Adaptive Binary PSO based Unit Commitment

R. K. Santhi
Professor
Department of Electrical Engineering
Annamalai University,
Annamalainagar, Tamil Nadu,
INDIA

S. Subramanian
Professor
Department of Electrical Engineering
Annamalai University,
Annamalainagar, Tamil Nadu,
INDIA

ABSTRACT

This paper presents a binary PSO based solution technique for power system unit commitment. The intelligent generation of initial population and the repairing mechanism ensure feasible solution that satisfies the spinning reserve and unit minimum up/down constraints. The algorithm adoptively adjusts the inertia weight and the acceleration coefficients in order to enhance the search process and arrive at the global optimum. Numerical results on systems up to 100 generating units demonstrate the effectiveness of the proposed strategy.

KEYWORDS

Unit commitment, particle swarm optimization, lambda iteration method.

1. NOMENCLATURE

ACT	Average Computation Time
ED	Economic Load Dispatch
EPM	EP based Method
ELRM	Enhanced LRM
GAM	GA based Method
LRM	Lagrangian Relaxation Method
PSO	Particle Swarm Optimization
PM	Proposed Method
UC	Unit Commitment
a,b,c	fuel cost coefficients
CST_i	cold start up cost of unit i ($)
c_1 & c_2	acceleration coefficients
$F(P_i^t)$	generator fuel cost function ($/hr)
HSTi	hot start up cost of unit i ($)
K^{max}	maximum number of iterations
k	iteration counter
N	number of generating units
n	number of particles in the population
P_i^{min} & P_i^{max}	minimum and maximum real power generation of unit i respectively
P_i^t	real power generation of unit i at hour- t
P_{load}^t	load demand at hour t
R^t	spinning reserve at hour t
r_1, r_2 & r_3	uniformly distributed random numbers in the range of [0,1]
ST_i^t	startup cost of unit i at hour t
$S(X_{ij}(k))$	sigmoid limiting transformation function
T	total number of hours
T_i^{cold}	cold start hour of unit i (hours)
T_i^{down}	minimum down time of unit i (hours)
T_i^{off}	continuously off time of unit i (hours)
T_i^{on}	continuously on time of unit- i (hours)
T_i^{up}	minimum up time of unit- i (hours)
$U_{i,t}$	on/off status of unit- i at hour- t
$V_i(k)$	velocity of i^{th} moving particle
$V_{ij}(k)$	velocity of j^{th} element in i^{th} particle
w(k)	inertia weight
$X_i(k)$	candidate solution of i^{th} particle
$X_{ij}(k)$	value of j^{th} element in i^{th} particle
$X^*(k)$	particle best
$X^{**}(k)$	global best
α	decrement constant smaller than but close to 1
λ	Lagrange multipliers
Φ(P,U)	objective function to be minimized over the scheduling period

superscripts *ini & fin* initial and final values respectively

2. INTRODUCTION

Unit Commitment (UC) is the most important function of energy control centers, which determines the on/off status as well as the real power outputs of the generators while minimizing the system operating cost over the planning period subject to various physical, operational and contractual constraints. This problem is a non-linear, large-scale, mixed-integer constrained optimization problem, which is quite difficult due to its inherent high dimensional, nonconvex, discrete and nonlinear nature [1]. Many methods with various degrees of near-optimality, efficiency, ability to handle difficult constraints and heuristics, have been suggested for UC in the literature. At one end of the spectrum, there are simple and fast but highly heuristic priority list [2,3] methods. At the other end, there are dynamic programming [4,5] and branch-and bound [6,7], which are in general flexible but often prone to the curse of dimensionality. Between the two extremes, there are Lagrangian relaxation methods (LRM) [8, 9], which are efficient and appear to be a desirable compromise, and well suited for large-scale UC. However under certain constraints such as crew constraints, these methods demand additional heuristics detrimental to efficiency of the method. An enhanced adaptive LRM (EALRM) and heuristic search for UC has been proposed [10]. Methods such as genetic algorithms [11,12] simulated annealing [13], evolutionary programming [14] and particle swarm optimization (PSO) [15-17] have been applied in solving UC. Having in common processes of natural evolution, these

algorithms share many similarities; each maintains a population of solutions that are evolved through random alterations and selection. The differences between these procedures lie in the representation techniques they utilize to encode candidates, the type of alterations they use to create new solutions, and the mechanism they employ for selecting the new parents. The algorithms have yielded satisfactory results across a great variety of power system problems. The main difficulty is their sensitivity to the choice of the parameters, such as temperature in SA, the crossover and mutation probabilities in GA and the inertia weight, acceleration coefficients and velocity limits in PSO. There exists a need for evolving simple and effective methods for obtaining optimal solution for the UC problem.
An attempt has been made to solve UC problem efficiently using binary PSO with a view to enhance the search process in this paper. The developed strategy has been tested to demonstrate the performance on systems up to 100 generating units and the results presented.

2. PROBLEM DESCRIPTION

The main objective of UC problem is to minimize the overall system production cost over the scheduled time horizon under the spinning reserve and operational constraints of generator units. This constrained optimization problem is formulated as

Minimize

$$\Phi(P,U) = \sum_{t=1}^{T}\sum_{i=1}^{N}\left\{F_i(P_i^t) + ST_i^t\left(1-U_{i,t-1}\right)\right\}U_{i,t} \quad (1)$$

where

$$F_i(P_i^t) = a_i {P_i^t}^2 + b_i P_i^t + c_i$$

Subject to

Power balance constraint

$$P^t{}_{load} - \sum_{i=1}^{N} P_i^t\, U_{i,t} = 0 \quad (2)$$

Spinning reserve constraint:

$$P^t{}_{load} + R^t - \sum_{i=1}^{N} P_i^{\max}\, U_{i,t} \le 0 \quad (3)$$

Generation limit constraints:

$$P_i^{\min} U_{i,t} \le P_i^t \le P_i^{\max} U_{i,t} \qquad \text{i=1,2,…N} \quad (4)$$

Minimum up and down time constraints:

$$U_{i,t} = \begin{cases} 1 & if\ T_i^{on} < T_i^{up} \\ 0 & if\ T_i^{off} < T_i^{down} \\ 0\ or\ 1 & otherwise \end{cases} \quad (5)$$

Startup Cost:

$$ST_i = \begin{cases} HST_i & if\ T_i^{down} \le T_i^{off} \le T_i^{cold} + T_i^{down} \\ CST_i & if\ T_i^{off} > T_i^{cold} + T_i^{down} \end{cases} \quad (6)$$

3. PARTICLE SWARM OPTIMIZATION

PSO was introduced by Kennedy and Eberhart as a modern heuristic optimizer. It is a population-based stochastic optimization technique modeled on swarm intelligence. Swarm-intelligence, also referred to as collective intelligence, is based on social-psychological principles and provides insights into social behavior, as well as contributing to engineering applications. The PSO system combines a social-only model and a cognition-only model [18].
In this approach, a population of m-individuals, called particles $X(k)$, is initialized with random guesses in the problem space. Each particle represents a candidate solution to the problem at hand. These particles fly around in a multidimensional search space with a velocity, $V(k)$.
These particles share their information with each other and run toward best trajectory to find optimal solution in an iterative process. In each iteration, the velocity and the position of particles are updated by

$$\begin{aligned} V_i(k) = w(t).V_i(k-1) &+ \\ c_1 r_1\left\{X_i^*(k-1) - X_i(k-1)\right\} &+ \\ c_2 r_2\left\{X^{**}(k-1) - X_i(k-1)\right\}& \\ i = 1,2,\cdots,n& \end{aligned} \quad (7)$$

$$X(k) = X(k-1) + V(k) \quad (8)$$

The inertia weight $w(k)$ is gradually decreased during the iterative process using the relation

$$w(k) = \alpha \cdot w(k-1) \quad (9)$$

The iterative process of updating the particle positions and velocities based on the objective function values is continued until the desired conditions are satisfied.

4. ADAPTIVE BINARY PARTICLE SWARM OPTIMIZATION

The binary version of the PSO (BPSO) , also suggested by Kennedy and Eberhart [19] enables the algorithm to operate in binary spaces. The particles in this version consist of binary 0's and 1's. Therefore, the main difference between the original PSO and the BPSO is that Eq. (8) is replaced by the following equation.

$$X_{ij}(t)=\begin{cases}1, & if\ r_3<S\left(X_{ij}(k)\right)\\ 0, & otherwise\end{cases} \qquad (10)$$

Where

$$S\left(X_{ij}(k)\right)=\frac{1}{1+\exp\left(-V_{ij}(k)\right)}$$

The time varying inertia weight that is linearly reduced during the iterations in order to enhance the computational efficiency is suggested in [20] instead of using Eq. (9).

$$w(k)=\left(w^{fin}-w^{ini}\right)\times\left(\frac{K^{\max}-k}{K^{\max}}\right)+w^{ini} \qquad (11)$$

The time-varying acceleration coefficients are introduced in [21] with a view to efficiently control the search process and convergence to the global solution. A large cognitive component and small social component at the beginning allows particles to move around the search space instead of prematurely moving towards the population best. A small cognitive component and a large social component during the latter stage allow the particles to converge to the global optimum.

$$c_1=\left(c_1^{fin}-c_1^{ini}\right)\left(\frac{k}{K^{\max}}\right)+c_1^{ini},\quad c_1^{fin}<c_1^{ini}$$
$$c_2=\left(c_2^{fin}-c_2^{ini}\right)\left(\frac{k}{K^{\max}}\right)+c_2^{ini},\quad c_2^{fin}>c_2^{ini} \qquad (12)$$

In the proposed formulation, the inertia weight and acceleration coefficients are adaptively changed with a view of enhancing the computational efficiency, improving the search capabilities and obtaining the global optimal solution.

5. PROPOSED ALGORITHM

The global solution of any optimization algorithm can be obtained by repeatedly running the algorithm with different initial values and choosing the best solution that minimizes the objective function as the global solution, whereas the PSO algorithm provides the global solution but is a time consuming process. The applications of PSO to UC problems have been proposed by various researchers [15-17], most of them differing in the method of representation, cost evaluation and handling of spinning reserve and minimum up/down constraints, which increase the complexity of problem formulation and solution methodology.

The method proposed in this paper uses a binary version of PSO along with an intelligent scheme for generating initial population and an efficient repairing mechanism to handle constraints with a goal of enhancing the search process, improving the computational efficiency and obtaining the global solution. The algorithm also adjusts adaptively the time varying inertia weight and acceleration coefficients in order to provide a balance between global and local explorations.

5.1 Representation of PSO variables

The binary variable $U_{i,t}$, which represents on/off status of unit i at hour-t is considered as the PSO variable. Each particle is therefore represented in matrix form as shown in Fig. 1

	1	2	3	4	…	T
1	0	0	1	1	…	0
2	0	1	1	1	…	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.
N	1	1	1	0	…	1

Fig.1 Representation of a particle

5.2 Generating Initial Population

It is difficult to generate feasible solution when initial population is generated at random. All unites are almost committed at heavy load while most of them are decommitted at light load. The initial population is therefore generated from the load curve [12] as shown in Fig.2.

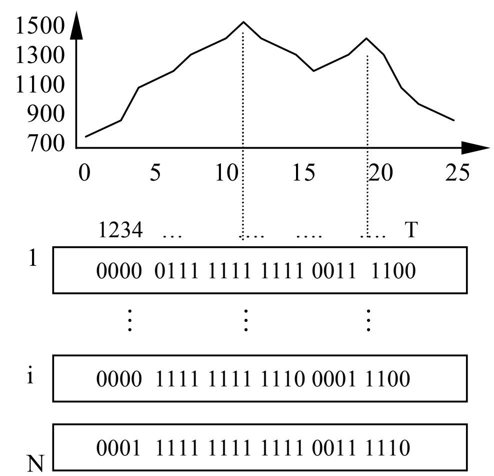

Fig. 2 Initial Population

5.3 Repair Algorithm

Spinning reserve, minimum up/down time constraints are important in UC problems. During iterative process, these constraints are often violated and the system may suffer from deficiency in units. At this stage, a repair algorithm can enhance the solution process. The proposed repair algorithm is outlined below.

1. If spinning reserve constraint is not satisfied, randomly change an off status unit to on $(0\rightarrow 1)$.
2. If the net minimum power generation of on status units is greater than the power demand, randomly change an on status unit to off $(1\rightarrow 0)$.
3. If minimum up/down time constraint is violated, identify the stream of bits that causes violation and alter them in order to overcome this violation. For example a string of 1111001111 may be modified either as 1111111111 or 1110001111 or 1111000111.

However, the one that requires least bit changes is chosen for repair.

4. Repeat steps 1-3 till all the constraints are satisfied.

5.4 Economic Load Dispatch

The economic load dispatch is an intensive computational part in UC problem. It is solved using lambda iteration method [1] based on the principle of equal incremental cost as the fuel cost is represented by a quadratic cost function. Lambda iteration method is carried out for various generating unit schedules of each particle using the expression.

$$P_i^t = \frac{\lambda}{2a_i + b_i} \tag{13}$$

5.5 Cost Evaluation

The PSO searches for the optimal solution by minimizing a cost function. The net fuel and start-up costs of the generating units, Eq. (1), are considered as the cost function to be minimized in the proposed approach.

5.6 Stopping Criteria

The process of generating new particles can be terminated either after a fixed number of iterations or if there is no further significant improvement in the global best solution.

5.7 Algorithm

The algorithm of the proposed solution methodology for UC problem is outlined.

1. Read the input data of the UC problem
2. Choose population size, m, and other PSO parameters
3. Set $k = 0$
4. Randomly generate initial population consisting m particles considering load curve.
5. Randomly generate m initial velocity values.
6. $k = k + 1$
7. Repair the particles to satisfy the spinning reserve and minimum up/down constraints.
8. For each particle, solve ELD and compute the cost function using Eq. (1).
9. Search for particle best and global best positions and store them.
10. Obtain values for $w(k), c_1$ & c_2 using Eqs. 11 and 12.
11. Update particle velocity and positions using Eqs. 7 and 10.
12. Check for convergence. If converged, stop and print the optimal solution corresponding to the global best position. Otherwise, go to step-6.

6. SIMULATION RESULTS

The proposed method (PM) has been tested on systems with 10, 20, 40, 60, 80 and 100 generating units. The unit data and load demand data for 24 hours for the system with 10 units are given in Tables A.1 and A.2 of the appendix respectively [11]. The data for other larger systems are obtained by duplicating the data of 10 unit system and adjusting the load demand in proportion to the system size. The population size is chosen as 30 for all the test problems. The maximum number of generations for convergence check is taken as 500, 1000, 2000, 3000, 4000 and 5000 for 10, 20, 40, 60, 80 and 100 unit systems respectively. The best production cost of the PM is compared with LRM [11], EALRM [10], genetic algorithm based method (GAM) [11], evolutionary programming based method (EPM) [14] in order to validate the results in Table 1. The analysis of this table indicates that the PM offers global optimal solution that corresponds to lower production cost than that of other methods. The average computation time (ACT) of the PM is graphically compared with evolutionary algorithms of GAM and EPM in Fig. 3. The computation times given in articles [11] and [14] for GAM and EPM were measured before a decade and hence are suitably scaled down using a factor of 0.5 with a view to compare with the computation times of PM executed using the present day fast computers. From this figure, it is very clear that the PM is reasonably faster than the other two methods.

Table 1 Comparison of total production cost

No of units	Best Production Cost ($)				
	LRM [11]	EALRM [10]	GAM [11]	EPM [14]	PM
10	565825	565508	565825	564551	563978
20	1130660	1126720	1126243	1125494	1125036
40	2258503	2249790	2251911	2249093	2246622
60	3394066	3371188	3376625	3371611	3367365
80	4526022	4494487	4504933	4498479	4505511
100	5257277	5615893	5627437	5623885	5623248

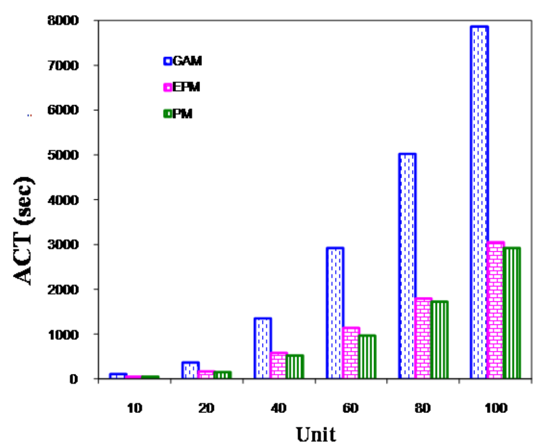

Fig. 3 Comparison of ACT

7. CONCLUSIONS

An adaptive binary PSO based algorithm has been proposed for unit commitment in this paper. This intelligent generation of initial population and the repairing mechanism has enabled the algorithm to provide faster solution. The adaptive adjustments of inertia weight and acceleration coefficients have made the algorithm to provide a robust solution. This method has been found to be ideally suitable for practical implementation.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the authorities of Annamalai University for the facilities offered to carryout this work.

APPENDIX

Table A.1. Unit data for the 10 unit system

Unit	1	2	3	4	5	6	7	8	9	10
P^{max}	455	455	130	130	162	80	85	55	55	55
P^{min}	150	150	20	20	25	20	25	10	10	10
a	1000	970	700	680	450	370	480	660	665	670
b	16.19	17.26	16.6	16.5	19.7	22.26	27.74	25.92	27.27	27.79
c	0.00048	0.00031	0.002	0.00211	0.00398	0.000712	0.00079	0.00413	0.00222	0.00173
T^{up}	8	8	5	5	6	3	3	1	1	1
T^{down}	8	8	5	5	6	3	3	1	1	1
HST	4500	5000	550	560	900	170	260	30	30	30
CST	9000	10000	1100	1120	1800	340	520	60	60	60
T^{cold}	5	5	4	4	4	2	2	0	0	0
Initial status	8	8	-5	-5	-6	-3	-3	-1	-1	-1

Table A.2. Load demand data

Hour	1	2	3	4	5	6	7	8	9	10	11	12
Load (MW)	700	750	850	950	1000	1100	1150	1200	1300	1400	1450	1500

Hour	13	14	15	16	17	18	19	20	21	22	23	24
Load (MW)	1400	1300	1200	1050	1000	1100	1200	1400	1300	1100	900	800

REFERENCES

[1] AJ. Wood and BF. Wollenberg. (1996). Power generation, operation and control. John Wiley and sons, New York.

[2] HH. Happ, RC. Johnson and WJ. Wright. (1971). Large Scale Hydrothermal unit Commitment method and results. IEEE Trans. On PAS, PAS – 90: 1373-1383.

[3] CJ. Baldwin, KM. Dale and RF. Dittrich. (1960). A study of econmic shutdown of generating units in daily dispatch. AIEE Tr on PAS, 78: 1272-1284.

[4] WL. Snyder, HD. Powell Jr and JC. Rayburn. (1987). Dynamic programming Approach to unit commitment. IEEE Trans. Power Syst., PWRS S-2(2): 339-350.

[5] WJ. Hobbs, G. Hermon, S. Warner and GB. Sheble. (1982). An Enhanced Dynamic programming Approach for unit commitment. IEEE Trans. on PAS, PAS-101: 79-86.

[6] TS. Dillon and KW. Edwin. (1978). Integer Programming approach to the problem of optimal unit commitment with probabilistic reserve Determination. IEEE Trans. on PAS, PAS – 97(6): 2154-2166.

[7] AI. Cohen and M.Yoshimura. (1983). A Branch and bound algorithm for unit commitment. IEEE Trans. on PAS, PAS-10: 444 – 451.

[8] FN. Lee. (1989). A Fuel constrained unit commitment method. IEEE Trans. Power Syst., 4(3): 691-698.

[9] CP. Cheng, CW. Liu and CC. Liu. (2000). Unit commitment by Lagrangian relaxation and genetic algorithem, IEEE Trans. Power Syst., 15: 707-714.

[10] Weerakorn Ongsakul and Nit petcharaks. (2004). Unit commitment by enhanced adaptive Lagrangian Relaxation, IEEE Trans. Power Syst., 19(1): 620-628.

[11] SA. Kazarlis, AG. Bakirtzis and V. Petidis. (1996). A genetic algorithm solution to unit commitment problem, IEEE Trans. Power Syst., 11: 83-92.

[12] T. Senjyu, H. Yamashiro, K. Shimabukuro, K. Uezato and T. Funabashi. (2003). Fast solution for large-scale unit commitment problem using genetic algorithm, IEE Proc.-Gener. Transm. Distrib., 150(6): 753-760.

[13] DN. Simopoulos, SD. Kavatza and CD. Vournas. (2006). Unit commitment by an enhanced simulated annealing algorithm , IEEE Trans. Power Syst., 21(1): 68-76.

[14] K.A. Juste, H. Kita, E. Tanaka and J. Hasegawa. (1999). An evolutionary programming solution to the unit commitment problem, IEEE Trans. Power Syst., (14): 1452-1459.

[15] Yun-Won Jeong and Jong-Bae Park. (2010)A new quantum-inspired binary PSO: Application to unit commitment problems for power systems, , IEEE Trans. Power Syst., (25(3): 1486-1495.

[16] B. Zhao, CX. Guo, BR. Bai and YJ. Cao. (2006). An improved particle swarm optimization algorithm for unit commitment, Elect. Power Energy Syst., 28(7): 482-490.

[17] X. Yuan, H. Nie, A. Su, L. Wang and Y. Yuan. (2009). An improved binary particle swarm optimization for unit commitment problem, Journal of Expert Systems with Applications, 36(4): 8049-8055.

[18] J. Kennady and RC. (1995). Eberhart.Particle swarm optimization, Proc. of IEEE Int. Conf. on Neural Networks, Piscataway, NJ, IV: 1942-48.

[19] J. Kennady and RC. Eberhart, (1997) A discrete binary version of the particle swarm algorithm, Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics, 5: 4104-4108.

[20] KT. Chaturvedi, M.Pandit and L.Srivastava. (2008). Self-organizing hierarchical particle swarm optimiation for nonconvex economic dispatch, IEEE Trans. Power Syst., 23(3): 1079-1089.

[21] Ratnaweera, SK. Halgamuge and HC. Watson. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. E Comput., 8(3): 240-255.