Analysis of Economic Load Dispatch Using Genetic Algorithm

Arunpreet Kaur1, Harinder Pal Singh2 and Abhishek Bhardwaj3

1Assistant Professor, Electrical Engineering Department, BUEST, Baddi- 173205, H.P., India
2Assistant Professor, Electrical Engineering, SBSSTC, Ferozpur-152004, Punjab, India
3Assistant Professor, Electrical Engineering Department, BUEST, Baddi- 173205, H.P., India

ABSTRACT

Genetic algorithm is a versatile method in the field of optimization over the past few years. This paper presents an economic load dispatch problem of thermal generator using Genetic Algorithm (GA) method. The proposed method has been applied on 3 generator system and 6 generator systems. The system considered here is the lossless system. The results obtained show a significant improvement in generator fuel cost while at the same time satisfying various equality and inequality constraints.

Keywords: Economic Load Dispatch, Genetic Algorithm.

1. INTRODUCTION

Electrical power systems are designed and operated to meet the continuous variation of power demand. In power system minimizing, the operation cost is very important. Economic Load Dispatch (ELD) is a method to schedule the power generator outputs with respect to the load demands, and to operate the power system most economically, or in other words, we can say that main objective of economic load dispatch is to allocate the optimal power generation from different units at the lowest cost possible while meeting all system constraints. Over the years, many efforts have been made to solve the ELD problem, incorporating different kinds of constraints or multiple objectives through various mathematical programming and optimization techniques. The conventional methods include Newton- Raphson method, Lambda Iteration method, Base Point and Participation Factor method, Gradient method, etc [1]. However, these classical dispatch algorithms require the incremental cost curves to be monotonically increasing or piece-wise linear [2]. The input/output characteristics of modern units are inherently highly nonlinear (with valve-point effect, rate limits etc) and having multiple local minimum points in the cost function. Their characteristics are approximated to meet the requirements of classical dispatch algorithms leading to suboptimal solutions and therefore, resulting in huge revenue loss over the time. Consideration of highly nonlinear characteristics of the units requires highly robust algorithms to avoid getting stuck at local optima [3]. The classical calculus based techniques fail in solving these types of problems. In this respect, stochastic search algorithms like genetic algorithm (GA) [4]-[9], evolutionary strategy (ES) [10]-[12], evolutionary programming (EP) [13], particle swarm optimization (PSO) [14] and simulated annealing (SA) may prove to be very efficient in solving highly nonlinear ELD problem without any restrictions on the shape of the cost curves. Although these heuristic methods do not always guarantee the global optimal solution, they generally provide a fast and reasonable solution (sub optimal or near global optimal).

2. ECONOMIC LOAD DISPATCH

2.1 Economic load dispatch

The Economic Dispatch can be defined as the process of allocating generation levels to the generating units, so that the system load is supplied entirely and most economically. For an interconnected system, it is necessary to minimize the expenses. The economic load dispatch is used to define the production level of each plant, so that the total cost of generation and transmission is minimum for a prescribed schedule of load. The objective of economic load dispatch is to minimise the overall cost of generation

2.2 Generator Operating Cost

The total cost of operation includes the fuel cost, cost of labour, supplies and maintenance. Generally, costs of labour, supplies and maintenance are fixed percentages of incoming fuel costs. The power output of fossil plants is increased sequentially by opening a set of valves to its steam turbine at the inlet. The throttling losses are large when a valve is just opened and small when it is fully opened.

Figure 1 Simple model of a fossil plant
Figure 1 shows the simple model of a fossil plant dispatching purposes. The cost is usually approximated by one or more quadratic segments. The operating cost of the plant has the form shown in Figure 2.

![Figure 2 Operating costs of a fossil fired generator]

The fuel cost curve may have a number of discontinuities. The discontinuities occur when the output power is extended by using additional boilers, steam condensers, or other equipment. They may also appear if the cost represents the operation of an entire power station, and hence cost has discontinuities on paralleling of generators. Within the continuity range the incremental fuel cost may be expressed by a number of short line segments or piece-wise linearization.

The $P_{gi}^{\text{min}}$ is the minimum loading limit below which, operating the unit proves to be uneconomical (or may be technically infeasible) and $P_{gi}^{\text{max}}$ is the maximum output limit.

### 3. FORMULATION OF ELD PROBLEM

#### 3.1 Objective Function

The objective of the economic dispatch problem is to minimize the total fuel cost of thermal power units subjected to the equality and inequality constraints of a power system. The simplified cost function of each generator can be represented as a quadratic function as given in (2).

$$F(P_{gi}) = \sum_{i=1}^{NG} F_i(P_{gi})$$

$$F_i(P_{gi}) = a_i P_{gi}^2 + b_i P_{gi} + c_i \text{ Rs/hr}$$

(1)  

(2)

Where $a_i, b_i, c_i$ are cost coefficients for $i^{th}$ unit, $F_i(P_{gi})$ is the total cost of generation, $P_{gi}$ is the generation of $i^{th}$ plant.

#### 3.2 Equality and Inequality Constraints

##### 3.1.1 Active power balance equation

For power balance, an equality constraint should be satisfied. The total generated power should be the same as total load demand plus the total transmission line loss.

$$\sum_{i=1}^{NG} P_{gi} = P_{D} + P_{\text{loss}}$$

(3)

where $P_D$ is the total load demand and $P_{\text{loss}}$ is the total line loss.

##### 3.1.2 Minimum and maximum power limits

Generation output of each generator should be lie between maximum and minimum limits. The corresponding inequality constraints for each generator are

$$P_{gi}^{\text{min}} \leq P_{gi} \leq P_{gi}^{\text{max}} \quad (i = 1, 2, \ldots, NG)$$

(4)

Where $P_{gi}^{\text{min}}$ is the lower permissible limit of real power generation, $P_{gi}^{\text{max}}$ is the upper permissible limit of real power generation.
4. GENETIC ALGORITHM

A Genetic Algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of Evolutionary Algorithms (EA) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover.

Genetic algorithms are implemented in a computer simulation in which a population of abstract representations (called chromosomes or the genotype of the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached. Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, and then improve it through repetitive application of mutation, crossover, inversion, and selection operators.

4.1 Representation

Genetic Algorithms are derived from a study of biological systems. In biological systems evolution takes place on organic devices used to encode the structure of living beings. These organic devices are known as chromosomes. A living being is only a decoded structure of the chromosomes. Natural selection is the link between chromosomes and the performance of their decoded structures. In GA, the design variables or features that characterize an individual are represented in an ordered list called a string. Each design variable corresponds to a gene and the string of genes corresponds to a chromosome. Chromosomes are made of discrete units called genes.

4.2 Encoding

Normally, a chromosome corresponds to a unique solution $x$ in the solution space. This requires a mapping mechanism between the solution space and the chromosomes. This mapping is called an encoding. In fact, GA works on the encoding of a problem, not on the problem itself. The application of a genetic algorithm to a problem starts with the encoding. The encoding specifies a mapping that transforms a possible solution to the problem into a structure containing a collection of decision variables that are relevant to the problem.

4.3 Decoding

Decoding is the process of conversion of the binary structure of the chromosomes into decimal equivalents of the feature values. Usually this process is done after de-catenation of the entire chromosome to individual chromosomes. The decoded feature values are used to compute the problem characteristics like the objective function, fitness values, constraint violation and system statistical characteristics like variance, standard deviation and rate of convergence. The stages of selection, crossover, mutation etc are repeated till some termination condition is reached.

The equivalent decimal integer of binary string $\lambda$ is obtained as

$$y^j = \sum_{i=1}^{2^J} b^j_i \quad (j = 1,2,\ldots,L)$$

(5)

Where $b^j_i$ is the $i^{th}$ binary digit of the $j^{th}$ string, $l$ is the length of the string, $L$ is the number of strings or population size.

The continuous variable $\lambda$ can be obtained to represent a point in the search space according to a fixed mapping rule, i.e.

$$\lambda^j = \lambda^{\min} + \frac{\lambda^{\max} - \lambda^{\min}}{2^l-1} y^j \quad (j = 1,2,\ldots,L)$$

(6)

Where $\lambda^{\min}$ is the minimum number of variable, $\lambda^{\max}$ is the maximum value of variable, $y^j$ is the binary coded value of the string.

4.4 Initialization

Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

4.5 Evaluation

Suitability of the solutions is determined from the initial set of solution of the problem. For this suitability determination, we use a function called fitness function. This function is derived from the objective function and used in successive genetic operation. The evaluation function is a procedure for establishing the fitness of each chromosome in the population and is very much application orientated. Since Genetic Algorithms proceed in the direction of evolving the
fittest chromosomes, and the performance is highly sensitive to the fitness values. In the case of optimization routines, the fitness is the value of the objective function to be optimized. Penalty functions can also be incorporated into the objective function, in order to achieve a constrained problem.

4.6 Fitness Function
The Genetic algorithm is based on Darwin’s principle that “The candidates, which can survive, will live, others would die”. This principal is used to find fitness value of the process for solving maximization problems. Minimization problems are usually transferred into maximization problems using some suitable transformations. Fitness value \( f(x) \) is derived from the objective function and is used in successive genetic operations. The fitness function for maximization problem can be used the same as objective function \( F(X) \).

The fitness function for the maximization problem is:

\[
f(x) = F(X)
\]  

(7)

For minimization problems, the fitness function is an equivalent maximization problem chosen such that the optimum point remains unchanged. The following fitness function is often used in minimization problems:

\[
F(X) = 1/(1 + f(x))
\]  

(8)

Here \( f(x) \) is fitness function and \( F(X) \) is objective function.

4.7 Selection
During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as this process may be very time-consuming.

Most functions are stochastic and designed so that a small proportion of less fit solutions are selected. This helps keep the diversity of the population large, preventing premature convergence on poor solutions. Popular and well-studied selection methods include roulette wheel selection and tournament selection.

4.8 Reproduction
The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation.

For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated. Although reproduction methods that are based on the use of two parents are more "biology inspired," some research suggests more than two "parents" are better to be used to reproduce a good quality chromosome.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

4.9 Termination
This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

1.) A solution is found that satisfies minimum criteria  
2.) Fixed number of generations reached  
3.) Allocated budget (computation time/money) reached  
4.) The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results  
5.) Manual inspection  
6.) Combinations of the above.

4.10 Flow Chart of Genetic Algorithm
5. PROPOSED ALGORITHM

The step-wise procedure is outlined below:

1. Read data, namely cost coefficients, $a_i$, $b_i$, $c_i$, no. of iterations, length of string, population size, probability of crossover and mutations, power demand and $P_{\text{min}}$ and $P_{\text{max}}$.
2. Create the initial population randomly in the binary form.
3. Decode the string, or obtain the decimal integer from the binary string using Eq. (5)
4. Calculate the power in MW generated from the decoded population by using Eq. (9)

\[ P_i^j = P_{\text{min}}^i + \frac{P_{\text{max}}^i - P_{\text{min}}^i}{2^L - 1} y_i^j \quad (i = 1,2,\ldots,NG, j=1,2,\ldots,L) \]  

(9)

Where $L$ is the number of strings or population size, $y_i^j$ is the binary coded value of the $i^{th}$ substring
5. Check $P_i^j$
   - If $P_i^j > P_{\text{max}}^i$, then set $P_i^j = P_{\text{max}}^i$
   - If $P_i^j < P_{\text{min}}^i$, then set $P_i^j = P_{\text{min}}^i$
6. Find fitness if \( (f_j > f_{\text{max}}) \) then set $f_{\text{max}} = f_j$ and
   - If $f_j < f_{\text{max}}$, then set $f_{\text{max}} = f_j$
7. Find population with maximum fitness and average fitness of the population.
8. Select the parents for crossover using stochastic remainder roulette wheel selection method.
9. Perform single point crossover for the selected parents.
10. Perform mutation
11. If the number of iterations reaches the maximum, then go to step 12. Otherwise, go to step 2.
12. The fitness that generates the minimum total generation cost is the solution of the problem.

6. NUMERICAL RESULTS AND DISCUSSION

The result of ELD after the implementation of proposed GA method is discussed. The programs are implemented in MATLAB 7.6.0. The performance is evaluated without considering losses using 6 generator and 3 Generator test system. The coefficients of fuel cost and maximum and minimum power limits are given in Table 1 [1], [15] and Table 3[16] respectively. The power demand is considered to be 450 (MW) and 850 (MW) respectively. The description of the results obtained by utilizing GA without considering losses is detailed herewith. The length of string in the population has been taken as 96 and 48 respectively and the population size has been assumed as 20 and 10 respectively. The single string represents three substrings, each of 16 bits. Therefore, population we have taken is 20 x 96 and 10 x 48 respectively in binary. The mutation probability is taken as 0.4. The crossover and mutation points are chosen randomly. The optimal result of GA is shown below in Table 2 and Table 4.
Table 1: Fuel Cost Coefficients for 3 generator System

<table>
<thead>
<tr>
<th>Unit No.</th>
<th>$a_i$</th>
<th>$b_i$</th>
<th>$c_i$</th>
<th>$P_{i\text{min}}$</th>
<th>$P_{i\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.001562</td>
<td>7.92</td>
<td>561</td>
<td>150</td>
<td>600</td>
</tr>
<tr>
<td>2</td>
<td>0.00194</td>
<td>7.85</td>
<td>310</td>
<td>100</td>
<td>400</td>
</tr>
<tr>
<td>3</td>
<td>0.00482</td>
<td>7.97</td>
<td>78</td>
<td>50</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 2: Optimal Result of GA for 3 generators

<table>
<thead>
<tr>
<th>$P_1$ (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>393.0103</td>
</tr>
<tr>
<td>$P_2$ (MW)</td>
</tr>
<tr>
<td>319.2256</td>
</tr>
<tr>
<td>$P_3$ (MW)</td>
</tr>
<tr>
<td>137.7642</td>
</tr>
<tr>
<td>Total Power(MW)</td>
</tr>
<tr>
<td>850.00</td>
</tr>
<tr>
<td>Total Cost(Rs/MWh)</td>
</tr>
<tr>
<td>8195.9790</td>
</tr>
</tbody>
</table>

Table 3: Fuel Cost Coefficients for 6 generator System

<table>
<thead>
<tr>
<th>Unit No.</th>
<th>$a_i$</th>
<th>$b_i$</th>
<th>$c_i$</th>
<th>$P_{i\text{min}}$</th>
<th>$P_{i\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.005</td>
<td>2.0</td>
<td>100</td>
<td>10</td>
<td>85</td>
</tr>
<tr>
<td>2</td>
<td>0.010</td>
<td>2.0</td>
<td>200</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>0.020</td>
<td>2.0</td>
<td>300</td>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>4</td>
<td>0.003</td>
<td>1.95</td>
<td>80</td>
<td>50</td>
<td>250</td>
</tr>
<tr>
<td>5</td>
<td>0.015</td>
<td>1.45</td>
<td>100</td>
<td>5</td>
<td>150</td>
</tr>
<tr>
<td>6</td>
<td>0.010</td>
<td>0.95</td>
<td>120</td>
<td>15</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 4: Optimal Result of GA for 6 generators

<table>
<thead>
<tr>
<th>$P_1$ (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>65.197850</td>
</tr>
<tr>
<td>$P_2$ (MW)</td>
</tr>
<tr>
<td>29.951560</td>
</tr>
<tr>
<td>$P_3$ (MW)</td>
</tr>
<tr>
<td>11.010200</td>
</tr>
<tr>
<td>$P_4$ (MW)</td>
</tr>
<tr>
<td>161.028600</td>
</tr>
<tr>
<td>$P_5$ (MW)</td>
</tr>
<tr>
<td>95.403810</td>
</tr>
<tr>
<td>$P_6$ (MW)</td>
</tr>
<tr>
<td>87.40800</td>
</tr>
<tr>
<td>Total Power(MW)</td>
</tr>
<tr>
<td>450.0000</td>
</tr>
<tr>
<td>Total Cost(Rs/MWh)</td>
</tr>
<tr>
<td>1971.0680</td>
</tr>
</tbody>
</table>

7. CONCLUSION
In this Study, Genetic Algorithm is successfully implemented to minimize the fuel cost of the generators on two different set of problems with two different load demands. It has been observed that genetic algorithm is capable of optimizing any kind of problems irrespective of load demand. The results discussed above are obtained after significant reduction in Fuel Cost of Generators and satisfies each and every constraint.

REFERENCES


