Economic Dispatch Problem using Shuffled Frog Leaping Algorithm

I. D. Soubache, P. Sudhakara Reddy

***Abstract*: *A new evolutionary algorithm known as the shuffled frog leaping algorithm is presented in this paper, to solve the economic dispatch (ED) problem of thermal plants. The proposed optimization technique can take care of economic dispatch problems involving constraints such as transmission losses, power balance and generation capacity. The feasibility of the proposed method is demonstrated for three units and six units systems, and is compared with Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) and methods in terms of the solution quality and computation efficiency. Compared with the other existing techniques, the proposed algorithm has been found to perform better in a number of cases. Considering the quality of the solution obtained, this method seems to be a promising alternative approach for solving the ED problems in practical power system*.**

***Index Terms*: *Shuffled frog leaping algorithm (SFLA), Economic Dispatch (ED), Particle Swarm Optimization (PSO), Genetic Algorithm (GA)*.**

I. INTRODUCTION

Economic dispatch problem is defined as the method of determining the optimal combination of power outputs for all generating units, which minimizes the total fuel cost of thermal power plants while satisfying load demand and operating constraints of a power system [1]. This makes the ED problem a large-scale non-linear constrained optimization problem.

In the traditional ED problem, each generator's cost function has been approximately represented by a single quadratic polynomial and can solved by using numerical programming based techniques such as lambda iteration method, gradient-based method [2].

These require incremental fuel cost curves which are piecewise linear and monotonically increasing to determine the global optimal solution. These techniques offer good results but when the search space is non-linear and it has discontinuities they become very complicated with a slow convergence ratio and not always seeking to the feasible solution. This makes the ED problem of finding the global optimum solution challenging. New numerical methods are needed to cope with these difficulties, especially those with high-speed search to the optimal and not being trapped in local minima.

Dynamic programming (DP) method [3] is one of the approaches to solve the non-linear and discontinuous ED problem, but it suffers from the problem of curse of dimensionality or local optimality.

In order to overcome this problem, the stochastic search algorithms such as genetic algorithm (GA) [4] [5], and simulated annealing (SA) [6] [7], may prove to be very effective in solving nonlinear ED problems without any restriction on the shape of the cost curves.

Although these heuristic methods do not always guarantee discovering the globally optimal solution in finite time, they often provide a fast and reasonable solution (suboptimal nearly global optimal).

SA is applied in many power system problems. But, the setting of control parameters of the SA algorithm is a difficult task and convergence speed is slow when applied to a real system [7]. Though, the GA method is usually faster than the SA method because the GA has parallel search techniques, which emulate natural genetic operations. The GA methods have been employed successfully to solve complex optimization problems [8] [9], recent research has identified some deficiencies in GA performance.

This degradation in efficiency is apparent in applications with highly epistatic objective functions. Moreover the premature convergence of GA degrades its performance and reduces its search capability that leads to a higher probability toward obtaining a local optimum [10].

The PSO has attracted many researchers' sights due to its simplicity and effectiveness. PSO, inspired from bird flocking and fish schooling, is a flexible, robust, population based algorithm [11] that are adopted by many people for solving ED problems and various power system problems [10] [12].

In this paper, a new integer-coded evolutionary algorithm known as shuffled frog leaping algorithm (SFLA) is used to solve the Economic dispatch problem. Two cases with three units and six units thermal power system are tested and compared with other approaches and found to be promising.

After the introduction, a brief description of the ED problem associated with its mathematical formulation is presented in Section II, while in Section III explains the shuffled frog leaping algorithm (SFLA). Section IV then details the proposed procedures. Cases study is presented in Section V. Finally, the conclusion is drawn in Section VI.

II. ECONOMIC LOAD DISPATCH PROBLEMS

The ELD may be formulated as a nonlinear constrained problem [2]. The convex ELD problem assumes quadratic cost function along with system power demand and operational limit constraints. As mentioned above, the objective of ED problems is to minimize the fuel cost of committed generators (units) subjected to operating constraints. Practically, the

Manuscript Received November, 2013.

Asst. Prof. I.D.Soubase, Department of Electrical Engineering, PRIST University, Vallam, Thanjavur, Tamil Nadu, India.

Dr. P. Sudhakara Reddy, Department of Electronics and Communication Engineering, Sri Kalahasteeswara Institute of Technology, Srikalahasti, Chittoor District, Andhra Pradesh, India.

economic power dispatch problem is usually formulated as,

Minimize

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

Subject to

$$\sum_{i=1}^{n} P_i = P_D + P_L \quad (2)$$

$$P_i^{Min} \le P_i \ge P_i^{Max} \quad (3)$$

where, n is the number of units, F_T is the total fuel cost, F_i and P_i are the cost function and the real power output of ith unit respectively, P_D is the total demand, P_L is the transmission loss. P_i^{Min} and P_i^{Max} are the lower and upper bounds of the ith unit respectively. The equality constraint, Eqn.(2) states that the total generated power should be balanced by transmission losses and power consumption while Eqn.(3) denoting unit's operation constraints.

Traditionally, the fuel cost of a generator is usually defined by a single quadratic cost function,

$$F_i(P_i) = \gamma_i P_i^2 + \beta_i P_i + \alpha_i \quad (4)$$

where, α_i, β_i, and γ_i are cost coefficients of the ith unit.

Conventionally, transmission loss is calculated using B-matrix loss formula [2], i.e.,

$$P_L = P^T BP + P^T B0 + B00 \quad (5)$$

where, P denotes the real power output of the committed units in vector form, and B, $B0$ and $B00$ are loss coefficients in matrix, vector and scalar respectively, which are assumed to be constant, and reasonable accuracy can be achieved when the actual operating conditions are close to the base case where the B-coefficients were derived. In the summary, the objective of economic power dispatch optimization is to minimize F_T subject to the constraints Eqn.(2) and (3).

III. SHUFFLED FROG LEAPING ALGORITHM

The SFLA is a meta-heuristic optimization method which is based on observing, imitating, and modeling the behavior of a group of frogs when searching for the location that has the maximum amount of available food [13]. SFLA, originally developed by Eusuff and Lansey in 2003, can be used to solve many complex optimization problems, which are nonlinear, non-differentiable, and multi-modal [14].

SFLA has been successfully applied to several engineering optimization problems such as water resource distribution [15], bridge deck repairs [16], job-shop scheduling arrangement [17], and traveling salesman problem (TSP) [18]. The most distinguished benefit of SFLA is its fast convergence speed [19]. The SFLA combines the benefits of the both the genetic-based memetic algorithm (MA) and the social behavior-based PSO algorithm [20]. In SFLA, there is a population of possible solutions defined by a set of virtual frogs partitioned into different groups which are described as memeplexes, each performing a local search. Within each memeplex, the individual frogs hold ideas, which can be infected by the ideas of other frogs. After a defined number of memetic evolution steps, ideas are passed between memeplexes in a shuffling process. The local search and the shuffling process continue until the defined convergence criteria are satisfied [19], [21].

The flowchart of SFLA is illustrated in Fig. 1. In the first step of this algorithm, an initial population of P frogs is randomly generated within the feasible search space. The position of the ith frog is represented as $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$, where D is the number of variables. Then, the frogs are sorted in descending order according to their fitness. Afterwards, the entire population is partitioned into m subsets referred to as memeplexes, each containing n frogs (i.e., $P = m \times n$). The strategy of the partitioning is as follows: the first frog goes to the first memeplex, the second frog goes to the second memeplex, the mth frog goes to the mth memeplex, the $(m+1)$th frog goes back to the first memeplex, and so forth. In each memeplex, the positions of frogs with the best and worst fitnesses are identified as X_b and X_w, respectively.

Also the position of a frog with the global best fitness is identified as X_g. Then, within each memeplex, a process similar to the PSO algorithm is applied to improve only the frog with the worst fitness (not all frogs) in each cycle. Therefore, the position of the frog with the worst fitness leaps toward the position of the best frog, as follows:

$$D_i = rand * (X_g - X_w) \quad (1)$$

$$X_w^{new} = X_w^{current} + D_i \quad (D_{i\min} < D_i < D_{i\max}) \quad (2)$$

Where, D_{imax} and D_{imax} are the maximum and minimum step sizes allowed for a frog's position, respectively.

If this process produces a better solution, it will replace the worst frog. Otherwise, the calculations in (1) and (2) are repeated but X_b is replaced by X_g. If there is no improvement in this case, a new solution will be randomly generated within the feasible space to replace it. The calculations will continue for a specific number of iterations [14], [17]. Therefore, SFLA simultaneously performs an independent local search in each memeplex using a process similar to the PSO algorithm. The flowchart of local search of SFLA is illustrated in Fig. 2.

After a predefined number of memetic evolutionary steps within each memeplex, the solutions of evolved memeplexes $(X_1, \ldots\ldots, X_P)$ are replaced into new population (new population = $\{X_k, k = 1, \ldots, P\}$); this is called the shuffling process. The shuffling process promotes a global information exchange among the frogs. Then, the population is sorted in order of decreasing performance value and updates the population best frog's position X_g, repartition the frog group into memeplexes, and progress the evolution within each memeplex until the conversion criteria are satisfied. Usually, the convergence criteria can be defined as follows [22]:

The relative change in the fitness of the global frog within a number of consecutive shuffling iterations is less than a pre-specified tolerance.

The maximum predefined number of shuffling iteration has been obtained.

IV. IMPLEMENTATION

In SFLA programming a number of parameters need to be adjusted to compute best optimal value of the variables *i.e.*, population size, number of memplexes, and number of global and local iteration.

1) Population size: it is a number of set of variables, defined as a total number of frogs. In this simulation procedure, population size has been taken as 100. Increase in number of population means good accuracy but it will lead to more propagation delay. After running the program with different number of population size, it has been observed that for this optimization problem, typically a population size of 100 is most suited for optimizing both processing time and value.

2) Number of memplexes: In this programming, number of memplexes is fixed at 10. As population size and number of memplexes are user input, the given input of number of memplexes is such that there exists a certain number of frogs (population size/total number. of memplexes) in each memplexes.

3) Number of global iteration: In this type of iteration, the cross-over between best frog & worst frog is done taking the whole population. One global iteration consists of local iterations as many as number of memplexes present. It is taken 10 here. Maximizing the number of global iteration gives more accurate results but it takes more time to process.

4) Number of local iteration: In this type of iteration, the cross-over between best frog & worst frog is done in every single memplexes. Number of local iterations are taken as 20 here. Maximizing the number of local iterations also gives more accuracy but it gives more delay.

All the SFLA parameters value discussed above is for three units and six units test system.

Shuffled Frog Leaping Algorithm

Step 1: start

Step 2: population size (n), no. of memeplexes (m), number of local search within each memeplexes and number of global search are given as inputs.

Step 3: generate population of frogs (F) randomly from the given data.

Step 4: evaluate fitness of F.

Step 5: sort F in descending order.

Step 6: cross-over between worst frog (Fw) and best frog (Fb) is done to get two new off-springs.

Step 7: replace Fb and Fw with two best frogs (according to their fitness) from four frogs (two parents and two offsprings).

Step 8: partition F into m memeplexes such that each memeplexes gets (F/m) frogs.

Step 9: find Fb and Fw from each memeplexes and do cross-over between them.

Step 10: get two new offspring from them and replace Fb and Fw with two best frogs (according to their fitness) from four frogs.

Step 11: check whether number of local search is completed or not, if not then go to step 9.

Step 12: if local search is completed then check whether number of global search is completed or not, if not then go to step 5.

Step 13: if global iteration is completed then get the best solution (best fitness) from F.

Step 14: end.

It has been observed from the above algorithm that proposed SFLA performs two simultaneous crossovers, i.e., global (Step 6 and 7) and local (Step 9 and 10) search to produce new offspring which gives better result compared to other optimization algorithms.

To verify the effectiveness and capability of the proposed method, case studies for two different units are conducted and the results are reported in the next section. The flowchart of SFLA is shown in the figure 1 and 2.

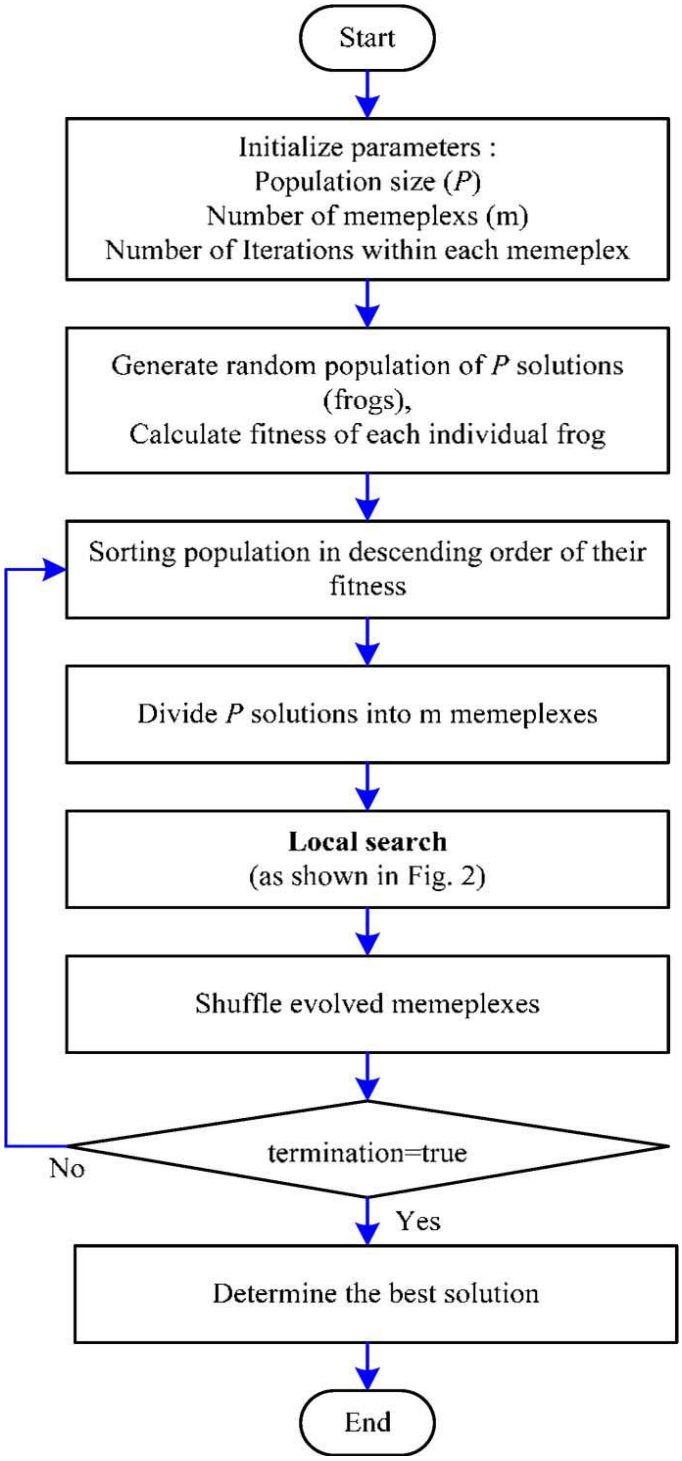

Figure 1 Flowchart of SFLA

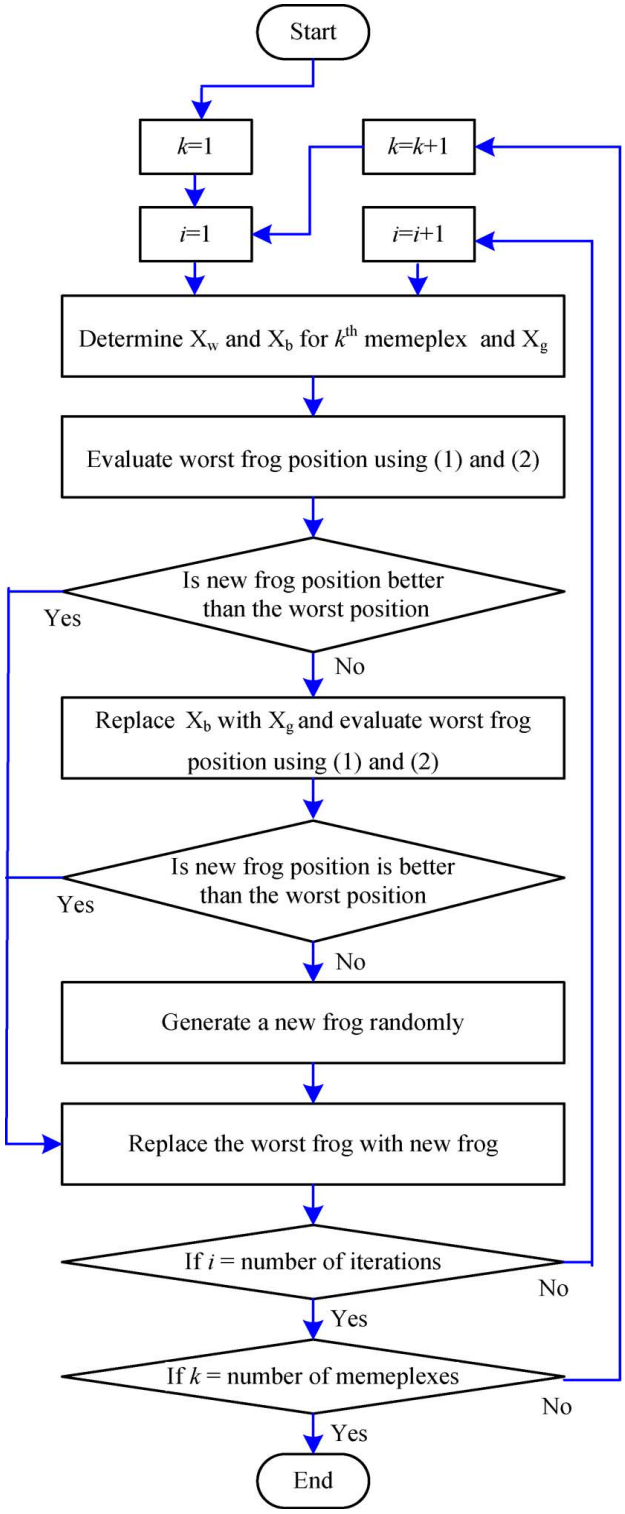

Figure 1 Flowchart of local search

V. EXPERIMENTAL RESULTS

Proposed SFLA Algorithm has been applied to ED problems in two different thermal unit systems for verifying its feasibility. These are a three units system and a six units system [4] [10]. The transmission losses will not take into account in all the case studies here for the sake of comparison with other algorithms presented in literature [4] [5] [6] [12] [23]. The stopping criterion, maximum number of iteration, varies for each case in considering the problem scale. The software has been written in MATLAB language and executed in Pentium ® Dual-Core personal computer with 2GB RAM.

A. Case study 1- Three units system

In this example, a simple system with three thermal units is used to demonstrate how the proposed approach works. The unit characteristics are given in Table 1. In this case, each individual P_g contains three generator power outputs, such as P_1, P_2, and P_3, which are generated randomly. The dimension of the population is equal to 3 X 100. Now, Table 2 provides the statistic results that involved the generation cost, evaluation value, and average CPU time. Figure 3 showed the distribution outline of the best solution for 500 iterations.

TABLE 1
GENERATING UNIT'S CAPACITY AND COEFFICIENTS

Unit	P_{min} MW	P_{max} MW	α $	β $/MW	γ $/MW^2
1	50	250	328.13	8.663	0.00525
2	5	150	136.91	10.040	0.00609
3	15	100	59.16	9.760	0.00592

In normal operation of the system, the loss coefficients with the 100-MVA base capacity are as follows,

$$B_{ij} = \begin{bmatrix} 0.000136 & 0.0000175 & 0.000184 \\ 0.0000175 & 0.000154 & 0.000283 \\ 0.000184 & 0.000283 & 0.00161 \end{bmatrix}$$

Load = 300 MW

TABLE 2
BEST POWER OUTPUT FOR 3-GENERATOR SYSTEM

Unit Output	GA	PSO	SFLA
P1 (MW)	208.99	209.001	204.34
P2 (MW)	86.0041	85.92	89.97
P3 (MW)	15.4163	15	15.01
Total Power Output (MW)	310.4099	309.9211	309.32
Total Generation Cost ($/h)	3624.28	3621.75	3618.64
Power Loss (MW)	10.4099	9.9833	9.82
Average CPU time (sec)	0.0028	0.064	4.758

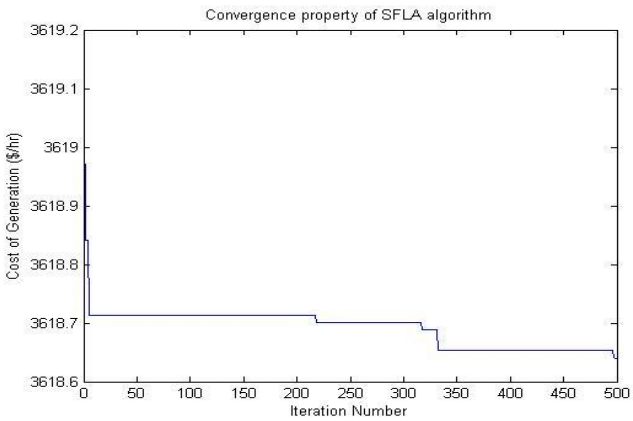

Figure 3 Convergence characteristic of Three-generator system

B. Case study 2- Six units system

The system contains six thermal units, 26 buses, and 46 transmission lines [10]. The load demand is 1263MW. The characteristics of the six thermal units are given in Tables 3. In this case, each individual P_g contains six generator power outputs, such as P_1, P_2, P_3, P_4, P_5 and P_6, which are generated randomly. The dimension of the population is equal to 6 X 100. Now, Table 4 provides the statistic results that involved the generation cost, evaluation value, and average CPU time. Figure 4 showed the distribution outline of the best solution for 500 iterations.

In normal operation of the system, the loss coefficients with the 100-MVA base capacity are as follows,

$$B_{ij} = \begin{bmatrix} 0.0017 & 0.0012 & 0.0007 & -0.0001 & -0.0005 & -0.0002 \\ 0.0012 & 0.0014 & 0.0009 & 0.0001 & -0.0006 & -0.0001 \\ 0.0007 & 0.0009 & 0.0031 & 0 & -0.001 & -0.0006 \\ -0.0001 & 0.0001 & 0 & 0.0024 & -0.0006 & -0.0008 \\ -0.0005 & -0.0006 & -0.001 & -0.0006 & 0.0129 & -0.0002 \\ -0.0002 & -0.0001 & -0.0006 & -0.0008 & -0.0002 & 0.015 \end{bmatrix}$$

$$B0 = 10^{-3}\begin{bmatrix} -0.3908 & -0.1297 & 0.7047 & 0.0591 & 0.2161 & -0.6635 \end{bmatrix}$$

$$B00 = 0.056$$

TABLE 3
GENERATING UNIT'S CAPACITY AND COEFFICIENTS

Unit	P_{min} MW	P_{max} MW	α $	β $/MW	γ $/MW^2
1	100	500	240	7.0	0.0070
2	50	200	200	10.0	0.0095
3	80	300	220	38.5	0.0090
4	50	150	200	11.0	0.0090
5	50	200	220	10.5	0.0080
6	50	120	190	12.0	0.0075

Load = 1263 MW

TABLE 4
BEST POWER OUTPUT FOR 6-GENERATOR SYSTEM

Unit Output	GA	PSO	SFLA
P1 (MW)	474.8066	447.497	447.12
P2 (MW)	178.6363	173.3221	172.00
P3 (MW)	262.2089	263.4745	261.98
P4 (MW)	134.2826	139.0594	143.04
P5 (MW)	151.9039	165.4761	164.64
P6 (MW)	74.1812	87.128	86.90
Total Power Output (MW)	1276.03	1276.01	1275.68
Total Generation Cost ($/h)	15,459	15,450	15447.44
Power Loss (MW)	13.0217	12.9584	12.88
Average CPU time (sec)	41.58	14.89	3.886

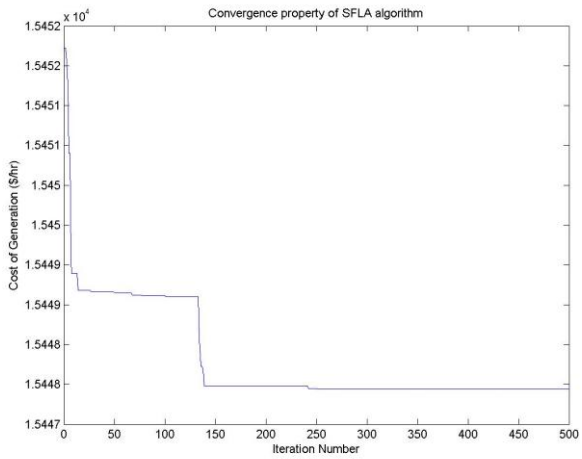

Figure 4 Convergence characteristic of Six-generator system

VI. CONCLUSION

The proposed shuffled frog leaping algorithm has been successfully implemented to solve ED problems with the generator constraints as linear equality and inequality constraints and also considering transmission loss. The algorithm is implemented for three units and six units system. From the result, it is clear that the proposed algorithm has the ability to find the better quality solution and has better convergence characteristics, computational efficiency and less average CPU time when compared to other methods such as PSO and GA.

ACKNOWLEDGMENT

The authors wish to thank the authorities of PRIST University, Vallam, Thanjavur, Tamil Nadu, India for the facilities provided to prepare this paper.

REFERENCES

1. B. H. Chowdhury and S. Rahman, "A review of recent advances in economic dispatch," IEEE Trans. Power Systems, vol. 5, no. 4, pp. 1248–1259, Nov. 1990.
2. A. J. Wood and B. F. Wollenberg, Power generation operation and control, 2nd ed. John Willy and Sons, 1996.
3. Z. X. Liang and J. D. Glover, "A zoom feature for a dynamic programming solution to economic dispatch including transmission losses", IEEE Trans. on Power Systems, vol. 7, no. 2, pp. 544-550, May 1992.
4. P. H. Chen and H. C. Chang, "Large Scale Economic Dispatch by Genetic Algorithm", IEEE Trans. on Power Systems, vol. 10, no. 4, Nov. 1995.
5. A. Bakirtzis, V. Petridis, and S. Kazarlis, "Genetic algorithm solution to the economic dispatch problem," Proc. Inst. Elect. Eng.–Gen., Transm. Dist., vol. 141, no. 4, pp. 377–382, July 1994.
6. K. P. Wong and Y. W. Wong, "Genetic and genetic/simulated — Annealing approaches to economic dispatch," Proc. Inst. Elect. Eng., pt. C, vol. 141, no. 5, pp. 507–513, Sept. 1994.
7. D. N. Simopoulos, S. D. Kavatza and C. D. Vournas, "Unit commitment by an enhanced simulated annealing algorithm," IEEE Trans. on Power Systems, vol. 21, No. 1, Feb. 2006.
8. T. Yalcionoz, H. Altun, and M. Uzam, "Economic dispatch solution using a genetic algorithm based on arithmetic crossover," in Proc. IEEE Proto Power Tech. Conf., Proto, Portugal, Sept. 2001.
9. C.-L. Chiang, "Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels," IEEE Trans. on Power Systems, vol. 20, no. 4, pp. 1690–1699, Nov. 2005.
10. Z. L. Gaing, "Particle Swarm Optimization to Solving the Economic Dispatch Considering the Generator Constraints", IEEE Trans. on Power Systems, vol. 18, No. 3, Aug. 2003.

11. J. Kennedy and R. Eberhart, "Particle swarm optimization", Proc. IEEE Int. Conf. Neural Networks, vol. 4, pp. 1942–1948, 1995.
12. J. B. Park, K. S. Lee, J. R. Shin and K. Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions", IEEE Trans. on Power Systems, vol.20, no.1, Feb 2005.
13. M. M. Eusuff, K. E. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization," Eng. Optimiz., vol. 38, no. 2, pp. 129–154, 2006.
14. X. Zhang, X. Hu, G. Cui, Y.Wang, and Y. Niu, "An improved shuffled frog leaping algorithm with cognitive behavior," in Proc. 7th World Congr. Intelligent Control and Automation, 2008.
15. M. Eslamian, S. H. Hosseinian, and B. Vahidi, "Bacterial foragingbased solution to the unit-commitment problem," IEEE Trans. Power Syst., vol. 24, no. 3, pp. 1478–1488, Aug. 2009.
16. H. Elbehairy, E. Elbeltagi, and T. Hegazy, "Comparison of two evolutionary algorithms for optimization of bridge deck repairs," Computer- Aided Civil Infrastr. Eng., vol. 21, pp. 561–572, 2006.
17. A. Rahimi-Vahed and A. H. Mirzaei, "Solving a bi-criteria permutation flow-shop problem using shuffled frog-leaping algorithm," in Soft Computing. New York: Springer-Verlag, 2007.
18. X.-H. Luo, Y. Yang, and X. Li, "Solving TSP with shuffled frog-leaping algorithm," in Proc. ISDA 2008, vol. 3, pp. 228–232.
19. E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," Adv. Eng. Informat., vol. 19, no. 1, pp. 43–53, 2005.
20. J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in Proc. IEEE Conf. Neural Networks, 1995, vol. 4, pp. 1942–1948.
21. M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," J. Water Resour. Plan. Manage., vol. 129, no. 3, pp. 210–225, 2003.
22. T. H. Huynh, "A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers," in Proc. ICIT 2008, pp. 1–6.
23. M. Vanitha and K. Thanushkodi, "Solution to Economic Dispatch Problem by Differential Evolution Algorithm Considering Linear Equality and Inequality Constrains," IJRRECE, vol. 1, no. 1, March 2011

AUTHOR PROFILE

I. D. Soubache received his B.Tech degree from Sri Manakula Vinayagar Engineering College, Madagadipet, Puducherry, India in 2004 and M.E degrees from Faculty of Engineering and Technology, Annamalai University, Annamalai Nagar, Chidambaram, India in 2006. Currently he is working as Head of the Department in the Department of Electrical Engineering, PRIST University, Puducherry campus, Puducherry. He is currently working towards the Ph.D. degree, Department of Electrical Engineering, PRIST University, Vallam, Thanjavur, Tamil Nadu, India. His research interest includes power system operation and control.

Dr. P. SUDHAKARA REDDY Associate Professor, Department of Electronics and Communication Engineering, Sri Kalahasteeswara Institute of Technology, Srikalahasti, Chittoor District, Andhra Pradesh. He received B.Tech degree in Electronics and Communications Engineering from N.B.K.R Institute of Science and Technology, Sri Venkateswara University, Tirupathi, Chittoor district, Andhra Pradesh, India in 1993; M.Tech degree in Digital Systems and Computer Electronics from JNTU Hyderabad, Andhra Pradesh, India in 1999 and Ph.D. degree received from Sri Venkateswara University, Tirupathi, Chittoor district, Andhra Pradesh, India in 2010. His area of research goes with VLSI Arrays for Signal/Image/Video Processing and Communications.