IOSR Journal of Engineering (IOSRJEN)
ISSN: 2250-3021 Volume 2, Issue 6 (June 2012), PP 1456-1461
www.iosrjen.org

Implementation of Artificial Intelligence Techniques for Economic Load Dispatch

K.C. Meher , A.K.Baliarsingh , A. K. Mangaraj , S.P.Rath
Department of Electrical Engineering, Orissa Engineering College, Bhubaneswar

Abstract: — *This paper presents the applications of computational intelligence techniques to economic load dispatch problems. The fuel cost equation of a thermal plant is generally expressed as continuous quadratic equation. In real situations the fuel cost equations can be discontinuous. In view of the above, both continuous and discontinuous fuel cost equations are considered in the present paper. First, genetic algorithm optimization technique is applied to a 6-generator 26-bus test system having continuous fuel cost equations. Results are compared to conventional quadratic programming method to show the superiority of the proposed computational intelligence technique. Further, a 10-generator system each with three fuel options distributed in three areas is considered and particle swarm optimization algorithm is employed to minimize the cost of generation. To show the superiority of the proposed approach, the results are compared with other published methods*

Keywords: — *Economic Load Dispatch, Continuous Fuel Cost, Quadratic Programming, Real-Coded Genetic Algorithm, Discontinuous Fuel Cost, Particle Swarm Optimization.*

I. INTRODUCTION

ECOMONIC load dispatch is defined as the process of allocating generation levels to the generating units in the mix, so that the system load is supplied entirely and most economically [1]. The objective of the economic dispatch problem is to calculate the output power of every generating unit so that all demands are satisfied at minimum cost, while satisfying different technical constraints of the network and the generators. In this problem, the generation costs are represented as curves and the overall calculation minimizes the operating cost by finding the point where the total output of the generators equals the total power that must be delivered. It is an important daily optimization task in the operation of a power system [2].

Several optimization techniques have been applied to solve the ED problem. To solve economic dispatch problem effectively, most algorithms require the incremental cost curves to be of monotonically smooth increasing nature and continuous [3-6]. For the generating units, which actually having non-monotonically incremental cost curves, the conventional method ignores or flattens out the portions of the incremental cost curve that are not continuous or monotonically increasing. Hence, inaccurate dispatch result may be obtained. To obtain accurate dispatch results, the approaches without restriction on the shape of fuel cost functions are necessary [7-8]. Most of conventional methods suffer from the convergence problem, and always get trap in the local minimum. Moreover, some techniques face the dimensionality problem especially when solving the large-scale system.

In recent years, one of the most promising research fields has been "Evolutionary Techniques", an area utilizing analogies with nature or social systems. Evolutionary techniques are finding popularity within research community as design tools and problem solvers because of their versatility and ability to optimize in complex multimodal search spaces applied to non-differentiable objective functions. Several modern heuristic tools have evolved in the last two decades that facilitate solving optimization problems that were previously difficult or impossible to solve. These tools include evolutionary computation, simulated annealing, tabu search, particle swarm, etc. Recently, genetic algorithm (GA) and particle swarm optimization (PSO) techniques appeared as promising algorithms for handling the optimization problems [9]. These techniques are finding popularity within research community as design tools and problem solvers because of their versatility and ability to optimize in complex multimodal search spaces applied to non-differentiable cost functions.

II. PROBLEM STATEMENT

The basic economic dispatch problem can described mathematically as a minimization of problem of minimizing the total fuel cost of all committed plants subject to the constraints [1].

$$\min \quad F = \sum_{i=1}^{N} F_i(P_i) \qquad (1)$$

Subject to the constraints

$$\sum_{i=1}^{N} P_i - P_D - P_L = 0 \quad (2)$$

$$P_{i\min} \le P_i \le P_{i\max},\ i = 1,2,\ldots\ldots,N \quad (3)$$

Where

F = Total operating cost

N = Number of generating units

P_i = Power output of i th generating unit

$F_i\,(P_i)$ = Fuel cost function of i th generating unit

P_D = Total load demand

P_L = Total losses

$P_{i\,min}$ = Minimum out put power limit of i th generating unit

$P_{i\,max}$ = Maximum out put power limit of i th generating unit

The total fuel cost is to be minimized subject to the constraints. The transmission loss can be determined form B_{mn} coefficients.

The conditions for optimality can be obtained by using Lagrangian multipliers method and Kuhn tucker conditions as follows:

$$2a_i P_i + b_i = \lambda(1 - 2\sum_{j=1}^{N} B_{ij}),\ i = 1,2,\ldots\ldots,N \quad (4)$$

With the following constraints

$$\sum_{i=1}^{N} P_i = P_D + P_L \quad (5)$$

$$P_L = \sum_{i=1}^{N} \sum_{i=1}^{N} P_i B_{ij} P_j \quad (6)$$

$$P_{i\min} \le P_i \le P_{i\max} \quad (7)$$

The following steps are followed to solve the economic load dispatch problem with the constraints:

Step-1:

Allocate lower limit of each plant as generation, evaluate the transmission loss and incremental loss coefficients and update the demand.

$$P_i = P_{i\min},\ X_i = 1 - \sum_{j=1}^{N} P_i B_{ij},\ P_D{}^{new} = P_D + P_L{}^{old} \quad (8)$$

Step-2:

Apply quadratic programming to determine the allocation P_i^{new} of each plant.

If the generation hits the limit, it should be fixed to that limit and the remaining plants only should be considered for next iteration.

Step-3:

Check for the convergence

$$\left|\sum_{i=1}^{N} P_i - P_D{}^{new} - P_L\right| \le \varepsilon \quad (9)$$

Where ε is the tolerance. Repeat until the convergence criteria is meet.

A brief description about the quadratic programming method is presented in the next section.

III. PARTICLE SWARM OPTIMIZATION APPROACH

A. Overview of Particle Swarm Optimization

The PSO method is a member of wide category of swarm intelligence methods for solving the optimization problems. It is a population based search algorithm where each individual is referred to as particle and represents a candidate solution. Each particle in PSO flies through the search space with an adaptable velocity that is dynamically modified according to its own flying experience and also to the flying experience of the other particles. In PSO each particles strive to improve themselves by imitating traits from their successful peers. Further, each particle has a memory and hence it is capable of remembering the best position in the search space ever visited by it. The position corresponding to the best fitness is known as *pbest* and the overall best out of all the particles in the population is called *gbest* [15-16].

The modified velocity and position of each particle can be calculated using the current velocity and the distances from the $pbest_{j,g}$ to $gbest_g$ as shown in the following formulas [11, 17-20]:

$$v_{j,g}^{(t+1)} = w * v_{j,g}^{(t)} + c_1 * r_1(\ ) * (pbest_{j,g} - x_{j,g}^{(t)}) + c_2 * r_2(\ ) * (gbest_g - x_{j,g}^{(t)}) \quad (21)$$

$$x_{j,g}^{(t+1)} = x_{j,g}^{(t)} + v_{j,g}^{(t+1)} \quad (22)$$

With $j = 1,2,\ldots,n$ and $g = 1,2,\ldots,m$

where,

n = number of particles in the swarm
m = number of components for the vectors v_j and x_j
t = number of iterations (generations)
$v_{j,g}^{(t)}$ = the g-th component of the velocity of particle j at iteration t , $v_g^{\min} \le v_{j,g}^{(t)} \le v_g^{\max}$;
w = inertia weight factor
c_1, c_2 = cognitive and social acceleration factors respectively
r_1, r_2 = random numbers uniformly distributed in the range (0, 1)
$x_{j,g}^{(t)}$ = the g-th component of the position of particle j at iteration t
$pbest_j$ = *pbest* of particle j
$gbest$ = *gbest* of the group

The j-th particle in the swarm is represented by a d-dimensional vector $x_j = (x_{j,1}, x_{j,2}, \ldots\ldots, x_{j,d})$ and its rate of position change (velocity) is denoted by another d-dimensional vector $v_j = (v_{j,1}, v_{j,2}, \ldots\ldots, v_{j,d})$. The best previous position of the j-th particle is represented as $pbest_j = (pbest_{j,1}, pbest_{j,2}, \ldots\ldots, pbest_{j,d})$. The index of best particle among all of the particles in the swarm is represented by the $gbest_g$. In PSO, each particle moves in the search space with a velocity according to its own previous best solution and its group's previous best solution. The velocity update in a PSO consists of three parts; namely momentum, cognitive and social parts. The balance among these parts determines the performance of a PSO algorithm. The parameters c_1 and c_2 determine the relative pull of *pbest* and *gbest* and the parameters r_1 and r_2 help in stochastically varying these pulls. In the above equations, superscripts denote the iteration number.

B. Parameter Selection for PSO

TABLE II: PARAMETERS USED IN PSO

Parameter	Value/Type
Maximum generations	150
Swarm size	10
Cognitive factors (c_1) & social acceleration factors (c_2)	c_1 =2.0 c_2=2.0
Inertia weights	w_{start}, =0.8

For the implementation of PSO, several parameters are required to be specified, such as c_1 and c_2 (cognitive and social acceleration factors, respectively), initial inertia weights, swarm size, and stopping criteria. These parameters should be selected carefully for efficient performance of PSO. The constants c_1 and c_2 represent the weighting of the stochastic acceleration terms that pull each particle toward *pbest* and *gbest* positions. Low values allow particles to roam far from the target regions before being tugged back. On the other hand, high values result in abrupt movement toward, or past, target regions. Hence, the acceleration constants were often set to be 2.0 according to past experiences. Suitable selection of inertia weight, w, provides a balance between global and local explorations, thus requiring less iteration on average to find a sufficiently optimal solution. As originally developed, w often decreases linearly from about 0.9 to 0.4 during a run [11, 15-20]. The parameters employed for the implementations of PSO in the present study are given in Table II.

IV. RESULTS AND DISCUSSIONS

A. Numerical Example 1

First, continuous quadratic cost curve for the plants is considered. The system consists of 26 bus, 6 units, and the demand of the system was divided into 12 small intervals as shown in Fig. 1. Generating units' data are given in Table 3.1. The cost function coefficients along with minimum and maximum generation capacity for each fuel option are given in Table III. Table IV, shows the optimal generators' power outputs for each hour including their corresponding fuel costs using quadratic programming method. Total production cost of 12 intervals is $156065.8. Table V, shows the same using RCGA method. Total production cost of 12 intervals is $151008. It is clear from Table IV and V that RCGA gives better solutions.

TABLE III: DATA FOR EXAMPLE - 1: 26-BUS 6-UNIT TEST SYSTEM

Unit/Cost	a ($\$/MW^2h$)	b ($/MWh)	c ($/h)	P_{min} (MW)	P_{max} (MW)
Unit-1	0.007	7	240	100	500
Unit-2	0.0095	10	200	50	200
Unit-3	0.009	8.5	220	80	300
Unit-4	0.009	11	200	50	150
Unit-5	0.008	10.5	220	50	200
Unit-6	0.0075	12	120	50	120

TABLE I IV: RESULTS OF QUADRATIC PROGRAMMING FOR EXAMPLE - 1: 26-BUS, 6-UNIT TEST SYSTEM

U/T	1	2	3	4	5	6	7	8	9	10	11	12
P_{g1}	350.315	363.153	399.336	401.934	436.308	462.628	467.039	464.833	396.740	378.617	352.879	381.200
P_{g2}	102.124	111.506	137.931	139.827	164.950	184.186	187.408	185.796	136.036	122.803	103.998	124.691
P_{g3}	183.725	193.286	220.174	222.101	247.633	267.157	270.426	268.791	218.247	204.788	185.636	206.708
P_{g4}	51.3537	60.8204	87.4125	89.3168	114.364	133.424	136.611	135.017	85.5089	72.2016	53.2457	74.1007
P_{g5}	84.4816	94.9211	124.221	126.317	153.650	174.333	177.787	176.06	122.124	107.465	86.5684	109.558
P_{g6}	50.00	50.00	50.00	50.00	69.6387	91.3661	94.9925	93.1791	50.00	50.00	50	50.00
Total cost in $	9902.8	10561.0	12479.0	12621.0	14812.0	16657.0	16972.0	16814.0	12339.0	11369.0	10033.0	11506.0

TABLE I V: RESULTS OF RCGA FOR EXAMPLE - 1: 26-BUS, 6-UNIT TEST SYSTEM

U/T	1	2	3	4	5	6	7	8	9	10	11	12
P_{g1}	359.803	373.383	412.102	415.102	450.27	478.807	483.107	481.75	409.134	388.793	363.135	393.862
P_{g2}	102.452	112.193	136.475	141.599	165.356	185.637	189.781	186.969	137.331	125.539	104.607	125.046
P_{g3}	185.283	196.693	222.84	222.39	249.633	270.131	271.469	268.597	218.677	204.206	186.459	107.169
P_{g4}	51.7959	60.2026	89.0846	88.6271	113.736	133.7	136.518	136.848	85.5349	73.2771	53.2834	73.4675
P_{g5}	80.8693	89.78	118.211	120.888	151.342	169.221	172.486	171.657	117.405	102.651	82.9496	105.735
P_{g6}	51.1326	51.1185	51.1152	51.7013	68.4418	88.9866	94.5453	91.3401	51.2375	51.5198	51.2862	51.1615
Total cost $	9683.1	10307.1	12111.1	12244.1	14278.1	15970.1	16258.1	16114.1	11980.1	11070.1	9807.1	11198.1

TABLE I VI: RESULTS OF RCGA FOR EXAMPLE - 2: 10-UNIT NEW ENGLAND TEST SYSTEM

Unit/Load	2400	2450	2500	2550	2600	2650	2700
1	188.6405	192.9906	205.419	210.4316	215.4442	212.919	217.1499
2	201.2427	203.4997	205.3573	207.5815	209.8058	208.6852	210.5626
3	252.7953	259.292	264.6391	271.0416	277.4441	274.2187	279.6228
4	231.9456	233.5405	234.8531	236.4249	237.9967	237.2049	238.5315
5	240.7297	249.6094	256.9177	265.6686	274.4194	270.011	277.3973
6	231.9456	233.5405	234.8531	236.4249	237.9967	237.2049	238.5315
7	252.175	260.7258	267.7635	276.1903	284.617	280.3718	287.4845
8	231.9456	233.5405	234.8531	236.4249	237.9967	237.2049	238.5315
9	319.2832	325.3744	330.3877	336.3906	342.3934	414.5581	427.4216
10	238.2969	246.8866	253.9562	262.4212	270.8861	266.6217	273.7667
Total cost	479.9326	501.8185	524.6588	548.2634	572.8008	597.3015	622.2292

V. CONCLUSION

This paper presents the applications of computational intelligence techniques to economic load dispatch problems considering both continuous and discontinuous fuel cost functions. First, a continuous fuel cost function is considered for a 26 bus, 6 unit test system and both conventional (quadratic programming method) and computational intelligence (real coded genetic algorithm) methods are applied to find the optimum generator allocation. It is seen that the results obtained by the computational intelligence method is better compared to the quadratic programming method. Further, a discontinuous fuel cost function is considered for a 10 unit New England test system and another computational intelligence technique (particle swarm optimization) is applied to find the optimum generator allocations. The results are compared with other published methods to show its superiority.

REFERENCE

[1.] J. Wood and B. F. Wollenberg, "Power Generation Operation and Control," 2nd edition, New York: Willey, 1996.

[2.] B. H. Chowdhury and S. Rahman, "A review of recent advances in economic dispatch," *IEEE Transactions on Power Systems*, vol. 5, no. 4, pp. 1248-1259, November 1990.

[3.] A. Jiang and S. Ertem, "Economic dispatch with non-monotonically increasing incremental cost units and transmission system losses",*IEEE Transactions on Power Systems*, vol. 10, no. 2, pp. 891-897, May 1995.

[4.] H.W. Dommel, "Optimal power dispatch", *IEEE Transactions on Power Apparatus and Systems*, PAS93 No. 3, pp. 820–830, 1974.

[5.] C.O. Alsac, J. Bright, M. Paris, and Stott, "Developments in LP-based optimal power flow, *IEEE Transaction of Power Systems*", Vol. 5 No. 3, pp. 697–711, 1990.

[6.] J. Nanda, D.P. Kothari, S.C. Srivastava, "New optimal power-dispatch algorithm using fletcher's quadratic programming method", *IEE Proceedings*, Vol. 136 No. 3, pp. 153–161, 1989.

[7.] H. T. Yang, P. C. Yang and C. L. Huang, "Evolutionary Programming Based Economic Dispatch For Units With Non-smooth Fuel Cost Functions," IEEE Transactions on Power Systems, Vol. 11, No. 1, pp. 112-118, 1996.

[8.] T. Jayabarathi, G. Sadasivam and V. Ramachandran, "Evolutionary programming based economic dispatch of generators with prohibited operating zones," Electric Power Systems Research, Vol. 52, No. 3, pp. 261-266, 1999.

[9.] Sidhartha Panda and N. P. Padhy, "Comparison of Particle Swarm Optimization and Genetic Algorithm for FACTS-based Controller Design", *Applied Soft Computing*. vol. 8, issue 4, pp. 1418-1427, 2008.

[10.] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning". Addison-Wesley, 1989.

[11.] Sidhartha Panda, S.C. Swain, P.K. Rautray, R. Mallik, G. Panda, "Design and analysis of SSSC-based supplementary damping controller", *Simulation Modelling Practice and Theory,* doi: 10.1016/j.simpat.2010.04.007.

[12.] Sidhartha Panda, C. Ardil, "Real-coded genetic algorithm for robust power system stabilizer design", *International Journal of Electrical, Computers and System Engineering*, vol. 2, no. 1, pp. 6-14, 2008.