International Journal of Power System Operation and Energy Management

Volume 1 | Issue 1 Article 13

July 2011

EMISSION CONSTRAINED ECONOMIC DISPATCH USING VARIOUS PSO ALGORITHM

C. Rani
School of Electrical Engineering Vellore Institute of Technology Vellore, Tamil Nadu, India,
royalrani@yahoo.co.in

Afshin Yazdani
School of Electrical Engineering Vellore Institute of Technology Vellore, Tamil Nadu, India,
afshinyazdani@gmail.com

EMISSION CONSTRAINED ECONOMIC DISPATCH USING VARIOUS PSO ALGORITHM

C.Rani[1], AfshinYazdani[2]
School of Electrical Engineering
Vellore Institute of Technology
Vellore, Tamil Nadu, India
Email: royalrani@yahoo.co.in

***Abstract*- The advancement in power systems has led to the development of economic load dispatch (ELD) that is difficult to solve by classical optimization method. The proposed paper work is to evolve a simple and effective method for an optimum generation dispatch to minimize the fuel cost and emission cost of power networks. The approach is based on Particle Swarm Optimization (PSO) technique, its variants and Hybrid Differential Evolution HDE. The proposed techniques are tested on a 6-generator, 30-bus standard test system. The effectiveness of the proposed optimization is verified in simulation studies using MATLAB software. The PSO based approach has been extended to evaluate the trade-off curve between the fuel cost of power production and the emission cost according to the multi-objective function.**

***Index terms*- economic load dispatch, hybrid differential evolution, multi-objective global optimization, particle swarm optimization.**

I. INTRODUCTION

Today's power systems are highly complex and their operations are unpredictable. The primary objective in the planning and operation of power systems is to provide quality supply to consumers at economical cost. Previous efforts on solving generation dispatch problem have employed the conventional methods which includes the lambda iteration and gradient method [1], [2]. The increasing energy demand and the decreasing energy resources have made optimization a great necessity in power system operation and planning.

Economic dispatch is the optimization scheme of generation system to determine the best generation schedule to supply a given load with minimum cost, while satisfying a set of constraints. Because of the increasing size and complexity of power system networks, such as multiple fuel options, environmental constraints, more attention is being given to develop optimization methods that automatically account for such practical constraints. Also the annual fossil fuel costs are in the order of several billions of dollars and even a small improvement in the economic dispatch function can lead to significant cost savings. A lot of efforts have also been devoted to the improvement of convergence and the reduction of computation time[12].

The paper proposes a method for optimization approach to determine the ELD while satisfying a set of constraints. The generation dispatch solution is based on PSO and formation of a bi-criterion objective function. The approach includes the evaluation of the trade-off curve between fuel cost and emission cost in power dispatch by solving the bi-criterion function and its effectiveness is demonstrated through a 6-generator, 30-bus standard test system. A MATLAB program was developed to implement the PSO algorithm for solving the bi-criterion problem.

II. EMISSION CONSTRAINED ECONOMIC DISPATCH

One method of handling an economic emission dispatch problem is by constraining the level of emissions. This is nothing but environmentally constrained economic dispatch problem (ECED). The characteristics of emissions of different pollutants are different and are usually highly nonlinear. This increases the complexity of the environmentally-constrained economic dispatch (ECED) problem. A solution procedure based on the LaGrange multiplier technique for the ECED problem was proposed by J. W. Lamont and E. V. Obessis [4]. Evolutionary-programming-based algorithm for solving the environmentally constrained economic dispatch (ECED) problem was proposed by Kit Po Wong and Jason Yuryevich [13].

A. Emission Constrained Economic Dispatch Formulation

Objective function is:

$$F_T = \sum_{i=1}^{l}\sum_{j=1}^{N}\left(c_{0j} + c_{1j}P_{ij} + c_{2j}P_{ij}^3\right) \quad (1)$$

NOx and SO_2 emissions are of the form:

$$\sum_{i=1}^{l}\sum_{j=1}^{N}(b_{0j} + b_{1j}P_{ij} + b_{2j}P_{ij}^3) \quad (2)$$

Constraints are:

$$\sum_{i=1}^{l} P_i - D_i - P_{loss} = 0 \quad ;$$

$$P_{i\min} \le P_i \le P_{i\max} \quad ; \quad E \le E_{\lim it}.$$

For the first interval, the emission limit E, of a pollutant is given by

$$E_1 = E_{\lim it}\left[\frac{D_1}{\sum_{i=1}^{l} D_i}\right] \quad (3)$$

For the second and the subsequent intervals, for example the ith interval, the emission limit is given by

$$E_i = \left[E_{\lim it} - \sum_{j=1}^{i-1} E_j \right] \left[D_i \Big/ \sum_{j=i}^{I} D_j \right] \quad (4)$$

Where, E (P_{ij}): Emission of generator j in interval i,
E_{limit}: Total emission limit over the horizon,
D_i: Demand in interval i.
P_{loss} : Transmission losses.

B. Basis for Solving Problem

In this method, one of the objective functions constitutes the primary objective function and all other objectives act as constraints. In other words, the ε-constraint approach replaces the (G-1) objective functions by (G-1) constraints as given below:

Minimize $f_i(x)$
Subject to
$f_j(x) <= \varepsilon_j$ (j=1, 2… G; j≠i); x ε X
Where, ε_j (j=1, 2… G; j≠i) are maximum tolerable levels.
G is number of objectives.
x is n-dimensional vector of decision variables.
X is the decision space.
f is an objective function.

The levels of satisfactory ε_j are varied parametrically to evaluate the impact on the single objective function $f_i(x)$. The ε-constraint approach facilitates the generation of non-inferior solutions as well as the trade-off functions. The number of solutions of the scalarized problem required to identify completely, or even approximately the non-inferior set, increases exponentially with the number of objectives. The major weakness of the constraint method is computational efficiency where there are several objective functions.

III. OVERVIEW OF PARTICLE SWARM OPTIMISATION

Kennedy and Eberhart first introduced PSO in year of 1995 [4]. PSO is motivated from the simulation of the behaviour of social systems such as fish schooling and birds flocking [5]. The PSO algorithm requires less computational time and less memory. The basic assumption behind the PSO algorithm is, birds find food by flocking and not individually. This leads to the assumption that information is owned jointly in flocking. Basically PSO was developed for two-dimension solution space by Kennedy and Eberhart [4]. The position of each individual is represented by XY axis position and its velocity is expressed by Vx in X direction and Vy in Y direction.

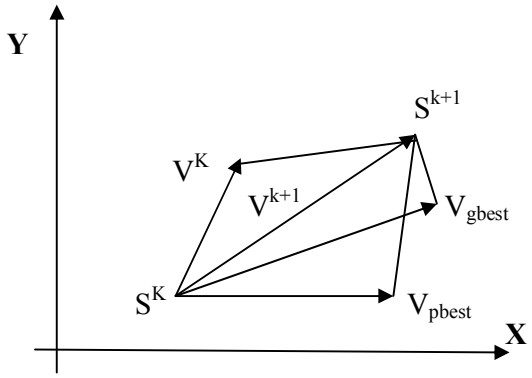

Fig. 1 Concept of Modification of Searching Point of PSO

Where S^K = Current searching point
S^{K+1} = Modified searching point
V^K = Current Velocity
V^{K+1} = Modified Velocity
Vpbest = Velocity based on 'pbest'
Vgbest = Velocity based on 'gbest'

The current position (searching point in the solution space) can be modified using the following equation and this is also shown in fig.1.

$$S_i^{K+1} = S_i^K + V_i^{K+1} \quad (5)$$

A. Optimization Algorithm

1. Represent the i^{th} particle
2. Pi = [Pi_1, Pi_2, Pi_3, Pi_4……..Pi_d,…Pin]
3. Generate the velocity V_{jmax}
4. Evaluate pbest and then identify gbest
5. Calculate new velocities
$Vij^{(Iter+1)} = w*Vij^{(Iter+1)} + c_1*rand_1*(pbest_i - p_{ij}^{(iter)}) + c_2*rand_2*(gbest_i - p_{ij}^{(iter)})$
6. Update generation
$P_{ij}^{(iter+1)} = P_{ij}^{(iter)} + V_{ij}^{(iter+1)}$

B. Implementation of Proposed Algorithm

The proposed method can be split into three major processes

1. Initialization Process
2. Fitness Evaluation Process
3. Updating Process

Initialization Process

Let P be the 'particle' co-ordinate (position) and V its speed (velocity) in search space. Consider i as a particle in the total population (swarm). Now i^{th} particle position can be

represented as Pi = [P_{i1}, P_{i2},…..P_{iN}] in the N dimensional space.

Fitness Evaluation Process

In this fitness evaluation process, each particle in the population is evaluated using the fitness function in the first iteration.

$$Fe = \sum (a_j P_j^{\,2} + b_j P_j + c_j) \quad \text{for } j = 1 \text{ to } n \qquad (6)$$

The best previous position of i^{th} particle is stored and represented as

Pbesti = ($Pbest_{i1}$, $Pbest_{i2}$, $Pbest_{i3}$,……Pbestij). All the pbest are evaluated by using a fitness function. The best particle among all pbest is represented as gbest.

Updating Process

In this updating process, modify each individual velocity V of the each particle P_i according to the equation shown below:

$$V_{ij}^{(Iter+1)} = w * V_{ij}^{(Iter+1)} + c_1 * rand_1 * (pbest_i - p_{ij}^{(iter)}) + c_2 * rand_2 * (gbest_i - p_{ij}^{(iter)}) \qquad (7)$$

i = 1,2,3,…n and d = 1,2,3,….n where n is the number of units. The use of linearly decreasing inertia weight factor w has provided improved performance in all the applications. Its value is decreased linearly from about 0.9 to 0.4 during a run. Suitable selection of the inertia weight factor provides a balance between global and local exploration and exploitation, and results in less iteration on an average to find a sufficiently optimal solution. Its value is set as

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{iter_{max}}\right) * iter \qquad (8)$$

Where iter indicates current iteration, itermax indicates maximum no of iterations.

However, after updating the velocity, the individual velocity may violate its velocity maximum or minimum constraints. This violation is corrected as follows:

If $V_{id}^{(Iter+1)} > V_{id\,max}$, then $V_{id}^{(Iter+1)} = V_{d\,max}$

$V_{id}^{(Iter+1)} < V_{id\,max}$, then $V_{id}^{(Iter+1)} = V_{d\,min}$

After this velocity updating process of all the individuals in each particle, modify the position (generator output level) of each individual in the particle P_i according to the following equation

$$p_{ij}^{(iter+1)} = p_{ij}^{(iter)} + V_{ij}^{(iter+1)} \qquad (9)$$

At the end of the updating process, if the evaluation value of each individual is better than the previous pbest, the current evaluation value is set to be as pbest and if the best pbest is better than gbest, that value is set to be gbest.

IV. VARIANTS OF PSO

A. Self Adaptive Particle Swarm Optimization

The conventional PSO has the uniform inertia weight and uniform acceleration coefficients, so the diversities of the population are limited. These coefficients are adjusted by self-adaptive strategies based on evaluating the results of the passed evolution. Presume the optimization is to find the global minimum of the object and the minimum is positive. If the minimum is negative, it can be set to positive by adding a positive constant big enough to the object. To consider the coefficient:

$$\xi_i^t = (f_i^{t-1} - f_G^{t-1})/f_i^{t-1} \quad , f_i^{t-1} \neq 0$$

$$\xi_i^t = 0 \quad , f_i^{t-1} = 0$$

where ξ_i^t is named as "related distance(RD)", where f_G and f_i^t are the global best fitness values found till the iteration t by all particles and the fitness values of particle i at the t^{th} iteration respectively, so $\xi_i^t \in [0,1]$. When it is close to 1, it means that the i^{th} particle is far from the known best position found by all particles so far; when it is close to 0, it means that it is near to the position. In this strategy the inertia weight (ω_i^t) and the social acceleration coefficient ($c_{i,2}^t$) are alterable. Firstly, the inertia weight (ω_i^t) is computed by

$$\omega_i^t = \omega_1 \times F(\xi_i^t) + \omega_2$$

Where ω_1 and ω_2 are positive constants. $F(\xi_i^t)$ is defined as follows:

$$F(\xi_i^t) = 2(1 - \cos((\pi/2) * \xi_i^t))$$

$F(\xi_i^t)$ is a function named "inertia weight adjust function", whose value increases when ξ_i^t increases from 0 to 1, and then ω_i^t will be increased with ξ_i^t to obtain stronger global search ability when the particle is far from gbest, and decreased to obtain stronger local search ability when it is close to gbest.

Secondly, the social acceleration coefficient ($c_{i,2}^t$) is computed by

$$c_{i,2}^t = c_f \times G(\xi i\, t) + c_e,$$

Where c_f and c_e are positive constants. $G(\xi_i^t)$ is defined as follows

$$G(\xi_i^t) = 2.5(1 - \cos((\pi/2) * \xi_i^t))$$

$G(\xi_i^t)$ is a function named "social adjust function", whose value increases while ξ_i^t increases from 0 to 1. So when the position of the i^{th} particle gets closer to gbest, $c_{i,2}^t$ will decrease to attract the particle to pbest from gbest to prevent the swarm from crowding in a local minimum; when it is far from gbest, $c_{i,2}^t$ will increase to attract the particle to gbest from pbest to speedup global convergence.

B. Dispersed Particle Swarm Optimization

Because the fitness value of the current position of each particle represents its adaptive capability, it is naturally selected as the performance to distinguish the characteristic differences. Thus, the better the fitness of current position is, the more adaptation that particle can get to enhance its survival ratio. Thus, a new index is introduced in this method which is based on performance differences, and the definition is provided as follows:

$$Grade_u(t) = \frac{f_{worst}(t) - f(x_u(t))}{f_{worst}(t) - f_{best}(t)}$$

Where, $f_{worst}(t)$ and $f_{best}(t)$ are the worst and best fitness values of the swarm at time t respectively. Occasionally, the swarm converges onto one point, that means $f_{worst}(t) = f_{best}(t)$. In this case, the value $Grade_u(t)$ of arbitrary particle u is set to

1. $Grade_u(t)$ is an information index to represent the differences of particle u at time t, according to its fitness value of the current position. The better the particle is, the larger $Grade_u(t)$ is and vice versa.

For the best solution, it should make a complete local search around its historical best position, as well as for the worst solution and it should make a global search around $\overrightarrow{P_g}$. Then, the dispersed social coefficient of particle j at time t is set as follows:

$$c_{2,j}(t) = c_{low} + (c_{up} - c_{low})Grade_j(t)$$

Mutation strategy

Since an explicit selection pressure $Grade_j(t)$ is introduced, the new social coefficient setting encounters local optima with a higher probability than PSO. To avoid premature convergence, a mutation strategy is introduced to enhance the ability to escape from the local optima. This mutation strategy is designed as follows. At each time, particle j is uniformly randomly selected within the whole swarm, as well as the dimensionality k is also uniformly randomly selected, then the $v_{jk}(t)$ is changed as follows:

$v_{jk}(t) = 0.5 \times x_{max} \times r1$, if r2 < 0.5,

$v_{jk}(t) = -0.5 \times x_{max} \times r1$, otherwise

Where r1 and r2 are two random numbers generated with uniform distribution within 0 and 1. In each generation, only one dimensional value of one particle is applied with mutation strategy. Thus, the mutation probability is 1/(Popsize*Dimension).

C. Chaotic Particle Swarm Optimization

Like evolutionary algorithms, PSO technique conducts search using a population of particles, corresponding to individuals. Each particle represents a candidate solution to the problem at hand.

During the calculation, the particle is affected by three factors when it is moving in space. One of the factors is the particle's current velocity V(t). Another is the optimal point X*(t) where the particle has reached before. The third factor is the optimal point X**(t) of the community or the sub-community. The particle's velocity is changed towards X*(t) and X**(t) in every iteration step. Meanwhile, Vi, X*(t) and X**(t) are assigned separately a weight at random. The velocity and position is updated according to the formulae below:

$$v_{i,j}^{(t)} = w \times v_{i,j}^{(t-1)} + c1 \times r1 \times (x_{i,j}^{(t-1)} - x_{i,j}^{(t-1)}) + c2 \times r2 \times (P_{gi,j}^{(t-1)} - x_{i,j}^{(t-1)})$$

$$x_{i,j}^{(t)} = x_{i,j}^{(t-1)} + v_{i,j}^{(t)} \qquad (i = 1, 2, \ldots, n \; j = 1, 2, \ldots, m)$$

Where c1 and c2 are the learning factors.
Generally, c1 = c2 =2.
w is the given weight,
r1, r2 are the random variables within the interval of [0, 1].
t is the number of iterations
n the number of particles and
m is the number of dimensions.

In fact, however, it can't ensure the optimization's ergodicity entirely in phase space, because it is random in traditional PSO. Therefore, the method introduces chaotic mapping with certainty, ergodicity and stochastic properties into particle swarm optimization so as to improve the global convergence. The parameter w is modified by the formula

$$w^{(t+1)} = 4.0 \times w^{(t)} \times (1 - w^{(t)}) \qquad w(t) \in (0, 1)$$

D. New Particle Swarm Optimization

The original PSO described above is basically developed for continuous optimization problems. However, lots of practical engineering problems are formulated as combinational optimization problems. Kennedy and Eberhart developed a discrete binary version of PSO for these problems. They proposed a model wherein the probability of a particle's deciding yes or no, true or false, or making some other decision, is a function of personal and social factors as follows:

$P(x_{id}^{t} = 1) = f(x_{id}^{t-1}, v_{id}^{t-1}, p_{id}, p_{gd})$

Where,

$P(x_{id}^{t} = 1)$ is the probability that individual I will choose 1 for the bit at the dth site on the bit string

x_{id}^{t} is the current state of the bit string site d of individual I

t means the current time step, and t −1 is the previous step

v_{id}^{t-1} is a measure of the individual's predisposition or current probability of deciding 1

p_{id} is the best state found so far; for example, it is 1 if the individual's best success occurred when x_{id} was 1 and 0 if it was 0

p_{gd} is the neighbourhood best, again 1 if the best success attained by any member of the neighbourhood was 1 when it was in the 1 state and 0 otherwise.

Both continuous and binary PSO have been used successfully in many optimization problems including the famous traveling salesperson problem.

From our personal experience we know that an individual not only learns from his or her own and other individuals' previous best, but also learns from his or her own and other individuals' mistakes. The idea of NPSO is based on this social behavior. Each particle tries to leave its previous worst position and its group's previous worst position. $P_i = (p_{i1}, p_{i2}, \ldots p_{Id})$ represents a particle's previous worst, and the index of the worst in the whole group is g, and the position change is represented as $\Delta X_i = \Delta x_{i1}, \Delta x_{i2}, \ldots \Delta x_{Id}$. The equations adopted here are similar to those of PSO

$\Delta x_{id} = \omega \, \Delta x_{id} + c_1 rand_1 * (X_{id} - P_{id}) + c_2 rand_2 * (X_{id} - P_{gd})$

$x_{id} = x_{id} + \Delta x_{id}$

c_1, c_2, rand1 ( ) and rand2 ( ) have the same meaning as in PSO.

V. CASE STUDY AND RESULTS

This problem is solved by using ε-constraint method. This problem contains only one objective function i.e. cost function of generator and emissions are considered as constraints. So, here cost is minimized while constraining the emissions in their limits. The limiting levels of emissions over a scheduling horizon represent additional operational constraints that are to be satisfied when finding the optimal solution for the economic dispatch problem. The EP-based ECED algorithm was applied to the 9-generator system, which was adopted as the study example in [l]. In the test example, the generation scheduling horizon is 6 hours with 6 intervals of 1 hour duration.

TABLE I

RESULTS GIVEN IN INFERENCE PAPER WHEN SOLVED USING EVOLUTIONARY PROGRAMMING FOR ECED PROBLEM

Unit	Interval loadings (MW)						
	1	2	3	4	5	6	
1	224.360	199.547	233.859	240.000	240.000	240.000	
2	226.771	202.872	240.000	240.000	240.000	240.000	
3	391.501	355.083	407.168	425.988	450.000	419.449	
4	268.545	244.483	277.670	286.469	303.936	285.207	
5	259.788	239.804	269.831	280.144	294.055	274.941	
6	350.000	350.000	360.984	390.109	439.820	380.964	
7	35.000	35.000	81.406	129.980	175.000	109.409	
8	89.431	35.000	115.758	136.923	175.000	132.778	
9	154.604	138.211	163.324	170.407	182.189	167.252	Total
SO_2	11.175	10.136	11.943	12.717	13.766	12.459	72.198
NOx	2.624	2.207	2.906	3.168	3.563	3.088	17.556
Cost ($)	3.0330 *10^4	2.7402 *10^4	3.2684 *10^4	3.5093 *10^4	3.8358 *10^4	3.4280 *10^4	1.98150 *10^4

TABLE II

RESULTS OF PARTICLE SWARM OPTIMIZATION FOR ECED PROBLEM

Unit	Interval loadings (MW)						
	1	2	3	4	5	6	
1	219.271	206.736	213.694	240.000	240.000	239.785	
2	191.296	211.882	226.380	237.420	240.000	239.791	
3	339.384	360.419	399.619	443.142	441.558	446.756	
4	274.235	243.806	289.720	294.804	271.897	291.723	
5	289.895	240.978	281.921	272.288	277.542	269.031	
6	350.000	350.000	350.000	393.205	524.777	383.778	
7	61.251	35.000	116.736	105.112	167.366	87.548	
8	156.457	35.000	124.619	137.872	143.397	122.615	
9	118.207	116.177	147.307	176.152	193.459	168.969	Total
SO_2	11.106	10.149	11.903	12.734	13.816	12.485	72.197
NOx	2.575	2.234	2.883	3.214	3.497	3.148	17.555
Cost ($)	3.0379 *10^4	2.73567 *10^4	3.28396* 10^4	3.50111* 10^4	3.8402 *10^4	3.4160 *10^4	1.9815 *10^4

TABLE III

RESULTS OF ADAPTIVE PARTICLE SWARM OPTIMIZATION FOR ECED PROBLEM

Unit	Interval loadings (MW)						
	1	2	3	4	5	6	
1	164.048	197.793	214.944	239.687	237.441	230.794	
2	179.963	198.781	234.665	240.000	239.435	237.158	
3	408.659	342.667	414.471	434.048	436.653	429.111	
4	278.155	253.737	307.046	294.163	281.626	282.339	
5	256.097	251.800	286.920	270.108	283.707	283.527	
6	350.000	350.000	350.000	400.366	519.182	367.801	
7	65.745	38.3613	80.659	117.307	168.423	123.757	
8	156.245	43.2692	103.777	133.617	143.274	132.808	
9	141.084	123.589	157.515	170.700	190.255	162.701	Total
SO_2	11.1071	10.1240	11.9372	12.729	13.8075	12.4492	72.154
NOx	2.5712	2.2153	2.9611	3.1916	3.4996	3.0989	17.537
Cost ($)	3.0322 *10^4	2.7439 *10^4	3.2667 *10^4	3.5038 *10^4	3.8402 *10^4	3.4283 *10^4	19.815 *10^4

TABLE IV

RESULTS OF DISPERSED PARTICLE SWARM OPTIMIZATION

Unit	Interval loadings (MW)					
	1	2	3	4	5	6
1	194.643	197.124	230.669	240.000	240.000	228.970
2	224.638	208.293	239.692	240.000	240.000	239.442
3	361.684	354.718	403.667	429.589	442.553	422.727
4	269.128	255.982	301.527	287.364	285.111	289.140
5	287.083	247.048	284.619	279.255	267.200	278.718

6	350.000	350.000	350.000	389.725	517.414	375.917	
7	35.000	35.000	77.283	123.785	154.848	118.245	
8	124.270	35.000	112.814	142.670	162.531	134.086	
9	153.550	116.832	149.724	167.608	190.340	162.750	Total
SO_2	11.1349	10.1390	11.9387	12.7183	13.8094	12.4497	72.189
NOx	2.5895	2.2347	2.9560	3.1850	3.4994	3.0886	17.553
Cost ($)	3.0319 *10^4	2.7385 *10^4	3.2664 *10^4	3.5080 *10^4	3.8400 *10^4	3.4304* 10^4	19.805 *10^4

TABLE V

RESULTS OF CHAOTIC PARTICLE SWARM OPTIMIZATION

Unit	Interval loadings (MW)						
	1	2	3	4	5	6	
1	201.251	198.372	227.408	239.173	240.000	237.248	
2	164.003	202.356	234.013	240.000	240.000	236.518	
3	376.414	359.832	413.333	431.516	441.911	422.310	
4	275.653	252.950	298.467	284.386	280.802	283.076	
5	298.271	253.501	282.258	283.135	272.416	275.991	
6	350.000	350.000	350.000	400.240	520.600	371.720	
7	44.301	35.0472	78.079	121.172	156.142	117.093	
8	140.854	35.000	111.237	135.938	157.488	139.951	
9	149.248	112.938	155.400	164.437	190.637	166.090	Total
SO_2	11.1137	10.1396	11.9390	1207250	13.8118	12.4499	72.1789
NOx	2.5930	2.2403	2.9506	3.01865	3.4999	3.0841	17.5544
Cost ($)	3.0330 *10^4	2.7384 *10^4	3.2664 *10^4	3.5063 *10^4	3.8400 *10^4	3.4311 *10^4	19.8151 *10^4

TABLE VI

RESULTS OF CHAOTIC PARTICLE SWARM OPTIMIZATION

Unit	Interval loadings (MW)						
	1	2	3	4	5	6	
1	197.054	240.000	240.000	240.000	240.000	240.000	
2	240.000	240.000	240.000	231.364	240.000	240.000	
3	336.850	330.758	366.850	450.000	450.000	450.000	
4	2358.671	237.547	307.773	285.170	271.331	265.485	
5	252.237	228.117	298.522	248.010	268.330	245.515	
6	354.233	350.000	350.000	350.000	500.337	350.000	
7	175.333	35.000	175.000	143.795	173.956	175.000	
8	38.894	35.000	35.000	175.000	162.604	143.863	
9	170.058	103.577	136.853	176.659	193.439	140.135	Total
SO_2	11.1361	10.1753	11.9372	12.7036	13.8009	12.4482	72.2013
NOx	2.5135	2.2779	2.9577	3.1904	3.4938	3.1008	17.5341
Cost ($)	3.0303 *10^4	2.7303 *10^4	3.2689 *10^4	3.5138 *10^4	3.8408 *10^4	3.4310 *10^4	19.8150 *10^4

TABLE VII

RESULTS OF HYBRID EVOLUTIONARY ALGORITHM FOR ECED PROBLEM

Unit	Interval loadings (MW)						
	1	2	3	4	5	6	
1	213.595	173.136	233.690	218.899	240.000	240.000	
2	222.965	168.561	240.000`	235.002	240.000	234.664	
3	388.287	382.919	400.483	407.333	450.000	399.160	
4	283.209	246.293	280.198	310.519	295.643	245.649	
5	277.777	242.622	292.663	282.511	285.445	265.994	
6	350.000	350.000	350.000	425.855	464.163	454.739	
7	67.447	35.000	82.116	88.416	163.689	89.983	
8	52.514	45.281	115.614	156.005	174.991	136.477	
9	144.202	156.184	155.233	175.455	186.067	183.331	Total
SO_2	11.1265	10.1228	11.9365	12.7185	13.7793	12.488	72.1720
NOx	2.6442	2.2073	2.9342	3.1512	3.6046	2.9888	17.5303
Cost ($)	3.0356 *10^4	2.7379 *10^4	3.2527 *10^4	3.5175 *10^4	3.8361 *10^4	3.4353 *10^4	19.8150 *10^4

TABLE VIII

COMPARISION TABLE FOR DIFFERENT ALGORITHMS FOR ECED PROBLEM

	Total cost ($)	**Total SO2 emission (tons)**	**Total NOx emission (tons)**
Reference	1.9815027*10^5	72.198	17.556

(EP)			
PSO	1.9815030*10^5	72.196	17.554
Self APSO	1.9815067*10^5	72.154	17.537
DPSO	1.9815166*10^5	72.189	17.553
CPSO	1.9815142*10^5	72.178	17.554
NPSO	1.9815086*10^5	72.190	17.534
HDE	1.9815081*10^5	72.194	17.545

Table VIII contains the fuel cost and emissions for Emission constrained economic dispatch problem (ECED) when solved using various evolutionary algorithms like PSO, APSO, DPSO, CPSO, NPSO and HDE. In Table VIII, the reference paper results are also mentioned, in which evolutionary programming is used to find the optimum fuel cost. The cost obtained when various evolutionary algorithms used is higher than the cost given in reference paper but the difference is very less which is almost negligible. Thus, the results obtained using various evolutionary algorithms are validated with those of reference paper results. Hence, we can infer that these evolutionary algorithms are efficient in solving an ECED problem effectively.

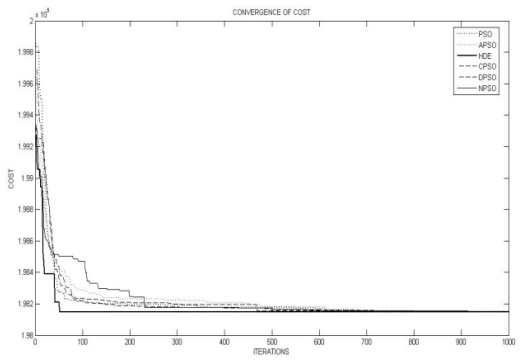

Fig. 2 Convergence of various algorithms for ECED problem

Fig. 2 shows the convergence of various algorithms for ECED problem. From Fig. 2 we observe that:

- All the evolutionary algorithms are converging to best solution but the main difference between these algorithms is their convergence time.
- Among all algorithms, HDE gives good solution and at the same time, it converges first than any other algorithm converges.
- Among all algorithms, APSO converges very slowly but it converges to best solution.

VI. CONCLUSION

An approach for the determination of the most appropriate dispatch solution, which best meets the economic and environmental objectives in power system operation has been proposed. The implementation of the approach is based on the formation of a bi-criterion objective and a global optimisation technique. The PSO, its alternatives and HDE method have been utilized in the present work. The solution that gives the best compromise among these objectives is chosen as the most appropriate generation dispatch solution. The trade-off curve evaluated is useful in providing alternate dispatch solution for engineers in daily operation.

ACKNOWLEDGMENT

We sincerely thank the management of Vellore Institute of Technology for giving us an opportunity to work towards our paper.

REFERENCES

[1] Bakirtzis, V. Petridis and S. Kazarlis" genetic algorithm solution to the economic dispatch problem" proc.inst.elect.engg-gen., trans., dist., vol141 no.4,pp 377-382 July 1994.
[2] Song.Y.H. Wang. G.S.,Wang. P.V.,Johns 1997 Environmental Economic Dispatch Using FUZZY Logic controlled Genetic Algorithm IEEE proceedings - Generation, Transmission Distribution, Vol.144 (4).
[3] Kit Po Wong, Bie Fan, C.S.Chang, A.C Liew " Multi-Objective generation dispatch using Bicriterion global optimisation", IEEE transaction on power system, Vol.10, No, November 1995.
[4] James kennedy,Russell Eberhat Particle swarm optimisation.Proceedings of IEEE conference on Neural Networks,Piscataway.NJ,IV:1942-1948.
[5] Eberhart.R.C.,Shi.Y. Particle swarm optimization Developments, Applications and Resources.Proceedings of the congress on evolutionary computation. 2001; vol.1:81-86.
[6] Effie Tsoi, Kit Po Wong, Chun Che Fung, " Hybrid GA/SA algorithms for evaluating trade-off between economic cost and environmental impact in generation dispatch, IEEE Nov 1995.
[7] Zwe-Lee gaing,"Particle swarm Optimisation to solving the economic dispatch considering the generation constraints, IEEE Transaction on power system, Vol.18, No.3, August 2003.
[8] A.J EL-Gallad, M.EL-Mawary, A.A.Sallam, A.Kalas,"Swarm Intelligence for hybrid cost dispatch problem".
[9] Cadogan, J.B and Eisenberg.L," Sulphur Oxide Emissions Management for Electric power system", IEEE Transaction on power application and systems, April 1977.
[10] Wong, K.P and Fung, C.C,"Simulated annealing based algorithm for minimum emission dispatch', Proc.International Power Engineering conference, March 1993.
[11] Gjengedal.T, Johanson.S and Hansen.O," A Qualitative approach to economic –environmental dispatch –treatment of multiple pollutants', IEEE trans. On energy conversion, Vol.7, No.3, Sept.1992, PP.367-373.
[12] C.Rani**,** M.Rajesh Kumar,K.Pavan "Multi-objective Generation Dispatch Using Particle Swarm Optimization" IEEE – IICPE 2006 Indian International Conference on power Electronics Dec 18-19, 2006, Chennai.
[13] Bahman J.Kermanshahi, Y.Wu, Keiichiro yasuda, Ryuichi yokoya ma,"Environmental marginal cost Evaluation by Non-Inferiority surface", IEEE Transaction on power system, Vol.5, N0.4, November 1990.