Research Article

Economic Load Dispatch Problem using Particle Swarm Optimization with Inertial Weight and Constriction Factor

Vinod Puri*, Yogesh K. Chauhan and Nidhi Singh
School of Engineering, Gautam Buddha University, Greater Noida, India

Abstract

The Economic Load dispatch (ELD) problem plays a significant role in the functioning of power systems. It is used in determining the optimal cost for satisfying the demand with available electric generation resources. The ELD problem of thermal units results in significant saving for electrical utilities. The formation of ELD has been discussed and solution is obtained by Particle swarm optimization (PSO). The algorithms based on various PSO variants such as inclusion of inertia weight (named as I-PSO); constriction factor (C-PSO) and constriction factor-inertia weight (CI-PSO) have been developed and have been tested on systems comprising four and six units. The results obtained with the proposed PSO techniques are matched with the existing results, which thereby show the correctness and effectiveness of the developed algorithm.

Keywords: Particle swarm optimization; economic load dispatch; power system; optimization; etc.

1. Introduction

The economic load dispatch problem plays an increasingly important role in utility industries to meet the demand while satisfying constraints imposed by different units. Different classical techniques have been used to find the optimal solution, e.g. lambda iteration method, dynamic programming, lagrangian relaxation method, etc. [1]. However, classical techniques become very complicated when dealing with increasingly complex problems [2]. They also suffer from numerical convergence and solution quality problems. The stochastic search algorithms such as particle swarm optimization, genetic algorithms, evolutionary programming, simulated annealing and ant-colony optimization can easily handle complex problems and provide high quality solutions [3]. Due to simplicity and less parameter tuning, particle swarm optimization is used to solve the economic load dispatch problem [4]. In this paper, the algorithm for solving the ELD problem has been formulated using IPSO, CPSO and CIPSO [5].

2. Formulation of Economic Load Dispatch Problem

The objective of economic dispatch is to determine the optimal share of load demand for each unit in the range of 3 to 5 minutes subjected to equality and inequality constraints [6]. The ELD problem can be expressed as

$$Min \sum_{i=1}^{n} F_i(P_i) \quad (1)$$

$$F_i(P_i) = (a_i + b_i \times P_i + c_i \times P_i^2) \quad (2)$$

Where a_i, b_i, c_i are the unit cost cofficient of the i^{th} generator and n is the number of generators committed to the operating system

*Correspondence : vinod_tu24@yahoo.com DOI 10.14456/tijsat.2016.14

[7]. The economic dispatch is subject to following constraints:
Inequality Constraints

$$P_{min,i} \leq P_i \leq P_{max,i} \quad (3)$$

Foro i =1,2,3……n
Equality Constraints

$$\sum_{i=1}^{n} P_i - D - L = 0 \quad (4)$$

$$L = \sum_{i=1}^{n} B_i P_i^2 \quad (5)$$

Where D is the load demand and L represents the transmission losses. B represents cofficients of transmission losses. $P_{min,i}$ and $P_{max,i}$ are minimum and maximum generation output of the i^{th} generator; for simplicity, the losses have been neglected and the problem is formulated without losses [8].

$$\sum_{i=1}^{n} (P_i - D) = 0 \quad (6)$$

3. Particle Swarm Optimization

Particle swarm optimization is a stochastic, population-based search and optimization algorithm for problem solving. It is a kind of swarm intelligence that is based on social-psychological principles and provides insights into social behaviour, as well as contributing to engineering applications. The particle swarm optimization algorithm was first described in 1995 by *James Kennedy and Russell C. Eberhart* [9]. The techniques have evolved greatly since then, and the original version of the algorithm is barely used at present. Social influence and social learning enable a person to maintain cognitive consistency. People solve problems by talking with other people about them and, as they interact, their beliefs, attitudes, and behaviour changes. The changes could typically be depicted as the individuals moving toward one another in a socio-cognitive space[10]. The basic PSO algorithm is studied as follows:

1. Initialize the swarm, p(t), of particles such that the position xi(t) of each particle . p(t) is random within the hyperspace, with t = 0.
2. Evaluate the fitness function for each particle and find the pbest.
3. For each individual particle, compare the particle's fitness value with its p_{best}. If the current value is better than the p_{best} value, then set this value as the p_{best} and the current particle's position, xi, as pi.
4. Identify the particle that has the best fitness value. The value of its fitness function is identified as gbest and its position as pi.
5. Update the velocities and positions of all the particles.

$$v_i(t) = v_i(t-l) + C_i\left(x_{pbest} - x_i(t)\right) + C_2\left(x_{gbest} - x_i(t)\right) \quad (7)$$

Where C_1 and C_2 are random variables, the second term above is referred to as the cognitive component, while the last term is the social component [11].

$$x_i(t) = x_i(t-1) + v_i(t) \quad (8)$$

3.1 PSO with Inertia weight (I-PSO)

To control the global and local exploration capabilities of the particle, a large inertia weight factor is used during initial exploration and its value is gradually reduced as the search proceeds [12]. The concept of time varying inertial weight is given by

$$v_i(t) = w \times v_i(t-1) + C_1 \times \left(x_{pbest} - x_i(t)\right) + C_2 \times \left(x_{gbest} - x_i(t)\right) \quad (9)$$

Where, w is the inertia weight.

$$w = \left((w_{max} - w_{\min}) \times \left(\frac{iter_{max} - iter}{iter_{max}}\right) + w_{min}\right) \quad (10)$$

w_{max}= 0.9 ; $w_{\min}$=0.4

where $iter_{max}$, is the maximum number of iterations.

3.2 PSO with constriction factor (C-PSO)

To improve the convergence of PSO algorithm, a constriction factor (K) is introduced [13].

$$v_i(t) = K \times \left(v_i(t-1) + C_1 \times \left(x_{pbest} - x_i(t)\right) + C_2 \times \left(x_{gbest} - x_i(t)\right)\right) \quad (11)$$

Where $K = \frac{2}{|2-\emptyset-\sqrt{(\emptyset^2-4)}|}$

where $4.1 \leq \emptyset \leq 4.2$

also $\emptyset \geq C_1 + C_2$

As Ø increases, the factor K decreases and convergence become slower because population diversity is reduced [14].

3.3. PSO with inertia weight and constriction factor (CI-PSO)

To combine both the improvement in the PSO algorithm where inertia weight and constriction factor both are used to update the velocity of the particle[15].

$$v_i(t) = K \times \left(w \times v_i(t-1) + C_1 \times \left(x_{pbest} - x_i(t)\right) + C_2 \times \left(x_{gbest} - x_i(t)\right)\right) \quad (12)$$

Where K and w are constriction factor and inertia weight, respectively.

4. Economic Dispatch Using PSO

Our main aim is to minimise the operating time and optimization is done by proposed PSO algorithms [16].

The following steps are used by the PSO technique to solve the economic dispatch problem. The flow chart of the above mentioned steps is developed and shown in Figure 1

1. Initialize a population of particles p_i and other variables. Each particle is usually generated randomly within the allowable range.

$$P_{min,i} \leq P_i \leq P_{max,i} \quad (13)$$

Here P_i represented as i^{th} unit in the power system.

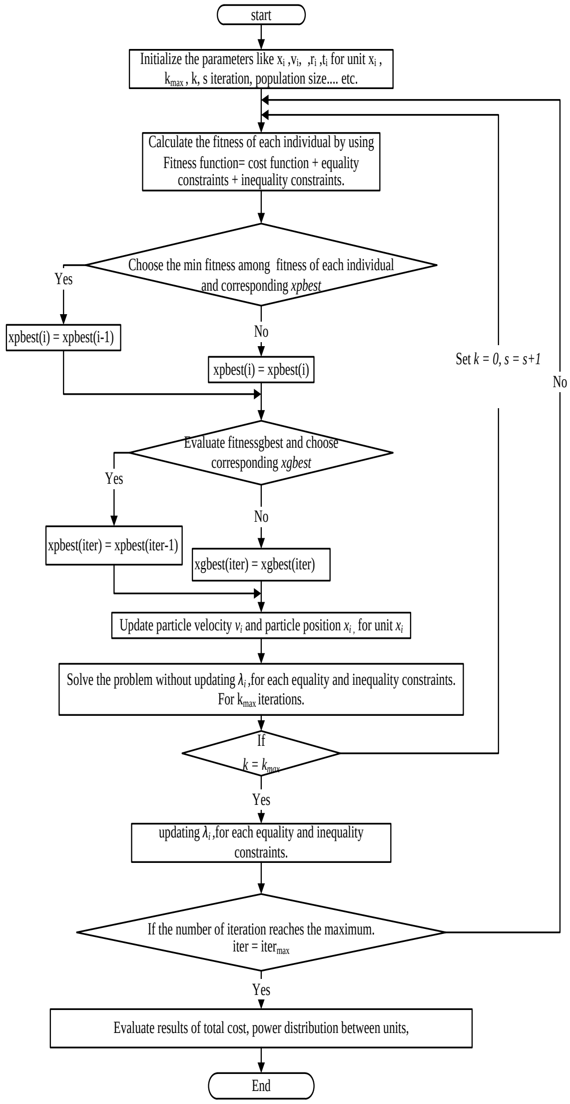

Figure 1. Economic Load Dispatch Using Particle Swarm Optimization.

2. Initialize the parameters such as the size of population, initial and final inertia weight, random velocity of particle, acceleration constant, the max generation, Lagrange's multiplier(λ_i), etc.

3. Calculate the fitness of each individual in the population using the fitness function or cost function.

$$F_T\left(\sum_{t=1}^{T}\sum_{i=1}^{n} F_i(P_i) + Equa.constraints + Inequa.Constraints\right) \quad (14)$$

Where $F_i(P_i)$ is represented as

$F_i(P_i) = (a_i + b_i \times P_i + c_i \times P_i^2)$ (15)
With equality constraints as
$\sum_{i=1}^{n}(P_i - D) = 0$ (16)
Where P_i is the i^{th} generator and P_D is the load or demand. And inequality constraints as
$P_{min,i} \leq P_i \leq P_{max,i}$ (17)
4. Update each individual unit's position and velocity; also update α_i and λ_i corresponding to equality and inequality constraints.
5. If the number of iterations reaches the maximum then go to step 6. Otherwise go to step 3.
6. The individual that generates the latest is the optimal generation power of each unit with the minimum total generation cost [17,18].

5. Simulation Results

The proposed methods (I-PSO, C-PSO and CI-PSO) have been implemented in Matlab and tested on the following test system comprising four and six units. The data for the respective test systems are given below.

5.1 Test system1

The four generating units considered have different characteristics. Their cost function characteristics are given by the following equations:

$F_1 = (0.0018P_3^2+20.74P_3+231)$ Rs/H
$F_2 = (0.0042P_2^2+16.95P_2+585.62)$ Rs/Hr
$F_3 = (0.0021P_1^2+16.83P_1+684.74)$ Rs/Hr
$F_4 = (0.0034P_4^2+23.60P_4+252)$ Rs/Hr (18)

In this problem the active power constraints are considered. The operating limit of maximum and minimum power are also different. The unit operating ranges are

80 (MW) $\leq P_1 \leq$ 25 (MW)
250 (MW) $\leq P_2 \leq$ 60 (MW)
300 (MW) $\leq P_3 \leq$ 75 (MW)
60 (MW) $\leq P_4 \leq$ 20 (MW) (19)

The load pattern for the given test system is [450, 530, 600, 800, 540, 400, 280, 290, 500]

5.2 Test system2

The six generating units considered have different characteristics. Their cost function characteristics are given by following equations:

$F_1 = (0.001562P_1^2+7.92P_1+561.0)$ Rs
$F_2 = (0.00194P_2^2+7.85P_2+310.0)$ Rs/Hr
$F_3 = (0.00482P_3^2+7.97P_3+78.0)$ Rs/Hr
$F_4 = (0.00139P_4^2+7.06P_4+500.0)$ Rs/Hr
$F5 = (0.00184P_5^2+7.46P_5+ 295.0)$ Rs
$F_6 = (0.00184P_6^2+7.46P_6+252)$ Rs/Hr (20)

In this problem the active power constraints are considered. The operating limit of maximum and minimum power are also different. The unit operating ranges are

100 (MW) $\leq P_1 \leq$ 600 (MW)
100 (MW) $\leq P_2 \leq$ 400 (MW)
50 (MW) $\leq P_3 \leq$ 200 (MW)
140 (MW) $\leq P_4 \leq$ 590 (MW)
110 (MW) $\leq P_5 \leq$ 440 (MW)
110 (MW) $\leq P_6 \leq$ 440 (MW) (21)

The load pattern for the given test system is [800, 1200, 1520, 2238, 3150, 3800, 4600, 1800]

6. Results

The results of the two test systems, i.e. test system 1 and test system 2, are given in Table 1 and Table 2, respectively. In Table 1 and Table 2, the results of the proposed methods of PSO i.e. I-PSO, C-PSO and CI-PSO have been evaluated for four and six units, respectively. The units with different characteristics are loaded such that minimum cost is obtained. It has been seen that the different methods compute the cost with negligible amount of improvement in the results. The results also show that among different proposed algorithms the CI-PSO converges fastest as compared to I-PSO and C-PSO. In Figure 2 the I-PSO method has been used for a 450 (MW) load. It can be seen that the curve optimizes to its optimal value within 66 iterations. In Figure 3 the C-PSO method has been used for the same load and the curve optimizes to its optimal value within 40 iterations. Similarly, in Figure 4 the CI-PSO has been used and the curve optimizes to its optimal value within 22 iterations. The same analysis has been carried out for the results of Table 2 where the proposed algorithms have been used for six units and it can again be seen that the CI-PSO converges fastest as compared to C-PSO and I-PSO. Figures 5, 6, and 7 confirm these results. Figure 5 shows the cost curve of the 800(MW) load by I-PSO. Figure 6 shows the cost curve for the given load by C-PSO and Figure 7 shows the cost curve for the given load by CI-PSO. It is seen that the number of iterations for the corresponding load is different and that the CI-PSO converges to its optimal value with fewer iterations than algorithms like I-PSO and C-PSO. Iterations(ITER)= 100, Particles=50, C1=2.05, C2=2.05, wmax= 0.9,wmin = 0.4, Upper Limit=1000, Lower Limit= (-)1000

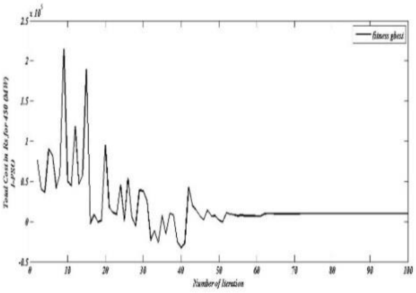

Figure 2. Cost optimizing Curve for 450 (MW) by (I-PSO).

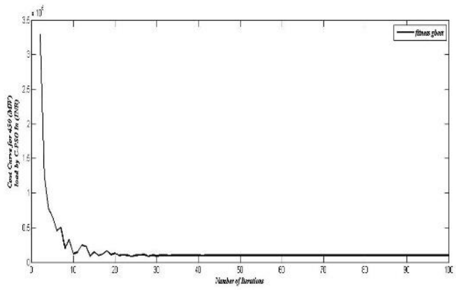

Figure 3. Cost optimizing Curve for 450 (MW) by (C-PSO).

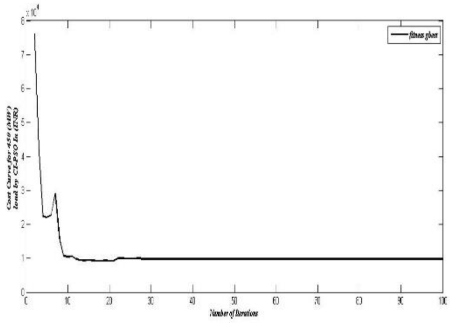

Figure 4. Cost optimizing Curve for 450 (MW) by (CI-PSO).

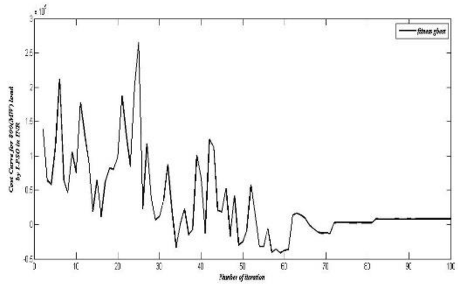

Figure 5. Cost optimizing Curve for 800(MW) by (I-PSO).

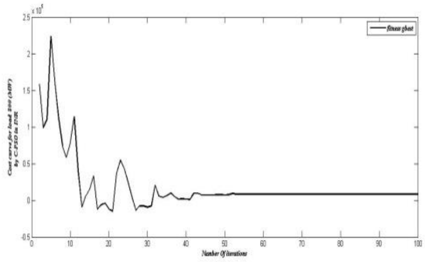

Figure 6. Cost optimizing Curve for 800 (MW) by (C-PSO).

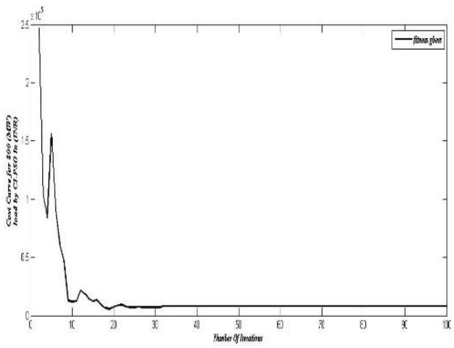

Figure 7. Cost optimizing Curve for 800 (MW) by (CI-PSO).

Table 1. The Results Of Economic Dispatch For 4 Units Using PSO Algorithms (I-PSO,C-PSO,CI-PSO).

S.NO	LOAD	PSO TECH	UNIT1	UNIT2	UNIT3	UNIT4	COST	ITER	CPU TIME (SEC)
1	**450**	**I-PSO (Iteration 1)**	24.75	135.43	270.38	19.64	9790.11	66	0.50664
		I-PSO (Iteration 2)	24.56	232.0996	173.7566	19.52	9861.46	60	0.508917
		C-PSO(Iteration 1)	24.97	124.63	280.13	20.04	9789.72	40	0.392135
		C-PSO(Iteration 2)	24.91	128.95	275.45	20.38	9789.46	42	0.5235
		CI-PSO(Iteration1)	25.00	125.06	279.93	19.99	9789.75	22	0.517252
		CI-PSO(Iteration2)	24.81	182.8674	222.6681	19.67	9811.37	20	0.5027
2	**530**	**I-PSO(Iteration 1)**	24.37	184.73	300.24	20.05	11245.19	54	0.510805
		I-PSO(Iteration 2)	25.56	230.75	250.99	21.94	11296.21	61	0.5163
		C-PSO(Iteration 1)	25.14	184.65	300.06	20.02	11245.80	42	0.39842
		C-PSO(Iteration 2)	25.20	184.95	299.94	19.84	11246.54	42	0.3982
		CI-PSO(Iteration1)	25.12	184.85	299.98	20.01	11245.85	21	0.403727
		CI-PSO(Iteration2)	24.62	185.866	298.93	20.32	11248.71	22	0.5845
3	**600**	**I-PSO(Iteration 1)**	29.65	249.97	300.22	20.10	12570.57	59	0.505002
		I-PSO(Iteration 2)	27.50	250.91	300.43	21.13	12580.53	72	0.5024
		C-PSO(Iteration 1)	29.92	250.02	299.97	20.01	12570.54	42	0.404131
		C-PSO(Iteration 2)	29.51	250.00	300.34	19.97	12571.69	21	0.5006
		CI-PSO(Iteration1)	29.95	250.00	300.01	20.01	12570.54	19	0.394495
		CI-PSO(Iteration2)	26.94	249.80	300.113	23.058	12579.46	22	0.5265
4	**540**	**I-PSO(Iteration 1)**	25.09	195.04	299.85	20.10	11431.31	64	0.506132
		I-PSO(Iteration 2)	25.69	201.19	293.03	19.96	11436.70	52	0.5042
		C-PSO(Iteration 1)	25.21	194.80	300.10	19.90	11431.30	39	0.400420
		C-PSO(Iteration 2)	25.09	194.67	300.09	20.07	11431.90	39	0.4049
		CI-PSO(Iteration 1)	25.03	194.96	300.00	20.02	11431.30	25	0.518381
		CI-PSO(Iteration 2)	24.85	194.91	300.52	19.79	11433.25	22	0.4702
5	**400**	**I-PSO(Iteration 1)**	24.62	111.64	244.02	20.48	8892.40	72	0.504728
		I-PSO(Iteration 2)	25.499	155.40	198.85	20.03	8907.8	52	0.50288
		C-PSO(Iteration 1)	25.00	113.00	241.84	20.14	8893.17	41	0.502480
		C-PSO(Iteration 2)	24.99	111.16	244.07	19.91	8893.06	32	0.5047
		CI-PSO(Iteration 1)	25.09	90.39	264.37	20.08	8895.26	22	0.504192
		CI-PSO(Iteration 2)	24.82	136.08	218.53	20.41	8899.72	21	0.4348
6	**280**	**I-PSO(Iteration 1)**	25.14	119.54	116.17	19.36	6785.70	72	0.509013
		I-PSO(Iteration 2)	25.71	97.84	132.98	20.24	6777.59	52	0.5095
		C-PSO(Iteration 1)	24.91	71.94	163.10	19.90	6769.63	41	0.391117
		C-PSO(Iteration 2)	24.78	66.35	168.90	20.06	6769.34	42	0.5060
		CI-PSO(Iteration 1)	25.03	65.06	170.04	19.87	6769.62	22	0.510926
		CI-PSO`(Iteration 2)	24.71	64.48	170.88	20.07	6769.80	21	0.44492
7	**290**	**I-PSO(Iteration 1)**	25.20	71.60	173.40	19.90	6944.96	70	0.503239

		I-PSO(Iteration 2)	25.03	93.64	151.31	19.95	6947.85	55	0.5082
		C-PSO(Iteration 1)	24.98	72.94	172.20	19.94	6944.95	45	0.390747
		C-PSO(Iteration 2)	25.058	86.51	159.0524	19.42	6946.07	39	0.5060
		CI-PSO(Iteration 1)	24.74	127.21	118.42	19.56	6966.47	34	0.505811
		CI-PSO(Iteration 2)	23.62	69.59	178.83	18.279	6995.17	21	0.50446
8	**500**	**I-PSO(Iteration 1)**	25.06	154.70	299.86	20.27	10694.39	53	0.509806
		I-PSO(Iteration 2)	24.94	168.92	286.18	19.91	10698.54	51	0.4010
		C-PSO(Iteration 1)	25.20	154.95	299.60	19.94	10694.57	45	0.391801
		C-PSO(Iteration 2)	25.08	154.63	300.12	20.03	10694.37	32	0.5043
		CI-PSO(Iteration 1)	24.99	155.10	299.90	19.98	10694.51	22	0.510490
		CI-PSO(Iteration 2)	25.70	183.62	270.42	20.163	10707.27	21	0.4217

Table 2. The Results Of Economic Dispatch For 6 Units Using PSO Algorithms(I-PSO,C-PSO,CI-PSO)

S.NO	LOAD	PSO WITH I,C&CI	UNIT1	UNIT2	UNIT3	UNIT4	UNIT5	UNIT6	COST	ITER	CPU TIME (SEC)
1	**800**	**I-PSO(ITR-1)**	109.57	99.65	57.27	221.54	167.82	143.51	8249.30	75	0.584604
		I-PSO(ITR-2)	122.933	106.80	61.59	215.64	121.53	171.53	8260.28	72	0.480298
		C-PSO(ITR-1)	99.08	99.74	50.16	277.56	135.06	139.81	8228.49	52	0.593148
		C-PSO(ITR-2)	107.50	100.54	50.16	274.96	131.67	134.31	8231.32	42	0.605622
		CI-PSO(ITR-1)	99.93	99.98	49.78	267.77	164.38	118.02	8232.42	22	0.598271
		CI-PSO(ITR-2)	106.04	130.59	86.85	162.95	182.36	131.33	8.302.76	21	0.482330
2	**1200**	**I-PSO(ITR-1)**	136.48	123.89	49.88	444.64	223.08	221.97	11477.64	52	0.584205
		I-PSO(ITR-2)	133.45	160.20	94.59	363.27	219.33	229.41	11509.04	49	0.581861
		C-PSO(ITR-1)	117.29	132.69	50.99	451.74	224.25	222.67	11477.75	39	0.580967
		C-PSO(ITR-2)	121.43	114.00	50.91	453.45	229.25	231.21	11477.54	42	0.591794
		CI-PSO(ITR-1)	118.71	132.96	49.91	472.80	221.39	204.15	11479.84	22	0.586441
		CI-PSO(ITR-2)	127.84	118.75	50.66	303.96	206.31	392.43	11556.53	22	0.483817
3	**1520**	**I-PSO(ITR-1)**	186.75	177.52	59.73	529.82	290.76	275.45	14169.99	55	0.580948
		I-PSO(ITR-2)	271.20	154.20	76.34	429.85	275.92	312.38	14194.27	55	0.483145
		C-PSO(ITR-1)	189.93	166.81	53.62	526.44	297.13	285.80	14169.86	49	0.475204
		C-PSO(ITR-2)	196.21	181.07	63.75	506.96	285.5256	286.83	14170.56	35	0.66871
		CI-PSO(ITR-1)	211.79	197.51	81.68	498.94	245.97	283.99	14178.51	21	0.581726

		CI-PSO(ITR-2)	129.20	265.25	51.91	475.04	238.77	359.74	14210.67	21	0.482689
4	**2238**	**I-PSO(ITR-1)**	397.02	300.61	133.58	588.18	424.09	394.53	20443.94	66	0.586628
		I-PSO(ITR-2)	408.02	305.86	142.59	547.81	406.71	427.42	20453.69	62	0.600988
		C-PSO(ITR-1)	397.42	322.40	101.32	584.71	439.36	391.67	20467.45	40	0.468436
		C-PSO(ITR-2)	367.18	313.25	112.94	590.16	432.32	422.32	20465.85	45	0.507430
		CI-PSO(ITR-1)	428.59	342.98	127.58	589.80	333.29	415.67	20489.70	21	0.585623
		CI-PSO(ITR-2)	566.18	289.24	107.08	578.21	448.61	248.07	20811.38	21	0.48057
5	**1800**	**I-PSO(ITR-1)**	229.90	293.32	80.12	550.88	340.30	305.69	16587.93	72	0.585138
		I-PSO(ITR-2)	240.37	262.62	99.09	591.65	259.96	345.07	16733.45	75	0.581864
		C-PSO(ITR-1)	250.10	204.03	82.04	585.87	328.84	349.02	16580.41	45	0.590126
		C-PSO(ITR-2)	242.55	225.87	69.99	585.51	330.81	345.37	16580.39	39	0.491607
		CI-PSO(ITR-1)	254.97	222.65	68.28	576.21	337.13	340.76	16579.93	22	0.593300
		CI-PSO(ITR-2)	224.90	330.83	73.68	511.07	282.49	377.03	16622.17	21	0.592007

7. Conclusion

In this paper, the various algorithms dealing with PSO have been discussed, namely, I-PSO, C-PSO and CI-PSO. The economic load dispatch problem is solved using those algorithms. The effectiveness of these algorithms has been tested on systems comprising four and six units on different load patterns and their total operating cost was evaluated. It is found that there is a very small variation on the total operating cost but the iteration needed to find an optimal solution is different in each case. Also, the CPU times varies with the change in the number of iterations. Two iterations of the solution have been included and there is very little variation in the results so, it is concluded that CI-PSO needed fewer iterations then C-PSO and I-PSO for achieving a similar solution.

8. Reference

[1] Wood A.J. and Wollenberg B.F., Power Generation Operation andControl, John Wiley & Sons, New York, 1984.

[2] Happ H.H., Optimal Power Dispatch – A Comprehensive Survey, IEEE Trans. Power App. Syst., vol. PAS-96, pp. 841-854, 1977.

[3] Yang H.T., Yang P.C. and Huang C.L., Evolutionary Programming Based Economic Dispatch for Units with Non-smooth Fuel Cost Functions, IEEE Trans. Power Syst., pp.112–118, February 1996.

[4] Sheble G.B. and Brittig K., Refined Genetic Algorithm – Economic Dispatch Example, IEEE Trans. on Power Systems, Vol. 10, No. 1, pp.117-124, 1995.

[5] Jeyakumar D.N., Jayabarathi T. and Raghunathan T., Particle Swarm Optimization for Various Types of Economic Dispatch Problems, Elect. Power Energy System, pp. 36–42, 2006.

[6] Selvakumar A.I. and Thanushkodi K., A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems, IEEE Trans. Power System, pp. 42-51, February 2007.

[7] Lin W.M., Cheng F.S. and Tsay M.T., An Improved Tabu Search for Economic Dispatch with Multiple Minima, IEEE Trans, Power Systems, pp. 108–112.87, February 2002.

[8] Yalcinoz T., Altun H. and Uzam M., Economic Dispatch Solution Using A Genetic Algorithm Based on Arithmetic Crossover, IEEE Porto Power Tech Conference, September, 2001.

[9] Gaing Zwe-Lee., Particle Swarm Optimization to Solving the Economic Dispatch Considering the Generator Constraints, IEEE Trans, Power Syst., pp. 1187–1195, November 2004.

[10] Selvakumar A.I. and Thanushkodi K., A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems, IEEE Trans. Power System, pp. 42-51, February 2007.

[11] Kennedy J. and Eberhart R. C., Swarm Intelligence, San Francisco, CA : Morgan Kaufmann, 2001.

[12] Attaviriyanupap P., Kita H., Tanaka E. and Hasegawa J., A hybrid EP and SQP for Dynamic Economic Dispatch with Non-smooth Fuel Cost Function, IEEE Trans, Power Syst., , pp. 411–416, May 2002.

[13] Singh S.N, Østergaard J. and Yadagiri J., Application of Advance Particle Swarm Optimization Technique to wind –Thermal Coordination, IEEE Conference, 2009.

[14] Kusic G. L. and Putnam H. A., Dispatch And Unit Commitment Including Commonly Owned Units, IEEE Transactions on Power Apparatus and Systems, Vol. 104, pp. 2408-2412, Sep 1985.

[16] Lim Shi Yao, Montakhab Mohammad, and Nouri Hassan, Economic Dispatch of Power System Using Particle Swarm Optimization with Constriction Factor, IJESP, Vol. 4, No. 2, pp. 29-34, October 2009.

[17] Yalcinoz T. and Altun H., Comparison of Simulation Algorithms for the Hopfield Neural Network: an Application of Economic Dispatch, Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 8, No.1, pp. 67-80, 2000.

[18] Puri Vinod, et al, Unit Commitment Using Particle Swarm Optimization, BIOINFO Computational Optimization, Vol. 2, No.1, pp. 09-16, 2012.

[19] Puri Vinod and Chauhan Y. K., A Solution to Economic Dispatch Problem Using Augmented lagrangian Particle Swarm Optimization, IJETAE , Vol. 2, No. 8, pp. 511-517, August, 2012.