FUZZIFIED PSO ALGORITHM FOR DC-OPF OF INTERCONNECTED POWER SYSTEM

[1] N.M JOTHI SWAROOPAN., [2] P. SOMASUNDARAM

[1] Research Scholar, Department of Electrical and Electronics Engineering, CEG, Anna University, India.

[2] Asst .Prof., Department of Electrical and Electronics Engineering, CEG, Anna University, India-600025

ABSTRACT

This paper present a new computationally efficient improved stochastic algorithm for solving multi-area DC Optimal Power Flow (DC-OPF) in interconnected power systems. This algorithm is based on the combined application of Fuzzy Logic strategy incorporated in Particle Swarm Optimization (PSO) algorithm, hence termed as Fuzzy PSO (FPSO). Multi-area DC-OPF calculations determine optimum generation schedule, optimal control variables and system quantities of each area with due consideration of generation and transmission system limitations for efficient power system operation. The proposed method is tested on IEEE 30-bus interconnected three area system. The investigation reveals that the proposed method can provide accurate solution with fast convergence and has the potential to be applied to other power engineering problems.

Keywords: DC-Optimal Power Flow, Fuzzy Logic, Particle Swarm Optimization.

NOMENCLATURE

F_T	Total fuel cost.
ng	Number of committed generating units excluding the slack bus generator.
a_j, b_j, c_j	Fuel cost coefficients of jth committed unit.
$P_{G j}$	Active power generation of jth generating unit.
$P_{G s}$	Slack bus active power generation.
a_s, b_s, c_s	Fuel cost coefficients of slack bus generator.
A	Area index.
AA	Index of the area adjacent to area A.
B^A	Area A network admittance matrix.
θ^A	Area A bus voltage phase angle vector.
P^A	Area A unit active power output vector.
D^A	Area A bus active power demand vector.
T^A	Set of area A tie-lines.
R^A	Area A node to tie-line incidence matrix. $\left[R^A_{i-j}\right] = \begin{cases} 1, \text{ for } i = j \\ 0, \text{ otherwise} \end{cases}$ (i: bus index of area A; j: tie-line index connecting bus i).
θ^A_{ref}	Area A reference bus voltage phase angle.
NL	Number of lines in area A.
NT	Number of tie-lines connecting area A.
x_{ij}	Reactance of the line connecting buses i and j.
LF_k	Power flow on the line k connecting buses i-j.
ST_t	Scheduled power flow on the tie-line t connecting area A (i^{th} bus) and area AA (j^{th} bus).
N_p	Number of individuals.
I_{pi}	pi^{th} parent population.
f_{pi}	Fitness function of pi^{th} parent population.

1. INTRODUCTION

ELECTRIC power systems are interconnected due to the fact that it is a better system to operate with more reliability, improved stability and less production cost than the isolated systems. Multi-area DC Optimal Power Flow (DC-OPF) problem is a large scale non-linear optimization problem with linear constraints. Multi-area DC-OPF calculations determine optimum generation schedule, optimal control variables and system quantities of each area with due consideration of generation and transmission system limitations. Many approaches [1-5] have been developed for solving multi-area DC-OPF problem. In [3], a new effective decomposition method for solving DC-OPF problem in large interconnected decentralized power systems is proposed. This method decomposes the overall multi-area OPF problem into independent OPF sub-problems, one for each area. In this paper, similar decomposition technique is adopted for resolving the multi-area DC-OPF into independent equivalent single area DC-OPF sub-problems by incorporating scheduled tie-line power flow.

Conventional optimization methods [3-5] have been used for economic dispatch and OPF calculations. Methods based on successive linearization (SLP) are popular. For medium size power systems, the conventional methods for OPF calculations may be fast and efficient enough. However, for large scale interconnected power systems the higher dimension of possible solution space and increase of constraints result in excessive computational burden.

With a view to reduce the computational burden, some stochastic techniques have been developed. The recent trend is on the application of modern and improved heuristic application techniques. Artificial intelligence techniques are the most widely used tool for many power system optimization problems. These methods (e.g., genetic algorithms, evolutionary programming and PSO etc.,) seem to be promising and are still evolving. In Genetic Algorithm (GA) the solution space is discrete in nature (binary representation) and hence it is difficult to effectively apply GA to multi-area DC-OPF problem in a continuous multi-dimensional space. Evolutionary Programming (EP) is capable of finding global or near global optimal solutions within reasonable computation time hence it has become increasingly popular in recent years in science and engineering disciplines. Particle Swarm Optimization (PSO) is a powerful optimization procedure that has been successfully applied to a number of combinatorial optimization problems. It has the ability to avoid entrapment in local minima by employing a flexible memory system. The multi-area DC-OPF problem is effectively solved using PSO algorithm in [6 -8]. The major drawback of PSO method is large number of iterations and very large computation time.

In the present trend, there has been an increasing interest in the application of Fuzzy model [12]. Fuzzy logic has been applied in PSO algorithm. This gives promising results especially in cases where the processes are too complex to be analyzed by conventional techniques or where the available information is inexact or uncertain. Hence in this paper an amendment based on fuzzy logic is incorporated in PSO technique for solving the multi-area DC-OPF problem. The fuzzy logic is implemented in this effective stochastic algorithm (PSO) for obtaining a much better (faster) convergence.

2. MULTI-AREA DC-OPF PROBLEM FORMULATION

The multi-area DC-OPF problem is decoupled to equivalent independent single area sub-problem, one for each area by the addition of related tie-line power flows. The resulting equivalent independent single area sub-problem (considering area *A*) is formulated as a mathematical optimization problem as follows:

Objective function

Mininize

$$F_T{}^A = \sum_{j=1}^{ng} (a_j P_{Gj}^{A^2} + b_j P_{Gj}^A + c_j) + a_s P_{Gs}^{A^2} + b_s P_{Gs}^A + c_s \quad (1)$$

Subject to

(i) Power flow equation

$$B^A \theta^A + R^A T^A = P^A - D^A \quad (2)$$

$$\theta_{ref}^A = 0 \quad (3)$$

(ii) Line flow limit

$$\left|\frac{1}{x_{ij}}(\theta_i^A - \theta_j^A)\right| \le LF_k^{A\ \max}; k = 1,2,...NL$$

(4)

(iii) Generator limit

$$P_{G\,j}^{A\ \min} \le P_{G\,j}^{A} \le P_{G\,j}^{A\ \max}; \ j = 1,2,......ng$$

(5)

(iv) Slack bus generator limit

$$P_{G\,s}^{A\ \min} \le P_{G\,s}^{A} \le P_{G\,s}^{A\ \max}$$

(6)

(v) Tie-line flow limit

$$\left|\frac{1}{x_{ij}}(\theta_i^A - \theta_j^{AA})\right| - ST_t = 0; t = 1,2,......NT \qquad (7)$$

3. OVERVIEW OF PSO

The PSO method was introduced in 1995 by Kennedy and Eberhart [6]. The method is motivated by social behavior of organisms such as fish schooling and bird flocking. PSO provides a population-based search procedure. Here individuals called as particles change their positions with time. These particles fly around in a multidimensional search space. During flight, each particle adjusts its position according to its own experience, and the experience of neighbouring particles. Thus each particle makes use of the best position encountered by itself and its neighbours. The direction of the particle is given by the set of particles neighbouring the particle and its past experience. Let x and v denote the particle position and its corresponding velocity in the search space. pbest is the best previous position of the particle and gbest is the best particle among all the particles in the group. The velocity and position for each element in the particle at $(t+1)^{th}$ iteration is calculated by using the following equations.

$$v_i^{t+1} = k * \begin{pmatrix} w * v_i^t + \varphi_1.rand\left(pbest - x_i^t\right) + \\ \varphi_2.rand\left(gbest - x_i^t\right) \end{pmatrix}$$

(8)

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

(9)

where x_i and v_i are the current position and velocity of the i^{th} particle, w is the inertia weight factor, φ_1 and φ_2 are acceleration constants, $rand()$ is the function that generates uniform random number in the range [0,1] and k is the constriction factor introduced by Eberhart and Shi to avoid the swarm from premature convergence and to ensure stability of the system. Mathematically, k can be determined as follows

$$k = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}$$

(10)

where $\varphi = \varphi_1 + \varphi_2$ and $\varphi > 4$.

The selection of w provides a balance between global and local explorations. In general, the inertia weight w is set as

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{t_{\max}} \times t$$

(11) where $t_{\max}$ is the maximum number of iterations or generations and $w_{\max}$ and $w_{\min}$ are the upper and lower limit of the inertia weight. The inertia weight balances global and local explorations and it decreases linearly from 0.9 to 0.4 in each run. The constants c_1 and c_2 pulls each particle toward pbest and gbest positions. Maximum Velocity $V_{\max}$ was set at 10 – 20 % of the dynamic range of variable on each dimension. The Flow chart for basic PSO approach is shown in Figure. 1.

4. PARTICLE SWARM OPTIMIZATION (PSO) BASED MULTI-AREA DC-OPF PROBLEM

PSO, as an optimization tool, provides a population-based search procedure in which individuals called particles change their positions (states) with time. It is similar to the other evolutionary algorithm in which each particle in the swarm is initialized randomly within the effective real power operating limits.

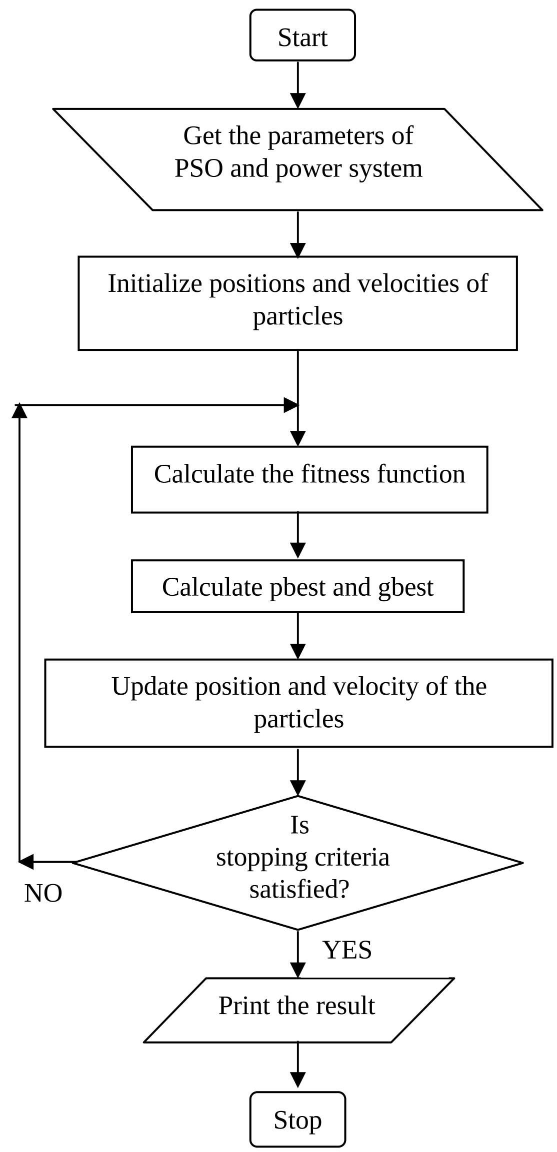

Fig. 1 Flow chart for basic PSO approach

These particles fly around in a multidimensional search space with a velocity which is dynamically adjusted according to the flying experiences of its own and its colleagues. The location of the j^{th} particle is represented as $X_j = \left(P_{11,j},.....P_{1n,j},......P_{mn,j}\right)$ for $j = 1,2,...N_p$ and $P_{mn,j} \in [P_{mn,\min}, P_{mn,\max}]$, $P_{mn,min}$ and $P_{mn,max}$ are the lower and upper bounds for the generation respectively. The best previous position (giving the best fitness value) of the j^{th} particle is recorded and represented as $X_j^p = \left(P^p_{11,j},.....P^p_{1n,j},......P^p_{mn,j}\right)$, for $j = 1,2,...N_p$ which is also called pbest. The index of the best particle among all the particles in the swarm is represented as $X_j^G = \left(P^G_{11,j},.....P^G_{1n,j},......P^G_{mn,j}\right)$, called as gbest. The velocity for the j^{th} particle is represented as $V_j = \left(v_{11,j}....,v_{1n,j}....,v_{nm,j}\right)$, is clamped to a maximum velocity $V_{\max} = \left(v_{\max,11}....,v_{\max,1n}....,v_{\max,mn}\right)$, which is specified by the user. In PSO, at each iteration(t), the velocity and location of each particle is changed toward its pbest and gbest locations according to the equations (12) and (13), respectively.

$$v^{(t+1)}_{mn,j} = w * v^{(t)}_{mn,j} + c_1 * rand\ (\ ) * \left(P^p_{mn,j} - P^{(t)}_{mn,j}\right) + c_2 * rand\ (\ ) * \left(P^g_{mn,j} - P^{(t)}_{mn,j}\right)$$

(12)

$$P^{(t+1)}_{mn,j} = P^{(t)}_{mn,j} + v^{(t+1)}_{mn,j}$$

(13)

where w is inertia weight, c_1 and c_2 are acceleration constants and rand() is a uniform random number in the range [0, 1]. In equation (12), the first part represents the inertia of pervious velocity; the second part is the "cognition" part, which represents the private thinking by itself; the third part is the "social" part, which represents the cooperation among the particles. If the sum of accelerations would cause the velocity $v_{mn,j}$ on that dimension to exceed $v_{\max,mn}$ then $v_{mn,j}$ is limited to $v_{\max,mn}$. $V_{\max}$ determines the resolution with which regions between the present position and the target position are searched.

The process for implementing PSO is as follows:

a) Initialization of particles: An initial swarm of size N_p is generated randomly within the feasible range.

$$X_j^{(t)} = [P^{(t)}_{11,j},.....P^{(t)}_{1n,j},......P^{(t)}_{mn,j}]$$

(14)

The elements of each particle X_j; j = 1,2, …., N_p are the real power output of committed generating units.

b)The fitness function for each particle is computed as,

$$f_j = F_j + k_1 \mid APBC_j \mid + k_2 \sum_{t=1}^{Nt} P_{t,j}^{\lim}; \; j = 1,2,..,N_P$$

(15)

Where *APBC* the area power balance constraint and the values of penalty factors k_1 and k_2 are chosen such that if there is any constraint violations the fitness function value corresponding to that particle will be ineffective. The maximum fitness function value among the particles is stored as $f_{max.}$

c) Determination of pbest and gbest particles: Compare the evaluated fitness value of each particle with its pbest. If current value is better than pbest, then set the current location as the pbest location. If the best pbest is better than gbest, the value is set to gbest.
d) Modification of member velocity: Change the member velocity of the each individual particle $v_{mn,j}$ according to the equations (12).
e) Modification of member position: The member position in each particle is modified according to (13).
f) If $t = t_{max}$ then the individual that generates the latest gbest is the optimal solution. Otherwise repeat the process from step b.

5. 5. FUZZFIED PSO (FPSO)

In the classical PSO technique the value of inertia weight is computed based on the iteration (t & t_{max}) alone which is independent of the problem being solved. However, for practical applications it may lead to slow and premature convergence. Hence there is a need for an adaptive inertia weight. The convergence depends on the relative fitness function value $fpi / fmax$. It is an essential factor which has a major influence in the convergence process. If the relative fitness value $fpi / fmax$ is low then the inertia weight is small and vice versa.

The other factor is the search range ($P_{mn,max}$ - $P_{mn,min}$) which is a constant throughout the whole search process. But actually the search range varies for each generation or iteration. Hence there is a need for an effective search range. Thus these factors need a certain control to obtain a better convergence. Moreover the relationship between them seems arbitrary, complex and ambiguous to determine, hence fuzzy logic strategy where the search criteria are not precisely bounded would be more appropriate than a crisp relation. Thus an adaptive inertia weight can be obtained from the fuzzy logic strategy thereby leading to an improved PSO technique termed as Fuzzy PSO (FPSO). The various sequential steps involved in the fuzzy implemented PSO based algorithm are as follows:

(i) The fuzzy logic inputs and output are decided and their feasible ranges are declared. The two fuzzy inputs are as follows:

$$Input\ 1 = f_{pi} / f_{max} \quad (16)$$

$$Input\ 2 = \text{Max} \{ (P_{mn,max} - P_{mn}^{pi}); (P_{nm}^{pi} - P_{mn,min}) \}$$

(17)

The *Input1* is the first essential factor and *Input 2* is an active search range determined as the maximum search distance or range pertaining to each element P_{mn} of particle I_{pi} in the present iteration from any of its corresponding limits (maximum or minimum). The output of the fuzzy logic strategy is the inertia weight w.
(ii) Fuzzification of inputs and output using triangular membership function. Five fuzzy linguistic sets have been used for each of the inputs and output as shown in Figure.2.

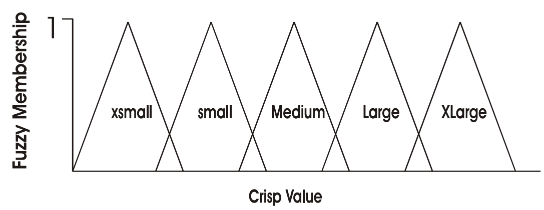

Figure.2. Fuzzy Membership function

(iii) The fuzzy rule base is formulated for all combinations of fuzzy inputs based on their ranges.

(iv) Defuzzification of output using Centroid method.

$$C = \sum_{i=1}^{5} x_i y_i \Big/ \sum_{i=1}^{5} y_i$$

(18)

Where x_i the mid-point of each fuzzy output set and y_i is its corresponding membership function value. The Centroid C is scaled (multiplied by its range) to obtain inertia weight value of each element in the particle.

6. SAMPLE SYSTEMS AND RESULTS

The proposed algorithm FPSO is tested on a standard IEEE 30-bus [3] test system, an interconnected two area test system formed by interconnecting two identical standard IEEE 30-bus systems through a tie-line of scheduled interchange (from area 1 to area 2) and an interconnected four area system. The standard IEEE 30-bus test system consists of 6 generating units, 41 lines and a total demand of 283.4 MW. The interconnected two area test system has a scheduled tie-line power flow of 20 MW between buses 3 and 26 corresponding to area 1 and 2 respectively. The four area interconnected system consists of four identical standard IEEE 30-bus systems with five tie-lines as shown in Figure. 3. The scheduled tie-line interchange $ST_{A\text{-}AA}$ are as follows :

$ST_{A1\text{-}A2}$ = 70 MW , $ST_{A2\text{-}A4}$ = 70 MW , $ST_{A4\text{-}A1}$ = 60 MW , $ST_{A1\text{-}A3}$ = 60 MW , $ST_{A3\text{-}A4}$ = 60 MW.

For all the test systems N_p is chosen as 10. The penalty factors k_1, k_2 and k_3 are chosen by trial and error. Initially a small value between 10 and 100 will be chosen. After the investigation if the constraint violated individuals have not been effectively eliminated then, the penalty factor values will be increased until a converged solution is reached with no constraint violations. Convergence is tested for 100 trial runs. The simulations were carried out on Pentium IV, 2.5 GHz processor.

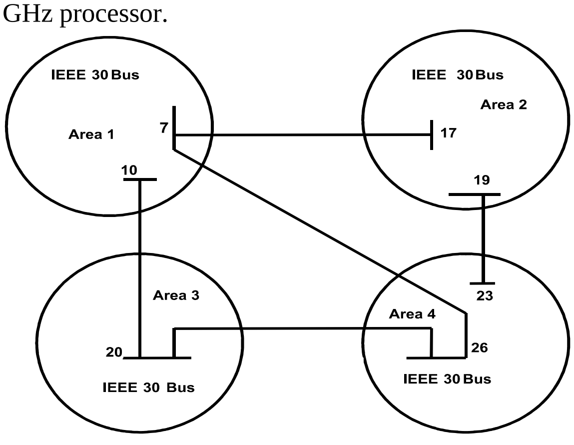

Fig.3 Four Area Interconnected system

The fuzzy logic data for mutation and recombination are presented in TABLE I and TABLE II respectively.

TABLE I

Data for Fuzzy Mutation

Fuzzy Set	Input 1	Input 2	Output
XSmall	0.00001 to 0.00004	10 to 30	0.001 to 0.005
Small	0.00003 to 0.006	25 to 50	0.004 to 0.06
Medium	0.005 to 0.05	40 to 80	0.04 to 0.08
Large	0.03 to 0.5	70 to 150	0.075 to 0.09
XLarge	0.4 to 1	140 to 190	0.085 to 0.1

TABLE II

Data for Fuzzy Recombination

Fuzzy Set	Input 3	Input 4	Output
XSmall	10 to 30	10 to 30	0.001 to 0.006
Small	25 to 50	25 to 50	0.004 to 0.08
Medium	40 to 80	40 to 80	0.07 to 0.09
Large	70 to 150	70 to 150	0.085 to 0.2
XLarge	140 to 190	140 to 190	0.15 to 0.3

The convergence characteristics of IEEE 30-bus test system corresponding to EP, FMEP, PSO and FPSO algorithms based single area DC-OPF (with same initial population) are shown in Figures. 4, 5 and 6 respectively. The convergence characteristics are drawn by plotting the minimum fitness value from the combined population across iteration or generation index.

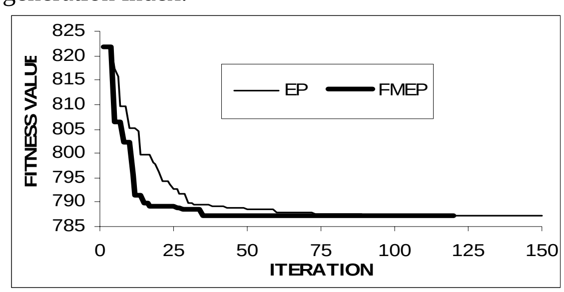

Fig.4 Convergence characteristic of EP and FMEP

From Figures. 4, 5 and 6 it is observed that the fitness function value converges smoothly to the optimum value without any abrupt oscillations, thus ensuring convergence reliability of the proposed algorithm and this algorithm have much better convergence than EP technique.

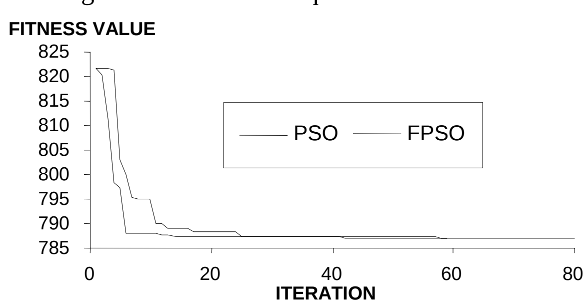

Fig.5 Convergence characteristic of PSO and FPSO

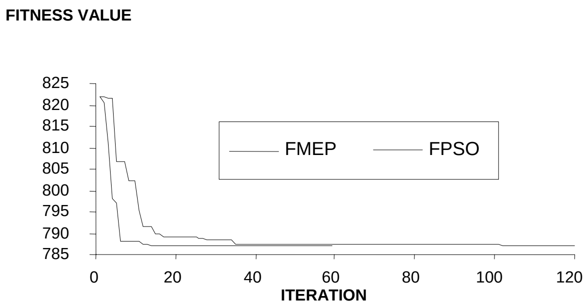

Fig. 6 Convergence characteristic of FMEP and FPSO

.

TABLE III

Optimal solution of IEEE 30-bus system

Algorithm	SLP	EP	FMEP	PSO	FGPSO
$P_{G\,S}$ (MW)	136.953	137.499	137.597	137.595	137.595
$P_{G\,2}$ MW)	59.13	60.5156	60.9731	60.9751	60.9759
P_{G5} (MW)	23.9	22.3237	22.4612	22.4610	22.4602
$P_{G\,8}$ (MW)	25.89	32.185	32.5847	32.5849	32.585
$P_{G\,11}$ MW)	14.61	15.8073	14.6537	14.6539	14.654
$P_{G\,13}$ MW)	22.88	15.0697	15.1303	15.1302	15.1302
Total Fuel cost ($/hr)	789.275	787.151	787.147	787.149	787.149
Max no of iterations	6	150	120	80	60
CPU time in ms	15	90	65	60	35

From Fig. 6 it is inferred that the FPSO has a faster convergence than FMEP. Similar convergence characteristics can be obtained for each decoupled equivalent single area DC-OPF sub-problem.

The optimal solutions of IEEE 30-bus test system using the proposed algorithms are compared with Successive Linear Programming (SLP)[3], EP [9] and PSO techniques and the results are presented in TABLE III.

From TABLE III it is inferred that for the same optimum the number of iterations or generations are low for FMEP[12] and FPSO algorithms than EP and PSO techniques respectively. Moreover the FPSO algorithm has a faster convergence (less number of iterations) than FMEP algorithm. Even though the number of iterations and CPU time are less for SLP, the optimal solution mainly depends on the initial conditions. It is observed that the deterministic SLP method initially suffers from oscillations and also the model becomes inaccurate when wider variations are allowed in the control variables. It is also observed that with infeasible initial solutions the proposed algorithm have smooth convergence

TABLE IV

Optimal solution of interconnected two area system using PSO and FPSO

Area	Area 1 sub-problem		Area 2 sub-problem	
Algorithm	PSO	FMPSO	PSO	FMPSO
$P_{G\,S}$ (MW)	136.649	136.998	136.593	136.35
$P_{G\,2}$ (MW)	67.3767	67.8788	57.4071	57.0314
P_{G5} (MW)	24.7875	25.6412	22.5134	21.7156
$P_{G\,8}$ (MW)	35	34.8767	23.4836	22.0071
$P_{G\,11}$ (MW)	18.7835	18.2925	11.4033	13.7489
$P_{G\,13}$ (MW)	20.803	19.7124	12	12.5471
Total Fuel cost ($/hr)	866.228	866.12	713.46	713.443
Max no of iterations	150	120	150	120
CPU time in ms	90	65	90	65

The optimal solutions of an interconnected two area test system using the proposed algorithms are compared with PSO and FPSO techniques and the results are presented in TABLE IV. From TABLE V it is observed that fuzzy implemented PSO algorithm have faster convergence.

The tie-line flow with respect to its scheduled value (20MW) is shown in Fig. 7 for FMEP and FPSO algorithms. Fig.7 shows the effectiveness of the proposed algorithms in maintaining the tie-line scheduled value.

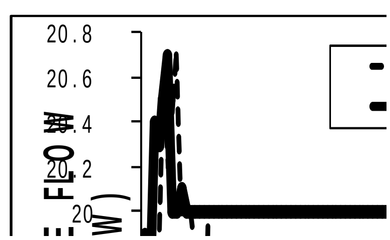

Fig. 7 Tie-line flow convergence using FMEP and FGPSO

In order to depict the effectiveness of this algorithm in maintaining the line flows within their limits, a critical line, 39 (between buses 29-30 with 1p.u flow limit) flow is shown in Fig. 7 for FMEP and FPSO algorithms.

TABLE V

Optimal solution of interconnected four area system for Area 1 and Area 2

Algorithm	EP	FMEP	PSO	FPSO
Area 1 sub-problem				
P_{GS} (MW)	142.83	141.307	141.55	142.131
P_{G2} (MW)	65.34	57.35	58.3	60.5
P_{G5} (MW)	50	50	50	50
P_{G8} (MW)	35	35	35	35
P_{G11} (MW)	10	19.76	18.51	15.15
P_{G13} (MW)	30.21	29.97	30.01	30.01
Total Fuel cost ($/hr)	1027.47	1027.09	1026.71	1026.2
Max no of iterations	150	120	80	60
CPU time in ms	90	65	50	35
Area 2 sub-problem				
P_{GS} (MW)	136.46	136.668	136.5	136.79
P_{G2} (MW)	57.01	57.187	57.21	59.38
P_{G5} (MW)	28.19	27.544	27.07	24.76
P_{G8} (MW)	23.609	22.144	23.194	31.86
P_{G11} (MW)	18.47	19.753	19.75	15.59
P_{G13} (MW)	19.64	20.102	19.35	15.25
Total Fuel cost ($/hr)	791.208	791.136	790.645	790.268
Max no of iterations	150	120	80	60
CPU time in ms	90	65	60	35

TABLE VI

Optimal solution of interconnected four area system for Area 3 and Area 4

Algorithm	EP	FMEP	PSO	FPSO
Area 3 sub-problem				
P_{GS} (MW)	137.218	137.465	137.465	137.69
P_{G2} (MW)	60.18	60.71	59.97	60.72
P_{G5} (MW)	24.23	22.77	23.08	22.95
P_{G8} (MW)	27.55	29.28	33.58	32.16
P_{G11} (MW)	19.21	17.08	14.13	16.29
P_{G13} (MW)	15.01	16.07	15.16	13.57
Total Fuel cost ($/hr)	788.054	787.314	787.261	787.162
Max no of iterations	150	120	80	60
CPU time in ms	90	65	60	35
Area 4 sub-problem				
P_{GS} (MW)	125.514	124.914	125.838	125.511
P_{G2} (MW)	35.908	35.3	34.73	35.88
P_{G5} (MW)	15	15.5	15.84	16.54
P_{G8} (MW)	10.572	12.64	11.98	10.26
P_{G11} (MW)	11.74	10	10	10.19
P_{G13} (MW)	12.65	12	12	12
Total Fuel cost ($/hr)	546.895	546.00	545.623	545.605
Max no of iterations	150	120	80	60
CPU time in ms	90	65	60	35

It can be observed that the proposed algorithm effectively eliminates the line limit violations.

The optimal solutions of an interconnected four area test system using the proposed algorithms are compared with EP and PSO techniques and the results are presented in TABLE V and TABLE VI respectively.

From TABLE V and TABLE VI it is observed that fuzzy implemented EP and PSO algorithms have faster convergence. The computation times taken by the proposed algorithms (FMEP and FPSO) are only 70% and 60% of the time taken by EP and PSO methods respectively. Also it is inferred that the proposed algorithms can be used to solve "n" number of areas / buses.

7. CONCLUSION

This paper presents a simple, efficient and reliable fuzzified PSO (FPSO) for solving multi-area DC-OPF problem. This paper demonstrates this algorithm with clarity, chronological development and by successful application of the proposed algorithm on standard test systems for solving multi-area DC-OPF problem. The results obtained from the proposed algorithms are compared with those obtained from SLP. The analysis reveals that PSO based algorithm converges faster than EP based algorithm and in both techniques the fuzzy implemented algorithms are much faster in convergence. The proposed algorithm have the potential to be applied to other power engineering problems such as, AC-OPF, SCOPF (Security Constrained Optimal Power Flow) since they can produce accurate optimum with fast convergence.

REFERENCES

[1] B. Stott and O. Alsac , "Optimal load flow with steady state security," *IEEE Trans on Power Apparatus and Systems*, Vol 93, pp. 745- 754,1974.

[2] James A Momoh, M.E. El-Hawary and Ramababu Adapa, " A review of Selected Optimal Power Flow literature to 1993 Part I and Part II, "*IEEE Trans on Power systems*, Vol 14, No.1, pp.96-111, Feb 1999.

[3] Anastasios G. Bakirtzis and Pandelis N. Biskas, "A decentralized solution to the DC-

OPF of interconnected power systems, " *IEEE Trans. Power Syst.*, vol. 13, pp.1007-1013, Aug.2003.

[4] A. J. Conejo and J. A. Aguado, "Multi-area coordinated decentralized DC Optimal Power Flow, " *IEEE Trans. Power Syst.*, vol. 13, pp. 1272-1278, Nov. 1998.

[5] J. Y. Fan and L. Zhang, "Real-time economic dispatch with line flow and emission constraints using quadratic programming, " *IEEE Trans. Power Syst.*, vol. 13, pp. 320-332, Feb. 1998.

[6] R.C. Eberhart, and J. Kennedy, "Particle swarm optimization". Proc. IEEE Int. Conf. Neural Netw., vol. 4, pp. 1942–1948, 1995.

[7] Abido M.A, "Optimal power flow using particle swarm optimization", Electrical Power and Energy Systems vol. 24 563-571, 2002.

[8] J.B. Park, K.-S. Lee, J.R. Shin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with non smooth cost functions," IEEE Trans. Power Syst., vol. 20(1), pp. 34–42, 2005.

[9] P. Venkatesh, R. Gnanadass, and N.P. Padhy, "Comparison and application of evolutionary programming techniques to combined economic and emission dispatch with line flow constraints", IEEE Trans. on Power Systems, vol.18(2), pp. 688- 697, 2003.

[10] K. P. Wong and J. Yuryevich, " Evolutionary-programming-based algorithm for environmentally-constrained economic dispatch, " *IEEE Trans. Power Syst.*, vol. 13, pp. 301-306, May. 1998.

[11] P. Somasundaram, K. Kuppusamy and R. P. Kumudini Devi, " Evolutionary Programming based security constrained optimal power flow, " *Electric Power Syst. Res.* 72, pp. 137-135, 2004.

[12] M.Y.El-sharkh, A.A.El-Keib,H.Chen. "A fuzzy evolutionary programming based solution methodology for security-constrained generation maintenance scheduling". *Electric Power system Research*, 2003. Vol. 67. PP67-72.