

Optimal Thermal Generation Scheduling by Rapid Genetic Algorithm

Er. Uttampreet Singh

M.Tech Student, Department of Electrical Engineering
Guru Nanak Dev Engineering College
Ludhiana, India

Er. Harmeet Singh Gill

Assistant Professor, Department of Electrical Engineering
Guru Nanak Dev Engineering College
Ludhiana, India

Abstract-This paper explores the implementation of Genetic Algorithm to thermal power stations having three generators to meet some load demand. The goal is to minimise the total cost of real power generation within generator limits by meeting equality and inequality constraints. It is shown that the total cost of three generator power systems to meet a demand of 300 MW can be reduced by 5.4% by using genetic algorithm. The genetic algorithm program is designed using MATLAB software.

Keywords- Genetic Algorithm; Optimal Power Flow; Fitness Function; Cost; Objective Function; Crossover; Mutation; Reproduction; String; Chromosome; Evolutionary Programming

I. INTRODUCTION

Current interest in the OPF centers on its ability to solve for the optimal solution that takes account of the security of the system. The optimal power flow is the process of determining the dispatch schedule of power generators with minimum cost while satisfying the system constraints like upper and lower power and reactive generation limit, upper and lower voltage level limit, and the line flow limit[1][2]. The minimization of generation cost will result to a lower cost of electricity paid by the consumer. In its most general formulation, the OPF is a nonlinear, non-convex, large-scale, static optimization problem with both continuous and discrete control variables. Even in the absence of non-convex unit operating cost functions, unit prohibited operating zones and discrete control variables; the OPF problem is non-convex due to the existence of the nonlinear (AC) power flow equality constraints.

Different classical techniques have been also employed to solve the OPF problem e.g. Lambda iteration method, Gradient method, Newton's method [3], linear programming method [5], Interior point method [4]. Lambda iteration method also called the equal incremental cost criterion (EICC) method has its roots in the common method of economic dispatch used since the 1930s [6]. In [7] James Daniel Weber solved the OPF problem using Newton's Method. In this marginal cost data are determined as a by product of the solution technique. Newton method and gradient method suffers from the difficulty in handling the inequality constraints. To apply LP, input output function is to be expressed as a set of linear functions which may lead to loss of accuracy. Moreover they are not guaranteed to converge to the global optimum of the general non-convex

OPF problem. Finally, all these methods cannot be applied with discrete variables which are transformer taps.

It seems that GA and EP (Evolutionary programming) are appropriate methods to solve this problem, which eliminates the above drawbacks. The genetic algorithms are part of the evolutionary algorithms family, which are computational models, inspired in the nature. Genetic algorithms are powerful stochastic search algorithms based on the mechanism of natural selection and natural genetics.

II. GENETIC ALGORITHM

A simple Genetic Algorithm is an iterative procedure, which maintains a constant size population P of candidate solutions. During each iteration step (generation) three genetic operators (reproduction, crossover, and mutation) are performing to generate new populations (offspring), and the chromosomes of the new populations are evaluated via the value of the fitness which is related to cost function. Based on these genetic operators and the evaluations, the better new populations of candidate solution are formed.

With the above description, a simple genetic algorithm is given as follow:

1. Generate randomly a population of binary string
2. Calculate the fitness for each string in the population
3. Create offspring strings through reproduction, crossover and mutation operation.
4. Evaluate the new strings and calculate the fitness for each string (chromosome).
5. If the search goal is achieved, or an allowable generation is attained, return the best chromosome as the solution; otherwise go to step 3.

In genetic algorithm, there are three main genetic operators:-

- A. Crossover
- B. Mutation
- C. Reproduction

A. Crossover

Crossover is the primary genetic operator, which promotes the exploration of new regions in the search space. For a pair of parents selected from the population the recombination operation divides two strings of bits into segments by setting a crossover point at random, i.e. Single Point Crossover. The crossover can also be Multipoint Crossover and Uniform Crossover depending upon the way of exchanging the bits.

B. Mutation

Mutation is a secondary operator and prevents the premature stopping of the algorithm in a local solution. The mutation operator is defined by a random bit value change in a chosen string with a low probability of such change. The mutation adds a random search character to the genetic algorithm, and it is necessary to avoid that, after some generations, all possible solutions were very similar ones. Mutation is the operator responsible for the injection of new information. With a small probability, random bits of the offspring chromosomes flip from 0 to 1 and vice versa and give new characteristics that do not exist in the parent population [8].

C. Reproduction

The first genetic algorithm operator is reproduction. The reproduction genetic algorithm operator selects good string in a population and form a mating pool. So, sometimes the operator is also named as the selection operator. The commonly used reproduction operator is the proportionate reproduction operator where a string selected for the mating pool with a probability proportional to its fitness.

III. OBJECTIVE FUNCTION

The most commonly used objective in the OPF problem formulation is the minimization of total cost of real power generation. The individual costs of each generating unit are assumed to be function, only of active power generation and are represented by quadratic curves of second order. The objective function of entire power system can then be written as the sum of the quadratic cost model at each generator.

$$F_i = a_i P_{gi}^2 + b_i P_{gi} + c_i \quad (1)$$

where

$i = 1, 2, 3, \dots, n_g$.

n_g is number of generators including the slack bus

P_{gi} is the generated active power at bus i .

a_i, b_i, c_i are the unit cost coefficients for i^{th} generator.

The minimization of objective functions is done while satisfying the constraints. The constraints of the OPF problem can be split into two parts: The equality constraints, representing the power flow equations and the demand variables and the inequality constraint set, representing all the operational constraints.

IV. PROBLEM FORMULATION

The objective is to develop a program in MATLAB software to minimize the cost of real power generation i.e. To minimize the cost function $F(x)$ as given by eqn. (1) and simultaneously satisfying the load flow equations (equality constraints) without violating the inequality constraints. The standard OPF problem can be written in the following form,

Minimise $F(x)$ (the objective function)

subject to: $h_i(x) = 0, \quad i = 1, 2, \dots, n$ (equality constraints)

$g_j(x) = 0, \quad j = 1, 2, \dots, m$ (inequality constraints)

where x is the vector of the control variables that is those which can be varied by a control center operator (generated active and reactive powers, generation bus voltage

magnitudes, transformers taps etc.). The objective is to make all the calculations very rapidly and tries to minimize the cost to its minimum value that it can.

V. FITNESS FUNCTION

Genetic algorithms mimic the survival of the fittest principle of nature to make a search process. Therefore, genetic algorithms are naturally suitable for solving maximizing problems. Minimization problems are usually converted to maximization problems using some suitable transformation. Fitness function $f(x)$ is derived from the objective function and used in successive genetic operations. The fitness function for maximization problems can be used the same way as the objective function $F(x)$, i.e.

$$f(x) = F(x) \quad (2)$$

The fitness function for the minimization problem can be obtained from the objective function using the following relation:-

$$f(x) = 1/(1+F(x)) \quad (3)$$

This transformation does not change the location of minimum. But it only converts minimization problem to an equivalent maximization problem. The fitness value of the string is termed as string fitness. In many case, the fitness value corresponds to the number of offspring that an individual can expect to produce in next generation. A commonly used transformation is of proportional fitness assignment. Normally, the value of the fitness to be selected lies between 0 and 1. More nearer the value of fitness to 1, more accurate the result will be.

A. Step-Wise Procedure for G.A. In Optimization:

- 1) Read data, namely cost coefficients, B-coefficients, convergence tolerance, error, step size, maximum allowed iterations (optional), length of string, population size, probability of crossover, probability of mutation, seed number, maximum and minimum value of incremental cost, etc.
- 2) Generate an array of random numbers. Generate the population by flipping the coin.
- 3) Set generation counter, $k=0$, $BIG=1.0$, $f(\max)=0.0$ and $f(\min)=1.0$.
- 4) Increment the generation counter, $k=k+1$ and set population counter, $j=0$.
- 5) Increment the population counter, $j=j+1$.
- 6) Decode the string.
- 7) Using Gauss Elimination Method, find P_i^j .
- 8) Find ϵ and check if $\epsilon < BIG$, then set $\epsilon = BIG$.
- 9) Find Fitness
- 10) If $j < L$ (population size) then go to step 5 and repeat.
- 11) If $BIG \leq \text{error}$ then go to step 18.
- 12) Find population with maximum fitness and average fitness of the population also find cost of generation corresponding to the fitness values.
- 13) Select the parent for crossover using stochastic remainder roulette wheel selection.
- 14) Perform single point crossover for the selected parents.
- 15) Perform the mutation.
- 16) If generation counter is less than maximum number of iterations then goes to step 4 and repeat.
- 17) Stop

VI. RESULTS AND DISCUSSION

The system specifications are summarised as:-

TABLE I. TEST SYSTEM SPECIFICATIONS

S.No.	Quantity	Unit 1	Unit 2	Unit 3
1	a_i	0.00525	0.00609	0.00592
2	b_i	8.663	10.040	9.760
3	c_i	328.13	136.91	59.16
4	P_{\min} (MW)	50	5	15
5	P_{\max} (MW)	250	150	100
6	Load Demand	300MW		

The Rapid Genetic Algorithm used for optimization of power flow is applied to a three generator system with its specifications such as unit cost coefficients of three units, maximum and minimum power limits and the load demand as given in TABLE I.

Fig. 1 shows a variation of new fitness achieved in various rounds. In this we have selected the value of fitness required is 0.99 i.e. the nearest value to 1. Because the value of fitness lies between 0 and 1. More the value nearer to 1, more accurate our objective will be achieved. The Fig. 1 shows various values of fitness in different rounds, since the required value of fitness is entered 0.99. Here the maximum value of fitness is 0.99988177279 which is very much near to 1. It is found that maximum value occur in approximately 16th round.

Fig. 2 shows the variation of incremental cost (Lambada) in various rounds. All these values of incremental cost i.e. Lambada are found according to the value of fitness. The maximum and minimum value of Lambada (Incremental Cost) is selected as 10 and 12.5 respectively. The value of incremental cost corresponding to maximum value of fitness is 10.890669108.

Fig. 3 shows the variation of error calculated in various rounds of iterations. The error found corresponding to maximum fitness is 0.035472356687. In Fig. 3, some values of error are shown as negative but in actual these values are to be taken as positive (Modulus of value). The value once calculated is used to find the fitness value again. This process is repeated till the maximum value of fitness is achieved.

Our aim is to maximize the value of fitness function or we can say to minimise the cost of real power generation. The program is designed using MATLAB software and applied to three generator system whose specifications are given in TABLE I. During running of this program the GA operators are applied to the input data and one value of fitness value is selected which is very near to 1. For each iteration, the maximum value of fitness is calculated along with the incremental cost is given in TABLE II.

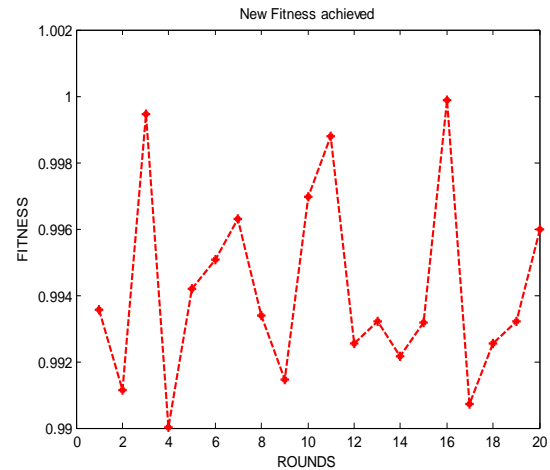


Fig.1. Maximum fitness achieved in various rounds.

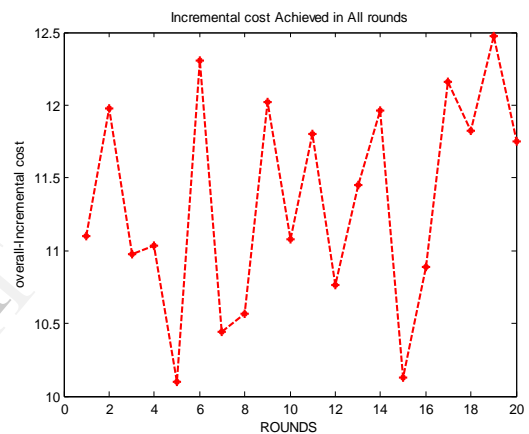


Fig. 2 Graph showing values of incremental cost.

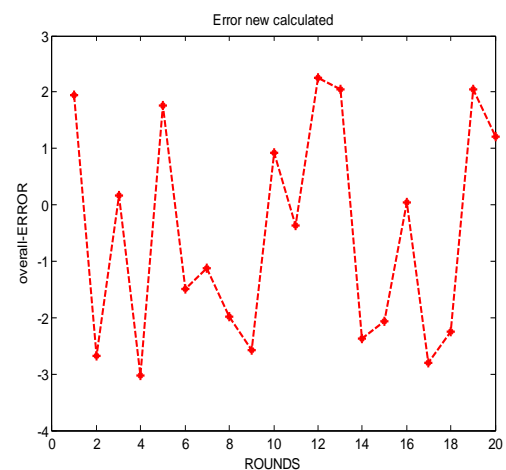


Fig. 3. Graph showing variation of error during various rounds.

TABLE II. MAXIMUM FITNESS CORRESPONDING STRING AND INCREMENTAL COST

New String Achieved	Lambda (Incremental Cost)	Fitness achieved
0011111100001110	11.103379873	0.99356215136
0011000101010011	11.978027008	0.99113553861
1001010000100110	10.978141451	0.99944908512
1110101110010110	11.033607995	0.99001947763
0001001001010000	10.100404364	0.99419309489
1100101111010111	12.303006027	0.99506346584
0010000010110100	10.439612420	0.99628237829
1010000001011100	10.566605630	0.99339837682
1010000011110011	12.021705958	0.99147089246
0101000001110110	11.074616617	0.99695073364
0100101000011101	11.800030518	0.99877544896
0100010001110010	10.763027389	0.99252718111
1100111100101001	11.454604409	0.99320193946
0101110100010011	11.960250247	0.99216474224
1100001010110000	10.129510948	0.99317383078
0010110011011010	10.890669108	0.99988177279
1001010010111011	12.159800106	0.99071111217
0101111101011101	11.825970855	0.99254569049
1110101010111111	12.474059662	0.99320193913
0111110011001101	11.750438696	0.99597719255

As a result of the program, the minimum value of cost of real power generation and the best fit powers of three different units is found. These are summarize in TABLE III.

TABLE III. FINAL RESULTS CALCULATED WITH BEST FITNESS VALUES

S. no	Parameter	Best calculated value	Theoretical Value[9]
1	Total Cost (Rs/h)	3420.72	3616.14
2	Best Fit Power Of Plant 1 (MW)	165.4750	202.4288
3	Best Fit Power Of Plant 2 (MW)	54.76060	80.94910
4	Best Fit Power Of Plant 3(MW)	73.67460	27.06991

VII. CONCLUSION

A Genetic algorithm solution to the optimal power flow problem has presented and applied to three generator power systems. It is concluded that the calculated value of total cost by rapid genetic algorithm is reduced by 5.4% as compared to theoretical value.

REFERENCES

1. H. R.Cai, C. Y. Chung and K. P. Wong "Application of differential evolution algorithm for transient stability constrained optimal power flow," IEEE Transactions on Power Systems, vol. 23, no. 2, 2007, pp. 719- 728.
2. B. Zhao, C. X. Guo and Y. J. Cao, "A multiagent-based particle swarm optimization approach for opimal reactive power dispatch," IEEE Transactions on Power Systems, vol. 20, no. 2,May 2005, pp.1070-1078.
3. T. Bouktir, M. Belkacemi and K. Zehar, "Optimal power flow using modified gradient method," Proceeding ICEL'2000, U.S.T.Oran, Algeria, vol. 2,2000, pp. 436- 442.
4. T. Bouktir, M. Belkacemi , L. Benfarhi, and A. Gherbi (2000), "Oriented Object Optimal Power Flow," The UPEC 2000, 35th Universities Power Engineering Conferences Belfast, Northern Ireland, 6-8.
5. L. Terra and M. Short, "Security constrained reactive power dispatch, IEEE transaction on power systems", vol. 6, Feb. 1991.
6. A.J. Wood and B. F. Wollenberg, "Power Generation Operation and Control", New York, NY: John Wiley & Sons, Inc., 1996.
7. James Daniel and Weber B.S , "Implementation of A Newton-Based Optimal Power Flow into a Power System Simulation Environment," University of Wisconsin – Platteville, 1995.
8. D. E. Goldberg , "Genetic Algorithms in Search, Optimization and Machine Learning, Reading". Reading, MA: Addison-Wesley,1989.
9. D.P. Kothari , " Power System Optimization", PHI Learning Pvt. Ltd 2011, pp 518-529.