Economic Load Dispatch Using Genetic Algorithm Technique

[1]**Jabinbanu M Nadaf**

[1]Lecturer (Selection Grade)
[1]Department of Electrical and Electronics,
[1]Govt Polytechnic Belgaum, Karnataka,INDIA

Abstract -*Economic power dispatch problem plays an important role in the operation of power systems. The objective of economic dispatch problem is to schedule output of the committed units such that the total fuel cost is minimized while meeting a set of operating constraints. This paper presents an approach based on genetic algorithm (GA) to solve the economic load dispatch (ELD) problem with losses for three and six thermal plant systems. Genetic algorithms are adaptive search methods that simulate some of natural process: selection, information, inheritance, random mutation and population dynamics. The performance of genetic algorithm – intelligent approach (GAS) is compared with the conventional method and is observed that this method is accurate and may replace effectively the conventional practices presently performed in different central load dispatch centers.*

Keywords - *Economic dispatch, Genetic algorithms, Prohibited zones, Lambda-iteration, Network losses.*

1. INTRODUCTION

Economic load dispatch (ELD) is a sub-problem of the optimal power flow (OPF) having the objective of fuel cost minimization. The lambda-iterative method has been used for ELD. Many other methods such as Gradient methods, Newton's methods, Linear and Quadratic programming, etc. have also been applied to the solution of ELD problems. However, all these methods are based on assumption of continuity and differentiability of cost functions. Hence, the cost functions have been approximated in the differentiable form, mostly in the quadratic form. Further, these methods also suffer on two main counts. One is their inability to provide global optimal solution and getting stuck at local optima. The second problem is handling the integer or discrete variables.

Genetic algorithms (GAS) have been proved to be effective and quite robust in solving the optimization problems. GAS can provide near global solutions and can also handle effectively the discrete control variables. GAS does not stick into local optima because GAS begins with many initial points and search for the most optimum is parallel. GAS considers only the pay-off information of objective function regardless whether it is differentiable or continuous. Consequently, the, most realistic cost characteristic of power plants can be formulated.

This paper proposes the application of GAS to solve the economic load dispatch for six thermal plants systems and results are compared with conventional method.

2. ECONOMIC LOAD DISPATCH PROBLEM

The objective of the ELD problem is to minimize the total fuel cost at thermal plants,

$$Obj=\sum_{i=1}^{n} F_i(P_i)= F_T$$

Subject to the constraint of equality in real power balance

$$\sum_{i=1}^{n} P_i - P_L - P_D = 0$$

The inequality constraints of real power limits of the generation outputs are

$$P_{i\,min} < P_i < P_{i\,max}$$

Where,

$F_i(P_i)$ is the individual generation production in terms of its real power generation.
P_i is the output generation for unit i, n the number of generators in the system.
P_L total system transmission losses.
P_D the total current system load demand.

The thermal plant can be expressed as input-output models (cost function), where the input is the fuel cost and output the power output of each unit, the cost function could be represented by a quadratic function.

$$F_i(P_i) = A_i * P_i^2 + B_i * P_i + C_i$$

The incremental cost curve data are obtained by taking the derivative of unit input-output equation resulting in the following equation for each generator:

$$d\,F_i(P_i)/\,dP_i = 2 * A_i * P_i + B_i$$

The transmission losses are a function of the unit generations and are based on the system topology. Solving the ELD equations for a specified system requires an iterative approach since all unit generation allocations are embedded in the equation for each unit. In practice, the loss penalty factors are usually obtained using on line power flow software. This information is updated to ensure accuracy. They can also be calculated directly using the Bmn matrix loss formula.

$P_L = P_i B_{ij} P_j$ Where B_{ij} are co-efficients, constants for certain conditions.

3. GENETIC ALGORITHMS

GAS is inspired from phenomena found in living nature. The phenomena incorporated so far in GA models include phenomena of natural selection as there are selection and production of variation by means of recombination and mutation, and rarely inversion, diploid and others. Most genetic algorithms work with one large panmictic population, i.e., in the recombination step each individual may potentially choose any other individual from the population as a mate. Then GA operators are performed to obtain the new child offspring.

3.1. Genetic Algorithm Steps

GA borrows the analogous biological terms for each step. GAS maintains a population of parameter set solutions and iterate on the complete population. Each iteration is called a generation.

Step 1: Random initialization of population

In the first step of the GA, an initial population is created (Possible solutions to the problem) randomly or heuristically. In general, there are ‘n’ individuals (i.e. possible solutions points in the search space) in this population and ‘n’ is an even number (since later, pairs will be formed from the individuals). An individual is characterized by a fixed length binary bit string, which is called a chromosome. Thus an individual is characterized by a string of zeros and ones (Ex: an individual is an initial population can be described by binary string 01101 another individual by 10011 etc. Each of the strings is decoded into a set of parameters that it represents the initial population is then a collection of randomly generated individual binary strings.

Step 2: Evaluation of fitness of individuals in the population

In this step, all the individuals of the initially created population are evaluated by means of a fitness function (f). The fitness function is obtained from the objective function (performance index). The fitness function is then used in the next step to create a genetic pool.

Step 3: New population generation

In a biological evolutionary process, which is based on the Darwinian principle of reproduction and survival of the fittest, the new population generation involves the simultaneous occurrence of three processes:

1. The selection of the fittest parents (Reproduction).
2. Crossover (mating).
3. Mutation (Gene modifications).

In a single out of produce an offspring. In nature an offspring may have some abnormalities, i.e, Mutations. These are usually disabling and inhibit the ability of the offspring to survive, but sometimes they improve the fitness of the individual.

Stage 1: Selection stage (obtaining mating pool, with fittest individuals)

Before applying the crossover and mutation genetic operators the reproduction operators are applied i.e. individuals are selected among the population (mating solution) with probabilities proportional to their fitness values (since probability of being reproduced is proportional to the fitness value). In other words, individuals of the initial population are selected (for mating) into the new population according to a probabilistic rule, which favours those individuals with higher fitness values. There are many selection rules, but one of the most commonly use techniques is the “Roulette wheel selection”, where the probability of an individual to be selected is

$$P_i = f_i / \sum_{k=1}^{n} f_k \qquad (k = 1,2,\ldots,n)$$

Where ‘n’ is the size of population (no. of individuals) and fi is the fitness function. In this way individual with above average values replace individuals with below average fitness value. Because the total number of strings in each generation is kept constant (for computation efficiency), strings with lower fitness values are eliminated (this individuals ‘die’) while others are reproduced (copied to the next generation).

Reproduction: It is a process in which individual strings are copied according to their objective function values ‘f’ called as the fitness function. The function ‘f ’can be thought of some measure of profit, utility or goodness that we want to maximize copying strings according to their fitness values means that strings with a higher value have a higher probability of contributing one or more offspring in the next generation. This operator is an artificial version of natural selection, a Darwinian survival of fittest among string creatures.

Stage 2: Crossover stage (creation of off springs, two new strings created from existing two strings)

The selection stage is followed by applying the genetic crossover (mating) operated to the mating pool, since it is the goal to generate new individuals (which will retain good features from the previous generation and which may also out perform their parents) individuals of their mating pool are paired randomly and n/2 genetic couples are obtained (that is why n has to be an even number). Crossover is thus applied to selected pairs (parents) with a crossover probability Pc.

In other words, according to a fixed probability of crossover (Pc), each couple undergoes a crossing of the string bits (i.e. if the crossover rate is 1, then crossover is always done on selected parents). When the most basic crossover operator is used, which is the one point crossover operator, a crossover point (location) in the string bits of the selected pair is randomly chosen and the bits of the two parents are interchanged at this point.

Ex: a) one point crossover

	Before crossover		After crossover	
Bit string of Individual 1	1 0	1 1 0 1 1	1 0	0 0 1 0 1
Bit string of Individual 2	1 1	0 0 1 0 1	1 1	1 1 0 1 1

This crossover process is similar to the mating of process in a biological system, where parents pass segments chromosomes to their off-springs and thus off-springs can out perform their parents if they got "Qwd" genes from both parents.

Stage 3: Mutation stage (modification of offspring: New string created from existing string

The crossover process is followed by a mutation process, which introduces further changes to a bit string (new genetic material is introduced into a population, but this new material does not originated from the parents and is not introduced by crossover). This is required, since if the population does not contain all the encoded information required to solve a specific problem, no amount of gene mixing can provide a satisfactory solution. By applying the mutation operator, it is possible to produce new chromosomes (individuals). This can be implemented in various ways, and the most common technique is to change (invert) a randomly chosen bit in the string of the individual to be mutated. Thus bit 1 changed into 0 and bit 0 is changed into bit 1. The mutation is implemented with a probability of Pm, which is equal to a low is given mutation rate. The mutation rate is low, so good chromosomes are preserved (in nature mutation takes place very rarely). By using the mutation operator, it is possible to prevent the population from converging and stagnating at any local optima.

Mathematically the mutation ensures that, given any population in entire search space is connected.

Ex: By using the population mutation is as follows – 01100, 11011, 11001, 10000, and assuming that the probability of mutation is 0.0015, since there are all together 4*5 = 20 bit positions, 20*0.0015 = 0.3 bits are subjected to mutation. So no bits are changed by mutation. The generation is 01100, 11011, 11001 and 10000.

In the simple genetic algorithm mutation is the occasional (with small probability) random alternation of the value of a string position. This simply means changing a '1' to '0' and vice versa.

Step 4: Evaluate population, Goto Step 2, and end search for optimal solution, if stop criteria reached.

It can be concluded that at each iteration step, structures in the current population are evaluated and new population are formed and over a sufficiently high number of populations, the best individuals (solutions) are selected.

Stop Criteria: Stopping criteria is given in the order of

(a). The number of iterations reached without improving the current based solution.

(b). Maximum allowable number of iterations reached.

Evaluation function (Fitness function):

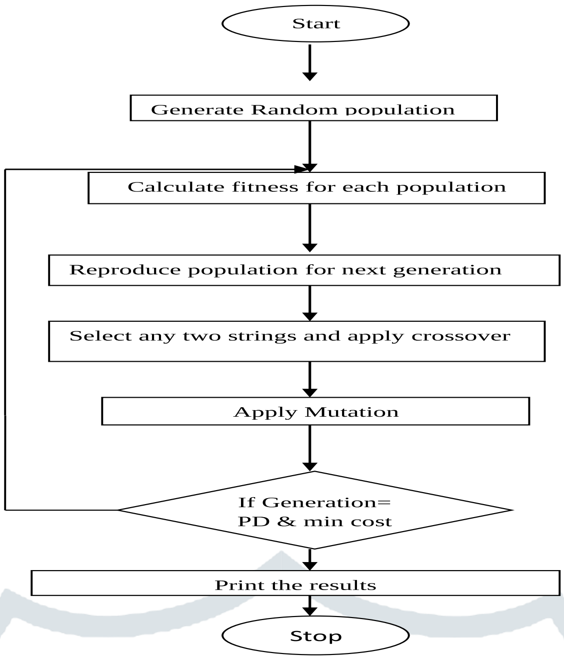
The evaluation function evaluates the 'fitness' of a chromosome as a solution to the optimization problem. It is also referred to as the "fitness function" and is maximized during the search for the global optimum solution. This function can be defined in terms of the objective function of the problem.

$$\text{Fitness Function (f)} = K / F_T$$

Where K is the maximum floating-point number is represented in the computer. It is used to amplify ($1 / F_T$), the value of which is usually small, so that the fitness values of the chromosomes will be in a wider range.

Methods	Load (MW)	Unit 1(MW)	Unit 2(MW)	U nit 3(MW)	Cost($/h)
Conventional Method	150	80	40	30	1607.9
Genetic Algorithm	150	33.2408	64.6303	54.8049	1600

4. Flow chart of GA applied to the ELD



5.TEST CASE STUDIES

Case1: The test system has three units and details of this test system are given as follows:

$F1 = 0.008P_1^2 + 7P_1 + 200$ $10 < P_1 < 85$
$F2 = 0.009P_2^2 + 6.3P_2 + 180$ $10 < P_2 < 80$
$F3 = 0.007P_3^2 + 6.8 P_3 + 140$ $10 < P_3 < 70$

B - Coefficients are:

B = [0.0218 0.0093 0.0028; 0.0093 0.022 0.0017; 0.0028 0.0017 0.0179];
B0 = [0.0003 0.0031 0.0015];
B00 = [0.00030523];

The parameters used in GA are as follows:

Population size	=	50
Generations	=	500
Time limit	=	200
Stall time limit	=	100
Crossover probability	=	0.5
Mutation probability	=	0.01

Case2: Consider the six units test system and details of this test system are given as follows:

$F_1 = 240 + 7.0 P_1 + 0.0070 P_1^2$ $100 < P < 500$
$F_2 = 200 + 10.0 P_2 + 0.0095 P_2^2$ $50 < P_2 < 200$
$F_3 = 220 + 8.5 P_3 + 0.0090 P_3^2$ $80 < P_3 < 300$
$F_4 = 200 + 11.0 P_4 + 0.0090 P_4^2$ $50 < P_4 < 150$
$F_5 = 220 + 10.5 P_5 + 0.0080 P_5^2$ $50 < P_5 < 200$
$F_6 = 190 + 12.0 P_6 + 0.0075 P_6^2$ $50 < P_6 < 120$
P_D = 1263 MW

The parameters used in GA are:

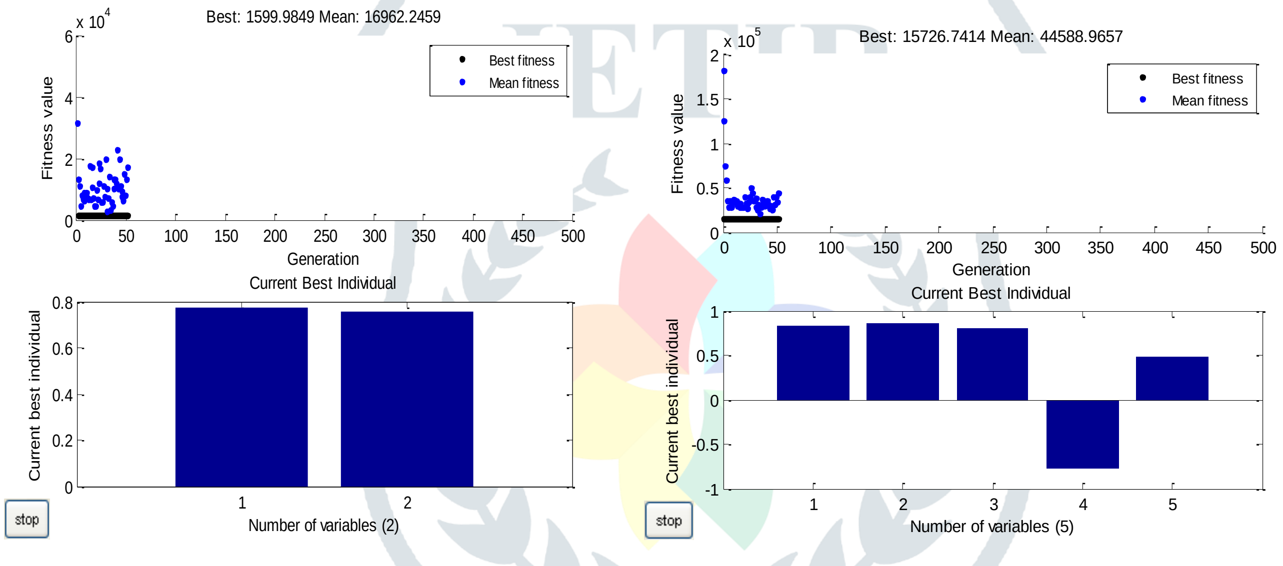
Population size	=	50
Generations	=	500
Time limit	=	200
Stall time limit	=	100
Crossover probability	=	0.5
Mutation probability	=	0.01

B=1e-4*[0.14 0.17 0.15 0.19 0.26 0.22
0.17 0.6 0.13 0.16 0.15 0.2
0.15 0.13 0.65 0.17 0.24 0.19
0.19 0.16 0.17 0.71 0.3 0.25
0.26 0.15 0.24 0.3 0.69 0.32
0.22 0.2 0.19 0.25 0.32 0.85];

RESULTS:

1. Three unit test system and Six unit test system

Method	Load (MW)	P_1 (MW)	P_2 (MW)	P_3 (MW)	P_4 (MW)	P_5 (MW)	P_6 (MW)	Cost ($/h)
Conventional Method	1263	446.7142	171.2631	264.1111	125.2222	172.125	83.6	16760.73
Genetic Algorithm	1263	477.9665	174.5534	268.1405	131.0530	165.6807	84.3175	15727



Genetic algorithm claims to provide near optimal solution for computationally intensive problems. Therefore the effectiveness of GA should always be evaluated by MATLAB was tested for three and six thermal plant systems. The performance of genetic algorithm approach is compared with conventional method. It is observed that this method is accurate and may replace effectively in the conventional practices presently performed in different central load dispatch centres.

CONCLUSION

This paper has attempted to solve economic load dispatch problem of the power system networks. The results are obtained for three and six thermal plant systems. This method shows that the fuel cost is reduced when compared with the conventional method.

FUTURE SCOPE: This method can be extended to one plant as combined cycle co-generation plant in multi thermal plant systems. Also output of genetic algorithm can be taken as input in neural network and trained in ANN tool. This method can be called as Neuro-Genetic Algorithm.

REFERENCES

[1] A.J. Wood and B.F. Wollenberg, " power generation operation and control" *John wiley and sons*, 1984.
[2] D E Goldberg and J H Holland , " Genetic algorithms in search optimization and Machine Learning " *Addison Wesley*, 1989 .
[3] Sheble G. B and Brittig K. Refined genetic algorithm -economic dispatch example. "*IEEE Trans. On power systems*, 1995 , 10(1) : 117-124.
[4] Chen P H and Chang H C large- scale economic dispatch approach by genetic algorithm ", *IEEE Trans. On power systems* 1995 , 10(4): 1919-1926.
[5] Wright A H "Genetic algorithms for real parameter optimization", *foundations of Genetic Algorithms.* J , E . Rawlins (Ed) , Morgan Kaufmann , 1991; 205-218.
[6] Wong K P and Wong Y W " Floating –point number coding method for genetic algorithms", *Proccedings of IEEE Australian and Newzealand conference on intelligent information systems, perth*, 1993, pp,512-516.
[7] Davis. L ," *Handbook of Genetic Algorithms* , van Nostrand Reinhold,1991.
[8] Yalcinoz T , Altun H and Uzam. M Economic dispatch solution using a genetic algorithm based on arithmetic crossover. *IEEE porto power Tech '2001 , port , Portuga"l*, 10-13 september 2001 . paper No. AIT1-166.