ISSN : 2248-9622, Vol. 4, Issue 3(Version 2), March 2014, pp.24-31

RESEARCH ARTICLE OPEN ACCESS

A Novel Approach for Economic Load Dispatch Problem Based On GA and PSO

Nishant Chaturvedi[1], A. S. Walkey[2]
[1]Research Scholar, Department of Electrical & Electronics Engineering
[2]Associate Professor, Department of Electrical & Electronics Engineering National Institute of Technical Teachers' Training and Research, Bhopal (M.P.)

Abstract
Economic load dispatch (ELD) is an important issue in the operation of power system, and several models by using different techniques have been used to solve these problems. Some traditional approaches are utilized to find out the optimal solution of non-linear problem. More recently, the soft computing techniques have received more attention and were used in a number of successful and practical applications. Genetic algorithm and particle swarm optimization are the most popular algorithms in term of optimization. The PSO techniques have drawn much attention from the power system community and been successfully applied in many complex optimization problems in power systems. This paper find out the advantages of application of Genetic algorithm (GA) and Particle Swarm Optimization (PSO) in specific to the economic load dispatch problem. Here, an attempt has been made to find out the minimum cost by using GA and PSO using the data of fifteen generating units. Comparison of both algorithm is shown here with a standard example when considering Loss and No Loss Conditions.
Keywords –Genetic algorithm, PSO, Economic Load Dispatch.

I. INTRODUCTION

The Economic Load Dispatch (ELD) problem is one of the fundamental issues in power system operation. The ELD problem involves the solution of two different problems. The first of these is the Unit Commitment or predispatch problem wherein it is required to select optimally out of the available generating sources to operate, to meet the expected load and provide a specified margin of operating reserve over a specified period of time. The second aspect of economic dispatch is the on-line economic dispatch wherein it is required to distribute the load among the generating units actually paralleled with the system in such manner as to minimize the total cost of supplying the minute-to-minute requirements of the system.

The main objective is to reduce the cost of energy production taking into account the transmission losses. While the problem can be solved easily if the incremental cost curves of the generators are assumed to be monotonically increasing piece-wise linear functions, such an approach will not be workable for nonlinear functions in practical systems. In the past decade, conventional optimization techniques such as lambda iterative method, linear programming and quadratic programming have been successfully used to solve power system optimization problems such as Unit commitment and Economic load dispatch. For highly non-linear and combinatorial optimization problems, the conventional methods are facing difficulties to locate the global optimal solution. To overcome these difficulties, some intelligent methods are used which are iterative techniques that can search not only local optimal solutions but also a global optimal solution depending on problem domain and execution time limit. They are general-purpose searching techniques based on principles inspired from the genetic and evolution mechanisms observed in natural systems and populations of living beings. These methods have the advantage of searching the solution space more thoroughly. The main difficulty is their sensitivity to the choice of parameters. Among intelligent methods, PSO is simple and promising. It requires less computation time and memory. It has also standard values for its parameters. In this paper the Particle Swarm Optimization (PSO) is proposed as a methodology for economic load dispatch. The results are compared with the traditional method i.e. Genetic Algorithm (GA).

II. FORMULATION OF ECONOMIC LOAD DISPATCH PROBLEM

Input Output Characteristic Parameters

The parameters of the input-output characteristic of any generating unit can be determined by the following approaches:

- Based on the experiments of the generating unit efficiency.

- Based on the historic records of the generating unit operation.
- Based on the design data of the generating unit provided by manufacturer.

In the Practical power systems, we can easily obtain the fuel statistic data and power output statistics data. Through analysing and computing data set (F_k, P_k), we can determine the shape of the input-output characteristic and the corresponding parameters.

A. System Constraints

Generally there are two types of constraints [1]:

1. Equality constraints
2. Inequality constraints

1. Equality Constraints

The equality constraints are the basic load flow equations of active and reactive power [1]

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

2. Inequality Constraints

Following are the inequality constraints:

i. Generator Constraints

The KVA loading of a generator can be represented as $\sqrt{P^2 + Q^2}$. The KVA loading should not exceed a pre-specified value to limit the temperature rise. The maximum active power generated 'P' from a source is also limited by thermal consideration to keep the temperature rise within limits. The minimum power generated is limited by the flame instability of the boiler. If the power generated out of a generator falls below a pre-specified value P_{min}, the unit is not put on the bus bar.

$$P_{min} \leq P \leq P_{max}$$

The maximum reactive power is limited by overheating of rotor and minimum reactive power is limited by the stability limit of machine. Hence the generator reactive powers Q should not be outside the range stated by inequality for its stable operation.

$$Q_{min} \leq Q \leq Q_{max}$$

ii. Voltage Constraints

The voltage magnitudes and phase angles at various nodes should vary within certain limits. The normal operating angle of transmission should lie between 30 to 45 degrees for transient stability reasons. A higher operating angle reduces the stability during faults and lower limit of delta assures proper utilization of the available transmission capacity.

iii. Running Spare Capacity Constraints

These constraints are required to meet:

- The forced outages of one or more alternators on the system &
- The unexpected load on the system.

The total generation should be such that in addition to meeting load demand and various losses a minimum spare capacity should be available i.e.

$$G \geq P_p + P_{so}$$

Where, G is the total generation and P_{so} is some pre-specified power. A well planned system has minimum P_{so} [1].

iv. Transmission Line Constraints

The flow of active and reactive power through the transmission line circuit is limited by the thermal capability of the circuit and is expressed as.

$$C_p \leq C_{pmax}$$

Where C_{pmax} is the maximum loading capacity of the P^{th} line [1].

v. Transformer tap settings

If an auto-transformer is used, the minimum tap setting could be zero and maximum one, i.e.

$$0 \leq t \leq 1.0$$

Similarly for a two winding transformer if tapping are provided on the secondary side,

$0 \leq t \leq n$ where n is the ratio of transformation [1].

vi. Network security constraints

If initially a system is operating satisfactorily and there is an outage, may be scheduled or forced one, it is natural that some of the constraints of the system will be violated. The complexity of these constraints (in terms of number of constraints) is enhanced when a large system is being analyzed. In this a study is to be made with outage of one branch at a time and then more than one branch at a time. The natures of the constraints are same as voltage and transmission line constraints [1].

B. Optimum Load Dispatch

The optimum load dispatch problem involves the solution of two different problems. The first of these is the unit commitment or pre dispatch problem wherein it is required to select optimally out of the available generating sources to operate to meet the expected load and provide a specified margin of operating reserve over a specified period time. The second aspect of economic dispatch is the on line economic dispatch whereas it is required to distribute load among the generating units actually paralleled with the system in such manner as to minimize the total cost of supplying the minute to minute requirements of the system. The objective of this work is to find out the solution of nonlinear on line economic dispatch problem by using PSO algorithm.

C. Cost Function

The Let$C_i$mean the cost, expressed for example in dollars per hour, of producing energy in the generator unit I. the total controllable system production cost therefore will be,

$$C = \sum_{i=1}^{N} C(i) \text{ INR/hr}$$

The generated real powerP_{Gi}accounts for the major influence onC_i. The individual real generation are raised by increasing the prime mover torques, and this requires an increased expenditure of fuel. The reactive generationsQ_{Gi}do not have any measurable influence on$C_i$because they are controlled by controlling by field current.

The individual production cost$C_i$of generators unit I is therefore for all practical purposes a function only ofP_{Gi}, and for the overall controllable production cost, we thus have,

$$C = \sum_{i=1}^{N} C(i)\, P_{Gi}$$

When the cost function$C$can be written as a sum of terms where each term depends only upon one independent variable.

III. PROPOSED METHODOLOGY

A. Genetic Algorithm

GA handles a population of possible solutions. Each solution is represented through a chromosome, which is just an abstract representation. Coding all the possible solutions into a chromosome is the first part, but certainly not the most straightforward one of a Genetic Algorithm. A set of reproduction operators has to be determined, too. Reproduction operators are applied directly on the chromosomes, and are used to perform mutations and recombination over solutions of the problem [12].
Appropriate representation and reproduction operators are really something determinant, as the behaviour of the GA is extremely dependents on it. Frequently, it can be extremely difficult to find a representation, which respects the structure of the search space and reproduction operators, which are coherent and relevant according to the properties of the problems.

Selection is supposed to be able to compare each individual in the population. Selection is done by using a fitness function. Each chromosome has an associated value corresponding to the fitness of the solution it represents. The fitness should correspond to an evaluation of how good the candidate solution is [13]. The optimal solution is the one, which maximizes the fitness function. Genetic Algorithms deal with the problems that maximize the fitness function. But, if the problem consists in minimizing a cost function, the adaptation is quite easy. Either the cost function can be transformed into a fitness function, for example by inverting it; or the selection can be adapted in such way that they consider individuals with low evaluation functions as better.

Once the reproduction and the fitness function have been properly defined, a Genetic Algorithm is evolved according to the same basic structure. It starts by generating an initial population of chromosomes. This first population must offer a wide diversity of genetic materials. The gene pool should be as large as possible so that any solution of the search space can be engendered. Generally, the initial population is generated randomly [15].Then, the genetic algorithm loops over an iteration process to make the population evolve. Each iteration consists of the following steps:

1. Evaluation

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, allowing the entire range of possible solutions. Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

2. Truncation Selection

Truncation selection is a selection method used in genetic algorithms to select potential candidate solutions for recombination.In truncation selection the candidate solutions are ordered by fitness, and some proportionof the fittest individuals are selected and reproduced 1/p times.

3. Crossover

Crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next.

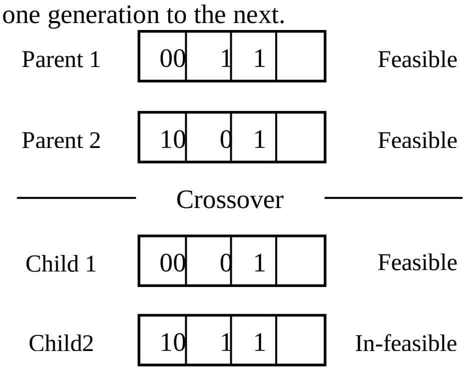

Figure 1: Crossover Operation

Assume a problem of four items has a full feasible random population. When it performs crossover using two feasible solution as parents, it generates to children, it could happen that one of it or both are not feasible as shown in figure 1.

It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. Cross over is a process of taking more than one parent solutions and producing a child solution from them [16].

4. Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation.

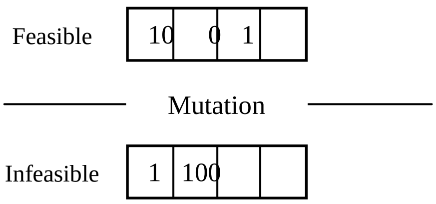

Figure 2: Mutation Operation

Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to better solution by using mutation.

The basic genetic algorithm is as follows:
-[start] Genetic random population of n chromosomes (suitable solutions for the problem)
- [Fitness] Evaluate the fitness f(x) of each chromosome x in the population
-[New population] Create a new population by repeating following steps until the New population is complete
- [selection] select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to get selected).
- [crossover] with a crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
- [Mutation] with a mutation probability, mutate new offspring at each locus (position in chromosome)
- [Accepting] Place new offspring in the new population.
- [Replace] Use new generated population for a further sum of the algorithm.
The Genetic algorithm process is discussed through the GA cycle [16]

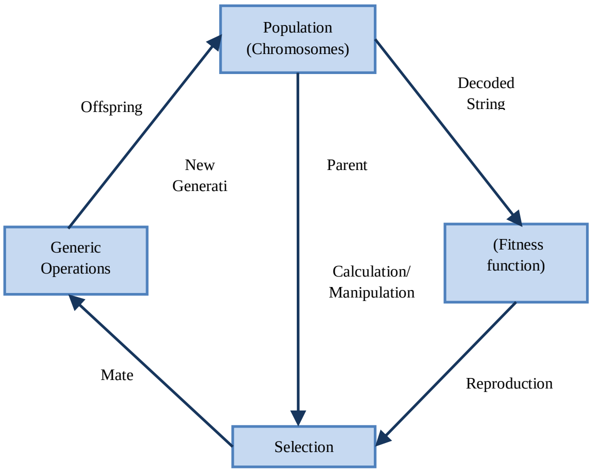

Figure 3: Genetic Algorithm cycle

Reproduction is the process by which the genetic material in two or more parent is combined to obtain one or more offspring. In fitness evaluation step, the individual's quality is assessed. Mutation is performed to one individual to produce a new version of it where some of the original genetic material has been randomly changed. Selection process helps to decide which individuals are to be used for reproduction and mutation in order to produce new search points.

B. Particle Swam Optimization

Particle Swarm Optimization (PSO) is a technique used to explore the search space of a given problem to find the settings or parameters required to maximize or minimize a particular objective.

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

This technique, first described by James Kennedy and Russell C. Eberhart in 1995, originates from two separate concepts: the idea of swarm intelligence based off the observation of swarming habits by certain kinds of animals (such as birds and fish); and the field of evolutionary computation. The PSO algorithm works by simultaneously maintaining several candidate solutions in the search space. During each iteration of the algorithm, each candidate solution is evaluated by the objective function being optimized, determining the fitness of that solution. Each candidate solution can be thought of as a particle "flying" through the fitness landscape finding the maximum or minimum of the objective function. Initially, the PSO algorithm chooses candidate solutions randomly within the search space. It should be noted that the PSO algorithm has no knowledge of the underlying objective function, and thus has no way of knowing if any of the candidate solutions are near to or far away from a local or global maximum or minimum.

The PSO algorithm simply uses the objective function to evaluate its candidate solutions, and operates upon the resultant fitness values. Each particle maintains its position, composed of the candidate solution and its evaluated fitness, and its velocity. Additionally, it remembers the best fitness value it has achieved thus far during the operation of the algorithm, referred to as the individual best fitness, and the candidate solution that achieved this fitness, referred to as the individual best position or individual best candidate solution.

Finally, the PSO algorithm maintains the best fitness value achieved among all particles in the swarm, called the global best fitness, and the candidate solution that achieved this fitness, called the global best position or global best candidate solution. The PSO algorithm consists of just three steps, which are repeated until some stopping condition is met:

1. Evaluate the fitness of each particle.
2. Update individual and global best fitness's and positions.
3. Update velocity and position of each particle.

The first two steps are fairly trivial. Fitness evaluation is conducted by supplying the candidate solution to the objective function. Individual and global best fitness and positions are updated by comparing the newly evaluated fitness against the previous individual and global best fitness, and replacing the best fitness and positions as necessary. The velocity and position update step is responsible for the optimization ability of the PSO algorithm. The velocity of each particle in the swarm is updated using the following equation:

$$v_i(t+1) = wv_i(t) + c_1r_1[\hat{x}_i(t) - x_i(t)] + c_2r_2[g(t) - x_i(t)]$$

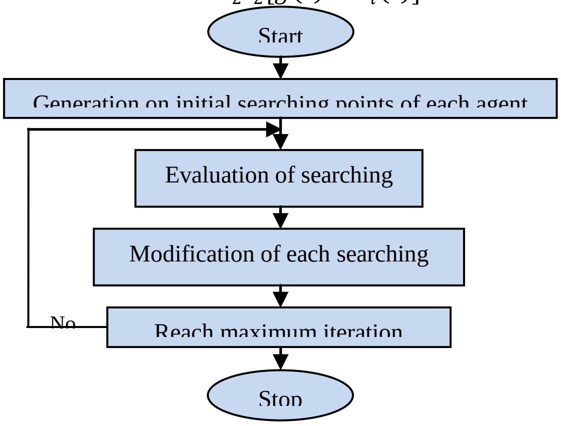

Figure 4: Flow chart of PSO

Each of the three terms of the velocity updateequation have different roles in the PSO algorithm. This process is repeated until some stopping condition is met.Somecommon stopping conditions include: a pre-set number of iterations of the PSO algorithm, a number of iterations since the last update of the global best candidate solution, or a predefined target fitness value.

IV. SIMULATION AND RESULTS

We considered a standard problem for fifteen generator system. The cost characteristic equation for all fifteen units are as given below:

UNIT 1: F1 = 0.000299*P1^2 + 10.1*P1 +671 Rs/Hr 150 MW < P1 < 455 MW
UNIT 2: F2 = 0.000183*P2^2 + 10.2*P2 + 574 Rs/Hr 150 MW < P2 < 455 MW
UNIT 3: F3 = 0.001126*P3^2 + 8.8*P3 + 374 Rs/Hr 20 MW < P3 < 130 MW
UNIT 4: F4 = 0.001126*P4^2 + 8.8*P4 + 374 Rs/Hr 20 MW < P4 < 130 MW
UNIT 5: F5 = 0.000205*P5^2 + 10.4*P5 + 461 Rs/Hr 150 MW < P5 < 470 MW
UNIT 6: F6 = 0.000301*P6^2 + 10.1*P6 + 630 Rs/Hr 135 MW < P6 < 460 MW
UNIT 7: F7 = 0.000364*P7^2 + 9.8*P7 + 548 Rs/Hr 135 MW < P7 < 465 MW
UNIT 8: F8 = 0.000338*P8^2 + 11.2*P8 + 227 Rs/Hr 60 MW < P8 < 300 MW
UNIT 9: F9 = 0.000807*P9^2 + 11.2*P9 + 173 Rs/Hr 25 MW < P9 < 162 MW
UNIT 10: F10 = 0.001203*P10^2 + 10.7*P10 + 175 Rs/Hr 25 MW < P10 < 160 MW
UNIT 11: F11 = 0.003586*P11^2 + 10.2*P11 + 186 Rs/Hr 20 MW < P11 < 80 MW
UNIT 12: F12 = 0.005513*P12^2 + 9.9*P12 + 230 Rs/Hr 20 MW < P12 < 80 MW
UNIT 13: F13 = 0.000371*P13^2 + 13.1*P13 + 225 Rs/Hr 25 MW < P13 < 85 MW
UNIT 14: F14 = 0.001929*P14^2 + 12.1*P14 + 309 Rs/Hr 15 MW < P14 < 55 MW
UNIT 15: F15 = 0.004447*P15^2 + 12.4*P15 + 323 Rs/Hr 15 MW < P15 < 55 MW

Transmission Loss B_{mn} matrix for the above equations is as follows:

$$B = \begin{bmatrix}
1.4 & 1.2 & 0.7 & -0.1 & -0.3 & -0.1 & -0.1 & -0.1 & -0.3 & -0.5 & -0.3 & -0.2 & 0.4 & 0.3 & -0.1 \\
1.2 & 1.5 & 1.3 & 0.0 & -0.5 & -0.2 & 0.0 & 0.1 & -0.2 & -0.4 & -0.4 & 0.0 & 0.4 & 1.0 & -0.2 \\
0.7 & 1.3 & 7.6 & -0.1 & -1.3 & -0.9 & -0.1 & 0.0 & -0.8 & -1.2 & -1.7 & 0.0 & -2.6 & 11.1 & -2.8 \\
-0.1 & 0.0 & -0.1 & 3.4 & -0.7 & -0.4 & 1.1 & 5.0 & 2.9 & 3.2 & -1.1 & 0.0 & 0.1 & 0.1 & -2.6 \\
-0.3 & -0.5 & -1.3 & -0.7 & 9.0 & 1.4 & -0.3 & -1.2 & -1.0 & -1.3 & 0.7 & -0.2 & -0.2 & -2.4 & -0.3 \\
-0.1 & -0.2 & -0.9 & -0.4 & 1.4 & 1.6 & 0.0 & -0.6 & -0.5 & -0.8 & 1.1 & -0.1 & -0.2 & -1.7 & 0.3 \\
-0.1 & 0.0 & -0.1 & 1.1 & -0.3 & 0.0 & 1.5 & 1.7 & 1.5 & 0.9 & -0.5 & 0.7 & 0.0 & -0.2 & -0.8 \\
-0.1 & 0.1 & 0.0 & 5.0 & -1.2 & -0.6 & 1.7 & 16.8 & 8.2 & 7.9 & -2.3 & -3.6 & 0.1 & 0.5 & -7.8 \\
-0.3 & -0.2 & -0.8 & 2.9 & -1.0 & -0.5 & 1.5 & 8.2 & 12.9 & 11.6 & -2.1 & -2.5 & 0.7 & -1.2 & -7.2 \\
-0.5 & -0.4 & -1.2 & 3.2 & -1.3 & -0.8 & 0.9 & 7.9 & 11.6 & 20.0 & -2.7 & -3.4 & 0.9 & -1.1 & -8.8 \\
-0.3 & -0.4 & -1.7 & -1.1 & 0.7 & 1.1 & -0.5 & -2.3 & -2.1 & -2.7 & 14.0 & 0.1 & 0.4 & -3.8 & 16.8 \\
-0.2 & 0.0 & 0.0 & 0.0 & -0.2 & -0.1 & 0.7 & -3.6 & -2.5 & -3.4 & 0.1 & 5.4 & -0.1 & -0.4 & 2.8 \\
0.4 & 0.4 & -2.6 & 0.1 & -0.2 & -0.2 & 0.0 & 0.1 & 0.7 & 0.9 & 0.4 & -0.1 & 10.3 & -10.1 & 2.8 \\
0.3 & 1.0 & 11.1 & 0.1 & -2.4 & -1.7 & -0.2 & 0.5 & -1.2 & -1.1 & -3.8 & -0.4 & -10.1 & 57.8 & -9.4 \\
-0.1 & -0.2 & -2.8 & -2.3 & -0.3 & 0.3 & -0.8 & -7.8 & -7.2 & -8.8 & 16.8 & 2.8 & 2.8 & -9.4 & 128.3
\end{bmatrix}$$

And the system load is 2630 MW.

A. Scenario 1: Neglecting System Loss

In this case B = 0.

On simulating our program the results we get are as follows:

For Genetic algorithm

G1	331.341
G2	351.055
G3	52.133
G4	47.497
G5	374.177
G6	380.865
G7	347.692
G8	209.330
G9	80.103
G10	89.362
G11	20.321
G12	20.300
G13	25.603
G14	15.135
G15	15.084
Cost:	**29953.6646 INR**
Loss:	**0 MW**

For Particle Swarm optimization

G1	422.069
G2	416.387
G3	130.000
G4	130.000
G5	150.000
G6	419.265
G7	465.000
G8	60.000
G9	25.000
G10	25.000
G11	21.249
G12	41.030
G13	25.000
G14	15.000
G15	15.000
Cost:	**29441.3778 INR**
Loss:	**0 MW**

B. Scenario 2: Considering System Loss

On simulating our program the results we get are as follows:

For Genetic algorithm

G1	330.556
G2	366.239
G3	46.937
G4	47.668
G5	379.599
G6	357.450
G7	366.297
G8	205.215
G9	87.093
G10	74.658
G11	20.359
G12	20.494
G13	25.863
G14	15.184
G15	16.396
Cost:	**29955.4757 INR**
Loss:	**0.0066798 MW**

For Particle Swarm optimization

G1	422.072
G2	416.384
G3	130.000
G4	130.000
G5	150.000
G6	419.271
G7	465.000
G8	60.000
G9	25.000
G10	25.000
G11	21.246
G12	41.031
G13	25.000
G14	15.000
G15	15.000
Cost:	**29441.4192 INR**
Loss:	**0.0039904 MW**

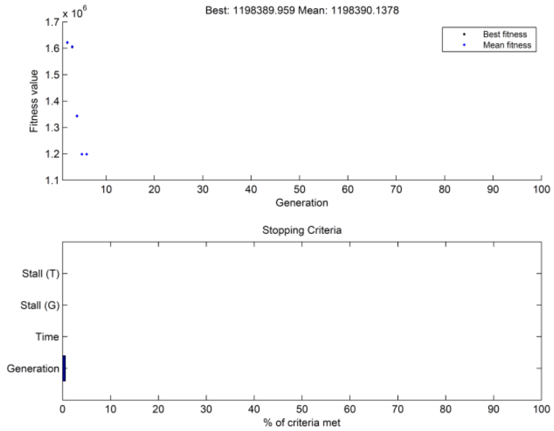

Figure 5: Convergence graph for Genetic Algorithm

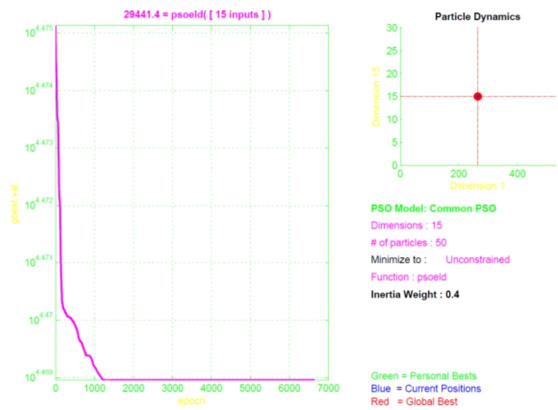

Figure 6: Convergence graph for PSO

V. CONCLUSION

Economic load dispatch in electric power sector is an important task, as it is required to supply the power at the minimum cost which aids in profit-making. As the efficiency of newly added generating units are more than the previous units the economic load dispatch has to be efficiently solved for minimizing the cost of the generated power.

In this paper both conventional GA and PSO based economic dispatch of load for generation cost reduction were comparatively investigated on two sample networks (15 generator system with loss and without loss). The results obtained were satisfactory for both approaches but it was shown that the PSO performed better than GA from the economic viewpoints. This is because of the better convergence criteria and efficient population generation of PSO.

A future recommendation can be made for GA and PSO to solve ELD problems as the use of new efficient operators to control and enhance the efficiency of instantaneous population for better and fast convergence.

REFERENCE

[1] Wadhwa, C.L, “Electrical Power System. New Age publishers”, 2009.

[2] Bishnu Sahu, Avipsa Lall, Soumya Das, T. Manoj Patra, “Economic Load Dispatch in Power System using Genetic Algorithm”, International Journal of Computer Applications (IJCA), ISSN: 0975–8887, Volume 67, No.7, April 2013.

[3] Shubham Tiwari, Ankit Kumar, G.S Chaurasia, G.S Sirohi, “Economic Load Dispatch Using Particle Swarm Optimization”, International Journal of Application or Innovation in Engineering & Management (IJAIEM), ISSN 2319 – 4847, Volume 2, Issue 4, April 2013.

[4] Kamlesh Kumar Vishwakarma, Hari Mohan Dubey, Manjaree Pandit and B.K. Panigrahi, “Simulated annealing approach for solving economic load dispatch problems with valve point loading effects”, International Journal of Engineering, Science and Technology, Vol. 4, No. 4, pp. 60-72, 2012.

[5] K. J. Vishnu Sudhan, R. S. Saravana Kumar & A. Rathina Grace Monica, “Economic Load Dispatch Problem Based Biogeography Algorithm”, Undergraduate Academic Research Journal (UARJ), ISSN: 2278 – 1129, Volume-1, Issue-1, 2012.

[6] N. Phanthuna V. Phupha N. Rugthaicharoencheep, and S. Lerdwanittip, “Economic Load Dispatch with Daily Load Patterns and Generator Constraints by Particle Swarm Optimization”, World Academy of Science, Engineering and Technology, pp. 71, 2012.

[7] Mohammad T. Ameli, Saeid Moslehpour, Massoud Pourhassan, “Educational Software for Economic Load Dispatch for Power Network of Thermal Units Considering Transmission Losses and Spinning Reserve Power”, American Society for Engineering Education (ASEE), Northeast Section Annual Conference, University of Hartford, 2011.

[8] B. Shaw, S. Ghoshal, V. Mukherjee, and S. P. Ghoshal, “Solution of Economic Load Dispatch Problems by a Novel Seeker Optimization Algorithm”, International Journal on Electrical Engineering and Informatics, Volume-3, Number-1, 2011.

[9] Y. Labbi, Ben Attous, “Hybrid GA–PS Method To Solve The Economic Load Dispatch Problem”, Journal of Theoretical and Applied Information Technology, 2010.

[10] Leandro dos Santos Coelho, Chu-Sheng Lee, “Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches”, ELSEVIER, International Journal of Electrical Power & Energy Systems, Volume 30, Issue 5, Pages 297–307, June 2008.

[11] Ismail Musirin, Nur Hazima Faezaa Ismail, Nur Hazima Faezaa Ismail, “Ant Colony Optimization (ACO) Technique in Economic Power Dispatch Problems”, Proceedings of the International Multi Conference of Engineers and Computer Scientists (IMECS), Hong Kong. Vol.-2, pp.-19-21 March, 2008.

[12] R.K. Gupta, “Genetic Algorithms-an Overview”, IMPULSE, ITM University, Vol. 1, 2006.

[13] Xavier Hue, “Genetic Algorithms for Optimisation Background and Applications”, The University of Edinburgh, Version 1.0, February 1997.

[14] http://www.saylor.org/site/wp-content/uploads/2011/06/CS411-3.1.pdf

[15] Jarmo T. Alander, “Genetic Algorithms and other “natural” optimization method to solve hard problems — a tutorial review”, Department of Electrical Engineering and Energy Technology University of Vaasa, Finland, 2012.

[16] Rohini V, “A Phased Approach to Solve the University Course Scheduling System”, International Journal of Computational Engineering Research, Vol. 03, Issue 4, 2013.

[17] Nitin S. Choubey, Madan U. Kharat, "Grammar Induction and Genetic Algorithms: An Overview", Pacific Journal of Science and Technology, Volume 10. Number 2. November 2009.

[18] Younes M, hadjeri.S, Zidi.S, Houari.S and Laarioua. M, ''Economic Power Dispatch using an Ant Colony Optimization Method", 10th International conference on Sciences and Techniques of Automatic control & computer engineering, Hammamet, Tunisia, 785-794, pp. 20-22 December, 2009.

[19] R. Chakrabarti, P. K. Chattopadhyay, M. Basu, and C. K. Panigrahi, "Particle swarm optimization technique for dynamic economic dispatch", IE (I) Journal-EL, vol. 87, pp. 48-54, 2006.

[20] B. N. S. Rahimullah, E.I. Ramlan and T.K.A. Rahman, "Evolutionary Approach for Solving Economic Dispatch in Power System", In Proceedings of the IEEE/PES National Power Engineering Conference, vol.1, pp. 32 – 36, Dec 2003.

[21] J. Rees and G.J. Koehler, "An investigation of GA performance results for different cardinality alphabets". In L.D. Davis, K. DeJong, M.D. Vose and L.D. Whitley (eds.), Evolutionary Algorithms: IMA Volumes in Mathematics and its Applications, Vol. 111. Springer-Verlag, New York, 191–206, 1999.

[22] Yuhui Shi and Russell Eberhart, "A modified particle swarm optimizer", Proceedings of the IEEE International Conference on Evolutionary Computation, pages 69–73, 1998.

[23] A Eberhart RC and Kennedy J, "A new optimizer using particle swarm theory", In Proceedings of 6th Internationl Symposium on Micro Machine and Human Science, Nagoya,Japan, IEEE Service Center, Piscataway, NJ, pp. 39–43, 1995.

[24] K.A. De Jong, W.M. Spears and D.F. Gordon, "Using Markov chains to analyze GAFOs". In D. Whitley and M. Vose (eds.), Foundations of Genetic Algorithms 3, Morgan Kaufmann, San Mateo, CA, 115–137, 1995.

[25] J.L. Shapiro, A. Prügel-Bennett and M. Rattray, "A statistical mechanics formulation of the dynamics of genetic algorithms. Lecture Notes in Computer Science, Vol. 865. Springer-Verlag, Berlin, pp. 17–27, 1994.