Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2015) 23: 354 – 369

doi:10.3906/elk-1212-36

A new heuristic method to solve unit commitment by using a time-variant acceleration coefficients particle swarm optimization algorithm

Arsalan NAJAFI*, **Mohsen FARSHAD**, **Hamid FALAGHI**
Department of Electrical Power Engineering, Faculty of Electrical and Computer Engineering,
University of Birjand, Birjand, Iran

Received: 06.12.2012 • **Accepted:** 28.02.2013 • **Published Online:** 23.02.2015 • **Printed:** 20.03.2015

Abstract: Unit commitment is one of the most important problems in power system operation. Because of the large amount of parameters and constraints, it contains a high level of complexity. In this paper a new method based on a time-variant acceleration coefficients particle swarm optimization algorithm has been proposed to solve the unit commitment problem. Integer coding (for satisfying minimum up/down constraints) and binary coding (for satisfying spinning reserve constraint) have been utilized in the proposed method. Simulations in the different cases have been done with different sizes. Numerical results have shown the superiority and better convergence of the proposed method in comparison with other methods.

Key words: Economic load dispatch, time-variant acceleration coefficients particle swarm optimization algorithm, unit commitment

1. Introduction

Unit commitment (UC) is a process that determines the optimum daily schedule of units in economic situations and operation constraints. This problem includes start up, shut down, and fuel costs over 24 h, 7 days, or maybe 1 month. UC is a complex problem that has various discrete and continuous variables. The power industry has used optimization techniques for solving the UC problem for many years. Therefore, millions of dollars are being saved every year [1].

Before the 1960s, the UC problem was limited to economic dispatch. In those times, the Kuhn–Tucker conditions characterized the optimal economic status. When these conditions were satisfied, all committed units were loaded according to their fuel, except those efficient units that were loaded with their peak power. Later, economic dispatch was developed by piecewise approximation of its cost function. Subsequently, a linear optimization method was suggested for the UC problem [2]. Dynamic programming could solve the unit commitment efficiently [3]. The Lagrange method was another way to solve the UC problem. By adding equal and unequal constraints to the objective function, the constraints of the problem were relaxed [4].

Recently, evolutionary methods have been utilized to solve the UC problem. These methods have attracted much attention because of their potential to reach global solutions. In [5], a method based on the genetic algorithm was suggested and applied to a sample case. An integer-coded genetic algorithm (ICGA) was used in [6] and an evolutionary method based on a shuffled frog leaping algorithm proposed in [7]. This method used integer coding for UC's solutions and used a penalty factor for satisfying some of the constraints.

*Correspondence: arsalan.najafi@birjand.ac.ir

Binary methods were suggested in [8] and [9]. A method based on particle swarm optimization (PSO) and the quantum concept was also suggested in [10]. Other methods to solve the UC problem include evolutionary programming [11], simulated annealing [12], bacterial foraging [13], the imperialistic competition algorithm (ICA) [14], the deterministic annular crossover genetic algorithm (DACGA) [15], and harmony search (HS) [16]. As a combination method, fuzzy dynamic programming was suggested for the UC problem [17].

In this paper, a new heuristic method based on a time-variant acceleration coefficients particle swarm optimization algorithm (TVACPSO) has been proposed. Early and wrong convergence is prevented in this variant of the PSO intelligent algorithm by varying its acceleration coefficients. In order to reach a better satisfaction of the UC's constraints, a new integer/binary coding has been utilized in the proposed method. Simulation results in some sample cases show the ability of the proposed method in comparison with other methods.

2. Problem formulation

2.1. Objective function

Operation cost in the UC problem is equal to the sum of fuel, start up, and shut down costs that must be minimized in the duration of planning. Therefore, the objective function is expressed by [18]:

$$min\sum_{t=1}^{T}\sum_{i=1}^{n}(FC_i(P_i^t)\cdot u_i^t)+SU_i+SD_i, \tag{1}$$

where P_i^t is the power generation of unit i at hour t, n is the number of units, T is the duration time of planning, FC_i is the fuel cost of unit i, and SU_i and SD_i are the start up and shut down costs of unit i, respectively. Additionally, u_i^t shows the state of unit i at hour t that equals zero or one. Fuel cost is the main part of the mentioned equation and it is calculated as follows:

$$FC_i(P_i^t)=a_i(P_i^t)^2+b_iP_i^t+c_i, \tag{2}$$

where a_i, b_i, and c_i are fuel cost coefficients of unit i.

Th shut down cost is a small term compared to the other costs and therefore is usually neglected. The start up cost is expressed mathematically as below [6]:

$$SU_i=\begin{cases} H_{\text{start up}} & T_i^{OFF}\le MDT_i+(T_{\text{cold}}) \\ C_{\text{start up}} & T_i^{OFF}> MDT_i+(T_{\text{cold}}) \end{cases}, \tag{3}$$

where $H_{\text{start up}}$ and $C_{\text{start up}}$ are respectively the hot and cold start up costs, MDT_i is the minimum down time of unit i, and T_{cold} determines the time of cold or hot start up.

2.2. Constraints

The constraints that must be considered in the UC problem are mentioned here:

1) Minimum up and down constraints: Units cannot be turned on or turned off immediately and need to be on or off for specific hours. Minimum up/down times are as below:

$$\begin{cases} T_i^{ON}\ge MUT_i \\ T_i^{OFF}\ge MDT_i \end{cases} \quad \text{for } i=1,2,..,n, \tag{4}$$

where T_i^{ON} and T_i^{OFF} are the duration for which unit i remains on or off and MUT_i is the minimum up time of unit i.

2) Load balance constraint: Total output generation of units must be equal to demand as below:

$$\sum_{i=1}^{n} u_i^t \cdot P_i^t = P_d^t for t = 1, 2, ..., T, \tag{5}$$

where P_d^t is the total system demand at hour t.

3) Generator output limitation: Output generation of units must be in their maximum and minimum limitations as below:

$$P_{i\min} \leq P_i \leq P_{i\max} for i = 1, 2, ..., n, \tag{6}$$

where $P_{i\min}$ and $P_{i\max}$ are the allowable minimum and maximum output of generator i, respectively.

4) Spinning reserve constraint: In order to minimize the probability of load interruption, spinning reserve must be available in the power system. It can be specified in terms of excess megawatt capacity expressed by:

$$\sum_{i=1}^{n} u_i^t \cdot P_{i\max}^t \geq P_d^t + SR^t, \tag{7}$$

where SR^t is the amount of spinning reserve at hour t.

5) Ramp up and ramp down rate constraints: Because of production limits, the increasing or decreasing of power could not be higher or lower than a specific amount. This amount is one of the generator characters and is a security constraint for the generator that prevents damaging the rotor. These constraints determine the maximum changing slope of generator output. These constraints are as follows [19]:

$$\begin{array}{l} P_{i\max}(t) = \min\left\{P_{i\max}, P_i^{t-1} + \tau \cdot RU_i\right\} \\ P_{i\min}(t) = \max\left\{P_{i\min}, P_i^{t-1} - \tau \cdot RD_i\right\} \end{array}, \tag{8}$$

where RU_i and RD_i are the ramp up and ramp down rates of unit i, respectively.

3. Optimization algorithm

3.1. Particle swarm optimization

PSO is a population-based intelligent algorithm introduced by Eberhart and Kennedy [20]. Each particle in PSO is a candidate solution in multidimensional space of the problem. Candidate solutions have two main parts, current position (X_k) and current velocity (V_k), which are expressed by:

$$\begin{array}{l} X_k(t) = \left(x_k^1(t), x_k^2(t), ..., x_k^{ns}(t)\right) \\ V_k(t) = \left(v_k^1(t), v_k^2(t), ..., v_k^{ns}(t)\right) \end{array}, \tag{9}$$

where ns is the dimension of the problem and t is the iteration index. The new position of each particle is created by its current position and new velocity. New velocity is produced by four factors, the current velocity and position, the particle's best position ($Pbest$), and the best position among all of particles in all iterations ($Gbest$). Therefore, new velocity is obtained as follows:

$$v_k^j = \omega v_k^j + c_1 r_1 (pbest_k^j - x_k^j) + c_2 r_2 (gbest^j - x_k^j), \tag{10}$$

where ω is the particle inertia coefficient, c_1 and c_2 are acceleration coefficients, r_1 and r_2 are random numbers between 0 and 1, and k and j are the particle and its dimension indices, respectively. The new position of the particle is obtained by:

$$x_k^j = x_k^j + v_k^j. \tag{11}$$

3.2. Time-variant acceleration coefficients particle swarm optimization

A PSO algorithm with variable inertia coefficient (ω) can achieve good solutions but is feeble in obtaining optimum solutions. This is because of lack of variety in the search space. In this paper, a variant of PSO called TVACPSO has been utilized. This variant of PSO has a widespread search space that is the result of changing its acceleration coefficients. This advantage prevents wrong and early convergence. The idea behind TVACPSO is to improve the global search in the early part of the optimization and to encourage the particles to converge towards the global optimum solution at the end of the search. This is done by changing acceleration coefficients c_1 and c_2. In the beginning of optimization, the value of c_1 is large and that of c_2 is small. Increasing the iterations, c_1 will decrease and c_2 will increase. This process means that the solutions are around *Pbest* at first, but they will be gathered around the global best finally. This process prevents early convergence. In the proposed algorithm, the acceleration coefficients vary as below [21]:

$$c_1 = (c_{1f} - c_{1i}) \cdot \frac{iter}{iter_{\max}} + c_{1i}, \tag{12}$$

$$c_2 = (c_{2f} - c_{2i}) \cdot \frac{iter}{iter_{\max}} + c_{2i}, \tag{13}$$

where c_{1i}, c_{2i}, c_{1f}, and c_{2f} are the initial and final values of the acceleration coefficients, respectively.

4. UC solving based on TVACPSO

UC is a complex problem because of its nonlinearity. It also has many constraints that further complicate the problem. In this paper, the TVACPSO algorithm has been used to solve the UC problem. This algorithm is very efficient for nonlinear problems. A new heuristic method based on the combination of binary and integer coding has been proposed to accommodate the UC constraints. The number of ON/OFF cycles of units is equal to the planning duration (here considered to be 24) because, by limiting the number of cycles to 5, which is common in the previous integer works, it is very difficult for an optimization algorithm to commit or decommit the units because it may increase the ON/OFF cycles over 5. Considering the number of cycles equal to 24 does not prevent committing or decommitting the units. According to this combination method, initializing the primary population, updating the new solutions, and satisfying the minimum up/down time constraints have been achieved by the integer coding. Load balancing and spinning reserve constraints have been satisfied by the binary coding, also. Figure 1 shows a flowchart of the proposed method.

4.1. Initial population

The initial population has been produced by integer coding. Active (ON) hours have been shown with positive integer codes and OFF hours by negative integer codes. Zeroes show that the period of planning has been finished. Positive and negative generated codes are produced such that the absolute sum of them is equal to 24. A sample particle in the initial population is shown in Figure 2. The superiority of this method over the others lies in the fact that many of the infeasible solutions (related to the minimum up/down time constraints) are eliminated in population generating. The spinning reserve constraint is not satisfied by the initial population and therefore a penalty factor is used for it in the dispatch and evaluation stage. According to mentioned points, the population is initialized as follows.

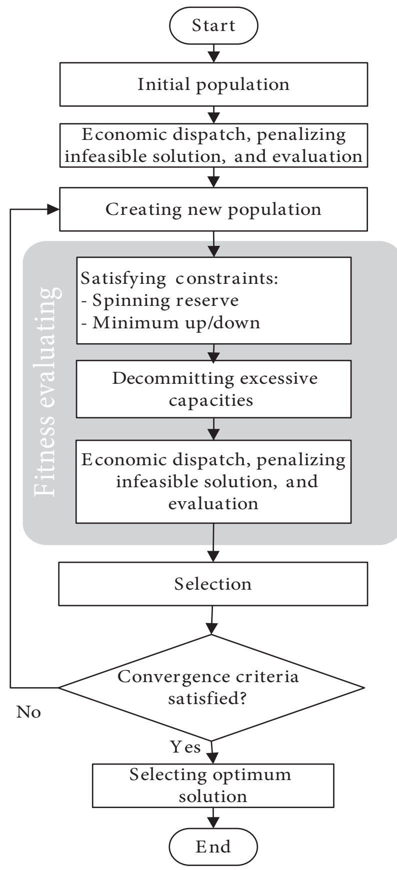

Figure 1. Flowchart of the proposed method.

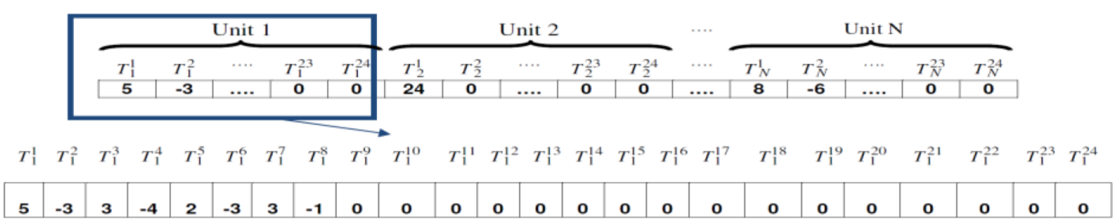

Figure 2. A sample particle in the initial population.

The first cycle is expressed by [7]:

$$T_i^1 = \begin{cases} rand((MUT_i - T_i^0), T) & if T_i^0 > 0 \\ -rand((MDT_i - T_i^0), T) & if T_i^0 < 0 \end{cases}, \tag{14}$$

where T_i^0 is the initial state of unit i. Other cycles are generated as follows.

If $T_i^{c-1} < 0$, the next cycle will change as follows:

$$T_i^c = \begin{cases} rand(MUT_i, RT_i^{c-1}) & RT_i^{c-1} > MUT_i \\ RT_i^{c-1} & otherwise \end{cases}, \tag{15}$$

and if $T_i^{c-1} > 0$, the next cycle will change as follows:

$$T_i^c = \begin{cases} -rand(MDT_i, RT_i^{c-1}) & RT_i^{c-1} > MDT_i \\ -RT_i^{c-1} & otherwise \end{cases}, \tag{16}$$

where RT_i^{c-1} is obtained by:

$$RT_i^{c-1} = T - \sum_{l=1}^{c-1} \left|T_i^l\right|. \tag{17}$$

4.2. Creating a new population

New populations are produced by the TVACPSO algorithm. As mentioned above, the cycle's absolute sum of any solution (particle) must be equal to the duration of planning. This constraint may not be satisfied by new solutions. For much greater or smaller values than 24, it must be divided between cycles proportionally.

The rand functions of the TVACPSO algorithm produce a random number between 0 and 1, and therefore new population is not an integer while the solution of the UC problem must be represented only by integer numbers. To solve this problem, a round function has been utilized. After rounding, maybe the sum of the ON/OFF cycles is not equal to 24. This difference will affect the last nonzero cycle.

4.3. Priority list

Units are ranked based on the priority list. The priority list is based on units' parameters of units. To create the priority list, it is assumed that the units operate with maximum power generation and then their increasing rates are calculated. At last, the priority list is expressed as follows [22]:

$$\alpha_i = 2a_i \cdot P_{i\max} + b_i, \tag{18}$$

where α_i is the priority of unit i for committing. Lower α_i values mean higher priority for committing.

4.4. Spinning reserve satisfaction

There are 24 binary digits for each unit during a day in the binary solving of the UC problem. In this solving pattern, digits 0 and 1 show OFF and ON states, respectively. In the proposed method, if the spinning reserve constraint is violated, deactivated units will commit according to the priority list based on cheaper units. This process continues until the spinning reserve constraint is satisfied. By using this method to satisfy the spinning reserve, expensive units are replaced with cheaper units. Figure 3 shows a flowchart of the proposed heuristic method to handle the spinning reserve constraint for G units in the T scheduling horizon.

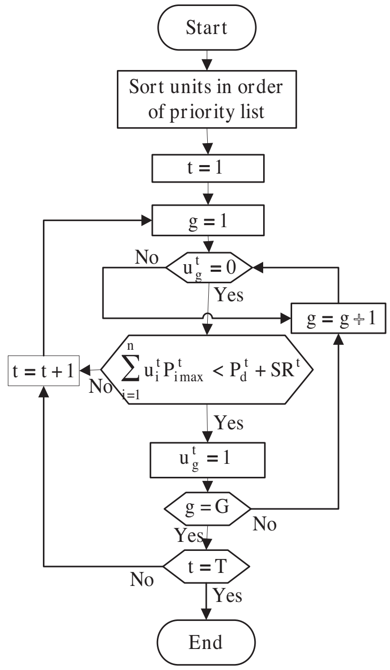

Figure 3. Flowchart of handling the spinning reserve constraint.

4.5. Satisfying minimum up/down constraints

After satisfying the spinning reserve constraint by new solutions, the minimum up/down constraints must be checked for all cycles. If these constraints are violated then they must be modified. Each unit has a feasible and an infeasible region and the proposed method converts the infeasible region to a feasible one (Figure 4). This technique is expressed as follows: in order to satisfy the minimum up time, if $T_i^c < MUT_i$, then the amount of unit minimum up time is added to T_i^c as follows:

$$T_i^c = T_i^c + MUT_i. \tag{19}$$

Similarly, to assure minimum down time, if $-T_i^c < MDT_i$, then:

$$T_i^c = T_i^c - MDT_i. \tag{20}$$

Meanwhile, if the remaining time is lower than the minimum up/down time in the last nonzero cycle, then:

$$T_i^c = RT_i^{c-1}. \tag{21}$$

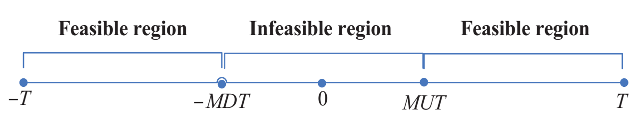

Figure 4. Positions of feasible and infeasible regions for minimum up/down constraints.

4.6. Decommitting of excessive capacity

The unit decommitment has been done after satisfying the minimum up/down constraint. As mentioned previously, the spinning reserve constraint is satisfied by committing new units. Therefore, it may create excessive capacity in the system. Excessive capacity is decommitted from the system according to the priority list. This process continues until active units are able to supply the load and spinning reserve and also the minimum up/down time constraint is not violated. Figure 5 illustrates a flowchart of the mentioned heuristic process.

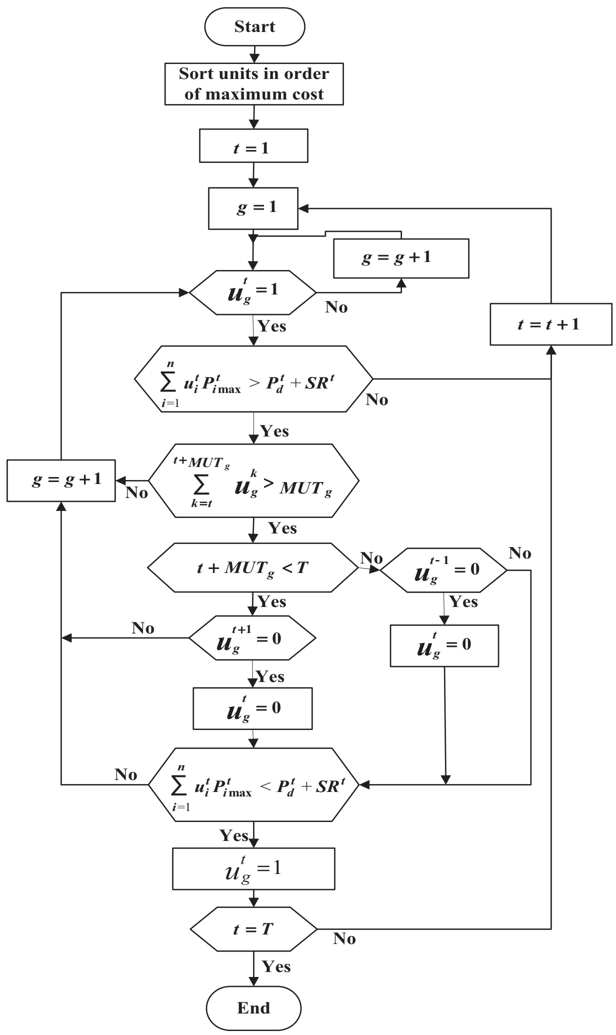

Figure 5. Flowchart of decommitting the excessive capacity.

4.7. Economic load dispatch and evaluation

Economic load dispatch is done by the classic lambda iteration method. Producing power of units at each hour is determined in this step. Production cost, which is specified by generated power, is indicated as represented by Eq. (2). It is possible that the load balancing and the spinning reserve constraints are violated again after satisfaction of the minimum up/down constraints. A penalty factor is used for compensation of violated spinning reserve as below [14]:

$$PX = pf * \sum_{t=1}^{T} \frac{1}{P_d^t} \cdot ((P_d^t + SR^t) - \sum_{i=1}^{n} u_i^t . P_{i\max}^t), \tag{22}$$

where PX is the penalty amount and pf is the constant penalty factor.

Therefore, the fitness function will be as follows:

$$fitness = FC_T + SU_T + PX, \tag{23}$$

where $fitness$, FC_T, and SU_T are the values of the fitness function, total fuel cost, and total start up cost, respectively.

4.8. Satisfying ramp rate constraints

Commitment of required units, decommitment of excessive units, and economic dispatch are done by considering the nominal maximum and minimum output power of units. The ramp rate constraints affect the maximum and minimum available power of units. Before satisfaction of the spinning reserve constraint and decommitment of excessive capacity, new maximum and minimum output power according to the ramp rate constraints must be determined at each planning hour. Execution of the economic dispatch at each hour determines these bounds.

5. Numerical result

5.1. Without considering ramp rate constraint

The proposed method has been tested on the UC problem for realistic power systems of different sizes, which consist of 10, 20, 40, 60, 80, and 100 units over a scheduling period of 24 h. The load data and details of a 10-unit system are given in [7]. For the 20-unit system, the data of the 10-unit system have been duplicated and the load data have been doubled. To construct the other test systems, the same procedure has been applied. The spinning reserve amount has been considered to be 10% of hourly demand. Simulations have been done on a Pentium IV 2.6 GHz with 2 gigabyte RAM. The results of 20 runs of the proposed method with different populations are given in Table 1. These results show the best, average, and worst values and standard deviations of solutions (with respect to dollars) and also one run time duration (with respect to seconds). Figure 6 shows 5 sequential runs of the 10-unit system with different initial populations so that all of these runs converge to optimum cost at last. Figure 7 displays the convergence of all particles and the *Gbest* particle to an optimum solution in a sample run. This figure shows that the particles are spread in the search space at first and converge to the best solution gradually. All of the units' production and start up costs in 24 h are given in Table 2. In order to increase the validation of the proposed method, 100 generator system results, including the situation of ON/OFF hours, are shown in Table 3. Best iterations of 20 runs for each case and run time are available in Table 4. Iterations and times are increased with increasing number of units.

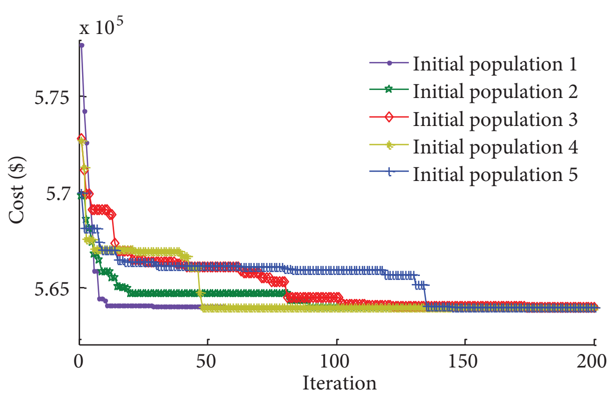

Figure 6. Five sequential runs of 10-unit system with different initial populations.

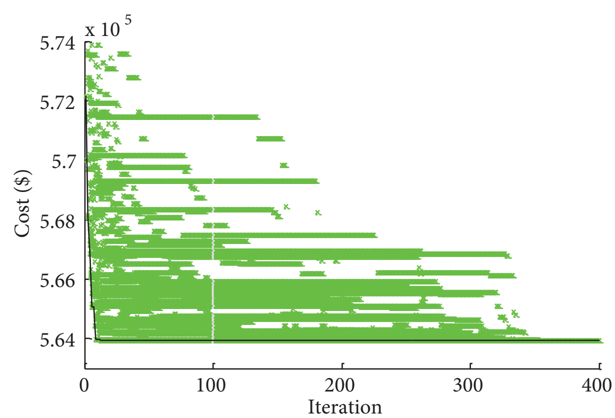

Figure 7. All particles and global best convergence in 10-unit system.

Table 1. Results of 20 runs for different populations.

Population size	Best solution	Mean solution	Worst solution	Standard deviation	Time
10	563,937.64	564,443.60	566,085.89	895.65	6.7
30	563,937.64	563,937.64	563,937.64	0	20
50	563,937.64	563,937.64	563,937.64	0	26.2
70	563,937.64	563,937.64	563,937.64	0	39.6

Table 2. Configuration of on/off cycle, operation cost, and schedule for 24 h in 10-unit system.

	Unit											
	1	2	3	4	5	6	7	8	9	10		
Duration of ON/OFF cycle	24	24	–5	–4	–2	–8	–8	–9	–10	-11		
			16	17	20	6	6	4	2	1		
			-3	–3	–2	–5	–5	–6	–12	–12		
						4	3	1				
						–1	–2	–4				
Time	Produced power (MW)										Start up cost ($)	Operation cost ($)
1	**455**	**245**	0	0	0	0	0	0	0	0	0	13,683.12
2	**455**	**295**	0	0	0	0	0	0	0	0	0	14,554.5
3	**455**	**370**	0	0	**25**	0	0	0	0	0	900	16,809.45
4	**455**	**455**	0	0	**40**	0	0	0	0	0	0	18,597.68
5	**455**	**390**	0	**130**	**25**	0	0	0	0	0	560	20,020.02
6	**455**	**360**	**130**	**130**	**25**	0	0	0	0	0	1100	22,387.04
7	**455**	**410**	**130**	**130**	**25**	0	0	0	0	0	0	23,261.99
8	**455**	**455**	**130**	**130**	**30**	0	0	0	0	0	0	24,150.32
9	**455**	**455**	**130**	**130**	**85**	**20**	**25**	0	0	0	860	27,251.07
10	**455**	**455**	**130**	**130**	**162**	**33**	**25**	**10**	0	0	60	30,057.54
11	**455**	**455**	**130**	**130**	**162**	**73**	**25**	**10**	**10**	0	60	31,916.07
12	**455**	**455**	**130**	**130**	**162**	**80**	**25**	**43**	**10**	**10**	60	33,890.15
13	**455**	**455**	**130**	**130**	**162**	**33**	**25**	**10**	0	0	0	30,057.54
14	**455**	**455**	**130**	**130**	**85**	**20**	**25**	0	0	0	0	27,251.07
15	**455**	**455**	**130**	**130**	**30**	0	0	0	0	0	0	24,150.32
16	**455**	**310**	**130**	**130**	**25**	0	0	0	0	0	0	21,513.64
17	**455**	**260**	**130**	**130**	**25**	0	0	0	0	0	0	20,641.83
18	**455**	**360**	**130**	**130**	**25**	0	0	0	0	0	0	22,387.04
19	**455**	**455**	**130**	**130**	**30**	0	0	0	0	0	0	24,150.32
20	**455**	**455**	**130**	**130**	**162**	**33**	**25**	**10**	0	0	490	30,057.54
21	**455**	**455**	**130**	**130**	**85**	**20**	**25**	0	0	0	0	27,251.07
22	**455**	**455**	0	0	**145**	**20**	**25**	0	0	0	0	22,735.54
23	**455**	**425**	0	0	0	**20**	0	0	0	0	0	17,645.36
24	**455**	**345**	0	0	0	0	0	0	0	0	0	15,427.43
	Total cost: 563,937.64										4090	559,847.6

Table 3. Situation of units for 24 h for 100-unit system.

Units	Hours																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1–19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
21	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
22–24	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
25	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
26–27	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
28–30	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
31–33	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
34	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1
35	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
36	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
37–40	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
41	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
42	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
43–45	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
46	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
47	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
48–49	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
50	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
51	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0
52	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0
53	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0
54	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0
55	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0
56	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0
57	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0
58	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0
59–60	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0
61–63	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0
64	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0
65	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	0	0
66–67	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0
68	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0
69	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	0	0
70	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0
71	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	0	1	0	0	0	0
72–74	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0
75	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0
76–77	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0
78	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0
79–82	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
83	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
84–88	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
89–97	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
98	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
99–100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4. Best iterations and run times for units.

No. of units	Best iteration	Time (s)
10	48	20
20	138	233
40	640	640
60	751	1050
80	819	1325
100	863	1700

Table 5 lists the comparison results of the TVACPSO algorithm and the other methods. These results illustrate that TVACPSO has a better convergence than the other methods in all of the studied systems. The ICA and ICGA methods considered five cycles to solve the UC problem. Comparing these methods and the proposed method shows the superiority of considering 24 cycles instead of 5 cycles to code the solutions. Running times of different methods are also given in this table.

Table 5. Comparison of results of case study 1.

Method	Best cost ($)	Average cost ($)	Worst cost ($)	Time (s)	Method	Best cost ($)	Average cost ($)	Worst cost ($)	Time (s)
10 units					60 units				
EP [11]	564,551	565,532	566,231	100	EP	3,371,611	3,376,255	3,381,012	2267
GA [5]	565,825	—	570,032	221	GA	3,376,625	—	3,384,252	5840
LR [5]	566,107	—	—	257	LR	3,374,994	—	—	1594
ICGA [6]	566,404	—	—	7.4	ICGA	3,378,108	—	—	117.3
SA [12]	565,828	565,988	566,260	3	SA	—	—	—	—
ICA [14]	563,938	564,406	—	48	ICA	3,371,722	—	—	366
HS [16]	565,828	—	—	—	HS	3,375,138	—	—	1021
DACGA [15]	563,987	—	—	—	DACGA	—	—	—	—
TVACPSO	**563,938**	**563,938**	**563,938**	**20**	**TVACPSO**	**3,365,250**	**3,368,005**	**3,370,917**	**1050**
20 units					80 units				
EP	1,125,494	1,127,257	1,129,793	340	EP	4,498,479	4,505,536	4,512,739	3584
GA	1,126,243	—	1,132,059	733	GA	4,504,933	—	4,510,129	10,036
LR	1,128,362	—	—	514	LR	4,496,729	—	—	2122
ICGA	1,127,244	—	—	22.4	ICGA	4,498,943	—	—	176
SA	1,126,251	127,955	1,129,112	17	SA	4,498,076	4,501,156	4,503,987	405
ICA	1,124,274	—	—	63	ICA	4,497,919	—	—	994
HS	1,127,377	—	—	92	HS	4,500,745	—	—	2157
TVACPSO	**1,123,759**	**1,124,297**	**1,124,480**	**233**	**TVACPSO**	**4,487,407**	**4,491,606**	**4,497,974**	**1325**
40 units					100 units				
EP	249,093	2,252,612	2,256,085	1176	EP	5,623,885	5,633,800	5,639,148	6120
GA	2251,911	—	2,259,706	2697	GA	5,627,437	—	5,637,914	15,733
LR	2,250,223	—	—	1066	LR	5,620,305	—	—	2978
ICGA	2,254,123	—	—	58.3	ICGA	5,630,838	—	—	242
SA	2,250,063	2,252,125	2,254,539	88	SA	5,617,876	5,624,301	5,628,506	316
ICA	2,247,078	—	—	151	ICA	5,617,913	—	—	1376
HS	2,250,968	—	—	467	HS	5,622,350	—	—	3710
TVACPSO	**2,245,000**	**2,245,951**	**2,247,237**	**640**	**TVACPSO**	**5,607,938**	**5,613,650**	**5,621,386**	**1700**

5.2. Considering ramp rate constraint

This test case has 26 units in a time horizon of 24 h. Test case data are given in [19] and Tables 6 and 7. Spinning reserve has been considered to be 5% of hourly demand. Ramp rate constraints have been considered. Of course, start up and shut down costs have been neglected according to the original reference. In order to show efficiency of the proposed method by considering ramp rate limits, TVACPSO has been run 20 times and results have been compared with PSO. Results show the superiority of TVACPSO compared to PSO. In order to show the difference between considering the ramp rate limit or not, TVACPSO has been run again 20 times without ramp rate considerations. Figure 8 and Table 8 display these three mentioned states.

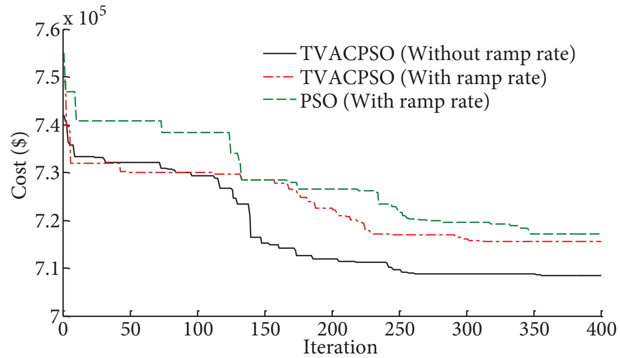

Figure 8. TVACPSO and PSO convergence in case of 26-unit system.

Table 6. Generating unit data for 26-unit system.

Unit	Pmin (MW)	Pmax (MW)	a	B	c	MUT	MDT	Initial state	UR	DR
Unit 1	2.4	12	0.02533	25.5472	24.3891	0	0	–1	48	60
Unit 2	2.4	12	0.02649	25.6753	24.411	0	0	–1	48	60
Unit 3	2.4	12	0.02801	25.8027	24.6382	0	0	–1	48	60
Unit 4	2.4	12	0.02842	25.9318	24.7605	0	0	–1	48	60
Unit 5	2.4	12	0.02855	26.0611	24.8882	0	0	–1	48	60
Unit 6	4	20	0.01199	37.551	117.7551	0	0	–1	30.5	70
Unit 7	4	20	0.01261	37.6637	118.1083	0	0	–1	30.5	70
Unit 8	4	20	0.01359	37.777	118.4576	0	0	–1	30.5	70
Unit 9	4	20	0.01433	37.8896	118.8206	0	0	–1	30.5	70
Unit 10	15.2	76	0.00876	13.3272	81.1364	3	2	3	38.5	80
Unit 11	15.2	76	0.00895	13.3588	81.298	3	2	3	38.5	80
Unit 12	15.2	76	0.0091	13.38085	81.4641	3	2	3	38.5	80
Unit 13	15.2	76	0.00932	13.4073	81.6259	3	2	3	38.5	80
Unit 14	25	100	0.00623	18	217.8952	4	2	–3	51	74
Unit 15	25	100	0.00612	18.1	218.335	4	2	–3	51	74
Unit 16	25	100	0.00598	18.2	218.7752	4	2	–3	51	74
Unit 17	54.25	155	0.00463	10.694	142.7348	5	3	5	55	78
Unit 18	54.25	155	0.00473	10.7154	143.0288	5	3	5	55	78
Unit 19	54.25	155	0.00481	10.7367	143.3179	5	3	5	55	78
Unit 20	54.25	155	0.00487	10.7583	143.5972	5	3	5	55	78
Unit 21	68.95	197	0.00259	23	259.131	5	4	–4	55	99
Unit 22	68.95	197	0.0026	23.1	259.649	5	4	–4	55	99
Unit 23	68.95	197	0.00263	23.2	260.176	5	4	–4	55	99
Unit 24	140	350	0.00153	10.8616	177.0575	8	5	10	70	120
Unit 25	100	400	0.00194	7.4921	310.0021	8	5	10	50.5	100
Unit 26	100	400	0.00195	7.5031	311.9102	8	5	10	50.5	100

Table 7. Load data for 26-unit system.

Hour	1	2	3	4	5	6	7	8	9	10	11	12
Load	1700	1730	1690	1700	1750	1850	2000	2430	2540	2600	2670	2590
Hour	13	14	15	16	17	18	19	20	21	22	23	24
Load	2590	2550	2620	2650	2550	2530	2500	2550	2600	2480	2200	1840

Table 8. Comparison of results of 26-unit system.

	Method	Best solution	Mean solution	Worst solution	Time (s)	Time/iteration (s)
With ramp rate	PSO	717,434	725,274	751,258	931	3.17
	TVACPSO	715,086	716,720	717,849	906	3.17
Without ramp rate	TVACPSO	708,474	713,703	717,066	748	2.98

5.3. Sensitivity analysis

As mentioned above, TVACPSO has been created by varying the acceleration coefficients of PSO. These coefficients change between their initial and final values.

Simulations have been done on a sample 20-unit system to show the sensitivity of TVACPSO with respect to the initial and final bounds of the acceleration coefficients (Table 9). The results show that the TVACPSO algorithm is very sensitive to the c_{1i} bound such that better results are obtained by increasing c_{1i}. It is also seen that the best results are obtained with c_{1i} equal to 2.5 and c_{2f} equal to 2.

Table 9. Results of sensitivity analysis of initial and final values of acceleration coefficients of TVACPSO in 20-unit system.

c_{1i}	c_{1f}	c_{2i}	c_{2f}	Best result	Average result	Std
1	0.5	0.5	1	1,124,274	1,125,467	1436
1.5	0.5	0.5	1	1,124,274	1,124,693	753
2	0.5	0.5	1	1,124,274	1,124,327	152
2.5	0.5	0.5	1	1,124,274	1,124,308	73
1	0.5	0.5	1.5	1,124,294	1,125,356	1172
1.5	0.5	0.5	1.5	1,124,274	1,124,724	1150
2	0.5	0.5	1.5	1,124,274	1,124,348	91
2.5	0.5	0.5	1.5	1,124,274	1,124,764	743
1	0.5	0.5	2	1,124,294	1,124,532	491
1.5	0.5	0.5	2	1,124,274	1,125,105	1052
2	0.5	0.5	2	1,124,274	1,124,638	596
2.5	0.5	0.5	2	1,124,274	1,124,353	88.6
1	0.5	0.5	2.5	1,124,294	1,124,876	793
1.5	0.5	0.5	2.5	1,124,274	1,124,360	157
2	0.5	0.5	2.5	1,124,294	1,124,475	561
2.5	0.5	0.5	2.5	1,124,294	1,124,403	70

6. Conclusion

This paper proposed a new heuristic method to solve the UC problem by using a new variant of the PSO algorithm known as TVACPSO. Varying the acceleration coefficients in the TVACPSO algorithm improved its performance. In the heuristic method proposed, integer and binary coding were used, which caused a better solution than with other popular methods. Integer coding satisfied the minimum up/down constraints without using a custom penalty factor. Binary coding and a penalty coefficient also satisfied the spinning reserve constraint well. The proposed method was applied to the UC problem with several test power systems consisting of up to 100 units and special 26 system units having the ramp rate constraint. The simulation results clearly showed better performance of the proposed method as compared to the others.

Nomenclature

1) Indices

i units counter
t hours counter
k solutions counter
j dimension of solutions counter
c cycles counter

2) Variables

P_i^t power generation of unit i
FC_i fuel cost of unit i
SU_i start up cost of unit i
SD_i shut down cost of unit i
u_i^t state of unit i at hour t
T_i^{ON} durations in which unit i remains on
T_i^{OFF} durations in which unit i remains off
X_k position of solution k
V_k velocity of solution k
$Pbest_k$ best position of particle k
$Gbest$ global best position
ω particle inertia coefficient
$iter$ current iteration
c_1, c_2 accelerate coefficients
c_{1i}, c_{1f} initial and final values of accelerate coefficient c_1
c_{2i}, c_{2f} initial and final values of accelerate coefficient c_2
T_i^c duration of operating cycle c for unit i
α_i priority of unit i for committing
PX amount of penalizing
$fitness$ fitness function value

3) Parameters

n number of units
T duration of planning
a_i, b_i, c_i fuel cost coefficients
MDT_i minimum down time of unit i
MUT_i minimum up time of unit i
$H_{\text{start up}}$ hot start up cost
$C_{\text{start up}}$ cold start up cost
T_{cold} cold start hour of unit i
P_d^t total system demand at hour t
$P_{i\,\min}$ minimum output power of unit i
$P_{i\,\max}$ maximum output power of unit i
SR^t spinning reserve at hour t
RU_i ramp up rate of unit i
RD_i ramp down rate of unit i

References

[1] H.Y. Yamin, "Review on methods of generation scheduling in electric power systems," Electric Power System Research, Vol. 69, pp. 227–248, 2004.

[2] H.W. Kuhn, A.W. Tucker, "Nonlinear programming," in Proceedings of the Second Berkeley Symposium on Mathematical Programming Statistics and Probability, Berkeley, CA, USA, University of California Press, 1951.

[3] C.K. Pang, G.B. Sheble, F. Albuyeh, "Evaluation of dynamic programming based methods and multiple area representation for thermal unit commitments," IEEE Transactions On Power Systems, Vol. 100, pp. 1212–1218, 1981.

[4] A.M. Geoffrion, "Lagrangian relaxation for integer programming problems," Mathematical Programming Study, Vol. 2, pp. 82–114, 1974.

[5] S.A. Kazarlis, A.G. Bakirtzis, V. Petridis, "A genetic algorithm solution to the unit commitment problem," IEEE Transactions on Power Systems, Vol. 11, pp. 83–92, 1996.

[6] I.G. Damousis, A.G. Bakirtzis, P.S. Dokopoulos, "A solution to the unit commitment problem using integer-coded genetic algorithm," IEEE Transactions on Power Systems, Vol. 19, pp. 1165–1172, 2004.

[7] J. Ebrahimi, S.H. Hosseinian, G.B. Gharehpetian, "Unit commitment problem solution using shuffled frog leaping algorithm," IEEE Transactions on Power Systems, Vol. 26, pp. 573–581, 2011.

[8] X. Yuan, H. Nie, A. Su, L. Wanga, Y. Yuan, "An improved binary particle swarm optimization for unit commitment problem," Expert Systems with Applications, Vol. 36, pp. 8049–8055, 2009.

[9] X. Yuan, H. Nie, A. Su, L. Wanga, Y. Yuan, "Application of enhanced discrete differential evolution approach to unit commitment problem," Energy Conversion and Management, Vol. 50, pp. 2449–2456, 2009.

[10] Y.W. Jeong, J.B. Park, S.H. Jang, K.Y. Lee, "A new quantum-inspired binary PSO: application to unit commitment problems for power systems," IEEE Transactions on Power Systems, Vol. 25, pp. 1486–1495, 2010.

[11] K.A. Juste, H. Kita, E. Tanaka, J. Hasegawa, "An evolutionary programming solution to the unit commitment problem," IEEE Transactions on Power Systems, Vol. 14, pp. 1452–1459, 1999.

[12] N.D. Simopoulos, D.S. Kavatza, D. Vournas, "Unit commitment by an enhanced simulated annealing algorithm," IEEE Transactions on Power Systems, Vol. 21, pp. 68–76, 2006.

[13] M. Eslamian, S.H. Hosseinian, B. Vahidi, "Bacterial foraging based solution to the unit-commitment problem," IEEE Transactions on Power Systems, Vol. 24, pp. 1478–1488, 2009.

[14] M.M. Hadji, B. Vahidi, "A solution to the unit commitment problem using imperialistic competition algorithm," IEEE Transactions on Power Systems, Vol. 27, pp. 117–124, 2012.

[15] B. Pavez-Lazo, J. Soto-Cartes, "A deterministic annular crossover genetic algorithm optimisation for the unit commitment problem," Expert Systems with Applications, Vol. 38, pp. 6523–6529, 2011.

[16] S. Najafi, Y. Pourjamal, "A new heuristic algorithm for unit commitment problem," Energy Procedia, Vol. 14, pp. 2005–2011, 2012.

[17] C.C. Su, Y.Y. Hsu, "Fuzzy dynamic programming: an application to unit commitment," IEEE Transactions on Power Systems, Vol. 6, pp. 1231–1237, 1991.

[18] K.A. Juste, H. Kita, E. Tanaka, J. Hasegawa, "An evolutionary programming solution to the unit commitment problem," IEEE Transactions on Power Systems, Vol. 14, pp. 1452–1459, 1999.

[19] C. Wang, S.M. Shahidehpour, "Effects of ramp rate limits on unit commitment and economic dispatch," IEEE Transactions on Power Systems, Vol. 8, pp. 1341–1350, 1993.

[20] J. Kennedy, R.C. Eberhart, "Particle swarm optimization," in Proceedings of IEEE International Conference on Neural Networks (ICNN'95), Perth, Australia, Vol. 4, pp. 1942–1948, 1995.

[21] K.T. Chaturvedi, M. Pandit, L. Srivastava, "Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch," International Journal of Electrical Power Energy Systems, Vol. 31, pp. 249–257, 2009.

[22] A.J. Wood, B.F. Wollenberg, Power Generation Operation and Control, New York, Wiley, 1984.