Volume : 3 | Issue : 1 | Jan 2014 • ISSN No 2277 - 8160

Research Paper | Engineering

The Matlab Solution of Mathematic Equation Using Genetic Algorithm for Optimum Result

F.M.PATEL — M.E. [Power System] Student, Department Of Electrical Engineering, Kalol Institute of Research and Technology, Kalol, Gujarat

N. B. PANCHAL — Asst. Professor, Department Of Electrical Engineering, Kalol Institute of Research and Technology, Kalol, Gujarat

ABSTRACT

The main objective of this paper is to use of Genetic Algorithm method for optimization techniques. In recent trends the concept and practice for electricity has been changed due to deregulation of power system. Better utilization of the existing system resources to increase capability with economic cost becomes essential. Genetic algorithm (GA) is an optimization and search technique based on the principles of genetics and natural selection. GAs operates on a population of potential solutions applying the principle of survival of the fittest to produce successively better approximations to a solution. In this paper efficient MATLAB code is generated for the solution of mathematical problem in which minimization of objective function. Real Genetic Algorithm is used and result is presented.

KEYWORDS : Optimization, Genetic Algorithm, MATLAB

I: INTRODUCTION

For better utilization of existing system with less cost and maximum efficiency, now a day's artificial intelligent is most suitable. There are different AI techniques, optimization of objective functions such Evolutionary Programming[2], Evolutionary Strategies, Simulated Annealing, Tabu search, Particle swarm optimization[4][8], Ant colony search[4] and Genetic Algorithm[1][2][3][4]. Genetic Algorithm is most suitable for results for nonlinear, non smooth and non convex complex problems. GA takes less time and give mass of possible optimum solution not only a single point solution compare to conventional system like Newton's method, Linear programming method, Non-linear programming method or Interior point method for complex system. GAs most suitable for multi objective function and parallel computer functions for more number of iterations on the same objective function.

II: OPTIMIZATION PROBLEM

Optimization is the process of adjusting the inputs or characteristics of a device, mathematical process or experiment to find the minimum or maximum output or result. Optimization varies the input to achieve a desired output. The input consists of variables; the process or function is known as the cost function, objective function or fitness function and the output is the cost or fitness. To maximize a function put minus sign on the front of the output and minimization (ex.1-X^2). For minimization put (X^2-1).

III. GENETIC ALGORITHM

The genetic algorithm (GA) is an optimization and search technique based on the principles of genetics and natural selection. GAs operates on a population of potential solutions applying the principle of survival of the fittest to produce successively better approximations to a solution. At each generation of a GA, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and reproducing them using operators borrowed from natural genetics. This process lead to the evolution of populations of individuals that is better suited to their environment than the individuals from which they were created, just as in natural adaption.

Case Study – Find the Optimum minimum solution of mathematical equation $X^2+4*X+25$ for 20 generation.

Genetic algorithm is solving using binary or continuous (real) value solution. Continuous value solution is the modification of the first investigation binary GA. Real GA reduces complex city for complex problem and less time and storage space required compare to Binary GA.

Initialization of Population

A single encoding of part of the solution space, i.e. either single bits or short blocks of adjacent bits that encode an element of the candidate solution is called **Gene.** A string of gene that represents a solution population is called **Chromosome.** A set of chromosomes is called **population.** To begin the GA, we define an initial population of Npop chromosomes. A matrix represents the population with each row in the matrix being a 1*Nvar array of continuous values. The MATLAB code for,

```
Pop=rand(Npop,Nvar)
```

If the range of values is between plo and phi, then the un normalized values are given by

```
P=(phi-plo)pnorm+plo
```

Where

Plo=highest number in the variable range

Phi=lowest number in the variable range

Pnorm=normalized value of variable

Loss and Fitness Calculation

As a objective function here we consider $X^2+4*X+25$ for 20 number of iterations.

Sorting

Here we sort all the initially generated chromosomes in ascending order as we want to minimize our function value by putting minimum at the top.

```
b=randi([0,1],5,4)

de=bi2de(b)

sort(de)
```

Selection

From this sorted matrix of chromosome we select some random number of chromosomes for crossover and mutation process, probably those which have good fitness value.

Selection process can be done by many ways like using roulette wheel, stochastic selection or tournament selection. One simple method of selection involves generation of random numbers. The MATLAB code for Roulette Wheel Selection method,

```
pop_size=5;
fitn=[28 18 14 9 26];
for i=1:pop_size
 prob(i)=fitn(i)/sum(fitn)
end
```

```
comprob= cumsum(prob)
parents=zeros(1:pop_size);
for i=1
r=0.1682;
for j=1:pop_size
if r<comprob(j)
parents(i)=j;
break;
end
end
end
```

Output:

comprob = 0.2947 0.4842 0.6316 0.7263 1.0000

prob = 0.2947 0.1895 0.1474 0.0947 0.2737

comprob = 0.2947 0.4842 0.6316 0.7263 1.0000

Crossover and Mutation

Selected chromosomes, 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 ffspring. 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.

A single offspring variable value, pnew, comes from a combination of the two corresponding offspring variable value, pnew, comes from a combination of the two corresponding offspring variable values.

Pnew=βpmin+(1-β)*pdn[4]

Where,

β =random number on the interval [0 1]

Pmn= nth variable in the mother chromosome

Pdn=nth variable in the father chromosome

Crossover Probability

For better results, it is advisable to select the crossover rate quite large than mutation rate. This is the usual practice to take crossover rate 20 times greater than the mutation rate .Crossover rate generally ranging from 0.25 to 0.95[4].

Mutation Probability

The mutation probability (Pm) is approximately inversely proportional to the population size. Mutation rate generally varying from 0.001 to 0.03[4].

Evaluation

After doing crossover and mutation, function value and fitness of all chromosomes is found, this is called Regeneration. This regeneration process has two starting criteria, first maximum generation and error *E* check.

In first case if generation has reached some predefine generation value then this process will stop and the best chromosome will be our answer. In second case if our error will become less than some predefine value then we will terminate the program and the best chromosome is our answer.

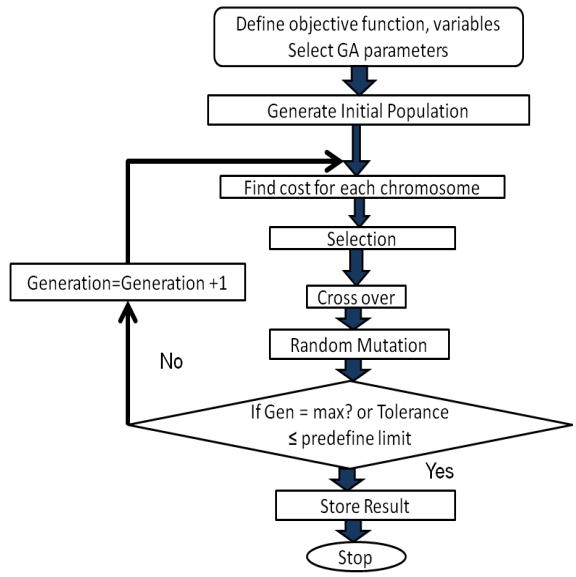


Figure 1(a) Flowchart of GA for mathematical equation $X^2+4*X+25$ for 20 generation problem[4]

IV: TEST CASE AND SIMULATION RESULTS

The example taken in section 3 is test case for this paper. Efficient MATLAB codes are generated to solve the same example.

Parameters for MATLAB code are as under.

pheno_length= 20

Dec_Point_pos=6;

INIT_POPULN= 80;

SEL_POPULN= 40;

GEN_POPLN = 40;

Genetic algorithms claim to provide near optimal or optimal solution for computationally intensive problems. Therefore, the effectiveness of genetic algorithm solutions should always be evaluated by experimental results. For solution of the objective function X^2+4*X+25 for minimum cost function. The table shows the minimum real value cost function for 20 iteration for time 0.062288 seconds.

Sr.No	Iteration No	Ans.
1	1	298.9168
2	2	298.9168
3	3	298.9168
4	4	298.9168
5	5	298.9168
6	6	298.9168
7	7	298.9168
8	8	111.9867
9	9	111.9867
10	10	111.9867
11	11	111.9867
12	12	111.9867
13	13	111.9867
14	14	111.9867
15	15	111.9867
16	16	111.9867
17	17	111.9867
18	18	111.9867
19	19	111.9867
20	20	111.9867

Elapsed time is 0.062288 seconds
The graph represents the total number of selected population and the minimum cost function possibility for 20 iteration.

V. CONCLUSION

The Genetic Algorithm, which is inspired from the biological genetics, is simple, powerful, domain free and probabilistic approach to general problem solving technique. It is best suited for the problems like load forecasting, economic load dispatch, transmission line reactive power [9]compensation for electrical power demand that is a type of non-linear, non smooth and non convex variations. Continuous GAs also are more compatible with other optimization algorithms.



Fig. Plot for Real Value Genetic Algorithm Solution Using Matlab Programming

REFERENCES

[1] D.P.Kothari and J.S.Dhillon, "Power System Optimization", 2nd Edition, PHI Learning Private Limited, 2011, pp 8-10,12,594-598,664-685, chapter 7. | [2] N.P.Padhy , "Artificial Intelligence and Intelligent Systems", First edition of Oxford University Press, 2005,pp 17-21, 460-508. | [3] David E. Goldberg, "Genetic Algorithm in search optimization and machine Learning", 11th edition Printed in India by Saurabh Printers Pvt. Ltd., pp 1-18. | [4] Randy L Haupt and Sue Ellen Haupt, "Practical Genetic Algorithms", 2nd Edition, A John Wiley & Sons Inc. Publication, 2004, pp 2-5,22-24,27-48,51-64,211-219. | [5] R. Narayan and S. Kumar," Determination of steady state operation of Power System using Genetic Algorithm",IJEEE, ISSN vol.2, Issue 1, Feb 2013, pp 13-20. | [6] M.S.Osman, M.A.Abo-sinna, A.A. Mousa, "A Solution to the optimal power flow using Genetic Algorithm", Applied Mathematics and computation, 2004,pp 391-405. | [7] H.Kim, N.Samann, D.Shin, B.Ko,G.Jang and J. Cha, "A New concept of Power Flow | analysis",Journal of Electrical Engineering and Technology, vol.2,2007, pp 312-319. | [8] M.Reddy, Y.P.Obulesh and S.S.Raju, "Particle Swarm Optimization based optimal Power Flow for Volt- Var Control",ISSN, January 2012,pp 20-25. | [9] J.Rahul, Y.Sharma and D.Birla, "A New attempt to Optimize Optimal Power flow based Transmission losses using Genetic Algorithm", Fourth International conference on computational Intelligence and Communication Networks, 2012, pp 566-570.