I.J. Modern Education and Computer Science, **2014, 7, 55-61**
Published Online July 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2014.07.08

Scheduling of Generating unit commitment by Quantum-Inspired Evolutionary Algorithm

Ebrahim Zare juybari
Mazandaran university of science and technology (USTMB), Babol, Mazandaran,Iran
E-mail: ebrahim.zare.juybari@gmail.com

Seyed Mehdi Hosseini
Department of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran
E-mail: mehdi.hosseini@nit.ac.ir

Abstract—An Quantum-Inspired Evolutionary Algorithm (QEA) is presented for solving the unit commitment problem. The proposed method has been used to achieve the schedule of system units by considering optimal economic dispatch. The QEA method based on the quantum concepts such as Q-bit, present a better population diversity compared with previous evolutionary approaches, and uses quantum gates to achieve better solutions. The proposed method has been tested on a system with 10 generating units, and the results shows the effectiveness of algorithm compared with Other previous references. Furthermore, it can be used to solve the large-scale generating unit commitment problem.

Index Terms—Evolutionary Algorithm, Quantum computing, Unit commitment.

I. INTRODUCTION

Generating unit commitment is one of the important problems, which plays an important role in optimal and economic performance of power systems during their operation. In this paper, a UC problem with the objective of minimum operation cost and constraint is solved. Among the method which has been previously used to solve the generating UC problem, priority list [1], dynamic programming [2], Lagrangian relaxation [3], mixed-integer programming [4] and the branch-and-bound algorithm [5] can be mentioned.In Recent years, evolutionary algorithms (EAs) like genetic algorithm (GA) [7], simulated annealing (SA) [8], evolutionary programming (EP) [9], particle swarm optimization (PSO) [10] and hybrid methods [11], have been employed successfully to solve the UC problems. These methods are based on the stochastic optimization techniques, and they operate with various search mechanisms on a group of selected solutions.

Quantum-Inspired Evolutionary computation is a branch of Evolutionary computation, which special principles of mechanic quantum like interference, superposition and uncertainty are, employed [12]. Quantum-Inspired Evolutionary algorithm will be reach to a better balance between the detection and exploitation of the solution space, by means of concepts and Fundamentals of quantum computing such as quantum bits (Q-bits), quantum gates (Q-gates) and conformity of these states. In addition, compared with the conventional Evolutionary algorithms in a small population, it can be reach to better solutions. In [13], conformity performance of QEA for combinatorial optimization problems has been showed. In this paper, QEA has been used to solve the UC problem. For this purpose, firstly the unit-scheduling problem is solved by QEA-based UC method (QEA-UC) and then economic dispatch problem is resolved. The presented algorithm has been compared with the proposed algorithm in [14]. In section 2 the formulation of the UC problem is presented, and in section 3 fundamentals and procedure of QEA is described. In section 4, we introduce the proposed QEA-UC method and in section 5 we will have the simulation results and finally in section 6 the conclusions are presented.

II. PROBLEM FORMULATION

The purpose of solving the UC problem in this paper is minimizing the production cost, which include the fuel cost and the start-up cost during a certain period of time (24 Hours). The formulation of this optimization problem is presented in equations (1) to (8):

Objective function:

$$\min .F_H = \sum_{k=1}^{H}\sum_{k=1}^{N}[F_{kh}(p_{kh}) + ST_{kh}(1-u_{k(h-1)})]u_{kh} \tag{1}$$

Fuel cost function:

$$F_{kh}(p_{kh}) = c_k(p_{kh})^2 + b_k(p_{kh}) + a_k \tag{2}$$

Start-up cost function:

$$ST_{kh} = \begin{cases} HSC_k,\ if\ MDT_k \le T_k^{off} \le MDT_k + CSH_k \\ CSC_k,\ if\ T_k^{off} > MDT_k + CSH_k \end{cases} \tag{3}$$

The constraints of UC problem which considered in this paper are as follows:

1) Load Request supply constraint

$$\sum_{k=1}^{N} p_{kh} u_{kh} = D_h \tag{4}$$

2) Spinning reserve constraint

$$\sum_{k=1}^{N} p_k(\max) u_{kh} \geq D_h + R_h \tag{5}$$

3) Unit production constraint

$$u_{nh}P_{n(max)} \geq P_n \geq u_{nh}P_{n(min)} \tag{6}$$

4) Minimum up time limit

$$T_n^{on} \geq MUT_n \tag{7}$$

5) Minimum down time limit

$$T_n^{off} \geq MDT_n \tag{8}$$

The notation list used in this paper is as follows:

N: Number of generating units;

H: The total hours of scheduling period;

n: Index of unit;

h: Index of time;

P_{nh} : Control variable for the generation of unit n at hour h;

U_{nh} : Control variable for the on/off status of unit n at hour h;

F_H : Total system production cost within h hours;

$F_{nh}(P_{nh})$: Fuel cost function of unit n at hour h;

a_n, b_n, c_n : Cost function parameters of unit n;

ST_{nh} : Start-up cost of unit n at hour h;

HSC_n/CSC_n: Hot/cold start-up cost of the nth unit;

MDT_n/MUT_n : Minimum down/up time of the nth unit;

CSH_n: Cold start hours of unit n;

T_n^{off} : Duration during which n unit is continuously OFF;

T_n^{on}: Duration during which unit is continuously ON;

D_h : System peak demand at hour h;

R_h : Spinning reserve at hour h;

$P_{n(max)}$: Maximum/minimum output limit of unit;

III. Quantum-Inspired Evolutionary Algorithm

The same as other evolutionary algorithms, the QEA method consist of computing functions and also specific population dynamics. In quantum computations, a Q-bit is smallest information unit that stored in two states, instead of binary numbers in the previous evolutionary methods. A string of Q-bits form an individual Q-bit that can include all the possible states in the search space. QEA only needs a small initial population to produce a variant and proper population to explore the solution. A Q-gate is a QEA variable parameter that becomes 0 or 1 for every Q-bit and gradually tends to the best solution by variation decrease, during the optimization process. So by definition of Q-bit, a unique Q-bit is represented and forms a search state. Moreover, for every individual Q-bit, Q-gate only shows one operation state (on or off). So the QEA mechanism creates a balance between exploration and extraction of solution space. In next section, the concept of Q-bit and steps of QEA method is described.

3.1 Presentation

A Q-bit is the smallest unit of information. It can be represented as:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{9}$$

Where α and β are two numbers that satisfy the equation (|α|2 +|β|2 =1) (The two numbers that is located on the first quarter of coordinate axis and sum of their square is equal to 1), where |α|2 and |β|2 are the probability of happenning “0” and “1” respectively.

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{10}$$

A unique Q-bit that consists of m Q-bit is showed as follows:

$$\begin{bmatrix} \alpha 1 & \alpha 2 & \alpha 3 & \ldots & \alpha m \\ \beta 1 & \beta 2 & \beta 3 & \ldots & \beta m \end{bmatrix} \tag{11}$$

The Q-bits exist in every unique Q-bits, show the possible states of nth unit during hth hour. So for m Q-bit, we have 2m states for all the h periods. The unique Q-bit for all the α and β is equal to the same value of $\sqrt{0.5}$ which all the possible states represent by the same probability as follows:

$$|\boldsymbol{\varphi}>= \sum_{\boldsymbol{k=1}}^{\boldsymbol{2m}} \frac{\boldsymbol{1}}{\sqrt{\boldsymbol{2m}}} |\boldsymbol{X_k} \tag{12}$$

Where Xk is the kth state of the binary string solution (X1, X2,…, Xm) and every Xi is either 0 or 1.

For example for a unique Q-bit with two Q-bit:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{13}$$

Where the Q-bit states are unique as follows:

$$\frac{1}{2}|00> + \frac{1}{2}|01> + \frac{1}{2}\Big|10> + \frac{1}{2}\Big|\, 11> \tag{14}$$

In which the probability of all states are ¼. So for a unique Q-bit with 2 Q-bits (particle), we have 4 states for all the h periods.

3.2 Qea Methods

QEA method uses Q-bit for population variation. According to QEA process, the binary solutions of X, obtain from the unique Q-bit and at the end of any iteration, the particles (Q-bit) will update for next step. The details of the QEA mechanism for solving the objective function f(x), and control the binary variation X are as follows:

Step 1: t is the generator counter, set t=0.

Step 2: Initialize (t) Q: (t) Q is a group of unique Q-bits that is initialized at t=0. So we will have: Q (t) = [Q1t, Q1t, … , Qnt]

Where subscript *n* is the total number of unique Q-bit and q_j^t is the *j*th particle (Q-bit) for *t*th generator.

$$q_j^t = \begin{bmatrix} \alpha_{j1}^t & \alpha_{j2}^t & \ldots & \alpha_{jm}^t \\ \beta_{j1}^t & \beta_{j2}^t & \ldots & \beta_{jm}^t \end{bmatrix} \tag{15}$$

Where j=1,2,…,n is the number of every particle and *m* is the particle string length. If all the α_{ij} and β_{ij}^t is initialized with the same value of $\sqrt{0.5}$, the probability of happening “0” or “1” state is equal for every Q-bit.

Step 3: X(t) is selected by observing Q(t):

X(t) =[X_1^t , X_2^t ,…, X_n^t] is a group of binary solutions which are determined by Q(t) observation. X_j is the binary solution that is obtained by q_j^t observation: X_{j1}^t =[x_{j1}^t,…, x_{j2}^t , x_{jn}^t] where X_{ij}^t is binary solution for particle number j, in the first iteration and for generator number *t*. It is determined by comparing $|\beta_{ij}^t|^2$ to a random number between 0 and 1, in which if the number was smaller than $|\beta_{ij}^t|^2$, X_{ij} will be equal to “1” and if not, will be equal to “0”:

$$A = \begin{cases} 1, \, random[0,1] < |\beta|^2 \\ 0, \qquad\qquad otherwise \end{cases}$$

Step 4: Evaluate X(t):

The cost function values which are the same as objective function is calculated by considering all the X(t)s, and the best X(t) with the least cost will be obtained.

Step 5: Storing the best solution of X(t) into B(t):

B(t) is a matrix including the best solutions among all of the particles.

Step 6: t = t+1.

Step 7: Determining X(t) by observing Q(t-1).

Step 8: Evaluating X(t).

Step 9: Updating Q(t) by Q-gate:

A Q-gate is a variable parameter in QEA that update Q-bits, and in updated Q-bit (αijt and βijt) the equation |α ijt| 2 + |βijt |2 = 1 must be satisfied. In QEA, the rotation gates are considered. The rotation gate is a conversion matrix that performs update operation:

$$U(\Delta\theta_{ji}^t) = \begin{bmatrix} \cos(\Delta\theta_{ji}^t) & -\sin(\Delta\theta_{ji}^t) \\ \sin(\Delta\theta_{ji}^t) & \cos(\Delta\theta_{ji}^t) \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} \alpha_{ji}^t \\ \beta_{ji}^t \end{bmatrix} = U(\Delta\theta_{ji}^t) \begin{bmatrix} \alpha_{ji}^{t-1} \\ \beta_{ji}^{t-1} \end{bmatrix} \tag{17}$$

Where$\Delta\theta_{ji}^t$ is the rotation angle that its magnitude can be obtained according to table 1 (reference [14]).

Table 1: The Rotation Angle

x_{ji}^t	b_i^t	$Quadrant$	$f(X_j^t) \leq f(B^t)$	$\Delta\theta_{ji}^t$
0	1	I/III	$false$	+θ
0	1	II/IV	$false$	−θ
1	0	I/III	$false$	−θ
1	0	II/IV	$false$	+θ
0	1	×	$true$	0
1	0	×	$true$	0
0	0	×	×	0
1	1	×	×	0

X is the final solution of every period, *b* is the best solution of every period, f(x) and f(b) are the objective function for X and *b* values. As it can be seen from table 1, when the values of f(x) and f (b) are equal, the rotation angle is zero and the conversion matrix will become a unit Matrix, which will be unchanged when it is multiplied by Q-bit. For the case that X and b are different and also f(x) is bigger than f (b), the rotation angle according to reference [14] will be equal to $\theta = 0.02\pi$ but If f (x) is less than f (b), this means the response obtained in this iteration was better than the previous Solution of and this Result as the best answer can be used in the next iteration.

Reference [15] introduces the D parameter for disclosing variations in the state of a solution matrix element by logical operator (XOR), which has been used to compare the difference between U0 and U_{best}. If they were the same, the D value will be zero, otherwise it will be 1. The conclusion of this comparison was showed in Fig.1.

$$D = ones\, U_g^0 \oplus U_{best} \mid \tag{18}$$

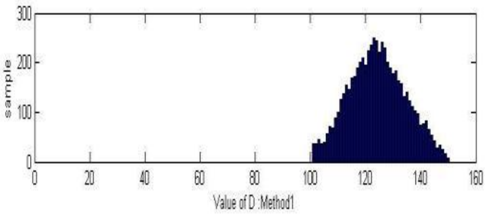

Fig 1. Comparison of variation disclosing in 2 states

Fig.1 is related to a system with 10 units and 5000 iterations for every element. As it can be seen, there is not any difference between X and b for around 100 elements (U0 and Ubest are the same), and for other elements it is changed less than 300 times in 5000 iterations (U0 and Ubest are not the same and D parameter equals 1). It means that their values were the same for more than 4700 times. On the other hands, using the updating step for large systems will slow the schedule, and due to the absence of phase variation for more than 98% of cases, it can be ignored for large systems.

Step 10: Storing the best solution into B(t):The best solution of X(t) and B(t-1) will be stored to B(t).

Step 11: Check whether the stopping conditions are provided:

If the stopping conditions are provided, it is end of algorithm; if not go to Step 6.

IV. UC Problem Solving By Proposed Method

In this section the process of using the QEA for solving the UC problem is described:

A) Using QEA for solving the UC problem:

Step 1: A unique Q-bit for UC problem: A population of unique Q-bit such as:

$$Q(t) = \left[\, q_1^t,\, q_2^t\,,\, \dots ,q_n^t \right] \tag{19}$$

are initialized, where q_j^t is the *j*th particle (unique Q-bit) for *t*th generator and j=1,2,…,n, where *n* is the number of population (number of the whole particles). For choosing QEA in UC problem, any unique Q-bit is like a H*N matrix, so that N is the total number of units and H is the total number of schedule time period (hours) in a certain scheduling horizon so h=1,2,…,H and K=1,2,…,N. So the unique Q-bit, q_j^t is as follows:

$$q_j^t = \begin{bmatrix} \alpha_{j11}^t & \alpha_{j12}^t & \dots & \alpha_{j1H}^t \\ \beta_{j11}^t & \beta_{j12}^t & & \beta_{j1H}^t \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \alpha_{jN1}^t & \alpha_{jN2}^t & \dots & \alpha_{jNH}^t \\ \beta_{jN1}^t & \beta_{jN2}^t & & \beta_{jNH}^t \end{bmatrix} \tag{20}$$

Step 2: The binary solution for unit schedules: U(t) is a group of unit schedules:

$$U(t) = \left[U_1^t,\, U_2^t\,,\, \dots ,\, U_n^t\right] \tag{21}$$

Where any U_j^t schedule is a N*H matrix. With observing q_j^t, a binary solution or a unit schedule will be formed:

$$U_j^t = \begin{bmatrix} u_{j11}^t & u_{j12}^t & \dots & u_{j1H}^t \\ u_{j21}^t & u_{j22}^t & \dots & u_{j2H}^t \\ \dots & \dots & \dots & \dots \\ u_{jN1}^t & u_{jN2}^t & \dots & u_{jNH}^t \end{bmatrix} \tag{22}$$

Step 3: Unit output variables: For any unit schedule in second step, an economic dispatch method is used to determine the optimum output power and we will have $P(t) = [\, P_1^t, P_2^t, \dots, P_n^t]$Where P (t) is the generator t output power in the unit schedules. The variable Pjt is the output power for jth schedule:

$$P_j^t = \begin{bmatrix} p_{j11}^t & p_{j12}^t & \dots & p_{j1H}^t \\ p_{j21}^t & p_{j22}^t & \dots & p_{j2H}^t \\ \dots & \dots & \dots & \dots \\ p_{jN1}^t & p_{jN2}^t & \dots & p_{jNH}^t \end{bmatrix} \tag{23}$$

Where P_{jkh} is the output power of unit number *k* at the *h*th hour and in the *j*th iteration.

Step 4: Function evaluation: Because the goal of a UC problem is minimizing the total operation cost, the cost function for any unit schedule is calculated and the P_{kh} value will be obtained.

4.1 The Steps of QEA-UC Method

Fig.2 shows the flowchart of QEA, where the process of flowchart is as follows:

Step 1: t is the generator counter, Set the generation counter (*t*=0).

Step 2: All of the α and β of a same group of unique Q-bit, will initialize with the same value of $\sqrt{0.5}$.

Step 3: Determining the unit schedules with observing unique Q-bit state.

Step 4: Determining the schedule cost with optimum economic dispatch.
Step 5: If t=0 go to step 7.
Step 6: Updating unique Q-bit by Q-gate.
Step 7: Comparing the schedule costs and storing the best schedule solution.
Step 8: t=t+1
Step 9: If t was bigger than the maximum number of generators, the algorithm will come to an end, otherwise go to step 3.

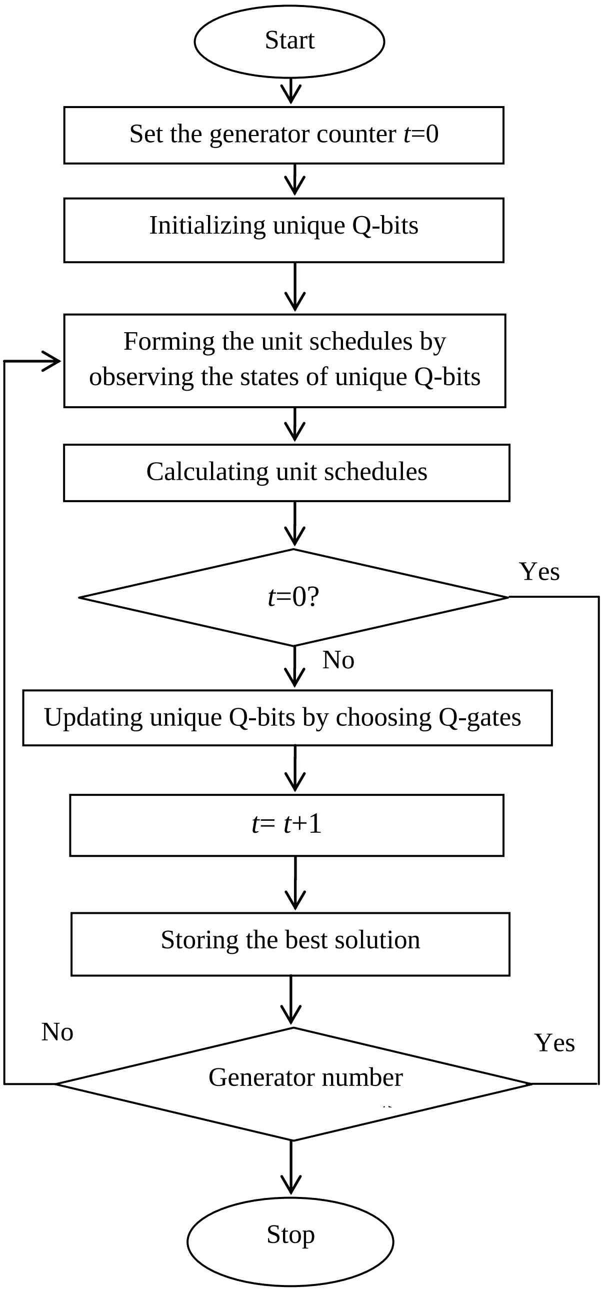

Fig 2. The flowchart of QEA method

V. Case Study And Simulation Results

For testing the effectiveness of QEA method to solve the UC problem, a system consist of 10 thermal unit has been considered, then the problem has been solved in a 24 hour period and the states of units has been determined. Units informations and load request value in the schedule period and the informations of production units has been presented in reference [16]. Unit power production has presented in table2. Convergence graphs for four different values of the particle Shown in Figure3. This results indicate that higher values of this parameter, the rate and extent of exploration of the search algorithm more Finally, after several trials, the best response to population size, the number is 18. The chart shows the number of small particles results in nearly optimal solution can be reached.

In this study, the operating costs are calculated by QEA-UC method. In Table3 of this method is compared with other methods. Since the purpose of UC, the units with the lowest cost of operation is determined at the Table3. optimize the performance of these methods can be seen.

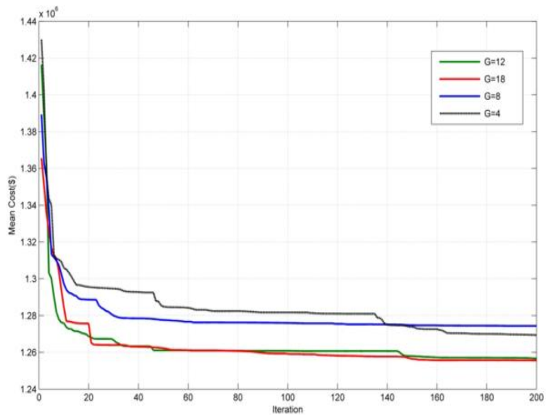

Fig 3. Sensitivity analysis algorithm to the population size

Table 2: The power production of units in 24 hours

HOUR	1	2	3	4	5	6	7	8	9	10	11	12
Unit 1	0	0	0	0	100	113.43	0	0	0	0	202.34	0
Unit 2	116.94	96.16	0	0	0	114.38	119.01	135.85	173.98	190.44	202.89	212.83
Unit 3	200.03	200	200.1	0	200	201.07	200.00	0	0	200	0	211.34
Unit 4	114.21	0	0	99	99	113.76	118.69	0	172.30	189.66	201.71	211.97
Unit 5	190.05	190	190.1	190	190	190.07	190.01	190	190.11	190	199.83	210.66
Unit 6	116.33	0	99.84	85	0	115.10	117.08	137.09	175.44	191.66	201.31	210.32
Unit 7	200.03	200	200.1	200	200	200.06	200.00	200	200.10	0	201.01	210.75
Unit 8	115.89	99	101.32	99	99	111.95	117.02	134.42	0	190.0	201.61	0
Unit 9	130.03	130	130.1	130	0	130.06	130.00	134.31	173.87	189.61	198.62	213.26
Unit 10	200.04	200	200.1	200	200	200.07	200.00	200	200.10	200	200.46	211.72

HOUR	13	14	15	16	17	18	19	20	21	22	23	24
Unit 1	0	0	0	171.99	157.93	184.03	0	0	205.29	196.79	0	0
Unit 2	0	198.01	0	0	0	0	204.92	0	0	0	180.37	0
Unit 3	215.83	200.01	200.50	200	200	200	202.33	214.75	204.29	200	200	200.19
Unit 4	217.73	197.14	193.79	174.14	156.05	183.17	203.15	215.84	204.79	196.88	180.08	153.10
Unit 5	216.31	196.93	195.01	190	190	190	202.10	214.26	207.47	195.40	0	190.22
Unit 6	0	200.28	193.11	172.03	155.87	182.36	203.75	213.30	204.38	0	179.62	150.14
Unit 7	215.68	200	200	200	200	200	202.68	214.20	204.83	200	0	200.19
Unit 8	216.26	201.09	185.85	173.54	156.48	0	204.43	214.37	203.92	198.62	179.83	149.13
Unit 9	0	197.85	193.97	173.80	156.82	183.79	200.17	214.10	202.95	198.16	179.74	148.74
Unit 10	214.24	200	202	200	200	200	202.62	213.63	207.99	200	200.02	200.19

Table 3: Operating costs for unit commitment problem

method	The minimum operation cost for 10 thermal units ($)
FA	570,215
EP	572,517
SA	566,208
ELR	568,131
GA	565,825
QEA	563,956

VI. CONCLUSION

A Quantum-Inspired Evolutionary Algorithm (QEA) has presented in this method and based on it, a new method (QEA-UC) has proposed to solve UC problems. The effectiveness and operation of this method has been showed in a 10 unit system. The results showed that the QEA-UC is so powerful and effective and unlike the previous evolutionary algorithms, has shown good performances even for small populations (particles). A linear relation exists between the dimensions of problem and the calculation period. This method has the ability to solve large scale UC problems in a certain time period, and from this view, it is better than the other methods.

REFERENCES

[1] R. M. Burns and C. A. Gibson, “Optimization of priority lists for a unit commitment program,” in Proc. IEEE/Power Engineering SocietySummer Meeting, Paper A, 1975, vol. 75, pp. 453–1.

[2] 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.

[3] H. H. Balci and J. F. Valenzuela, “Scheduling electric power generators using particle swarm optimization combined with the Lagrangian relaxation method,” Int. J. Appl. Math. Comput.Sci., vol. 14, no. 3, pp. 411–421, 2004.

[4] J. A. Muckstadt and R. C. Wilson, “An application of mixed-integer programming duality to scheduling thermal generating systems,” IEEE Trans. Power App. Syst., vol. PAS-87, pp. 1968–1978, 1968.

[5] 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.

[6] K. P.Wong and K. Doan, “An artificial intelligence algorithm for daily scheduling of thermal generators,” Proc. Inst. Elect. Eng. Part C, vol.138, no. 6, pp. 518–534, 1991.

[7] P. C. Yang, H. T. Yang, and C.-L. Huang, “Solving the unit commitment problem with a genetic algorithm through a constraint satisfactiontechnique,” Elect. Power Syst. Res., vol. 37, pp. 55–65, 1996.

[8] A. Viana, J. P. Sousa, and M. Matos, “Simulated annealing for the unitcommitment problem,” presented at the The IEEE Porto Power Tech.Conf., Porto, Portugal, Sep. 10–13, 2001, unpublished.

[9] K. A. Juste, H. Kita, E. Tanaka, and J. Hasegawa, “An evolutionary programming solution to the unit commitment problem,” IEEE Trans.Power Syst., vol. 14, no. 4, pp. 1452–1459, Nov. 1999.

[10] 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 Trans. Power Syst., vol. 21, no. 1, pp. 411–418, Feb. 2006.

[11] G. K. Purushothama and L. Jenkins, “Simulated annealing with local search—A hybrid algorithm for unit commitment,” IEEE Trans. PowerSyst., vol. 18, no. 1, pp. 273–278, Feb. 2003.

[12] K. H. Han and J. H. Kim, “Quantum-inspired evolutionary algorithm for a class of combinatorial optimization,” IEEE Trans. Evol. Comput, vol. 6, pp. 580–593, Dec. 2002.

[13] K. H. Han and J. H. Kim, “On setting the parameters of quantum-inspired evolutionary algorithm for practical applications,” in Proc. 2003 Congr. Evolutionary Computation, Canberra, Australia, Dec. 2003, pp.178–18.

[14] T. W. Lau, C. Y. Chung, K. P. Wong, T. S. Chung, and S. L. Ho,“Quantum-inspired evolutionary algorithm approach for unit commitment,”IEEE Trans. Power Syst., vol. 24, no. 3, pp. 1503–1512, Aug.2009.

[15] T. W. Lau, C. Y. Chung, K. P. Wong, T. S. Chung, and S. L. Ho,“ An Advanced Quantum-inspired evolutionary algorithm approach for unit commitment,”IEEE TRANSACTIONS ON POWER SYSTEMS, VOL. 26, NO. 2, MAY 2011.

[16] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, “A genetic algorithmsolution to the unit commitment problem,” IEEE Trans. Power Syst.,vol. 11, no. 1, pp. 83–92, Feb. 1996.

Ebrahim Zare juybari received B.Sc degree from University of Mazandaran, Iran in 2011, he received M.Sc degree from University of Science and Technology, iran in 2013. Currently, he is Research on calculation of transient flow in high-voltage lines.

Seyed Mehdi Hosseini received B.Sc degree from University of Mazandaran, Iran in 2002, he received M.Sc and Ph.D degree from Iran University of Science and Technology in 2002, 2009 respectively.Currently, he is conducting research work on Reliability of Distribution Systems, Distributed Generation, FACTS Devices.