Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)

Unit Commitment Using Particle Swarm-Based-Simulated Annealing Optimization Approach

Nasser Sadati, *Member*, *IEEE*, Mahdi Hajian and Majid Zamani

Intelligent Systems Laboratory
Electrical Engineering Department
Sharif University of Technology
Tehran, Iran.
e-mail: sadati@sina.sharif.edu, m_hajian@ee.sharif.ir, majid_zamani@ee.sharif.ir

***Abstract* – In this paper, a new approach based on hybrid Particle Swarm-Based-Simulated Annealing Optimization (PSO-B-SA) for solving thermal unit commitment (UC) problems is proposed. The PSO-B-SA presented in this paper solves the two sub-problems simultaneously and independently; unit-scheduled problem that determines on/off status of units and the economic dispatch problem for production amount of generating units. Problem formulation of UC is defined as minimization of total objective function while satisfying all the associated constraints such as minimum up and down time, production limits and the required demand and spinning reserve. Simulation results show that the proposed approach can outperform the other solutions.**

I. Introduction

Unit commitment (UC) is the problem of scheduling of generating units over a given time period so that the total operational cost is minimized and all operational constraints are satisfied [1]. UC involves two decision processes; First, the "unit scheduling" that determines on/off status of generating units in each hour of planning horizon subject to system capacity requirements, including the spinning reserve and the constrained on start-up and shut-down of units. Second, the "economic dispatch" decision involves the allocation of the system demand and reserve capacity among the operating units in each specified hour.

Mathematically, UC is a nonconvex, nonlinear, large-scale, mixed-integer optimization problem with a great number of 0–1 scheduling variables, continuous and discrete control variables, and a series of prevailing equality and inequality constraints [1]. The global optimal solution can be obtained by complete enumeration, which is not applicable to large power systems due to its excessive computational time requirements. Therefore, research interest, have been focused on efficient, near optimal solutions. Up to now many methods have been developed for solving UC problems such as priority list methods [2], [3], integer programming [4], [5], dynamic programming [6]-[11], mixed-integer programming [12], branch-and-bound methods [13], and Lagrangian relaxation (LR) methods [14], [15]. Priority list method is simple and very fast, but gives schedules with relatively high operation cost. Dynamic programming methods are flexible but are computationally expensive. Branch-and-bound method uses a linear function to represent the fuel consumption and time-dependent start cost and obtains the required lower and upper bounds. However this method has the danger of a deficiency of storage capacity and increasing the calculation time enormously as being a large scale problem. The integer and mixed-integer methods adopt linear programming techniques to solve and check for an integer solution. These methods have only been applied to small UC problems and have required major assumptions that limit the solution space. The LR method concentrates on finding an appropriate coordination technique for generating feasible primal solutions, while minimizing the duality gap. The main problem with the LR methods is the difficulty encountered in obtaining feasible solutions.

In this paper, we combined the two optimization methods for solving the UC problems; Particle Swarm Optimization and Simulated Annealing. In [16] it is shown that in solving the optimization problem, the PSO might have deficiency in finding the global solution and get trapped in local minima. So the idea was to combine the PSO with SA in order to enhance the performance of algorithm for finding the optimal solution [16].

The paper is organized as follows; A brief description of the proposed method is presented in Section II. The UC formulation is given in Section III. Section IV presents a detailed explanation of PSO-B-SA approach for UC problem. Numerical results for ten unit system are presented in Section V. Finally, Section VI concludes the paper.

II. Particle Swarm-Based-Simulated Annealing

A. Particle Swarm Optimization

Assuming that the search space is D-dimensional, the i-th particle of the swarm is represented by the D-dimensional vector $X_i = (x_{i1}, x_{i2}, ..., x_{id})$ and the best particle in the swarm, i.e. the particle with the smallest function value, is denoted by the index g (p_g). The best previous position of the i-th particle is recorded and represented as $P_i = (p_{i1}, p_{i2}, ... p_{id})$, while the position change (velocity) of the i-th particle is represented as $V_i = (v_{i1}, v_{i2}, ..., v_{id})$, which is clamped to a maximum velocity $V_{max} = (v_{max1}, v_{max2}, ..., v_{maxd})$ specified by the user. Following this notation, the particles are manipulated according to the following equations

$$v_{id}^{(t+1)} = w v_{id}^{(t)} + c_1 rand(.)(p_{id} - x_{id}) + c_2 rand(.)(p_{gd} - x_{id}) \quad (1)$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \quad (2)$$

where w can be expressed by the inertia weights approach,

1-4244-0708-7/07/$20.00 ©2007 IEEE

c_1 and c_2 are the acceleration constants which influence the convergence speed of each particle, and *rand(.)* is a random number in the range of [0,1]. For equation (1), the first part represents the inertia of the previous velocity, the second part is the "cognition" part which represents the private thinking by itself, and the third part is the "social" part which represents the cooperation among the particles. If the summation in (1) would cause the velocity v_{id} on that dimension to exceed v_{maxd}, then v_{id} is limited to v_{maxd}. V_{max} determines the resolution with which regions between the present position and the target position are searched. If V_{max} is too large, the particles might fly the past good solutions. If V_{max} is too small, the particles may not explore sufficiently beyond local solutions. In many experiences with PSO, V_{max} is often set to maximum dynamic range of the variables on each dimension. The constants c_1 and c_2 represent the weighting of the stochastic acceleration terms that pull each particle toward p_i and p_g positions. Low values allow particles to roam far from the target regions before being tugged back. On the other hand, high values result in abrupt movement toward, or past, the target regions. Hence, the acceleration constants c_1 and c_2 are often set to be 2.0 according to the past experiences. Suitable selection of inertia weight w provides a balance between global and local explorations, thus requiring less iterations on average to find a sufficiently optimal solution. As originally developed, w often decreases linearly from about 0.9 to 0.4 during a run. In general, the inertia weight w is set according to the following equation

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}}.iter \tag{3}$$

where $iter_{\max}$ represents the maximum number of iterations, and $iter$ is the current number of iterations or generations. Moreover, $w_{\max}$ and $w_{\min}$ are the maximum and minimum weight values, respectively. From the above discussion, it is obvious that PSO resembles, to some extent, the "mutation" operator of Genetic Algorithms through the position update equations (1) and (2). However, it should be noted that in PSO, the "mutation" operator is guided by the particle's own "flying" experience and benefits from the swarm's "flying" experience. In other words, PSO is considered as performing mutation with a "*conscience"* as pointed out by Eberhart and Shi [17].

B. Binary Particle Swarm Optimization (BPSO)[18]

In binary particle swarm, X_i and P_i can only take on values of 0 or 1. The velocity V_i will determine a probability threshold. If V_i is higher, the individual is more likely to choose 1, but lower values favor the 0 choice. Such a threshold needs to stay in the range [0.0,1.0]. One straightforward function for accomplishing this is common in neural networks. The function is called the sigmoid function and is defined as follows

$$s(V_i) = \frac{1}{1+\exp(-V_i)} \tag{4}$$

The function squashes its input into the requisite range and has properties that make it agreeable to be used as a probability threshold. A random number (drawn from a uniform distribution between 0.0 and 1.0) is then generated, whereby X_i is set to 1, if the random number is less than the value from the sigmoid function, as is illustrated below

$$\textit{If } rand(.) < s(V_i), \textit{ Then } U_i = 1, \textit{ Else } U_i = 0. \tag{5}$$

In the UC problems, U_i represents the on or off state of generator *i*. In order to ensure that there is always some chance of a bit flipping (on and off of generators), a constant $V_{\max}$ can be set at the start of a trial to limit the range of V_i. A large $V_{\max}$ value results in a low frequency of changing state of generator, whereas a small value increases the frequency of on/off of generator. In practice, $V_{\max}$ is often set to ± 4.0, so that there is always at least a good chance that a bit will change the state. This is to limit V_i so that $s(V_i)$ does not approach too close to 0.0 or 1.0. In this binary model, $V_{\max}$ functions are similar to the mutation rates of *GAs*.

C. The PSO-B-SA

The PSO-B-SA is an optimization algorithm which combines the PSO with the SA. In fact by combining PSO with SA, the strong points of SA can be used in PSO. This is the basic idea of the PSO-B-SA. The PSO-B-SA algorithm's searching process is started from initializing a group of random particles. In this paper, only p_g which is the leader of the swarm is based on SA, independently from other particles. This algorithm is named as the PSO-B-SA1 [16]. This process evolves through time until the terminating condition is satisfied.

In the process of simulated annealing, the new individuals are generated randomly around the original individuals.

$$present = present + r_1\, rand(.) \tag{6}$$

In the above equation, the $r_1\, rand(.)$ is a random number between 0 and 1. Now, to find the global minimum of the following optimizing problem

$$\min f(x_1, x_2, \ldots, x_n) \tag{7}$$

$$s.t. \quad x_i \in [a_i, b_i] \quad ; \; i = 1, 2, \ldots, n$$

the steps taken in the particle swarm-based-simulated annealing optimization is as follows

(1) Initialize a group of particles (the scale is m), including random position and velocity.
(2) Evaluate each particle's fitness.
(3) p_g is based on SA independently and a new

global best position (p_g) is obtained.

(4) For each particle, compare its fitness and its personal best position (p_i). If its fitness is better, replace p_i with its fitness.

(5) For each particle, compare its fitness and the global best position (p_g). If its fitness is better, then replace p_g with its fitness.

(6) Transform each particle's velocity and its position according to the expressions (1) and (2).

(7) This process evolves through time until the terminating condition is satisfied.

III. Problem Formulation of UC

The problem formulation of a UC problem can be organized as follows:

1. Objective function: the total cost over the entire scheduling period is the sum of fuel cost and start-up cost for all units. Accordingly total production cost for *N* units over *H* number of operating hours is

$$TPC = \sum_{h=1}^{H}\sum_{n=1}^{N}[F_i(P_{ih}) + ST_i(1-U_{i(h-1)}]U_{ih} \tag{8}$$

Generally, the fuel cost $F_i(P_{ih})$ is a function of the generator power output. Usually it is expressed as a quadratic polynomial as follows

$$F_i(P_{ih}) = \alpha_i P_{ih}^2 + \beta_i P_{ih} + \gamma_i \tag{9}$$

The generator start up cost depends on the time that the unit has been off prior to start up

$$SC_i = \begin{cases} h-\cos t: & MD_i \le X_i^{off} \le MD_i + c-s-hour \\ c-\cos t: & X_i^{off} > MD_i + c-s-hour \end{cases} \tag{10}$$

2. Constraints: UC problem has some constraints representing system constraints and generating unit constraints. The overall objective is minimizing TPC subject to a number of constraints:

A. System Power Balance

In each hour, the total power generated must supply the load demand

$$\sum_{i=1}^{N} P_{ih}U_{ih} = D_h \tag{11}$$

B. System Reserve Requirements

The hourly spinning reserve requirements , R_h, must be met

$$\sum_{i=1}^{N} P_{i(\max)}U_{ih} \ge D_h + R_h \tag{12}$$

C. Generation Limits

The unit rated minimum and maximum capacity must not be violated

$$P_{i(\min)} \le P_{ih} \le P_{i(\max)} \tag{13}$$

D. Unit Minimum Up/Down Time

The unit up/down time must satisfy the following conditions

$$\begin{aligned} X_i^{on}(t) &\ge MU_i \\ X_i^{off}(t) &\ge MD_i \end{aligned} \tag{14}$$

where the notification used are

TPC	total production cost;
N	number of generators;
H	number of hours;
P_{ih}	generation output of the i th unit at the h th hour;
ST_i	start-up cost of the i th unit;
U_{ih}	on/off status of the i th unit at the h th hour. $U_{ih}=0$ when off, $U_{ih}=1$ when on;
$h-\cos t$	i th unit hot start cost;
$c-\cos t$	i th unit cold start cost;
$c-s-hour$	cold start time of unit i ;
D_h	load demand at the h th hour;
R_h	spinning reserve at the h th hour (set to 10% of D_h);
$P_{i(\min)}$	minimum generation limit of the i th unit;
$P_{i(\max)}$	maximum generation limit of the i th unit;
MU_i	minimum up-time of the i th unit;
MD_i	minimum down-time of the i th unit;
$X_i^{on}(t)$	duration during which the i th unit is continuously on;
$X_i^{off}(t)$	duration during which the i th unit is continuously off;

IV. PSO-B-SA1 Approach to UC

In [19] the concept of hybrid particle swarm optimization was introduced, so that the economic dispatch and the UC problem are solved independently and simultaneously. Economic dispatch and UC is solved by real valued PSO and binary PSO, respectively. In our approach, we have used the PSO-B-SA1 algorithm [16] for our optimization problem.

The aforementioned steps in subsection C, are applied to this problem. The termination condition is considered as reaching to a specific number of iterations. The generation combination and the associated power output of each online unit , at last iteration , will be announced as an optimal solution.

A. Representation of Individual Particles

Before using the PSO-B-SA1 algorithm to solve the problem, representation of a particle must be defined. Hence, we define the generators status (ON('1')/OFF('0')) and correspond the power outputs as a sub-particle. There are 24 sub-particles in one day comprising a particle. A particle would display the generators commitment schedule in one day. For example, if there are ten generating units to supply the power to meet the demand in a system, the dimension of an individual is 24×20. When the size of the population is *ps***,** then the dimension of the population is equal to *ps*×24×20.

B. Constraints Satisfaction

The demand and reserve constraints are satisfied by penalty functions. In each hour, if committed generation and sum of the power output of each online unit can not meet the reserve and the demand, respectively, we add a penalty function, corresponding to the violated constraints, to the objective function. Consequently the formulation of the objective function (O.F.) can be written as

$$O.F. = TPC + \sum_{h=1}^{H}[\frac{s}{2}[c_1(D_h - \sum_{n=1}^{N} P_{ih}U_{ih})^2 + c_2(D_h + R_h - \sum_{i=1}^{N} P_{i(\max)}U_{ih})^2]] \quad (15)$$

where s is a penalty factor that is considered as $s = s_0 + \log(t+1)$. Also t is the number of generation. The value of s_0 must be determined so that the speed and the convergence of solution will be guaranteed. From the experiment a value of 50 for s_0 is selected. In (15), c_1 is set to 1 if a violation to constraint (11) occurs and $c_2 = 0$ whenever (11) is not violated. Likewise, c_2 is also set to 1 whenever a violation of (12) is detected, and it remains 0 otherwise.

For satisfying the generation limit constraints, the initial power output is generated randomly within the power limits of a generator. After each iteration, if the power output of a generator violates its power limit, we set the power output at the boundary, which is violated.

For satisfying min-up and min-down time, we enforce these constraints to each particle. After updating the position of particles, these constraints are checked. The state of a generator is changed whenever either MU_i or MD_i is violated. For example if a generator is committed at hour *h* and $X_i^{off}(t)$ is lower than MD_i, the generator will be kept off-line. It is possible that after this action generation schedule of a particle cannot satisfy the demand. Therefore for evaluating fitness function, penalty functions must added to objective function. Furthermore, this particle might be gbest, so gbest must be revaluated during fitness function calculations.

To increase the speed of algorithm for finding the optimal solution, a set of initial conditions based on priority list is generated so that all constraints will be met. For this purpose, full average production cost of units is calculated, then based on this index units are committed sequentially, until the demand and required spinning reserve of associated hour is satisfied. It should be noted that during this commitment the minimum up/down time of units must be considered. It means that this initial generation satisfies all conditions of the problem, but it is not the optimal solution necessarily.

V. Numerical Results

The PSO program was developed in MATLAB M-file and the simulation was carried out on a ten generating systems; the data for this is given in Table I and II. According to the experience, the following PSO and SA parameters are used:
Population size: 30;
Maximum iterations: 1000;
Dimension: 24×20;
Maximum Velocity (for discrete variables): 4;
Maximum Velocity (for continuous variables): $P_{i(\max)} - P_{i(\min)}$;
The acceleration constants: $c_1 = c_2 = 2$;
Inertia weight: $w = 1$;
Annealing schedule: $r = 0.9$;
Initial temperature: $T_{in} = 1$;
Stopping temperature: $T_{stop} = 0.0001$

Table III shows the best solution found by our proposed method after 100 runs. As shown in Table III, the best cost is 563938. Table IV gives a comparison between PSO-B-SA1 and the several other techniques. In this Table the best average and worst cost for several methods are given. Deterministic methods such as DP and LR have the same values for different runs, so these methods do not have average and worst cost. The average cost for GA was not

TABLE I
DATA FOR THE SYSTEM OF 10 UNITS

	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5
Pmax (MW)	455	455	130	130	162
Pmin (MW)	150	150	20	20	25
a ($/h)	1000	970	700	680	450
b ($/MW h)	16.19	17.26	16.60	16.50	19.70
c ($/MW² h)	0.00048	0.00031	0.002	0.00211	0.00398
min up (h)	8	8	5	5	6
min down (h)	8	8	5	5	6
hot start cost ($)	4500	5000	550	560	900
Cold start cost ($)	9000	10000	1100	1120	1800
Cold start hrs (h)	5	5	4	4	4
Initial status (h)	8	8	-5	-5	-6

	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10
Pmax (MW)	80	85	55	55	55
Pmin (MW)	20	25	10	10	10
a ($/h)	370	480	660	665	670
b ($/MW h)	22.26	27.74	25.92	27.27	27.79
c ($/MW² h)	0.00712	0.00079	0.00413	0.00222	0.00173
min up (h)	3	3	1	1	1
min down (h)	3	3	1	1	1
hot start cost ($)	170	260	30	30	30
Cold start cost ($)	340	520	60	60	60
Cold start hrs (h)	2	2	0	0	0
Initial status (h)	-3	-3	-1	-1	-1

available. As shown in Table IV, the average cost of PSO-B-SA1 is 564115 and the worst cost is 564985. It should be noted that in this case the best result of HPSO [19] and PSO-B-SA1 do not differ significantly. However, the average and the worst cost of PSO-B-SA1 is much better than the PSO, which implies that the solutions are generally closer to the global optima. From the simulation results, it can be seen that the proposed method has better results than the other techniques in term of total cost. Since the simulations were carried out by different computers, the simulation time is not compared here.

VI. Conclusion

This paper presents a new methodology for solving the UC problems. The proposed algorithm is the combination of particle swarm optimization and simulated annealing methods. A test system consisted of ten units is simulated to demonstrate the effectiveness of the proposed method compared with other approaches. From the numerical results, it can be concluded that the proposed method provides a cheaper cost than those obtained from other methods.

TABLE II
LOAD DEMAND FOR 24 HOURS

Hour	D_h	Hour	D_h
1	700	13	1400
2	750	14	1300
3	850	15	1200
4	950	16	1050
5	1000	17	1000
6	1100	18	1100
7	1150	19	1200
8	1200	20	1400
9	1300	21	1300
10	1400	22	1100
11	1450	23	900
12	1500	24	800

TABLE III
THE BEST SOLUTION OFTHE PSO-B-SA1

Hour	Load (MW)	Unit Number										Total Cost	Start-up Cost
		1	2	3	4	5	6	7	8	9	10		
1	700	455	245	0	0	0	0	0	0	0	0	13683.1	0
2	750	455	295	0	0	0	0	0	0	0	0	14554.5	0
3	850	455	370	0	0	25	0	0	0	0	0	17709.5	900
4	950	455	455	0	0	40	0	0	0	0	0	18597.7	0
5	1000	455	390	0	130	25	0	0	0	0	0	20580	560
6	1100	455	360	130	130	25	0	0	0	0	0	23487	1100
7	1150	455	410	130	130	25	0	0	0	0	0	23262	0
8	1200	455	455	130	130	30	0	0	0	0	0	24150.3	0
9	1300	455	455	130	130	85	20	25	0	0	0	28111.1	860
10	1400	455	455	130	130	162	33	25	10	0	0	30117.6	60
11	1450	455	455	130	130	162	73	25	10	10	0	31976.1	60
12	1500	455	455	130	130	162	80	25	43	10	10	33950.2	60
13	1400	455	455	130	130	162	33	25	10	0	0	30057.6	0
14	1300	455	455	130	130	85	20	25	0	0	0	27251.1	0
15	1200	455	455	130	130	30	0	0	0	0	0	24150.3	0
16	1050	455	310	130	130	25	0	0	0	0	0	21513.7	0
17	1000	455	260	130	130	25	0	0	0	0	0	20641.8	0
18	1100	455	360	130	130	25	0	0	0	0	0	22387	0
19	1200	455	455	130	130	30	0	0	0	0	0	24150.3	0
20	1400	455	455	130	130	162	33	25	10	0	0	30547.6	490
21	1300	455	455	130	130	85	20	25	0	0	0	27251.1	0
22	1100	455	360	0	0	145	20	25	0	0	0	22735.5	0
23	900	455	420	0	0	0	20	0	0	0	0	17645.4	0
24	800	455	345	0	0	0	0	0	0	0	0	15427.4	0
Total												563938	4090

TABLE IV
COMPARISON OF RESULTS

Method	Total cost		
	Best	Average	Worst
BCGA[24]	567367	N/A	N/A
ICGA[24]	566404	N/A	N/A
SA[23]	565828	565988	566260
DP[20]	565825	N/A	N/A
LR[20]	565825	N/A	N/A
PSOLR[25]	565275	N/A	N/A
LRGA[22]	564800	N/A	N/A
ACSA[26]	564049	N/A	N/A
EPL[27]	563977	N/A	N/A
ELR[27]	563977	N/A	N/A
UCC-GA[21]	563977	N/A	565606
HPSO[19]	563942	564772	565785
PSO-B-SA1	**563938**	**564115**	**564985**

REFERENCES

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

[2] R. M. Burns and C. A. Gibson, "Optimization of priority lists for a unit commitment program," in *Proc. IEEE/Power Engineering Society Summer Meeting*, Paper A 75 453-1, 1975.

[3] G. B. Sheble, "Solution of the unit commitment problem by the method of unit periods," *IEEE Trans. Power Syst.*, vol. 5, no. 1, pp. 257–260, Feb. 1990.

[4] T. S. Dillon, K. W. Edwin, H. D. Kochs, and R. J. Taud, "Integer programming approach to the problem of optimal unit commitment with probabilistic reserve determination," *IEEE Trans. Power App. Syst.*, vol. PAS-97, no. 6, pp. 2154–2166, Dec. 1978.

[5] L. L. Garver, "Power generation scheduling by integer programming development of theory," *IEEE Trans. Power App. Syst.*, vol. PAS-18, pp. 730–735, Feb. 1963.

[6] W. L. Snyder Jr., H. D. Powell Jr., and J. C. Rayburn, "Dynamic programming approach to unit commitment," *IEEE Trans. Power App. Syst.*, vol. PAS-2, pp. 339–350, May 1987.

[7] P. G. Lowery, "Generation unit commitment by dynamic programming,"*IEEE Trans. Power App. Syst.*, vol. PAS-102, pp. 1218–1225, 1983.

[8] C. K. Pang and H. C. Chen, "Optimal short-term thermal unit commitment," *IEEE Trans. Power App. Syst.*, vol. PAS-95, no. 4, pp. 1336–1346, Aug. 1976.

[9] C. K. Pang, G. B. Sheble, and F. Albuyeh, "Evaluation of dynamic programming based methods and multiple area representation for thermal unit commitment," *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 3, pp. 1212–1218, May 1981.

[10] C. C. Su and Y. Y. Hsu, "Fuzzy dynamic programming: An application to unit commitment," *IEEE Trans. Power Syst.*, vol. 6, no. 3, pp. 1231–1237, Aug. 1991.

[11] Z. Ouyang and S. M. Shahidehpour, "An intelligent dynamic programming for unit commitment application," *IEEE Trans. Power Syst.*, vol. 6, no. 3, pp. 1203–1209, Aug. 1991.

[12] J. A. Muckstadt and R. C. Wilson, "An application of mixed-integer programming duality to scheduling thermal generating systems," *IEEE Trans. Power App. Syst.*, pp. 1968–1978, 1968.

[13] A. I. Cohen and M. Yoshimura, "A branch-and-bound algorithm for unit commitment," *IEEE Trans. Power App. Syst.*, vol. PAS-102, no. 2, pp. 444–451, 1983.

[14] A. Merlin and P. Sandrin, "A new method for unit commitment at Electricite de France," *IEEE Trans. Power App. Syst.*, vol. PAS-102, pp. 1218–1255, May 1983.

[15] F. Zhuang and F. D. Galiana, "Toward a more rigorous and practical unit commitment by Lagrangian relaxation," *IEEE Trans. Power Syst.*, vol. 3, no. 2, pp. 763–770, May 1988

[16] N. Sadati, M. Zamani, H. Mahdavian, "Hybrid Particle Swarm-Based-Simulated Annealing Optimization Techniques," *IEEE* Inter. Conf. on Indus. Elect. (IECON 2006), Paris, France, 2006

[17] R. Eberhart and Y.H. Shi, "Evolving artificial neural networks," Proc. Int. Conf. on Neural Networks and Brain, 1998.

[18] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. Int. Conf. Systems, Man, Cybernetics*. Piscataway, NJ, 1997, pp. 4104–4109.

[19] T. O. Ting, M. V. C. Rao, and C. K. Loo, "A Novel Approach for Unit Commitment Problem via an Effective Hybrid Particle Swarm Optimization" IEEE transactions on power systems, vol. 21, no. 1, February 2006

[20] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, "A genetic algorithm solution to the unit commitment problem," *IEEE Trans. Power Syst.*, vol. 11, no. 1, pp. 83–92, Feb. 1996.

[21] T. Senjyu, K. Shimabukuro, K. Uezato, and T. Funabashi, "A unit commitment problem by using genetic algorithm based on unit characteristic classification," in *Proc. IEEE Power Engineering Society Winter Meeting*, vol. 1, 2002, pp. 58–63.

[22] C. P. Cheng, C.W. Liu, and C. C. Liu, "Unit commitment by Lagrangian relaxation and genetic algorithms," *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 707–714, May 2000.

[23] Dimitris N. Simopoulos, Stavroula D. Kavatza, and Costas D. Vournas, "Unit Commitment by an Enhanced Simulated Annealing Algorithm, " *IEEE Trans. Power Syst.*, VOL. 21, NO. 1, FEBRUARY 2006

[24] Ioannis G. Damousis et al., "A Solution to The Unit Commitment Problem Using Integer-Coded GA," *IEEE Trans. Power Sys.* vol. 19, NO. 2, MAY 2004

[25] P. Sriyanyong and Y.H.Song, "Unit Commitment Using Particle Swarm Optimization Combined with Lagrange Relaxation," *IEEE Trans.* 2005.

[26] T. Sum-im, W. Ongsakul, "Ant Colony Search Algorithm for Unit Commitment," *IEEE Con. ICIT 2003.*

[27] T. Senjyu,*et al* , "A Fast Technique for Unit Commitment Problem by Extended Priority List," *IEEE Trans. on Power Sys.,* vol. 18, no. 2, May 2003.