A New Quantum-Inspired Binary PSO: Application to Unit Commitment Problems for Power Systems

Yun-Won Jeong, Jong-Bae Park, *Member, IEEE*, Se-Hwan Jang, and Kwang Y. Lee, *Fellow, IEEE*

***Abstract*—This paper proposes a new binary particle swarm optimization (BPSO) approach inspired by quantum computing, namely quantum-inspired BPSO (QBPSO). Although BPSO-based approaches have been successfully applied to the combinatorial optimization problems in various fields, the BPSO algorithm has some drawbacks such as premature convergence when handling heavily constrained problems. The proposed QBPSO combines the conventional BPSO with the concept and principles of quantum computing such as a quantum bit and superposition of states. The QBPSO adopts a Q-bit individual for the probabilistic representation, which replaces the velocity update procedure in the particle swarm optimization. To improve the search capability of the quantum computing, this paper also proposes a new rotation gate, that is, a coordinate rotation gate for updating Q-bit individuals combined with a dynamic rotation angle for determining the magnitude of rotation angle. The proposed QBPSO is applied to unit commitment (UC) problems for power systems which are composed of up to 100-units with 24-h demand horizon.**

***Index Terms*—Binary particle swarm optimization, combinatorial optimization, constraint treatment technique, quantum computing, quantum evolutionary algorithm, unit commitment.**

I. Introduction

PARTICLE swarm optimization (PSO) suggested by Kennedy and Eberhart in 1995 is based on the analogy of swarm of bird and school of fish [1]. In the PSO, particles are drawn stochastically toward new positions based on the present velocity of each particle, its own previous best performance, and the best previous performance of their neighbors [1], [2]. In 1997, Kennedy and Eberhart showed that the binary particle swarm optimization (BPSO) was able to successfully optimize the De Jong's suite of five test functions [3].

Quantum computing is a new paradigm which has been proposed as a consequence of applying quantum mechanics to computer science [4]–[7]. Research on merging evolutionary computation and quantum computing has being carried out since the late 1990s and can be classified into two fields: 1) new quantum algorithms using automatic programming techniques such as genetic programming [4], 2) quantum-inspired evolutionary computing for a digital computer as a branch of study on evolutionary computation that is characterized by certain principles of quantum mechanics such as uncertainty, superposition, and interference, etc. [5]–[7]. Quantum inspired computing was first introduced in [5]. Narayanan and Moore [6] proposed quantum-inspired genetic algorithms, where concepts and principles of quantum mechanics are used to inform and inspire more efficient evolutionary computing methods. Han and Kim [7] proposed a quantum-inspired evolutionary algorithm (QEA) and Vlachogiannis and Lee applied it to a real and reactive dispatch problem in power systems [8].

Unit commitment (UC) problem has a significant influence on secure and economic operation of power systems. Optimal commitment scheduling can save huge amount of costs to electric utilities and improve reliability by keeping proper spinning reserves. The UC problems involve scheduling the on/off states of generating units, which minimizes the operating cost, start-up cost and shut-down cost for a given horizon under various operating constraints [9]. In the UC problem, the decisions are the selection of the time for each unit to be on- and/or offline (binary variables) as well as the unit's economic generation level (continuous variables). Thus, the UC problem can be formulated as a nonlinear mixed integer combinatorial optimization problem [10]–[26]. The number of combinations of 0-1 binary variables grows exponentially for a large-scale UC problem, which makes it difficult to solve in practice. Over the past decades, many salient methods have been developed for solving the UC problems. The exact solution to the problem can be obtained by complete enumeration, which cannot be applied to realistic power systems due to its computational burdens [8]. The solution methods for UC problems can be divided into two classes: One is the numerical optimization techniques such as priority list methods [10], [11], dynamic programming [12], [13], Lagrangian relaxation methods [14], [15], branch-and-bound methods [16], and mixed-integer programming [17]. The other is the stochastic search methods such as genetic algorithms [18], [19], evolutionary programming [20], [21], simulated annealing [22], [23], and particle swarm optimization [24]. Recently, quantum-inspired evolutionary algorithms have also been introduced as solution methods for UC problems [25], [26].

Among many modern heuristic optimization algorithms [27], however, the PSO algorithm has attracted much attention because of its simplicity and powerful performances [28]–[32]. Recently, quantum-behaved PSO approaches, inspired by the fundamental theory of particle swarm and features of quantum

Manuscript received September 21, 2009; revised December 24, 2009. First published March 11, 2010; current version published July 21, 2010. Paper no. TPWRS-00750-2009.

Y.-W. Jeong, J.-B. Park, and S.-H. Jang are with the Department of Electrical Engineering, Konkuk University, Seoul 143-701, Korea (e-mail: ywjeong@konkuk.ac.kr; jbaepark@konkuk.ac.kr; shjang@konkuk.ac.kr).

K. Y. Lee is with the Department of Electrical and Computer Engineering, Baylor University, Waco, TX 76798 USA (e-mail: Kwang_Y_Lee@baylor.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2010.2042472

mechanics, have been developed [33]–[35]. This paper proposes a quantum-inspired BPSO (QBPSO) which is based on the concept and principles of quantum computing such as a quantum bit and superposition of states to enhance the performance of the conventional BPSO. The proposed QBPSO introduces a Q-bit individual for the probabilistic representation of a particle, thereby replacing the velocity update procedure in the traditional PSO. In QBPSO, therefore, an inertia weight factor and two acceleration coefficients can be removed and only one factor, rotation angle, is needed when modifying the position of particles. To improve the conventional rotation gate for the *Q-bit* individual update, this paper proposes a new rotation gate in which two effective techniques are included: 1) a coordinate rotation gate for updating Q-bits, 2) a dynamic rotation angle approach for determining the magnitude of rotation angle. Therefore, the proposed QBPSO can obtain an efficient balance between exploration and exploitation with a smaller population size and shorter computation time. Furthermore, the rule-based heuristic constraint treatment techniques [25] are adopted to effectively satisfy the minimum up/down time and spinning reserve constraints in the UC problem.

The QBPSO is applied to UC problems for realistic power systems of different sizes, which consist of the 10-, 20-, 40-, 60-, 80-, and 100-units along with 24-h load demands. And their results are compared with those of previous works which used the same benchmarks [18], [20], [23]–[26].

II. Overview of Particle Swarm Optimization

The PSO, suggested by Kennedy and Eberhart [1], is a population-based parallel search algorithm using a group of particles. It is based on the behavior of individuals of a swarm and its roots are in the zoologist's modeling of the movement of individuals within a population. It has been noticed that members of a group seem to share information among them, a fact that leads to increased efficiency of the group [2]. A particle moves toward the optimum based on its present velocity, its previous experience, and the experience of its neighbors. In an n-dimensional search space, the position and velocity of the ith particle are represented as vectors $X_i = \{x_{i1}, \cdots, x_{in}\}$ and $V_i = \{v_{i1}, \cdots v_{in}\}$, where each element has real values. Let $Pbest_i = \{x_{i1}^P, \cdots, x_{in}^P\}$ and $Gbest = \{x_1^G, \cdots, x_n^G\}$ be the best position of the ith particle and the group's best position so far, respectively. The velocity and position of each particle is updated as follows:

$$\begin{aligned} V_i^{k+1} =& \omega \cdot V_i^k + c_1 \cdot rn_1 \cdot \left(Pbest_i^k - X_i^k\right) + c_2 \cdot rn_2 \\ & \cdot \left(Gbest^k - X_i^k\right) \quad (1) \\ X_i^{k+1} =& X_i^k + V_i^{k+1} \quad (2) \end{aligned}$$

where V_i^k is the velocity of the ith particle at iteration k, ω is the inertia weight factor, c_1 and c_2 are the acceleration coefficients, rn_1 and rn_2 are random numbers between 0 and 1, and X_i^k is the position of the ith particle at iteration k. In the velocity updating process, the values of parameters such as ω, c_1, and c_2 should be determined in advance, which makes it cumbersome to solve large-scale optimization problems.

The BPSO, also introduced by Kennedy and Eberhart [3], enables the PSO to operate in binary spaces. The structure of the BPSO is effectively the same as that of the real-valued PSO. In BPSO, however, the position vector of a particle is a binary one. The velocity of the jth element in the ith particle is related to the possibility that the position of the particle takes a value of 1 or 0. It is implemented by defining an intermediate variable $S(v_{ij}^{k+1})$, called a sigmoid limiting transformation, as follows:

$$S\left(v_{ij}^{k+1}\right) = \frac{1}{1 + \exp\left(-v_{ij}^{k+1}\right)}. \quad (3)$$

The value of $S(v_{ij}^{k+1})$ can be interpreted as a probability threshold. If a random number rn_{ij}, selected from a uniform distribution in [0, 1], is less than the value of $S(v_{ij}^{k+1})$ the position of the jth element in the ith particle at iteration $k+1$ (i.e., x_{ij}^{k+1}) is set to 1 and otherwise, set to 0. In the BPSO, therefore, (2) for modifying the position vector is replaced as follows:

$$x_{ij}^{k+1} = \begin{cases} 1, & \text{if } rn_{ij} < S\left(v_{ij}^{k+1}\right) \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

III. Quantum-Inspired BPSO Algorithm

A. *Quantum Computing*

The smallest unit of information stored in a quantum computer is called a *quantum bit* or *Q-bit* [7]. A Q-bit is analogous to a bit of storage in a traditional computer. A Q-bit may be in the "1" state, in the "0" state, or in any superposition of the two, while a bit in traditional computing can only hold a single state, either 0 or 1. To illustrate this, the traditional 0 and 1 values are written as $|0\rangle$ and $|1\rangle$, and the state of a Q-bit is represented as follows:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (5)$$

where α and β are complex numbers that specify the probability amplitudes of the corresponding states. Here, $|\alpha|^2$ and $|\beta|^2$ denote the probability that the Q-bit will be found in "0" state and "1" state, respectively. Normalization of the state to unity guarantees $|\alpha|^2 + |\beta|^2 = 1$. The state of a Q-bit can be changed by the operation with a quantum gate such as NOT gate, rotation gate, and Hadamard gate, etc. [4].

In [7], Han and Kim suggested a novel QEA, inspired by the concept of quantum computing, in which a Q-bit representation is designed to represent a linear superposition of states (i.e., binary solutions). A Q-bit is defined as the smallest unit of information, which is defined with a pair of numbers (α, β) as $\left[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right]$, where $|\alpha|^2 + |\beta|^2 = 1$. A *Q-bit individual* as a string of n Q-bits is defined as

$$q = \left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \beta_1 & \beta_2 & \cdots & \beta_n \end{array}\right] \quad (6)$$

where $|\alpha_j|^2 + |\beta_j|^2 = 1, j = 1, 2, \cdots, n$. The Q-bit representation has the advantage of representing a linear superposition of states. All possible combinations of the decision variables

TABLE I
EXAMPLE OF LOOKUP TABLE FOR DETERMINING ROTATION ANGLE

x_j	b_j	*Fitness* $(X) \geq$ *Fitness* (B)	$\Delta\theta_j$
0	0	False	θ_1
0	0	True	θ_2
0	1	False	θ_3
0	1	True	θ_4
1	0	False	θ_5
1	0	True	θ_6
1	1	False	θ_7
1	1	True	θ_8

can be derived from a single representation, while a system of n bits has $2n$ possible single states in the classical computing. Evolutionary computing with Q-bit representation has a better characteristic of population diversity than other representations, since it can represent linear superposition of states probabilistically.

The following rotation gate is used as a variation operator, by which a Q-bit individual is updated:

$$U(\Delta\theta_j) = \begin{bmatrix} \cos(\Delta\theta_j) & -\sin(\Delta\theta_j) \\ \sin(\Delta\theta_j) & \cos(\Delta\theta_j) \end{bmatrix} \tag{7}$$

where $\Delta\theta_j$ is a rotation angle of the jth Q-bit toward either 0 or 1 state. As described in Table I, the value of $\Delta\theta_j$ can be determined through a pre-defined lookup table [7], where $B = (b_1, \cdots, b_n)$ is the best solution.

In [7], the values of $\Delta\theta_j$ were set as $\theta_1 = 0$, $\theta_2 = 0$, $\theta_3 = 0.01\pi$, $\theta_4 = 0$, $\theta_5 = -0.01\pi$, $\theta_6 = 0$, $\theta_7 = 0$, and $\theta_8 = 0$ through experimental tests on the knapsack problems.

B. BPSO With Quantum Computing

In the proposed QBPSO, the state of each element in a particle takes a value of 0 or 1 by the probability of $|\alpha|^2$ or $|\beta|^2$. In other words, the velocity update process (1) in the traditional BPSO is replaced by the quantum computing. In the proposed QBPSO, therefore, an inertia weight factor (i.e., ω) and two acceleration coefficients (i.e., c_1 and c_2) can be removed, and only a rotation angle is added. The position vector of the ith particle (i.e., $X_i = \{x_{i1}, \cdots, x_{in}\}$) is updated by probability of $|\beta_i|^2$ stored in the ith Q-bit individual (i.e., q_i). The jth element of the ith particle takes a value of 0 or 1 by the following:

$$x_{ij} = \begin{cases} 1, & \text{if } rn_{ij} < |\beta_{ij}|^2 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

for $i = 1, 2, \cdots, NP$, $j = 1, 2, \cdots, n$. Here,rn_{ij} is the uniformly distributed random number between [0, 1] and NP is the population size.

To enhance the rotation gate for updating Q-bit individuals, this paper proposes a new rotation gate as a variation operator where two effective techniques are deployed: One is a *coordinate rotation gate* for updating Q-bits and the other is a *dynamic rotation angle approach* for determining the magnitude of rotation angle. The conventional rotation gate requires a pre-specified lookup table to determine the rotation angle $\Delta\theta$ to obtain new (α, β). However, the proposed coordinate rotation gate determines the rotation angle without the lookup table information since it uses the current position, $Pbest$, and $Gbest$ of a swarm as in the following:

$$\Delta\theta_{ij} = \theta \times \left\{\gamma_{1i} \times \left(x_{ij}^P - x_{ij}\right) + \gamma_{2i} \times \left(x_j^G - x_{ij}\right)\right\} \tag{9}$$

where θ is the magnitude of rotation angle, and γ_{1i} and γ_{2i} can be obtained by comparing the fitness of current position of particle i with those of $Pbest_i$ and $Gbest$, respectively, as follows:

$$\gamma_{1i} = \begin{cases} 0, & \text{if } Fitness(X_i) \geq Fitness(Pbest_i) \\ 1, & \text{otherwise} \end{cases} \tag{10}$$

$$\gamma_{2i} = \begin{cases} 0, & \text{if } Fitness(X_i) \geq Fitness(Gbest) \\ 1, & \text{otherwise.} \end{cases} \tag{11}$$

Therefore, each particle can approach to the optimum solution through its own experience and its neighbor's experiences.

The magnitude of rotation angle (i.e., θ) can give an effect on the quality of solution and the speed of convergence. Therefore, the proper selection of θ may not only lead to a balance between global exploration and local exploitation, but also result in less iteration in finding the optimal solution. In general, the values from 0.001π to 0.05π are recommended for the magnitude of the rotation angle, although they depend on problems [7]. This paper proposes a *dynamic rotation angle approach* for the magnitude of rotation angle to enhance the convergence characteristics. In the proposed approach, the magnitude of the rotation angle decreases monotonously from $\theta_{\max}$ to $\theta_{\min}$ along the iteration as follows:

$$\theta = \theta_{\max} - (\theta_{\max} - \theta_{\min}) \times \frac{k}{iter_{\max}} \tag{12}$$

where $iter_{\max}$ is the maximum iteration number and k is the current iteration number.

The procedure of the proposed QBPSO algorithm can be summarized as the following pseudo-code:

```
Begin
    Initialize Q-bit individual and position of a population.
    Set initial $Pbest$ and $Gbest$.
    Do while
        For $i = 1$ to Population size
            Update Q-bit individual of the $i$th particle.
            Modify position of the $i$th particle.
            Update $Pbest$ of the $i$th particle.
        Next $i$
        Update $Gbest$.
    Until termination criterion is met
End
```

The inevitable problem of the conventional BPSO is how to set the suitable parameters (i.e., a weight factor and two acceleration coefficients). It is requires significant effort for user to set these parameters; often it involves trial and error methods for each new UC problem. However, the proposed QBPSO can search the solution without these parameters. Therefore, the computation efficiency of the QBPSO is clearly expected to be better than BPSO. As was the superior performance of QEA for combinatorial optimization problems was demonstrated in [7]. Additional computational efficiency of QBPSO is also based on the concept and principles of quantum computing, which can achieve a better balance between exploration and exploitation of the solution space and obtain better solutions, even with a small population, compared with the conventional PSOs.

IV. Application of QBPSO to Unit Commitment Problem

A. Formulation of Unit Commitment Problem

1) Objective Function: The objective of the UC problem is to minimize the total operating cost, fuel cost, start-up cost and shut-down cost, of all generating units during a time horizon, subject to a number of system and unit constraints [9].

Fuel cost function: For all committed generating units, the total fuel cost is minimized by economically dispatching the units. The fuel cost function of unit j at hour t can be expressed as a second-order polynomial as follows:

$$F_j(P_{j,t}) = a_j + b_j P_{j,t} + c_j P_{j,t}^2 \tag{13}$$

where $P_{j,t}$ is the power generation of unit j at hour t and a_j, b_j, c_j are the cost coefficients of unit j.

Start-up cost: Start-up cost for restarting a de-committed generating unit, which is related to the temperature of the boiler, is included in the objective function. The start-up cost is associated with the number of hours during which the unit has been off. Start-up cost will be high, defined as the *cold cost* $(SU_{C,j})$, when down time duration exceeds the *cold-start hour* $(T_{cold,j})$ in excess of the minimum down time; and will be low, defined as the *hot cost* $(SU_{H,j})$, when down time duration does not exceed the cold-start hour in excess of the minimum down time. In general, the start-up cost is described as follows:

$$SU_{j,t} = \begin{cases} SU_{H,j}, & \text{if } MDT_j \le TOFF_{j,t} \le MDT_j + T_{cold,j} \\ SU_{C,j}, & \text{if } TOFF_{j,t} > MDT_j + T_{cold,j} \end{cases} \tag{14}$$

where $TOFF_{j,t}$ is the duration for which unit j has remained offline at hour t and MDT_j is the minimum down-time of the jth unit.

Shut-down cost: Shut-down cost is usually modeled as a constant value for each unit per shutdown. In this paper, the shut-down costs have not been taken into consideration.

Consequently, the objective function of the UC problem is given by the minimization of the following cost function:

$$\min \sum_{t=1}^{T} \sum_{j=1}^{N} \left[F_j(P_{j,t}) u_{j,t} + SU_{j,t}(1 - u_{j,t-1}) u_{j,t}\right] \tag{15}$$

where T is the number of scheduling period, N is the number of generating units, and $u_{j,t}$ is the on/off status of unit j at hour t (i.e., $u_{j,t} = 1$ when unit j is online, and $u_{j,t} = 0$ when unit j is offline).

2) System and Unit Constraints:

Load balance constraints: The sum of unit generation outputs at each hour must satisfy the system load demand requirement of the corresponding hour as follows:

$$\sum_{j=1}^{N} P_{j,t} u_{j,t} = PD_t \tag{16}$$

where PD_t is the total system demand at hour t.

Generation limit constraints: The power produced by each unit must be within its limits as indicated below:

$$u_{j,t} P_{j,\min} \le P_{j,t} \le u_{j,t} P_{j,\max} \tag{17}$$

where $P_{j,\min}$ and $P_{j,\max}$ are the minimum and maximum generation limits of unit j, respectively.

Spinning reserve constraints: Spinning reserve must be provided so as to minimize the probability of load interruption. The spinning reserve is considered to be a pre-specified amount or a given percentage of the forecasted peak demand. Spinning reserve can be specified in terms of excess megawatt capacity, which is expressed by

$$\sum_{j=1}^{N} P_{j,\max} u_{j,t} \ge PD_t + SR_t \tag{18}$$

where SR_t is the required spinning reserve at hour t.

Generation ramping constraints: Due to the mechanical characteristics and thermal stress limitations of a generating unit, the operating range of all online units is restricted by their corresponding ramp-rate limits as follows:

$$RD_j \le P_{j,t} - P_{j,(t-1)} \le RU_j \tag{19}$$

where RD_j and RU_j are the ramp-down and ramp-up limits of unit j, respectively.

Minimum up-time/down-time constraints: The unit cannot be turned on/off immediately once it is committed or de-committed. The minimum up/down time constraints indicate that a unit must be on/off during a certain number of hours before it becomes shut-down or start-up, respectively. These constraints are given by

$$u_{j,t} = \begin{cases} 1, & \text{if } 1 \le TON_{j,t-1} < MUT_j \\ 0, & \text{if } 1 \le TOFF_{j,t-1} < MDT_j \\ 0 \text{ or } 1, & \text{otherwise} \end{cases} \tag{20}$$

where $TON_{j,t}$ is the duration for which unit j remains online at hour t and MUT_j is the minimum up-time of unit j.

B. Implementation of QBPSO for UC Problems

Since the UC problems involve determining the on/off states of generating units, the decision variables are the on/off status of

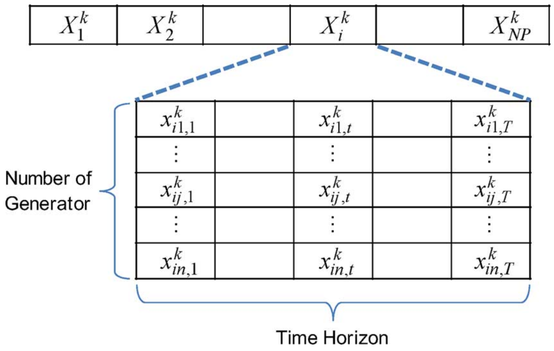

Fig. 1. Structure of a population of QBPSO for UC problems.

generating units. The structure of a population of the proposed QBPSO for UC problems is depicted in Fig. 1. The $x_{ij,t}$ is set to be 1 if the jth generator in the ith particle at hour t is ON; otherwise, $x_{ij,t}$ is set to be 0.

After determining the optimal commitment scheduling, the optimal power outputs of the units are determined through the conventional economic dispatch (ED) procedure. Since the fuel cost function of a generating unit is approximately represented as the quadratic function as (13), ED problem can be easily solved by numerical techniques. In the subsequent sections, the detailed procedures of QBPSO for scheduling the on/off states of units are described.

1) Creating Initial Q-Bit Individual and Position of Particles: In the initialization process, $\alpha_{ij,t}^0$ and $\beta_{ij,t}^0$ of all Q-bit individuals are set to be $1/\sqrt{2}$. It means that a Q-bit individual represents the linear superposition of all possible states with the same probability. The initial position of a set of particles is determined by the probability stored in the initialized Q-bit individuals. After generating a random number $rn_{ij,t}$, an initial value of the jth element in the ith particle at hour t (i.e., $x_{ij,t}^0$) takes a value of 1 if $r_{ij,t}$ is less than 1/2; otherwise, it is set to be 0. The initial $Pbest$ of each particle is set as its initial position, and the initial $Gbest$ is determined as the position of the particle with the minimum cost.

2) Q-Bit Individual Update: Q-bit individuals are updated by the rotation gate. After setting the magnitude of rotation angle by (12), the proposed coordinate rotation gate determines the rotation angle $\Delta\theta$ for each Q-bit as follows:

$$\Delta\theta_{ij,t}^{k+1} = \theta\times\left\{\gamma_{1i}^k \times \left(x_{ij,t}^{P,k} - x_{ij,t}^k\right) + \gamma_{2i}^k \times \left(x_{j,t}^{G,k} - x_{ij,t}^k\right)\right\}. \tag{21}$$

Then a new pair of (α, β) of each Q-bit in Q-bit individuals is obtained by

$$\begin{bmatrix} \alpha_{ij,t}^{k+1} \\ \beta_{ij,t}^{k+1} \end{bmatrix} = \begin{bmatrix} \cos\left(\Delta\theta_{ij,t}^{k+1}\right) & -\sin\left(\Delta\theta_{ij,t}^{k+1}\right) \\ \sin\left(\Delta\theta_{ij,t}^{k+1}\right) & \cos\left(\Delta\theta_{ij,t}^{k+1}\right) \end{bmatrix} \begin{bmatrix} \alpha_{ij,t}^{k} \\ \beta_{ij,t}^{k} \end{bmatrix} \tag{22}$$

where t is the index of time $(t = 1, 2, \cdots, T)$. The updated Q-bit should satisfy the normalization condition, $|\alpha_{ij,t}^{k+1}|^2 + |\beta_{ij,t}^{k+1}|^2 = 1$.

3) Modification of Position of Particles: The position vector of the ith particle at iteration k (i.e., X_i^k) is modified by the probability stored in the ith Q-bit individual as follows:

$$x_{ij,t}^k = \begin{cases} 1, & \text{if } rn_{ij,t} < \left|\beta_{ij,t}^k\right|^2 \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

4) Update of $Pbest$ and $Gbest$: If X_i^{k+1} yields a smaller cost function value than $Pbest_i^k$, then $Pbest_i^{k+1}$ is set to X_i^{k+1}. Otherwise, $Pbest_i^k$ is retained:

$$Pbest_i^{k+1} = \begin{cases} X_i^{k+1}, & \text{if } f\left(X_i^{k+1}\right) \le f\left(Pbest_i^k\right) \\ Pbest_i^k, & \text{otherwise.} \end{cases} \tag{24}$$

Also, $Gbest^{k+1}$ is set as the best evaluated position among the $Pbest_i^{k+1}$.

5) Stopping Criteria: The proposed QBPSO algorithm is terminated if the iteration reaches a pre-specified maximum iteration.

C. Constraint-Handling Techniques

This paper applies the rule-based heuristic constraint-handling techniques for the minimum up/down time and the spinning reserve constraints [25]. In evolutionary process for solving UC problem, random bits-flipping of state variables has occurred, thereby the constraints may be frequently violated. Therefore, heuristic-based repair algorithms are applied to accelerate the solution quality and to avoid infeasible solutions. To reduce the operating costs incurred by the excessive spinning reserve, the unit de-commitment approach is also introduced.

1) Minimum Up-Time/Down-Time Constraints: While modifying the binary solution of each individual, the minimum up/down time constraints should be satisfied. To do this, this paper introduces a heuristic-based constraint treatment technique, as illustrated in the following pseudo-code:

```
Begin
    For $j = 1$ to MaxUnit
        If unit $j$ is set to be ON at hour $t$ (i.e., $u_{j,t} = 1$)
        then
            If $u_{j,t-1} = 0$ then
                If $TOFF_{j,t-1} < MDT_j$ then $u_{j,t} = 0$
                Elseif $TOFF_{j,t-1} \ge MDT_j$ then
                $u_{j,t} = 1$
                Endif
            Elseif $u_{j,t-1} = 1$ then $u_{j,t} = 1$
            Endif
        Elseif $u_{j,t} = 0$ then
            If $u_{j,t-1} = 1$ then
                If $TON_{j,t-1} < MUT_j$ then $u_{j,t} = 1$
                Elseif $TON_{j,t-1} \ge MUT_j$ then
                $u_{j,t} = 0$
```

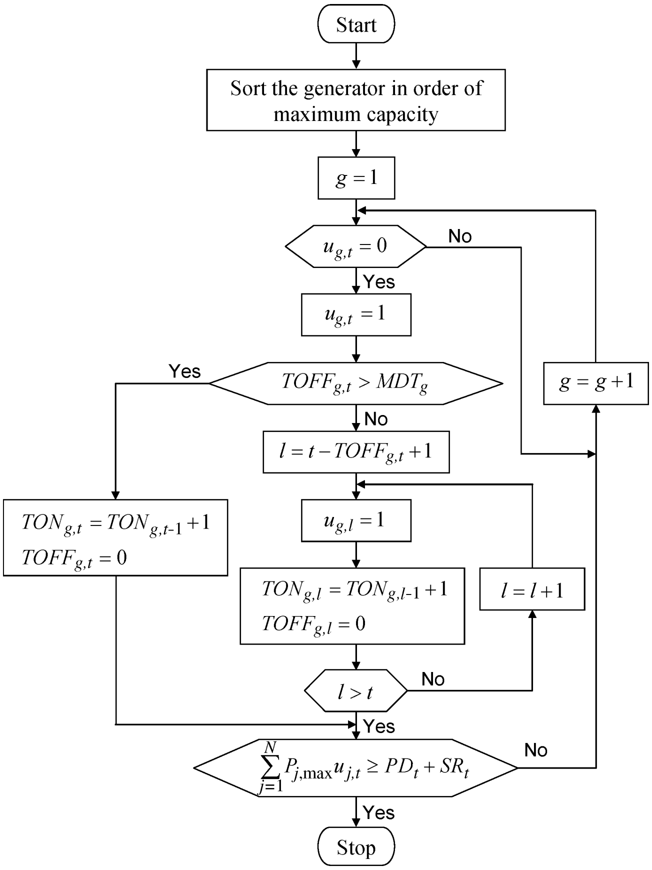

Fig. 2. Flowchart of repair algorithm for handling spinning reserve constraint.

```
                Endif
            Elseif $u_{j,t-1} = 0$ then $u_{j,t} = 0$
            Endif
        Endif
    Next $j$
End
```

2) Spinning Reserve Constraints: Adequate spinning reserves are required to maintain the system reliability for a given time horizon. If the spinning reserve constraint is violated, the system suffers from deficiency of units. This paper introduces an efficient heuristic-based repair method, which is launched when the spinning reserve is deficient at any scheduling period to avoid infeasible solutions. In the repair process, de-committed units are forced to turn on until the spinning reserve constraint is satisfied, as shown in Fig. 2.

3) Unit De-Commitment for Excessive Spinning Reserve: Excessive spinning reserve is not desirable due to the high operation costs. Therefore, this paper introduces a heuristic-based unit de-commitment process to reduce the excessive spinning reserve, leading to cost savings, as illustrated in Fig. 3. The unit de-commitment process is performed after obtaining the solutions satisfying the minimum up/down time and the spinning reserve constraints.

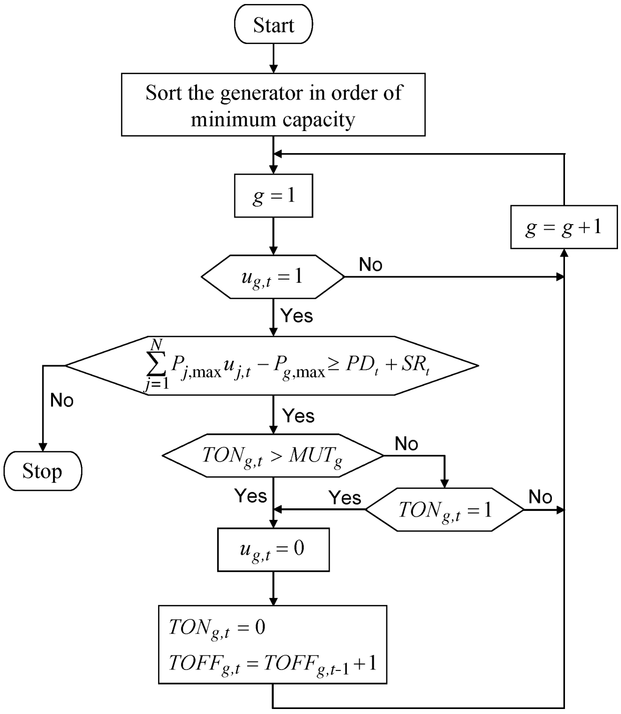
Start
Sort the generator in order of minimum capacity
$g = 1$
$g = g+1$
$u_{g,t} = 1$
No
Yes
$\sum_{j=1}^{N} P_{j,\max} u_{j,t} - P_{g,\max} \geq PD_t + SR_t$
No
Yes
Stop
$TON_{g,t} > MUT_g$
No
Yes
Yes
$TON_{g,t} = 1$
No
$u_{g,t} = 0$
$TON_{g,t} = 0$
$TOFF_{g,t} = TOFF_{g,t\text{-}1} + 1$

Fig. 3. Flowchart of unit de-commitment for prevention of excessive spinning reserve.

V. Numerical Tests

The QBPSO is applied to UC problems for realistic power systems of different sizes, which consist of the 10-, 20-, 40-, 60-, 80-, and 100-units along with 24-h load demands. Also, their results are compared with those of previous works which used the same testbeds [18], [20], [23]–[26]. For each test case, 50 independent trials are conducted to compare the solution quality and convergence characteristics. Numerical tests have been executed on a Pentium IV 2.0-GHz computer.

A. Parameter Sensitivity Analysis

The performance of the proposed QBPSO is influenced by the rotation angle and the population size. The maximum iteration number is set as 1000. In [7], the value from 0.05π to 0.001π is recommended for the magnitude of rotation angle. The proposed dynamic rotation angle approach determines the magnitude of rotation angle from $\theta_{\max}$ to $\theta_{\min}$ along the iteration. In order to find the optimal combination of rotation angles (i.e., $\theta_{\max}$ and $\theta_{\min}$), eight cases are tested on the 40-unit system with 10 and 30 population sizes, respectively, as shown in Fig. 4. Through this experiment, $\theta_{\max}$ and $\theta_{\min}$ are selected as 0.05π to 0.01π, respectively.

The population size NP is determined through the experiments for the 40-unit system with different population sizes. As shown in Fig. 5, the solution quality is continuously and marginally improved when increasing the population size, while the computation time is linearly increased. Through the heuristic

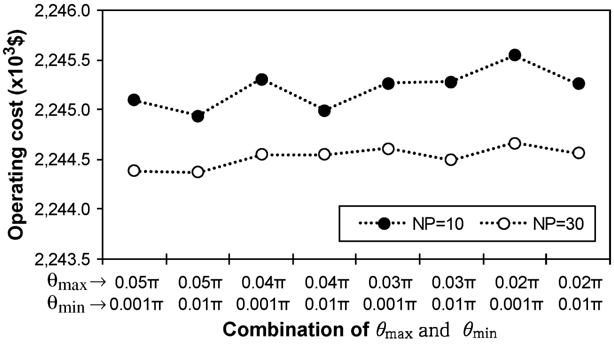

Fig. 4. Average cost of the QBPSO by combinations of rotation angles.

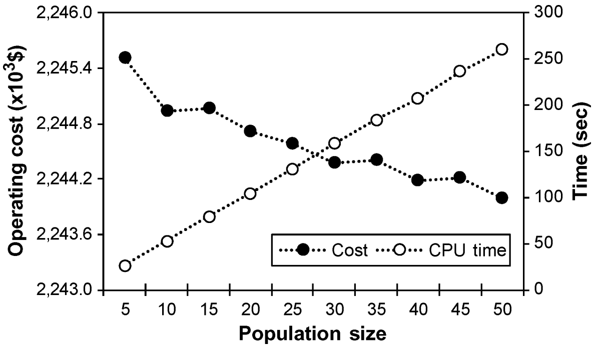

Fig. 5. Average cost and execution time of the QBPSO on the 40-unit system by population sizes.

TABLE II
GENERATING UNIT DATA FOR THE TEN-UNIT BASE SYSTEM

Unit	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10
$P_{\max}$ (MW)	455	455	130	130	162	80	85	55	55	55
$P_{\min}$ (MW)	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	0.00712	0.00079	0.00413	0.00222	0.00173
MUT (Hr)	8	8	5	5	6	3	3	1	1	1
MDT (Hr)	8	8	5	5	6	3	3	1	1	1
SU_H ($)	4500	5000	550	560	900	170	260	30	30	30
SU_C ($)	9000	10000	1100	1120	1800	340	520	60	60	60
T_{cold} (Hr)	5	5	4	4	4	2	2	0	0	0
Initial status (Hr)	8	8	-5	-5	-6	-3	-3	-1	-1	-1

trade-off analysis between the solution quality and the computation time, the population size (i.e., NP) is selected as 30.

TABLE III
DEMAND DATA WITH 24-H TIME HORIZON

Hour	Demand (MW)	Hour	Demand (MW)	Hour	Demand (MW)
1	700	9	1,300	17	1,000
2	750	10	1,400	18	1,100
3	850	11	1,450	19	1,200
4	950	12	1,500	20	1,400
5	1,000	13	1,400	21	1,300
6	1,100	14	1,300	22	1,100
7	1,150	15	1,200	23	900
8	1,200	16	1,050	24	800

TABLE IV
SIMULATION RESULTS OF EACH METHOD FOR TEST SYSTEMS

Units	Method	Best Cost ($)	Average Cost ($)	Worst Cost ($)	Standard Deviation
10	IQEA	563,977	563,977	563,977	0.00
	BPSO	563,977	563,977	563,977	0.00
	QBPSO	563,977	563,977	563,977	0.00
20	IQEA	1,123,890	1,124,320	1,124,504	126.28
	BPSO	1,123,783	1,124,242	1,124,294	156.68
	QBPSO	1,123,297	1,123,981	1,124,294	376.92
40	IQEA	2,245,151	2,246,026	2,246,701	377.90
	BPSO	2,243,210	2,244,634	2,245,982	450.57
	QBPSO	2,242,957	2,244,657	2,245,941	673.62
60	IQEA	3,365,003	3,365,667	3,366,223	309.36
	BPSO	3,363,649	3,365,301	3,367,171	982.42
	QBPSO	3,361,980	3,363,763	3,365,707	795.64
80	IQEA	4,486,963	4,487,985	4,489,286	501.35
	BPSO	4,487,388	4,488,725	4,489,793	538.02
	QBPSO	4,482,085	4,485,410	4,487,168	1053.87
100	IQEA	5,606,022	5,607,561	5,608,525	577.74
	BPSO	5,608,172	5,609,705	5,611,005	701.69
	QBPSO	5,602,486	5,604,275	5,606,178	951.99

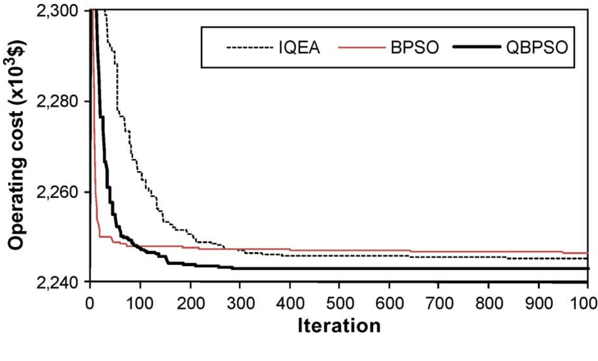

Fig. 6. Convergence characteristics of the IQEA, BPSO, and QBPSO for the 40-unit system.

B. Experimental Results

The proposed QBPSO is initially tested on a simple ten-unit base system with a 24-h time horizon. The unit characteristics of the ten-unit system and the demand are given in Tables II and III, respectively. Subsequently, the 20-, 40-, 60-, 80-, and 100-unit data are obtained by duplicating the base case, and the load demands are adjusted in proportion to the system size. In all cases, the spinning reserve requirements are assumed to be 10% of the hourly demand.

In Table IV, the best, average and worst costs, and standard deviation for test systems obtained by the proposed QBPSO algorithm are summarized and compared with those of the conventional BPSO and improved quantum evolutionary algorithm (IQEA) [25]. Here, the BPSO employed the same constraint

TABLE V
COMPARISON OF SIMULATION RESULTS OF EACH METHOD

Method	Best Cost ($)	Average Cost ($)	Worst Cost ($)	Difference (%)	Average Time (sec)
10-Unit					
GA [18]	565,825	-	570,032	0.74	221
EP [20]	564,551	565,352	566,231	0.30	100
SA [23]	565,828	565,988	566,260	0.08	3
IPSO [24]	**563,954**	564,162	564,579	0.11	-
IQEA [25]	563,977	563,977	563,977	0.00	15
QEA-UC [26]	**563,938**	563,969	564,672	0.13	19
QBPSO	563,977	563,977	563,977	0.00	18
20-Unit					
GA [18]	1,126,243	-	1,132,059	0.52	733
EP [20]	1,125,494	1,127,257	1,129,793	0.38	340
SA [23]	1,126,251	1,127,955	1,129,112	0.25	17
IPSO [24]	1,125,279		1127643	0.21	-
IQEA [25]	1,123,890	1,124,320	1,124,504	0.05	42
QEA-UC [26]	1,123,607	1,124,689	1,125,715	0.19	28
QBPSO	**1,123,297**	1,123,981	1,124,294	0.09	50
40-Unit					
GA [18]	2,251,911	-	2,259,706	0.35	2,697
EP [20]	2,249,093	2,252,612	2,256,085	0.31	1,176
SA [23]	2,250,063	2,252,125	2,254,539	0.20	88
IPSO [24]	2,248,163	-	2252117	0.18	-
IQEA [25]	2,245,151	2,246,026	2,246,701	0.07	132
QEA-UC [26]	2,245,557	2,246,728	2,248,296	0.12	43
QBPSO	**2,242,957**	2,244,657	2,245,941	0.13	158
60-Unit					
GA [18]	3,376,625	-	3,384,252	0.23	5,840
EP [20]	3,371,611	3,376,255	3,381,012	0.28	2,267
SA [23]	-	-	-	-	-
IPSO [24]	3,370,979	-	3379125	0.24	-
IQEA [25]	3,365,003	3,365,667	3,366,223	0.04	273
QEA-UC [26]	3,366,676	3,368,220	3,372,007	0.16	54
QBPSO	**3,361,980**	3,363,763	3,365,707	0.11	328
80-Unit					
GA [18]	4,504,933	-	4,510,129	0.12	10,036
EP [20]	4,498,479	4,505,536	4,512,739	0.32	3,584
SA [23]	4,498,076	4,501,156	4,503,987	0.13	405
IPSO [24]	4,495,032	-	4,508,943	0.31	-
IQEA [25]	4,486,963	4,487,985	4,489,286	0.05	453
QEA-UC [26]	4,488,470	4,490,128	4,492,839	0.10	66
QBPSO	**4,482,085**	4,485,410	4,487,168	0.11	554
100-Unit					
GA [18]	5,627,437	-	5,637,914	0.19	15,733
EP [20]	5,623,885	5,633,800	5,639,148	0.27	6,120
SA [23]	5,617,876	5,624,301	5,628,506	0.19	696
IPSO [24]	5,619,284	-	5,633,021	0.24	-
IQEA [25]	5,606,022	5,607,561	5,608,525	0.04	710
QEA-UC [26]	5,609,550	5,611,797	5,613,220	0.07	80
QBPSO	**5,602,486**	5,604,275	5,606,178	0.07	833

TABLE VI
UNIT SCHEDULING AND CORRESPONDING COSTS FOR THE TEN-UNIT SYSTEM

Hr	Generation Output										Total Power	Fuel Cost	Startup Cost
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10			
1	455	245	0	0	0	0	0	0	0	0	700	13,683	0
2	455	295	0	0	0	0	0	0	0	0	750	14,554	0
3	455	370	0	0	25	0	0	0	0	0	850	16,809	900
4	455	455	0	0	40	0	0	0	0	0	950	18,598	0
5	455	390	0	130	25	0	0	0	0	0	1,000	20,020	560
6	455	360	130	130	25	0	0	0	0	0	1,100	22,387	1,100
7	455	410	130	130	25	0	0	0	0	0	1,150	23,262	0
8	455	455	130	130	30	0	0	0	0	0	1,200	24,150	0
9	455	455	130	130	85	20	25	0	0	0	1,300	27,251	860
10	455	455	130	130	162	33	25	10	0	0	1,400	30,058	60
11	455	455	130	130	162	73	25	10	10	0	1,450	31,916	60
12	455	455	130	130	162	80	25	43	10	10	1,500	33,890	60
13	455	455	130	130	162	33	25	10	0	0	1,400	30,058	0
14	455	455	130	130	85	20	25	0	0	0	1,300	27,251	0
15	455	455	130	130	30	0	0	0	0	0	1,200	24,150	0
16	455	310	130	130	25	0	0	0	0	0	1,050	21,514	0
17	455	260	130	130	25	0	0	0	0	0	1,000	20,642	0
18	455	360	130	130	25	0	0	0	0	0	1,100	22,387	0
19	455	455	130	130	30	0	0	0	0	0	1,200	24,150	0
20	455	455	130	130	162	33	25	10	0	0	1,400	30,058	490
21	455	455	130	130	85	20	25	0	0	0	1,300	27,251	0
22	455	455	0	0	145	20	25	0	0	0	1,100	22,736	0
23	455	420	0	0	25	0	0	0	0	0	900	17,685	0
24	455	345	0	0	0	0	0	0	0	0	800	15,427	0
Total	10,920	9,680	2,080	2,210	1,540	332	225	83	20	10	27,100	559,887	4,090

treatment techniques used in the IQEA and QBPSO. The simulation results show that the proposed QBPSO provides much better solutions than the IQEA and BPSO.

The convergence characteristics of the IQEA, BPSO, and QBPSO for the 40-unit system are illustrated in Fig. 6. We have observed that the QBPSO was improving the solution quality continuously while the BPSO experienced a premature convergence.

In Table V, the best results of the proposed QBPSO are compared with those of genetic algorithm (GA) [18], evolutionary programming (EP) [20], simulated annealing (SA) [23], improved particle swarm optimization (IPSO) [24], IQEA [25], and QEA-based UC method (QEA-UC) [26]. Table V reveals that the proposed QBPSO is clearly superior to all existing methods. In the 100-unit system, for example, the QBPSO can save the operating costs of $3536 in a 24-h period compared to the IQEA [25] which is the best solution until now. The QBPSO works well for larger systems. For 10 gen unit, IPSO [24] and QEA [26] seem to provide better results than QBPSO. This is because the overhead for QBPSO is the same, independent of system sizes, and the effect of computational efficiency is not apparent for small system. However, the differences in the best cost among the results of IPSO [24], QEA [26], and QBPSO are very small. It should be noticed that the QBPSO has been terminated earlier than QEA [26] due to the termination criterion, which could have been refined for better operating cost.

For the ten-unit and 100-unit systems, the commitment schedules during the planning horizon obtained by the proposed QBPSO are described in Tables VI and VII, respectively.

Fig. 7 illustrates the execution time of the QBPSO with the system size. As shown in the figure, the execution time increases in a quadratic way with the number of units.

VI. CONCLUSIONS

This paper presents a new BPSO inspired by quantum computing and it is applied to the UC problems in power systems.

TABLE VII
COMMITMENT SCHEDULING FOR THE 100-UNIT SYSTEM OBTAINED BY QBPSO

Hr	Unit Schedule
1	1111111111111111111100000000000000000000000000000000000000000000000000000000000000000000000000000000
2	1111111111111111111100000000000000000000000000000000000000000000000000000000000000000000000000000000
3	1111111111111111111100000000000000000000000011000000000000000000000000000000000000000000000000000000
4	1111111111111111111100000000000000110000001011111100000000000000000000000000000000000000000000000000
5	1111111111111111111100000100011011100000111111111100000000000000000000000000000000000000000000000000
6	1111111111111111111101000100011111110011111111111100000000000000000000000000000000000000000000000000
7	1111111111111111111111000100111111111111111111111100000000000000000000000000000000000000000000000000
8	1111111111111111111111111111111111111111111111111100000000000000000000000000000000000000000000000000
9	1111111111111111111111111111111111111111111111111100111101111101010000000000000000000000000000000000
10	1111111111111111111111111111111111111111111111111111111111111111111111100101000100100000000010100001
11	1111111111111111111111111111111111111111111111111111111111111111111111111101111010110000000111100111
12	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111011111101111
13	1111111111111111111111111111111111111111111111111111111111111111111111101010101000100100010000000000
14	1111111111111111111111111111111111111111111111111111001011111000001110000000000000000000000000000000
15	1111111111111111111111000011111110111111111111111110010111001000000000000000000000000000000000000000
16	1111111111111111111111000011111110111111111111111100000000000000000000000000000000000000000000000000
17	1111111111111111111111000011111110111111111111111100000000000000000000000000000000000000000000000000
18	1111111111111111111111000011111110111111111111111100000000000000000000000000000000000000000000000000
19	1111111111111111111111000011111110111111111111111100110100000011011100000000000000000000000000000000
20	1111111111111111111111111111111111111111111111111111111111111011011110101100101000111010000110000110
21	1111111111111111111110111101111111111110111111111111111111111110110111000000000000000000000000000000
22	1111111111111111111110111001000111000101110111100011001011111000000010000000000000000000000000000000
23	1111111111111111111110111000000000100000000000100000000000000000000000000000000000000000000000000000
24	1111111111110111110110111100000000100000000000000000000000000000000000000000000000000000000000000000

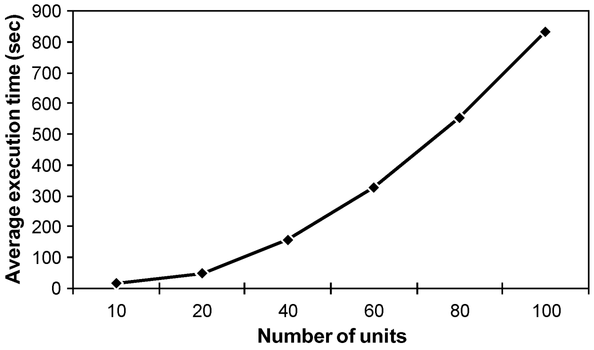

Fig. 7. Scaling of the average execution time of the QBPSO.

The proposed QBPSO is based on the concept and principles of quantum computing, and developed to enhance the conventional BPSO in solving the combinatorial optimization problems. The QBPSO introduces a Q-bit individual for the probabilistic representation, which replaces the velocity update procedure in the traditional PSO, in order to derive a swarm toward promising regions in a search space with a simple mechanism. In the QBPSO, therefore, a weight factor and two acceleration coefficients in the PSO framework are removed and only one factor, rotation angle, is considered. In updating Q-bit individuals, this paper also proposes a new rotation gate, which includes a coordinate rotation gate for updating Q-bits and a dynamic rotation angle approach for determining the magnitude of rotation angle. The proposed QBPSO is applied to UC problems in which several test power systems consisting of up to 100-units along with 24-h load demands and the results were compared with those of previous works. The simulation results clearly reveal that the proposed QBPSO algorithm can be used as an excellent optimizer in solving large-scale UC problems.

REFERENCES

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks (ICNN'95)*, Perth, Australia, 1995, vol. IV, pp. 1942–1948.

[2] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann, 2001.

[3] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, 1997, vol. 5, pp. 4104–4108.

[4] L. Spector, H. Barnum, H. J. Bernstein, and N. Swamy, "Finding a better-than-classical quantum AND/OR algorithm using genetic programming," in *Proc. 1999 Congr. Evolutionary Computation*, Piscataway, NJ, Jul. 1999, vol. 3, pp. 2239–2246, IEEE Press.

[5] M. Moore and A. Narayanan, Quantum-Inspired Computing, Dept. Comput. Sci., Univ. Exeter. Exeter, U.K., 1995.

[6] A. Narayanan and Moore, "Quantum-inspired genetic algorithms," in *Proc. IEEE Int. Conf. Evolutionary Computation*, Japan, 1996, pp. 61–66.

[7] K. H. Han and J. H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002.

[8] J. G. Vlachogiannis and K. Y. Lee, "Quantum-inspired evolutionary algorithm for real and reactive power dispatch," *IEEE Trans. Power Syst.*, vol. 23, no. 4, pp. 1627–1636, Nov. 2008.

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

[10] R. M. Burns and C. A. Gibson, "Optimization of priority lists for a unit commitment program," presented at the IEEE Power Engineering Society Summer Meeting, 1975, Paper A, 75 453-1.

[11] 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.

[12] W. L. Snyder, Jr., H. D. Powell, Jr., and J. C. Rayburn, "Dynamic programming approach to unit commitment," *IEEE Trans. Power Syst.*, vol. 2, no. 2, pp. 339–350, May 1987.

[13] 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.

[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] A. I. Cohen and M. Yoshimura, "A branch-and-bound algorithm for unit commitment," *IEEE Trans. Power App. Syst.*, vol. PAS-102, pp. 444–451, Feb. 1983.

[17] 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.

[18] 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.

[19] K. S. Swarup and S. Yamashiro, "Unit commitment solution methodology using genetic algorithm," *IEEE Trans. Power Syst.*, vol. 17, no. 1, pp. 87–91, Feb. 2002.

[20] 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.

[21] H. Chen and X. Wang, "Cooperative coevolutionary algorithm for unit commitment," *IEEE Trans. Power Syst.*, vol. 16, no. 1, pp. 128–133, Feb. 2002.

[22] F. Zhuang and F. D. Galiana, "Unit commitment by simulated annealing," *IEEE Trans. Power Syst.*, vol. 5, no. 1, pp. 311–317, Feb. 1990.

[23] D. N. Simopoulos, S. D. Kavatza, and C. D. Vournas, “Unit commitment by an enhanced simulated annealing algorithm,” *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 68–76, Feb. 2006.

[24] B. Zhao, C. X. Guo, B. R. Bai, and Y. J. Cao, “An improved particle swarm optimization algorithm for unit commitment,” *Elect. Power Energy Syst.*, vol. 28, no. 7, pp. 482–490, Sep. 2006.

[25] Y.-W. Jeong, J.-B. Park, J.-R. Shin, and K. Y. Lee, “A thermal unit commitment approach using an improved quantum evolutionary algorithm,” *Elect. Power Compon. Syst.*, vol. 37, no. 7, pp. 770–786, Jul. 2009.

[26] 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.

[27] , K. Y. Lee and M. A. El-Sharkawi, Eds., *Modern Heuristic Optimization Techniques With Applications to Power Systems*. New York: Wiley/IEEE Press, 2008.

[28] J.-B. Park, K.-S. Lee, J.-R. Shin, and K. Y. Lee, “A particle swarm optimization for economic dispatch with non-smooth cost functions,” *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 34–42, Feb. 2005.

[29] J. G. Vlachogiannis and K. Y. Lee, “Determining generator contributions to transmission system using parallel vector evaluated particle swarm optimization,” *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1765–1774, Nov. 2005.

[30] J. S. Heo, K. Y. Lee, and R. Garduno-Ramirez, “Multiobjective control of power plants using particle swarm optimization techniques,” *IEEE Trans. Energy Convers.*, vol. 21, no. 2, pp. 552–561, Jun. 2006.

[31] J. G. Vlachogiannis and K. Y. Lee, “A comparative study on particle swarm optimization for optimal steady-state performance of power systems,” *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1718–1728, Nov. 2006.

[32] J. G. Vlachogiannis and K. Y. Lee, “Economic dispatch—A comparative study on heuristic optimization techniques with an improved coordinated aggregation-based PSO,” *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 991–1001, May 2009.

[33] L. S. Coelho and V. C. Mariani, “Particle swarm approach based on quantum mechanics and harmonic oscillator potential well for economic load dispatch with valve-point effects,” *Energy Convers. Manage.*, vol. 49, no. 11, pp. 3080–3085, Nov. 2008.

[34] S. L. Sabat, L. S. Coelho, and A. Abraham, “MESFET DC model parameter extraction using quantum particle swarm optimization,” *Microelectron. Reliab.*, vol. 49, no. 6, pp. 660–666, Jun. 2009.

[35] J. Sun, W. Fang, D. Wang, and W. Xu, “Solving the economic dispatch problem with a modified quantum-behaved particle swarm optimization method,” *Energy Convers. Manage.*, vol. 50, no. 12, pp. 2967–2975, Dec. 2009.

Yun-Won Jeong received the B.S., M.S., and Ph.D. degrees from Konkuk University, Seoul, Korea, in 2002, 2004, and 2007, respectively.

His research interests include power system operation and planning, electricity markets, and computational intelligence and their application to power systems.

Jong-Bae Park (M’98) received the B.S., M.S., and Ph.D. degrees from Seoul National University, Seoul, Korea, in 1987, 1989, and 1998, respectively.

During 1989–1998, he was with Korea Electric Power Corporation, and during 1998–2001, he was an Assistant Professor at Anyang University, Anyang, Korea. During 2006–2008, he was a guest researcher of EPRI. Since 2001, he has been with the Electrical Engineering Department at Konkuk University, Seoul, as an Associate Professor. His major research topics include power system operation, planning, economics, and markets.

Se-Hwan Jang received the B.S. and M.S. degrees from Pusan National University, Busan, Korea, in 2006 and 2008, and he is pursuing the Ph.D. degree at Konkuk University, Seoul, Korea.

His research interest includes optimal operation and planning of power system resources, deregulation and restructuring of power industry, electricity market analysis, and intelligent systems applications to power systems.

Kwang Y. Lee (F’01) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1964, the M.S. degree in electrical engineering from North Dakota State University, Fargo, ND, in 1967, and the Ph.D. degree in systems science from Michigan State University, East Lansing, in 1971.

He was on the faculties of Michigan State University, Oregon State University, University of Houston, and The Pennsylvania State University, and he is currently a Professor and Chair of Electrical and Computer Engineering, Baylor University, Waco, TX. His research interests include control theory, computational intelligence and their application to power systems, and power plant control.

Dr. Lee is a Vice-Chair of the IFAC Technical Committee on Power Plants and Power Systems Control, an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS, and an Editor of the IEEE TRANSACTIONS ON ENERGY CONVERSION.