International Journal of Latest Trends in Engineering and Technology
Vol.(9)Issue(2), pp.017-024
DOI: http://dx.doi.org/10.21172/1.92.05
e-ISSN:2278-621X

IMPLEMENTATION OF PARTICLE SWARM OPTIMIZATION AND DIFFERENTIAL EVOLUTION APPLICATION TO LARGE-SCALE UNIT COMMITMENT PROBLEM

Palla Shailesh Kumar[1] & M. Ramu[2]

Abstract- This paper suggests the solving of unit commitment problem (UCP) using Lagrangian relaxation incorporating particle swarm optimization (PSO) and differential evolution (DE). The objective is to obtain optimal cost meeting all the required constraints. PSO is applied for initializing and Lagrangian multiplier update purpose. For large scale problems PSO as modified in operation i.e. Particles "flying out" of the simplex vicinity is a primary limitation of PSO because the computational time is increased while bringing back the "flown away" particle or by completely neglecting that particular particle, vice versa. This limitation is suppressed by modifying the method by operating it in an enclosed hyper cube before it attains a non-feasible point. Another method Differential evolution technique is quite popular among various evolutionary techniques existing, due to its impeccable accuracy and swift responses. But in some cases due to its swift response abilities, DE may tend to drop regional optimums. We introduce a dynamic mutation factor in the solution algorithm for surpassing this limitation for large scale problems. In this paper we implement Particle Swarm Optimization and Differential Evolution together in large-scale Unit Commitment Problems.The mitigation of the objective function for satisfying the minimum up and down time constraints, start-up cost and spinning reserve is regarded as the problem formulation of the unit commitment. The suggested method is tested for 10 units to 100 units with 24 hours' demand horizon and was verified with various pre-existing methods to verify its effectiveness.

Keywords- Unit Commitment (UC), Modified Particle Swarm Optimization (MPSO), Improved Differential Evolution (IDE)

1. INTRODUCTION

In Recent times, one of the most important commodities for human life indeed is electricity. Shortage of electricity creates a lot of havoc in daily life of people. Along with increase in population, the need for electricity is picking up a quick pace (i.e. the required load demand). Due to this, tremendous pressure is building up on the electrical utility for maintaining a reliable power supply while simultaneously maintaining their economical stand. For this to happen unit commitment is playing a vital role. The study of unit commitment displays different methods underlying for starting or shutting down of generating units based on forecasted load demand requirements [1]. Because of its strong importance in electrical power generation, many researchers came up with various methods for solving this problem.

Some methods include: Priority list method [2] which is a primitive and fastest way to solve UCP, it consists of priority wise arrangement of generation units based on the load but its operational cost is substantially high. Dynamic programming approach [3] is a classical method which is being extensively employed due to its pragmatic nature, its main drawback is dimensionality i.e. on increase in number of units' leads to a very complex problem. Branch and bound [4] is also a classical method which takes an unusually more time for solving large problems and suffers from lacking of memory storages. Coming to Lagrangian relaxation method [5] has significantly reduced the duality gap for finding the feasible outcomes but the efficiency in these outcome depends entirely on the choice of Lagrangian multiplier and corresponding updating methodologies employed in the problem. Few non classical methods are also existing; among them: Fuzzy Logics [6], Neural Networks [7], Genetic algorithms [8], Simulated Annealing [9], Tabu search method [10], Particle swarm optimization [11] is a sensible and noble approach that can be considered for solving which is stably being applicable from a decade due to its simplicity and strength of optimising. Lastly, differential evolution [13]is a good evolutionary algorithm etc.; are worth to be mentioned.

Though there are many new methods coming into picture for solving UCP but unfortunately no method can be considered as absolutely flawless to study this particular area. In this paper, these flaws have been nullified by making the requisite modifications in PSO and DE [14] and are being presented together as Implementation of Particle Swarm Optimization and Differential Evolution Application to large-scale Unit Commitment Problem.

[1] Department of Electrical and Electronics Engineering, Gitam University, Visakhapatnam, A.P. 530045, India.
[2] Department of Electrical and Electronics Engineering, Gitam University, Visakhapatnam, A.P. 530045, India.

2. PROBLEM FORMULATION

The primary function of UCP is to totally curtail the generation costs in a stipulated time (i.e. one day) under specific constraints like spinning reserve, as its boundaries. Objective function of UC to be minimized is [9].

$$F(S_i^t, V_{i,t}) = \sum_{t=1}^{T} \sum_{i=1}^{N} [F_i(S_i^t) + ST_{i,t}(1 - V_{i,t-1})]V_{i,t} \tag{1}$$

Subject to following constraints

2.1 Power Balance Constraint –

$$\sum_{i=1}^{N} S_i^t V_{i,t} = S_D^t \tag{2}$$

2.2 Spinning Reserve Constraint –

$$\sum_{i=1}^{N} S_{i,\max} V_{i,t} \geq S_D^t + R^t \tag{3}$$

2.3 Generation Limit Constraint –

$$S_{i,\min} V_{i,t} \leq S_i^t \leq S_{i,\max} V_{i,t}, \; i = 1, \ldots, N \tag{4}$$

2.4 Min up time and down time Constraints –

$$V_{i,t} = \begin{cases} 1, & if\, T_{i,on} < T_{i,up}, \\ 0, & if\, T_{i,off} < T_{i,down}, \\ 0\, or\, 1, & otherwise \end{cases} \tag{5}$$

2.5 Start up Cost –

$$ST_{i,t} = \begin{cases} HSC_i \; if\, T_{i,down} \leq T_{i,off} \leq T_{i,cold} + T_{i,down}, \\ CSC_i \; if \; T_{i,off} > T_{i,cold} + T_{i,down} \end{cases} \tag{6}$$

3. LAGRANGIAN RELAXATION

Mitigation or tranquilizing of coupling constraints present in UCP is accomplished by LR, which is indeed realized through dual optimization method [9].

$$L(S,V,\lambda,\mu) = F(S_i^t, V_{i,t}) + \sum_{i=1}^{T} \lambda^t (S_D^t - \sum_{i=1}^{N} S_i^t V_{i,t}) + \sum_{t=1}^{T} \mu^t (S_D^t + R^t - \sum_{i=1}^{N} S_{i,\max} v_{i,t}) \tag{7}$$

With respect to nonnegative λt and μt, whereas minimizing it with respect to other control variables in problem, that is

$$q^*(\lambda,\mu) = Max_{\lambda,\mu}\, q\,(\lambda,\mu) \tag{8}$$

Where

$$q\,(\lambda,\mu) = Min\, S_i^t, U_{i,t}\; L\,(S,S,\lambda,\mu) \tag{9}$$

Equations (2) & (3) are coupling constraints across the thermal units.
Lagrangian function is rewritten as

$$L = \sum_{i=1}^{N} \sum_{t=1}^{T} \{[F(S_i^t) + ST_{i,t}(1 - V_{i,t-1})]V_{i,t} - \lambda^t S_i^t V_{i,t} - \mu^t S_{i,\max} V_{i,t}\} + \sum_{t=1}^{T} (\lambda^t S_D{}^t + \mu^t (S_D{}^t + R^t)) \tag{10}$$

$$\sum_{t=1}^{T} \{[F(S_i^t) + ST_{i,t}(1 - V_{i,t-1})]V_{i,t} - \lambda^t S_i^t V_{i,t} - \mu^t S_{i,\max} V_{i,t}\}$$

Since coupling constraints are excluded, thermal units can mitigate this term individually afterwards. Over the stipulated amount of time the best value for LR function is found out for every individual unit i.e,

$$Min\, S_i, V_{i,t}\, L(S,V,\lambda,\mu) = \sum_{t=1}^{N} \min \sum_{t=1}^{T} \{[F(S_i^t) + ST_{i,t}(1 - V_{i,t-1})]V_{i,t}\} - \lambda^t S_i^t V_{i,t} - \mu^t S_{i,\max} V_{i,t} \tag{11}$$

Subjected to $V_{i,t} S_{i,\min} \leq S_i^t \leq V_{i,t} S_{i,\min}$

For t=1,...,T and the constraints in equation (5)

On/Off commitment guidelines:

Here, Dynamic programming is employed for achieving a dual outcome, for each and every individual unit as depicted in fig1, showing the only two possible states for unit i (i.e. $U_{i,t} = 0 or 1$). At $U_{i,t} = 0$ state, the value of the function to the minimized is unimportant (i.e., it equals zero), at the state where $U_{i,t} = 1$, the mitigated function is $\lfloor F_i(P_i^t) - \lambda^t P_i^t \rfloor$ the start-up cost and the term $\mu^t P_{i,\max}$ are dropped here since the minimization is with respect to P_i^t

Min $\lfloor F_i(P_i^t) - \lambda^t P_i^t \rfloor$ is mitigated to calculate dual power with the help of optimality condition

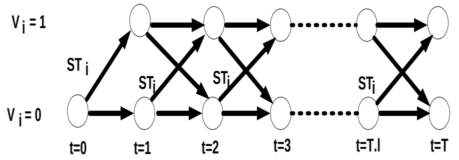

Figure 1. Two-state dynamic programming

$$\frac{d}{dP_i^t}\left[F_i(P_i^t) - \lambda^t P_i^t\right] = 0 \qquad (12)$$

The solution to this equation is

$$\frac{dF_i(S_i^{t,dual})}{dS_i^t} = \lambda^t \qquad (13)$$

The dual power is obtained

$$S_i^{t,dual} = \frac{\lambda^t - b_i}{2C_i} \qquad (14)$$

Three cases to check $P_i^{t,opt}$ against its limits

A. If $S_i^{t,dual} < S, then S_i^t = S_{i\min}$

B. If $S_{i\min} \leq S_i^{t,dual} \leq S_{i\max}$

Then $S_i^t \leq S_i^{t,dual} \leq S_{i\min}$

C. If $S_i^{t,dual} > P_{i\max}$ Then $S_i^t = S_{iMax}$

Dynamic programming is employed to decide the optimal schedule of each unit over the scheduled horizon. More specifically, for each state in each hour, the on/off decision making is needed to select the lowest cost among them by conducting standard evaluation of the start-up cost and accumulated costs. The dual power calculated will be substituted in the new On/Off decision criterion.

$$\lfloor F_i(S_i^t) + ST_{i,t}(1 - V_{i,t-1}) \rfloor - \lambda^t S_i^t - \mu^t S_i^t - \mu^t S_{i,\max}^t \qquad (15)$$

i = 1... NP

To mitigate the above term in equation (15) at every individual hour,

If

$$\{\lfloor F_i(S_i^t) + ST_{i,t}(1 - V_{i,t-1}) \rfloor - \lambda^t S_i^t - \mu^t S_i^t - \mu^t S_{i,\max}^t\} \leq 0$$

This unit will be committed, if it does not violate the minimum downtime constraint $U_{i,t} = 1$.

4. PROPOSED MODIFIED PARTICLE SWARM OPTIMIZATION FOR LARGE SCALE UC PROBLEM

This ability of PSO to obtain optimal outcomes in n-dimensional search space with a good response rate, makes PSO stand unique globally.

PSO concentrates on timely modifying velocities of particles so that they march towards their respective P_{best} and L_{best} values upon each time step. Whereas random numbers are fed as inputs for necessary acceleration towards attaining P_{best} and L_{best} values. G_{best} (i.e global best) is best value of particle among total population implying to condition that all neighbour hoods are considered.

$$v_i^{k+1} = w_k v_i^k + c_1 rand_1 (P_{besti} - s_i^k) + c_2 rand_2 (G_{best} - s_i^k) \tag{16}$$

And $s_i^{k+1} = s_i^k + v_i^{k+1}$ (17)

In large scale problems due to its fast response characteristics, some particles tend to fly away from their vicinities which indeed causes stress on the optimizer in deciding either to bring back the particle or to neglect them. This entire process eventually contributes to increase in the calculation time to a large extent. So, this serious limitation of PSO for large scale system is surpassed by modifying the optimizer, by making it optimize initially, in a regular hypercube before it attains a feasible point. Then in our search space an optimal function “F” is assigned to quickly identify the target optimal value.
For illustration, consider a “r” factored model in which we want to find out t- point optimal value.

Let $m = t*(r+1)$ and let $\varpi = [0,1]^m$ denote the m-dimensional hypercube. Define the m*1 vector, $\tilde{\psi} = (s_1', ..., s_t', f') \in \varpi$, where x_i is a r*1 vector in $[0,1]^r, i = 1, ..., t, f \in [0,1]^t$ and define $\varpi^* = \varpi \backslash \{\tilde{\psi} = (s_1', ..., s_t', f')' \in \varpi | 1_t'.f = 0 or 1_r'.s_i = 0 for some i\}$. to transform $\tilde{\psi}$ into a proper model ψ we define the function $F : \varpi^* \to (S^{r-1})^t * S^{t-1}$ by

$$F(\tilde{\psi}) = \left(\frac{s_1'}{(1_r'.s_1)}, ..., \frac{s_t'}{(1_r'.s_t)}, \frac{f'}{(1_t'.f)} \right)' \tag{18}$$

The function F is invariant in the sense that $F \circ F(\tilde{\psi}) = F(\tilde{\psi})$ and the model ψ is based on $\tilde{s}_i' = s_i' / (1_r'.s_i), i = 1, ..., t$ and the components in $\tilde{f}' = f' / (1_k'.f)$ are their corresponding weights. The notation $\psi = F(\tilde{\psi})$ signifies that the design ψ is transformed from $\tilde{\psi}$ via the function F.
Now our modified PSO algorithm is dependent on the optimal function F in Eq (18) which is employed as:
Firstly, we take a random population of “n” with “t” design points from ϖ^*. For differentiating the modification, we introduce two new notations i.e. let $\tilde{\psi}_i^{pbes}$ be the best position of ith particle as of P_{bes}. Then let $\tilde{\psi}^{gbes}$ be the global best i.e. best value of all particles among the total population as of G_{besi} previously. The procedure of (k+1)th iteration is illustrated as follows:

- Generate a new velocity v_i^{k+1} to reach to next position given by:

$$v_i^{k+1} = w_k v_i^k + c_1 rand_1 (\tilde{\psi}_i^{gbest} - \tilde{\psi}_i^k) + c_2 rand_2 (\tilde{\psi}_i^{pbest} - \tilde{\psi}_i^k) \tag{19}$$

  Where v_i^k was the velocity used to get to the (k)th iteration, is the inertia weight, c_1 and c_2 are two pre-specific positive constants, and $rand_1$, $rand_2$ are m*1 uniform random vectors.
- The next location for the ith particle is

$$\tilde{\psi}_i^{k+1} = \tilde{\psi}_i^k + v_i^{k+1} \tag{20}$$

If $\tilde{\psi}_i^{k+1}$ is not in ϖ^*, we project $\tilde{\psi}_i^{k+1}$ to a location closest to the boundary of ϖ^*.

- Project $\tilde{\psi}_i^{k+1}$ onto the regular r-simplex using i.e. $\psi_i^{k+1} = P(\tilde{\psi}_i^{k+1})$ and evaluate $\Delta(\psi_i^{k+1})$.
- Update the current best for each particle $\tilde{\psi}_i^{pbes}$.
- In this similar fashion, we find the $\tilde{\psi}^{gbes}$ after all particles are updates and continue the procedure.

The entire procedure is put to an end after completion of certain user defined numbers of iterations. Finally, we obtain the required ψ^{gbes} for large scale systems.

5. PROPOSED MODIFIED DIFFERENTIAL EVOLUTION FOR LARGE SCALE UC PROBLEM

According to Price [13, 14], the primary pros of DE include fast application and modification, simple and easy to implement, effective global optimization capability, parallel processing nature, self-referential mutation operation, ability to handle non differentiable, noisy, and/or time-dependent objective functions etc.
Mutation, crossover and selection are the three operations involved in DE for obtaining an appropriate solution among a population set. In DE, candidate solutions are identified by vectors and set of vectors generate the population. The basic notation is to form new vector by means of the weighted difference between the two population vectors. These three vectors

are chosen randomly. Then the fitness of the new vector is checked. If the fitness of the new vector is better than the previous two, then exchange takes place.

Implementation of DE

The implementation of DE can be illustrated in the following four steps:

5.1 Initialization –

In this step, initialization of population vector of size N_p in random in the D-dimensional search space over a generation G as follows:

$$x_{ij} = rand*(x_j^{max} - x_j^{min}) + x_j^{min} \tag{21}$$

Where i = 1, 2, N_p denotes the individual's population index and j = 1, 2, D signifies the D-dimensional search space position. $rand$ is a uniformly distributed random number varies between 0 to 1. The upper bound and lower bound of the decision parameter are symbolized by x_j^{min} and x_j^{max} respectively.

Now based on obtained fitness value either of the two cases are carried out, case a: deceleration factor is indulged if fitness is not up to the mark a (or) case b: acceleration factor is indulged if fitness is better than existing ones. These inputs are fed into mutation.

5.2 Mutation –

A mutant vector, for each target vector x_i^G is formed as:

$$\begin{aligned} DE/rand\,V_i^{(G+1)} &= x_k^{(G)} + F*(x_l^{(G)} - x_m^{(G)}) \\ DE/Best\,V_i^{(G+1)} &= x_{best} + F*(x_l^{(G)} - x_m^{(G)}) \end{aligned} \tag{22}$$

Where k, l, and m are randomly chosen vectors {1, 2….}. Further k, l and m should be different so that N > 4 is required. The mutation factor F is an experimentally chosen parameter that is used to regulate the amplification of the difference between two individuals to escape search stagnation. A special modification to this mutation factor is discussed explicitly in upcoming theory.

5.3 Crossover –

After mutation, crossover is applied to the population. For each mutant vector, a trial vector is generated as follows:

$$\begin{aligned} U_i^G = u_{ij}^G &= v_{ij}^G, if\ rand_j(0,1) < C_r\ or\ j = j_{rand} \\ &= x_{ij}^G, otherwise \end{aligned} \tag{23}$$

Where C_r is a crossover probability and it is fixed parameter used to create trial vectors at all generations, j_{rand} a newly generated random value for each i.

5.4 Selection –

The selection procedure compares the trial vector U_i^G and target vector x_{ij}^G of current position and the vector with the better fitness are allowed to enter the next generation.

$$\begin{aligned} x_{ij}^{G+1} &= U_i^G, if\ f(U_i^G) < f(x_{ij}^G) \\ &= x_{ij}^G,\ otherwise \end{aligned} \tag{24}$$

In case of unsatisfied termination, this process is repeated again from mutation until there is a satisfactory outcome from the operation.

But in some cases, especially for large scale problems, due to swift response abilities, DE may tend to drop regional optimums. We introduce a dynamic mutation factor in the solution algorithm for surpassing this limitation.

$$F = s*\sqrt{r(0,1)^2*d} - b \tag{25}$$

Table 1 &2: illustrate the values to be taken for each variable discussed.

Table -1 Parameter Set-up of DE

Mutation Factor (F)	0.5
Crossover rate (CR)	0.9
Upper bound	100
Lower bound	-100
Number of vector (NP)	50
Dimension	30/60
Maximum of generation	1000/2000

Table -2 Parameter Set-up of MDE

Linear decreasing factor (d)	1~0
Random variable (r)	rand (0,1)
Acceleration factor (s)	1.5
Deceleration factor (b)	0.2

6. RESULTS AND DISCUSSION

The test system containing ten power generating units and a time horizon of 24hours is taken from [9]. In PSO the population consists of 100 individuals where as in DE also the population is of 100 individuals. In lambda iteration, the tolerance is assigned to 0.0001. The fitness values of each and every individual are calculated by adding the fuel cost, the start-up cost and the penalty value. For every hour based on whether the start-up is cold start or a hot start, the appropriate cost is added to the total cost. Depending upon generation unit status, lambda iteration is used for their cost calculations. Under violation of user defined constraints such as spinning reserve requirement, t_{up} and t_{down} constraints etc. we employ the necessary penalty term. All parameter values are calculated using the best settings formed as a result of a series of 10 runs. The fitness function is given below:

$$f = \sum_{t=1}^{T} \{ \sum_{i=1}^{N} [F_i(S_i) + ST_{i,t}(1 - V_{i,t-1})] V_{i,t} + K_s \sum (S_p - S_{plim})^2 \} + K_u \sum (T_U - T_{ulim})^2 + K_d \sum (T_D - T_{dlim})^2 \} \qquad (26)$$

The results are depicted in Table -3, Table -4 and Table -5. The cost variations and fitness variation are depicted in Figure 2 and Figure 3 respectively for 150 iterations. The CPU run time for the following test system run is approximately 128 sec.

Table -3 Unit commitment schedule of 100 generator system (Large-scale)

Hour	Units On/Off Schedule
1	111111111111000000000000000000000000000000000000000000000000
2	111111111111000000000000000000000000000000000000000000000000
3	111111111111000000000000100000000000000000000000000000000000
4	111111111111000000000000111110000000000000000000000000000000
5	111111111111000000000011111111000000000000000000000000000000
6	111111111111000001111111111111000000000000000000000000000000
7	111111111111000111111111111111000000000000000000000000000000
8	111111111111111111111111111111000000000000000000000000000000
9	111111111111111111111111111111111111110000000000000000000000
10	111111111111111111111111111111111111111111111110000000000000
11	111111111111111111111111111111111111111111111111111110000000
12	111111111111111111111111111111111111111111111111111111111111
13	111111111111111111111111111111111111111111111110000000000000
14	111111111111111111111111111111111111110000000000000000000000
15	111111111111111111111111111111100000000000000000000000000000
16	111111111111111111111111111111000000000000000000000000000000
17	111111111111111111111111111111000000000000000000000000000000
18	111111111111111111111111111111000000000000000000000000000000
19	111111111111111111111111111111000000000000000000000000000000
20	111111111111111111111111111111111111110000111111111110000000
21	111111111111111111111111111111111111110000000000000000000000
22	111111111111000000111111000111111111110000000000000000000000
23	111111111111000000001111000000000000000000000000000000000000
24	111111111111000000000000000000000000000000000000000000000000

Table -4 Simulation outputs of the suggested DE method (Large-scale)

No. of Units	Best cost ($)	Average cost ($)	Worst cost ($)
10	564,180	565,922	570,509
20	1,123,321	1,124,283	1,129,199
40	2,244,193	2,250,522	2,254,122
60	3,366,628	3,372,331	3,378,352
80	4,489,722	4,492,634	4,505,124
100	5,610,001	5,621,821	5,630,851

Table -5 Comparison of various methods

No of generators	Total cost ($)				
	LR[9]	GA[9]	EP[9]	PSO(Large-scale)	DE(Large-scale)
10	565,825	565,825	564,551	564,030	564,180
20	1,130,660	1,126,243	1,125,494	1,123,325	1,123,321
40	2,258,503	2,251,911	2,249,093	2,244,267	2,244,193
60	3,394,066	3,376,625	3,371,611	3,366,720	3,366,628
80	4,526,022	4,504,933	4,498,479	4,489,800	4,489,722
100	5,657,277	5,627,437	5,623,885	5,610,122	5,610,001

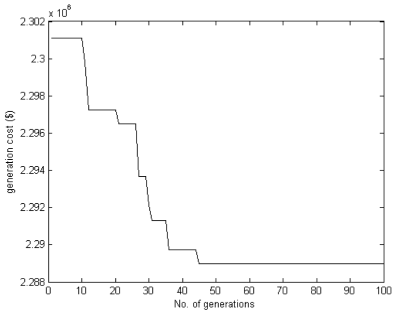

Figure 2. Convergence of total cost of 100-unit system with DE algorithm (Large-scale)

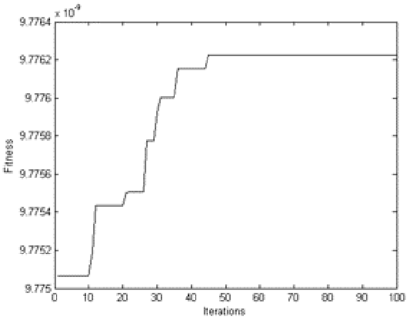

Figure 3. Convergence of fitness of 100-unit system with DE algorithm (Large-scale)

7. CONCLUSIONS

The first suggested method exhibits good performance in solving the unit commitment problem when it is enclosed in a hypercube before attaining a non-feasible point for a large scale problem. In parallel to this, we solved the model with DE by adding a suitable dynamic mutation factor, for obtaining the optimal settings of Lagrangian multipliers. The method has been tested for hundred units with 24 hours' time horizon inclusive of constraints. Moreover, the test results of the suggested methods are verified with the various pre-existing methods. Hence, it could be concluded that the approach gives optimal commitment schedule of units in large scale for any given demand that satisfies the defined constraints as well as the demand with minimum cost.

7. REFERENCES

[1] A. J. Wood and B. F. Wollenberg, Power Generation, Operation &Control, 2nd Ed. New York: Wiley, 1996.

[2] Yang Ting fang, T. O. Ting, “Methodological Priority List for Unit Commitment Problem”, IEEE computer society, pp.176-179, 2008.

[3] Prateek Kumar Singhal and R. Naresh Sharma, “Dynamic Programming Approach for Solving Power Generating Unit Commitment Problem”, ICCCT, pp.298-303, 2011.

[4] Arthur I. Cohen, Miki Yoshimura, “A Branch and Bound Algorithm for Unit Commitment”, IEEE Trans. Power Syst., Vol.2, pp.444-451, February 1983.

[5] Fulin Zhuang and F.D. Galiana, “Towards a more rigorous and practical unit commitment by Lagrangian relaxation”, Vol. 3, No. 2, pp.763-771, May. 1988.

[6] D. P. Kadam, P. M. Sonwane, V. P. Dhote, B. E. Kushare, “Fuzzy Logic Algorithm for Unit Commitment Problem”, International conference on control, automation, communication and energy conservation, pp1-4,2009.

[7] Musoke H. Sendaula, Saroj K Biswas, Ahmed Eltom and Cliff Parten Wilsonkazibwe, “Application of Artificial Neural Networks to Unit Commitment”, IEEE Trans. Power Syst., pp.256-260, 1991.

[8] S. A. Kazarlis, A. G. Bakirtzis and V. Petridis, “A genetic algorithm solution to the unit commitment problem”, IEEE Trans. Power Syst., vol. 11, pp. 83–92, Feb. 1996.

[9] Dimitris, N. Simopoulos, Stavroula D. Kavatza, D. Vournas, “Unit Commitment byan Enhanced Simulated Annealing Algorithm”, PSCE, pp.193-201, 2006.

[10] C. Christober, Asir Rajan and M. R. Mohan, “An Evolutionary Programming-Based Tabu Search Method for Solving the Unit Commitment Problem”, IEEE Trans. Power Syst., pp-557-585, vol.19. Feb.2004.

[11] Balci H.H. and Valenzuela J.F. “Scheduling Electric power generators using particle swarm optimization combined with the Lagrangian relaxation method”, Int.J. App Math. Comput. Sci., 14(3), pp.411421,2004.

[12] WengKee Wong, Ray-Bing Chen, Chien-Chih Huang, Weichung Wang, “A Modified Particle Swarm Optimization Technique for Finding Optimal Designs for Mixture Models”, DOI: 10.1371/journal.pone.0124720 June 19, 2015.

[13] M.Ramu, L. Ravi Srinivas, S. Tara Kalyani, “Unit Commitment by Lagrangian Relaxation Incorporating Differential Evolution”, Journal of Electrical Engineering: Volume 15 / 2015 - Edition: 3.

[14] Ching-Wei Chien, Zhan-Rong Hsu, Wei-Ping Lee, “Improving the Performance of Differential Evolution Algorithm with Modified Mutation Factor”, 2009 International Conference on Machine Learning and Computing IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore.