed and the procedure begins again with the function calculation of the population and so on.

chromosome =

0 1 1 0 1 1
0 0 0 0 0 0
0 1 1 1 1 1
0 0 0 1 0 1

Original population

Single Point CrossOver

chromosome =

0 1 1 0 1 1
0 0 0 0 0 0
0 1 1 0 0 0
0 0 0 1 0 1

3rd string
Cross over point is at 3.

1st string

2nd string

chromosome =

0 1 1 0 1 1
0 0 0 0 0 0
0 1 1 0 0 0
0 0 0 0 1 1

4th string crossover point is at 3.

2nd string

1st string

Random Bitwise Mutation

mutrow =

3 4

mutcol =

4 3

chromosome =

0 1 1 0 1 1
0 0 0 0 0 0
0 1 1 1 0 0
0 0 0 0 1 1

Bit number (3,4) was originally 0 and mutate to 1.

chromosome =

0 1 1 0 1 1
0 0 0 0 0 0
0 1 1 1 0 0
0 0 1 0 1 1

Bit number (4,3) was originally 0 and mutate to 1.

IV. SIMULATION RESULT

Variable	Value
V1(p.u)	1.0221
V2(p.u)	1.0190
V5(p.u)	1.0182

V8(p.u)	0.9888
V11(p.u)	0.9752
V13(p.u)	0.9803
Pgen1 (MW)	1.4570
Pgen2 (MW)	0.2584
Pgen3 (MW)	0.1732
Pgen1 (MW)	0.1340
Pgen1 (MW)	0.1784
Pgen1 (MW)	0.1263
Tap 9	0.9130
Tap 10	1.0935
Tap 12	0.9661
Tap 27	0.9600
Qsh 10	0.2229
Qsh 24	0.2390

Table – 2 shows the results of MATLAB program on IEEE 30-bus test system.

Variables	Value using GA	Value using NR load flow
V1 (p.u.)	1.0175	1.06
V2 (p.u.)	1.0182	1.045
V3 (p.u.)	1.0035	1.01
V6 (p.u.)	0.9968	1.07
V8 (p.u.)	1.0038	1.08
Pgen1 (MW)	21.065	0
Pgen2 (MW)	70	40
Pgen3 (MW)	50	0
Pgen6 (MW)	70	0
Pgen8 (MW)	50	0
Tap8	0.998	0.978
Tap9	0.989	0.969
Tap10	0.995	0.932

Table – 3 shows comparison of optimal values of variables generated using genetic algorithm and values of the same variables in normal load flow analysis using NR method.

V. CONCLUSION

The proposed Genetic algorithm approach to obtain the optimum values of the reactive power Variables including the voltage stability constraint. The effectiveness of the proposed method is demonstrated on IEEE-14 and IEEE-30 bus system with promising results. The performance of the proposed algorithm is demonstrated through its voltage stability enhancement by simulation. From this multi-objective reactive power optimization solution, the application of GA has performed well when it was used to characterize Pareto optimal front and leads to global search with fast convergence rate and a feature of robust computation. From the simulation work, it is concluded that GA performs better results than other evolutionary methods.

VI. REFERENCE

[1] Liu, S. C., Zhang, J. H., Liu, Z. Q., & Wang, H. Q. (2010, October). Reactive power optimization and voltage control using an improved genetic algorithm. In *Power System Technology (POWERCON), 2010 International Conference on*(pp. 1-5). IEEE.

[2] Krost, G., & Bakare, G. A. (1999, August). A genetic algorithm based approach for improvement in voltage profile and real power loss minimization. In *Electric Power Engineering, 1999. PowerTech Budapest 99. International Conference on*(p. 153). IEEE

[3] Padhy, N. P., Abdel-Moamen, M. A., & Praveen Kumar, B. J. (2004, June). Optimal location and initial parameter settings of multiple TCSCs for reactive power planning using genetic algorithms. In *Power Engineering Society General Meeting, 2004. IEEE* (pp. 1110-1114). IEEE.

[4] Wang, R., Lin, F., You, X., & Zheng, T. Q. (2008, June). Research on the reactive power control of grid-connected inverter of distributed generation system based on genetic algorithm. In *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on* (pp. 1096-1099). IEEE.

[5] Hong, Y. Y., & Luo, Y. F. (2009). Optimal VAR control considering wind farms using probabilistic load-flow and gray-based genetic algorithms. *Power Delivery, IEEE Transactions on*, *24*(3), 1441-1449.

[6] Zhao, F., Ge, L., & Li, W. (2012, March). Application of Ant-Genetic Algorithm in Reactive Power Optimization of Distribution Network. In *Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific* (pp. 1-4). IEEE

[7] Dai, C., Chen, W., Zhu, Y., & Zhang, X. (2009). Seeker optimization algorithm for optimal reactive power dispatch. *Power Systems, IEEE Transactions on*,*24*(3), 1218-1231

[8] Mathew, T. V., & Radhakrishnan, P. (2010). Calibration of microsimulation models for nonlane-based heterogeneous traffic at signalized intersections.*Journal of Urban Planning and Development*, *136*(1), 59-66.

[9] Younes, M., Rahli, M., & Abdelhakem-Koridak, L. (2007). Optimal Power Flow Based on Hybrid Genetic Algorithm. *Journal of Information Science & Engineering*, *23*(6).

[10] Kumari, M. S., & Maheswarapu, S. (2010). Enhanced genetic algorithm based computation technique for multi-objective optimal power flow solution.*International Journal of Electrical Power & Energy Systems*, *32*(6), 736-742.

[11] Wei, H., Chunli, X., Jianhua, Z., & Shan'ang, H. (2003, September). Study of reactive power optimization based on immune genetic algorithm. In*Transmission and Distribution Conference and Exposition, 2003 IEEE PES*(Vol. 1, pp. 186-190). IEEE

[12] Sreejaya, P., & Rejitha, R. (2008, October). Reactive power and Voltage Control in Kerala Grid and Optimization of Control Variables Using Genetic Algorithm. In*Power System Technology and IEEE Power India Conference, 2008. POWERCON 2008. Joint International Conference on* (pp. 1-4). IEEE.

