Quantum-Inspired Evolutionary Algorithm Approach for Unit Commitment

T. W. Lau, C. Y. Chung, Senior Member, IEEE, K. P. Wong, Fellow, IEEE, T. S. Chung, and S. L. Ho

Abstract—This paper presents a novel method for solving the unit commitment (UC) problem based on quantum-inspired evolutionary algorithm (QEA). The proposed method applies QEA to handle the unit-scheduling problem and the Lambda-iteration technique to solve the economic dispatch problem. The QEA method is based on the concept and principles of quantum computing, such as quantum bits, quantum gates and superposition of states. QEA employs quantum bit representation, which has better population diversity compared with other representations used in evolutionary algorithms, and uses quantum gate to drive the population towards the best solution. The mechanism of QEA can inherently treat the balance between exploration and exploitation and also achieve better quality of solutions, even with a small population. The proposed method is applied to systems with the number of generating units in the range of 10 to 100 in a 24-hour scheduling horizon and is compared to conventional methods in the literature. Moreover, the proposed method is extended to solve a large-scale UC problem in which 100 units are scheduled over a seven-day horizon with unit ramp-rate limits considered. The application studies have demonstrated the superior performance and feasibility of the proposed algorithm.

IndexTerms—Evolutionaryalgorithm,quantumcomputing,quantum-inspired evolutionary algorithm, unit commitment.

I. INTRODUCTION

UNIT COMMITMENT (UC) is an important optimization problem in power system operation. Its objective is to schedule the generating units online or offline over a scheduling horizon such that the power production cost is minimized with the load demand fully met and the operation constraints satisfied. The UC problem is combinatorial in nature and consists of many hard constraints. In solving this problem, generator schedules are first found and their costs are evaluated through the economic dispatch calculation.

Various numerical optimization approaches have been applied to deal with the UC problem in the literatures. Deterministic techniques previously used include the priority list [1], dynamic programming [2]–[4], Lagrangian relaxation [5], [6], mixed-integer programming [7], and the branch-and-bound method [8]. The priority list approach is simple and fast, but it usually yields high production cost. The dynamic programming is flexible but suffers from the problem of high dimensionality. The branch-and-bound method uses a linear function to represent the fuel consumption and time-dependent start-up cost and obtains the required lower and upper bounds. However, its computational time increases exponentially with the increment of the dimension of the UC problem. The mixed-integer programming method employs linear programming technique to solve and check for an integer solution. But it also suffers from an excessive computational time requirement. While the Lagrangian relaxation method offers a faster solution, it may encounter numerical convergence problems. Artificial intelligence method based on heuristic depth-first search method [9], [10] was also developed but it can be limited in its application to large-scale UC problem.

Recently, evolutionary algorithms (EAs) have been successfully applied to solve the UC problem, such as genetic algorithm (GA) [11], [12], simulated annealing (SA) [13]–[16], evolutionary programming (EP) [17], particle swarm optimization (PSO) [18], and hybrid methods [19], [20]. These approaches are general-purpose stochastic optimization techniques, and they operate on a group of candidate solutions with different search mechanisms. These techniques have been reported to be capable of attaining global/near-global solution search. They have attracted much attention, owing to their great potential to escape from local convergence, easy implementation and accommodation of complex problem characteristics. Nevertheless, EAs are parameter-sensitive and computationally expensive. They often require a considerable amount of computational time when solving the UC problem. Although powerful GA and hybrid genetic/simulated annealing algorithms for economic dispatch [21]–[23], generator scheduling and fuel scheduling [24]–[26] have previously been developed by the third author and his colleagues, the ability of these algorithms to generate large amount of diversified candidate solutions so that the solution space can be thoroughly explored is limited.

Research on combining evolutionary computation and quantum computing started in the late 1990s. Quantum-inspired evolutionary computing is a branch of study on evolutionary computation and employs the certain principles of quantum mechanics, such as superposition, interference and uncertainty [27]–[30]. Based on the concept and principles of quantum computing, such as quantum bits (Q-bits), quantum gates (Q-gates) and superposition of states, Han and Kim [30] developed a quantum-inspired evolutionary algorithm (QEA), which can achieve a better balance between exploration and exploitation of the solution space and also obtain better solutions.
even with a small population, compared with the conventional EAs. The superior performance of QEA for combinatorial optimization problems was demonstrated in [30] and [31].

This paper applies QEA to solve the UC problem and proposes a novel QEA-based UC method (QEA-UC) in which the unit-scheduling problem is handled by QEA and the economic dispatch problem is solved by the commonly-used method, Lambda-iteration technique. The formulation of the UC problem is first presented in Section II. Section III describes the Lambda-iteration technique. The formulation of the UC dispatch problem is solved by the commonly-used method, optimization problems was demonstrated in [30] and [31].

II. PROBLEM FORMULATION

A. Notation

\( N \)  
Number of generating units.

\( H \)  
Number of hours.

\( k \)  
Index of unit \((k = 1, 2, \ldots, N)\).

\( h \)  
Index of time \((h = 1, 2, \ldots, H)\).

\( p_{kh} \)  
Control variable for the generation of unit \(k\) at hour \(h\).

\( u_{kh} \)  
Control variable for the on/off status of unit \(k\) at hour \(h\).

\( F_H \)  
Total system production cost within \(H\) hours.

\( F_{kh}(p_{kh}) \)  
Fuel cost function of unit \(k\) at hour \(h\).

\( a_k, b_k, c_k \)  
Cost function parameters of unit \(k\).

\( ST_{kh} \)  
Start-up cost of unit \(k\) at hour \(h\).

\( HSC_k/CSC_k \)  
Hot/cold start-up cost of the \(k\)th unit.

\( MDT_{k}/MUT_k \)  
Minimum down/up time of the \(k\)th unit.

\( CSH_k \)  
Cold start hours of unit \(k\).

\( T_k^{on} \)  
Duration during which unit \(k\) is continuously ON.

\( T_k^{off} \)  
Duration during which unit \(k\) is continuously OFF.

\( D_h \)  
System peak demand at hour \(h\).

\( R_h \)  
Spinning reserve at hour \(h\).

\( p_k^{(max)}/p_k^{(min)} \)  
Maximum/minimum output limit of unit \(k\).

B. Objective Function

The objective of UC problem is to minimize the total power production cost, comprising the fuel cost and the start-up cost, over a specified period of time or a scheduling horizon. The objective function can be expressed by

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

where \(F_{kh}(p_{kh})\) and \(ST_{kh}\) are given by

\[
F_{kh}(p_{kh}) = c_k(p_{kh})^2 + b_k(p_{kh}) + a_k 
\]

\[
ST_{kh} = \begin{cases} 
HSC_k, & \text{if } MDT_{k} \leq T_k^{off} \leq MDT_{k} + CSH_k \\
CSC_k, & \text{if } T_k^{off} > MDT_{k} + CSH_k .
\end{cases} 
\]

C. Constraints

The minimization of the objective in the UC problem is subjected to the following system and unit constraints:

1) Power balance constraint

\[
\sum_{k=1}^{N} p_{kh}u_{kh} = D_h. 
\]

2) Spinning reserve constraint

\[
\sum_{k=1}^{N} p_k^{(max)}u_{kh} \geq D_h + R_h. 
\]

3) Unit output constraint

\[
p_k^{(max)} \geq p_{kh} \geq p_k^{(min)}. 
\]

4) Minimum up time limit

\[
T_k^{on} \geq \text{MUT}_k. 
\]

5) Minimum down time limit

\[
T_k^{off} \geq \text{MDT}_k. 
\]

III. QUANTUM-INSPIRED EVOLUTIONARY ALGORITHM

Like other EAs, QEA [30] consists of the representation of individuals, evaluation functions as well as population dynamics. In quantum computing, a Q-bit is the smallest unit of information stored in a two-state quantum computer. QEA employs a Q-bit as a probabilistic representation, instead of binary, numeric or symbolic representation used in other EAs. A Q-bit
individual is defined by a string of Q-bits and can represent a linear superposition of all the possible states in the search space. With this property, QEA requires only a small population size to provide good population diversity to effectively explore the solution space. A Q-gate is defined as a variation operator of QEA to drive the probability of each Q-bit to either 1 or 0 and toward the best single state with a gradual diminishing diversity property in the optimization process. By using the concept of Q-bit representation and superposition principle, each Q-bit individual can represent and explore all possible states. Moreover, with Q-gate operation, each Q-bit is driven to exploit a single state. Thus, the mechanism of the QEA method can inherently treat the balance between exploration and exploitation. In the following sections, the Q-bit representation and the principle and procedure of the QEA method are described:

A. Representation

A Q-bit, which is defined as the smallest unit of information, can be represented as

\[
\begin{pmatrix}
\alpha \\
\beta
\end{pmatrix}
\]  

(9)

where \(\alpha\) and \(\beta\) are a pair of numbers with \(|\alpha|^2 + |\beta|^2 = 1\). \(|\alpha|^2\) and \(|\beta|^2\) give the probabilities that the Q-bit will be found in the “0” and “1” states, respectively.

The state of a Q-bit may be “0”, “1”, or a linear superposition of the two and is expressed by

\[
|\psi\rangle = |0\rangle + \beta |1\rangle
\]

(10)

where \(|0\rangle\) and \(|1\rangle\) mean the states “0” and “1”, respectively. \(|\alpha|^2\) and \(|\beta|^2\) determine the probabilities of states \(|0\rangle\) and \(|1\rangle\), respectively. Specifically, the larger the \(|\alpha|^2\) value is, the higher the probability of the state \(|0\rangle\) will be observed.

A Q-bit individual with a string of \(m\) Q-bits is defined as

\[
\begin{pmatrix}
\alpha_1 \\
\beta_1 \\
\alpha_2 \\
\beta_2 \\
\vdots \\
\alpha_m \\
\beta_m
\end{pmatrix}
\]

(11)

where \(|\alpha_i|^2 + |\beta_i|^2 = 1\) for \(i = 1, 2, \ldots, m\).

The merit of Q-bit representation is that a Q-bit individual can represent a linear superposition of states. By adopting the concept of Q-bit representation and superposition principle, a system with \(m\) Q-bits can represent \(2^m\) states at the same time. When a Q-bit individual contains all \(\alpha\) and \(\beta\) values equal to \(1/\sqrt{2}\), the linear superposition of all possible states with the same probability can be represented by

\[
|\psi\rangle = \sum_{k=1}^{2^m} \frac{1}{\sqrt{2^m}} |X_k\rangle
\]

(12)

where \(X_k\) is the \(k\)th state and represented by the binary string \(x_1x_2\ldots x_m\) and \(x_i\) is either 0 or 1.

For instance, a Q-bit individual with two Q-bits is given by

\[
\begin{pmatrix}
\frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}}
\end{pmatrix}
\]

(13)

The states of the Q-bit individual can be represented as

\[
\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle.
\]

(14)

The results of (14) indicate that the probabilities of all states are \(1/4\), and the individual with two Q-bits holds the information of four states at the same time.

B. QEA

QEA is characterized by the Q-bit representation for the population diversity, the observation process for making binary solutions from Q-bit individuals, the update process for driving the individuals toward better solutions by the rotation Q-gate, and termination conditions. The detailed procedure and mechanism of QEA, for solving a minimization problem with the objective function \(f(X)\) and the binary control variables \((X)\), are described as follows:

Step 1) Set the generation counter \(t = 0\).
Step 2) Initialize \(Q(t)\):

\(Q(t)\) represents a group of Q-bit individuals, which is initialized at \(t = 0\), and \(Q(t) = Q_0, Q_1, \ldots, Q_m\), where subscript \(n\) is the total number of Q-bit individuals and \(Q_j\) is the \(j\)th Q-bit individual at generation \(t\) which is defined as

\[
Q_j = \begin{pmatrix}
\alpha_{j1} \\
\beta_{j1} \\
\alpha_{j2} \\
\beta_{j2} \\
\vdots \\
\alpha_{jm} \\
\beta_{jm}
\end{pmatrix}
\]

(15)

where \(j = 1, 2, \ldots, n\) and \(m\) is the string length. If all \(\alpha_{ji}\) and \(\beta_{ji}\), for \(i = 1, 2, \ldots, m\), are initialized with \(1/\sqrt{2}\), then the probability of observing the state “1” or “0” of each Q-bit is the same.

Step 3) Determine \(X(t)\) by observing \(Q(t)\):

\(X(t)\) is a group of binary solutions and are obtained by observing \(Q(t)\). \(X(t) = [X_1^t, X_2^t, \ldots, X_n^t]\), where \(X_i^t\) is a binary solution and obtained by observing \(Q_j^t\).

\[
X_i^{t_j} = [x_{i1}^t, x_{i2}^t, \ldots, x_{in}^t]
\]

where \(x_{ij}\) is binary and determined by comparing \(|\beta_{ji}|^2\) to a uniformly distributed random number in the range of 0 to 1. Here, \(x_{ij}\) is set to 1 if random \([0, 1] < |\beta_{ji}|^2\); otherwise \(x_{ij}\) is set to 0.

Step 4) Evaluate \(X(t)\):

The fitness or objective function values of the solutions in \(X(t)\) are evaluated.

Step 5) Store the best solution in \(X(t)\) into \(B(t)\):

\(B(t)\) is a matrix that stores the best solution in the whole population. It should be noted that the local best solutions in subpopulations can also be considered. The details can be found in [30].

Step 6) Set \(t = t + 1\).
Step 7) Determine \(X(t)\) by observing \(Q(t - 1)\).
Step 8) Evaluate $X(t)$.
Step 9) Update $Q(t)$ using Q-gates:

Q-bit individuals are updated by using Q-gates. A Q-gate is a variation operator of QEA to update the Q-bits, and the updated Q-bit at generation $t$ ($\alpha_{ji}^t, \beta_{ji}^t$) should meet the normalization condition, $|\alpha_{ji}^t|^2 + |\beta_{ji}^t|^2 = 1$. Rotation gates are considered in QEA. The rotation gate $U(\Delta \theta_{ji}^t)$ and the update operation are expressed as

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

where $\Delta \theta_{ji}^t$ is a rotation angle which determines the magnitude and direction of rotation. Fig. 1 illustrates the polar plot of the rotation gate for Q-bit individuals.

At generation $t$, the rotation angle $\Delta \theta_{ji}^t$ is updated according to the criteria summarized in Table I, where $x_{ji}^t$ and $b_{ji}^t$ are the binary control variables in solution $X_j^t$ and the best solution $B(t)$, respectively. $f(X_j^t)$ and $f(B(t))$ represent the objective function values of $X_j^t$ and $B(t)$. For example, when $x_{ji}^t$ and $b_{ji}^t$ are 0 and 1, and $f(X_j^t)$ is larger than $f(B(t))$, the rotation angle $\Delta \theta_{ji}^t$ is updated according to the following conditions:

a) if the Q-bit is in the first or third quadrant in Fig. 1, the value of $\Delta \theta_{ji}^t$ is set to a positive value or $+\theta$ to increase the probability of the state “1”;

b) if the Q-bit is in the second or fourth quadrant, the value of $\Delta \theta_{ji}^t$ is set to a negative value or $-\theta$ to increase the probability of the state “1”.

It is noted that the same lookup table can be used for the maximization problem. The details can be found in [30].

Step 10) Store the best solution into $B(t)$:
The best solution among $X(t)$ and $B(t-1)$ is stored into $B(t)$.

Step 11) Check whether the stopping conditions are met:
Terminate if the stopping conditions are met; else go to Step 6.

IV. PROPOSED APPROACH TO THE UC PROBLEM

In this paper, the economic dispatch of each UC schedule is calculated by the Lambda-iteration method to determine the optimal generation outputs of committed units. The QEA method is applied to solve the unit-scheduling problem. Representation for the UC problem, constraint handling, and the procedures of the QEA-UC method are described below.

A. Representation for the UC Problem

1) Q-Bit Individuals for the UC Problem: A population of Q-bit individuals is initialized, $Q(t) = [q_0^t, q_1^t, \ldots, q_k^t]$, where $q_j^t$ is the $j$th Q-bit individual at generation or iteration $t$; and $j = 1, 2, \ldots, n$. Here “n” is the population size. To adopt QEA to the UC problem, each Q-bit individual is given by a 2N-by-H matrix. $N$ here is the total number of units and $H$ is the total number of scheduling intervals in the scheduling horizon, $k = 1, 2, \ldots, N$ and $h = 1, 2, \ldots, H$. Thus Q-bit individual $q_j^t$ is represented by

$$q_j^t = \begin{bmatrix} \alpha_{j11}^t & \alpha_{j12}^t & \cdots & \alpha_{j1H}^t \\ \beta_{j11}^t & \beta_{j12}^t & \cdots & \beta_{j1H}^t \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{jNH}^t & \alpha_{jN2}^t & \cdots & \alpha_{jNH}^t \\ \beta_{jNH}^t & \beta_{jN2}^t & \cdots & \beta_{jNH}^t \end{bmatrix}.$$

2) Binary Solutions for Unit Schedules: $U(t)$ is a group of unit schedules, $U(t) = [U_1^t, U_2^t, \ldots, U_n^t]$, and each schedule $U_j^t$ is an N-by-H matrix. By observing $q_j^t$, a binary solution or unit schedule $U_j^t$ is formed as

$$U_j^t = \begin{bmatrix} u_{j11}^t & u_{j12}^t & \cdots & u_{j1H}^t \\ u_{j21}^t & u_{j22}^t & \cdots & u_{j2H}^t \\ \vdots & \vdots & \ddots & \vdots \\ u_{jN1}^t & u_{jN2}^t & \cdots & u_{jNH}^t \end{bmatrix}.$$

3) Variables for Unit Outputs: For the unit schedules obtained in (2) above, the Lambda-iteration economic dispatch method is used to decide the optimal generation outputs of committed units, $P(t) = [P_1^t, P_2^t, \ldots, P_n^t]$, where $P(t)$ represents...
the power generation of unit schedules at iteration \( t \). The variables for the power generation of the \( j \)th schedule \( p_{j} \) are given by

\[
p_{j} = \begin{bmatrix}
p_{j11} & p_{j12} & \cdots & p_{j1H} \\
p_{j21} & p_{j22} & \cdots & p_{j2H} \\
\vdots & \vdots & \ddots & \vdots \\
p_{jN1} & p_{jN2} & \cdots & p_{jNH}
\end{bmatrix}
\]  

(20)

where \( p_{jkh} \) represents the generation of unit \( k \) at time interval \( h \) of the \( j \)th unit schedule.

4) Evaluation Function: Since the minimization of the total operation cost is the objective of the UC problem, the objective function in (1) is used as the evaluation function of each unit schedule and the corresponding \( p_{jkh} \) is obtained the Lambda-iteration method.

B. Constraint and Over-Commitment Handling

Although the penalty-based constraint handling method is commonly used to help generating feasible unit schedules, it has no guarantee that the penalty-based constraint handling approach will produce feasible unit schedules, especially in large-scale UC problems. Moreover, with the penalty approach, there is a need to find the appropriate values for the penalty factors. For different problems, the penalty factors must be tuned again. To alleviate the problems affiliated with penalty factors, an alternative constraint handling approach is adopted here to ensure any unit schedule generated by QEA is feasible. The procedure of the constraint handling method is described as follows:

Step 1) Satisfying the spinning reserve constraints:
If the spinning reserve constraint at any scheduling interval is violated, further commitment in that interval is required and the start-up order is based on the full load average cost as in the priority list method [1]. Decommitted unit with the highest priority are selected to be online first and the further commitment procedure stops immediately when the constraint is met.

Step 2) Handling over-commitment:
Excessive generation capacity may result in expensive production cost. When the total maximum generation capacity in a scheduling interval is higher than the summation of the load demand and spinning reserve, units are selected offline in the reverse order according to their priority orders until any further decommitment will lead to deficiency in generation capacity.

Step 3) Satisfying the minimum up/down time constraint:
Extra units are committed over a period of time to observe the minimum up time constraint whenever the minimum up/down time constraint is dissatisfied.

Step 4) Improving unit schedules:
Excessive commitment may be caused by the action of Step 3, but it can be efficiently solved by repeating Steps 1 to 3. In our experiments, one iteration should be sufficient to produce potential solutions that meet constraints (5), (7), and (8).

Step 5) Examining unit schedules:
For every schedule, examine whether or not the generation capacity in all scheduling intervals is adequate. If not, a new schedule is created by observing the corresponding Q-bit individual, and repeats the above steps until a feasible schedule is produced.

C. Procedure of the QEA-UC Method

The procedure of the proposed QEA-UC approach is presented in the following steps and the corresponding flowchart is provided in Fig. 2.

Step 1) Set the generation counter \( t = 0 \).
Step 2) Initialize a group of Q-bit individuals with all \( \alpha \) and \( \beta \) values equal to \( 1/\sqrt{2} \).
Step 3) Determine unit schedules by observing the states of Q-bit individuals.
Step 4) Improve the obtained unit schedules by commitment handling method in Section IV-B.
Step 5) Determine the cost of the schedule by determining the optimal economic dispatch of the units in each schedule by the Lambda-iteration method.
Step 6) If \( t = 0 \), then go to Step 8.
Step 7) Update Q-bit individuals by using Q-gates.
Step 8) Compare the costs of the schedules and store the best solution schedule.
Step 9) Set \( t = t + 1 \).
Step 10) Terminate if \( t \) is larger than the maximum number of generations; else go to Step 3.

Fig. 2. Flowchart of the QEA-UC method.
V. NUMERICAL RESULTS

The proposed QEA-UC method is tested on systems with the number of units in the range of 10 to 100 and considering a 24-h scheduling horizon. The ten-unit system data and load demands are given in [11]. The 20-, 40-, 60-, 80- and 100-unit data are obtained by duplicating the ten-unit case and adjusting the load demands in proportion to the size of the system. The spinning reserve requirements are assumed to be 10% of the load demand. For each test case, totally 30 trial runs are performed to verify the robustness of the QEA-UC method. The proposed QEA-UC method has been developed based on MATLAB and executed on a computer with Intel Core of 2.39 GHz and 1.99 GB RAM.

In this section, parameter sensitivity analysis is first performed. Case studies on the performance of the QEA-UC method on different test systems are then reported. The results obtained are compared with some published methods in the literatures. Finally, the proposed method is extended to solve a large-scale UC problem in which 100 units are scheduled over a seven-day horizon with unit ramp-rate limits considered.

A. Parameter Sensitivity Analysis

The effects of the magnitude of rotation angle in radians and the population size are studied on the ten-unit test system. The maximum number of generations is set to 100 for the parameter sensitivity tests.

1) Determination of the Rotation Angle: The value of $\theta$ is problem-dependent [30], [32]. In this study, the population size is set to 4 and the values of $\theta$ from $0.005\pi$ to $0.05\pi$ with a step size of $0.005\pi$ are examined. The results are tabulated in Table II. It can be observed that the QEA-UC method is sensitive to the magnitude of $\theta$. Since large angles may cause premature convergence, small angles generally produce better solutions. The performance is the best when $\theta = 0.02\pi$.

2) Determination of the Population Size: The effects of the population size are investigated by varying the size from 2 to 30 with a step size of 2, and $\theta = 0.02\pi$. The results are shown in Table III and Fig. 3.

Noticeably, in Table III, the best solutions are the same for the use of different population sizes. A large population size can slightly improve the mean value of the solutions, but it increases the computational time. From the size of 16 to 30, the improvement of the solutions is not pronounced, but the computational time increases linearly as shown in Fig. 3. Although the population size is problem-dependent, the size of 18 is the best choice in compromise between computational time and solutions. It is also agreed with the range of population size from 10 to 30 suggested in [32].

B. Case Studies

The performance of the proposed QEA-UC method is tested on the system with the number of units from 10 to 100. Based on the selected settings in Section V-A. (i.e., the population size equal to 18 and $\theta = 0.02\pi$) and the maximum number of generations equal to 100, the results of different case studies are obtained and tabulated in Table IV. The population size of 4 is also included for comparison. Obviously, the QEA-UC method with the population size of 18 outperforms the one with the size equal to 4 in terms of the best, mean as well as worst costs, and the standard deviation of the results on the systems with different problem scales. As shown in Table V, when the maximum number of generations is set to 200, the better solutions can be obtained, especially for small population and large system sizes, but longer computation is needed. The average cost convergence curves of the QEA-UC method for different cases are presented in Fig. 4. It can be observed that the convergence behavior of the QEA-UC method is very smooth. For the ten-unit system, the convergence curves become nearly steady after 100 generations so the improvement of solutions are not significant for
Fig. 4. Average cost convergence curves of the proposed QEA-UC method with different population sizes (PopSize) for different sizes of test systems. (a) Ten-Unit-Case, (b) 20-Unit-Case, (c) 40-Unit-Case, (d) 60-Unit-Case, (e) 80-Unit-Case, (f) 100-Unit-Case.

TABLE IV
RESULTS OF PROPOSED METHOD WITH MAXIMUM NUMBER OF GENERATIONS SET TO 100

<table>
<thead>
<tr>
<th>No. of Units</th>
<th>Population Size</th>
<th>Maximum Generation</th>
<th>Mean Time (s)</th>
<th>Best ($)</th>
<th>Mean ($)</th>
<th>Worst ($)</th>
<th>S.D.</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4</td>
<td>100</td>
<td>2.18</td>
<td>563,938</td>
<td>564,289</td>
<td>564,714</td>
<td>294</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td></td>
<td>9.62</td>
<td>563,938</td>
<td>563,994</td>
<td>564,672</td>
<td>137</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>100</td>
<td>3.05</td>
<td>1,124,244</td>
<td>1,125,777</td>
<td>1,126,672</td>
<td>551</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td></td>
<td>13.71</td>
<td>1,123,933</td>
<td>1,125,048</td>
<td>1,125,926</td>
<td>439</td>
</tr>
<tr>
<td>40</td>
<td>4</td>
<td>100</td>
<td>4.42</td>
<td>2,247,036</td>
<td>2,248,207</td>
<td>2,249,702</td>
<td>801</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td></td>
<td>19.83</td>
<td>2,246,381</td>
<td>2,247,154</td>
<td>2,249,242</td>
<td>560</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>100</td>
<td>5.82</td>
<td>3,369,351</td>
<td>3,371,039</td>
<td>3,373,062</td>
<td>917</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td></td>
<td>26.23</td>
<td>3,367,186</td>
<td>3,369,203</td>
<td>3,370,699</td>
<td>883</td>
</tr>
<tr>
<td>80</td>
<td>4</td>
<td>100</td>
<td>7.30</td>
<td>4,491,896</td>
<td>4,494,710</td>
<td>4,497,278</td>
<td>1,621</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td></td>
<td>32.78</td>
<td>4,490,537</td>
<td>4,491,903</td>
<td>4,494,269</td>
<td>895</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
<td>100</td>
<td>8.75</td>
<td>5,615,242</td>
<td>5,618,308</td>
<td>5,622,678</td>
<td>1,776</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td></td>
<td>39.24</td>
<td>5,616,696</td>
<td>5,614,434</td>
<td>5,616,478</td>
<td>1,203</td>
</tr>
</tbody>
</table>

the use of higher maximum number of generation. For 100-unit system, the solution can be further improved after 100 generations so higher number of generation can achieve a solution with lower cost. However, the curve becomes nearly steady after 200 generations and further increment of the number of generation only improves the solution slightly. A compromise between them can be considered because of the linear relationship between the computational time and the number of generation.

C. Comparison of Results Among Various Approaches

Table VI summarizes the study results on the test systems in the last section above obtained by the proposed QEA-UC method and other methods including LR [34], GA [11], EP [17], HPSO [18], SA [16], and GAUC [20]. For the 10-unit test case, the best UC schedule obtained by the QEA-UC method is given in Table VII. In Table VI, for the QEA-UC method, the results obtained with both population size of 4 and 18 are tabulated to show further the effects of the population size. In addition, the results obtained for both the maximum number of generations of 100 and 200 are summarized for the same reason.

In Table VI, it can be observed that the solutions of the QEA-UC method are more attractive than those obtained by other techniques. Besides, it can be observed that the population size and maximum number of generations required by
the QEA-UC method are much smaller than that of the other methods in all the study cases. Although the mean times consumed by various approaches cannot be directly compared due to different computing machines used by other researchers, it can still be able to indicate that the computational time of the QEA-UC method increases linearly with the system size while that of other techniques increases dramatically. This efficient characteristic of the QEA-UC method indicates that QEA-UC has large capability in solving large-scale UC problems. Even for the case of population size of 4 and the maximum generation of 100, QEA-UC finds better solution than all the other methods considered and with a very short computational time. As expected, with the population size of 18 and maximum generation of 200, a much better solution is found by the proposed method. Owing to the linear relationship between the computational time and the system size, the superiority of the proposed QEA-UC method over the other methods considered...
in terms of solution quality and computation time is more significant in the 100-unit system.

D. Unit Commitment Considering Ramp-Rate Limits in a Seven-Day Horizon

The QEA-UC method is now further extended to solve a UC problem with ramp-rate limits. The ramp-rate limits can be easily handled using the ramp-rate limit handling method in [33] by constraining all online units to operate within their feasible output limits. The details can be found in [33]. In this study, 100 units are scheduled over a horizon of seven days. The up-ramp/down-ramp limits of unit 1 to unit 10 of the 10-unit system [11] are set to 160, 160, 100, 100, 100, 60, 60, 40, 40, and 40 in MW, respectively. The same limiting values in [11] are set to 160, 160, 100, 100, 100, 60, 60, 40, 40, and 40 in MW, respectively. The same limiting values for the corresponding units are assumed in the 100-unit study case. The population size and maximum number of generations of the proposed method are 18 and 200. Table VIII and Fig. 5 present the scheduling results obtained either with or without unit ramp-rate limits. It is obvious that the generating cost is increased due to the incorporation of unit ramp-rate characteris- tics in the UC problem. The effectiveness of the proposed method is again observed in terms of the speed of convergence and the computational time. The results illustrate the power and the feasibility of the proposed QEA algorithm in solving large-scale UC problems.

VI. CONCLUSION

This paper has introduced a quantum-inspired evolutionary algorithm and, based on it, established a new method, QEA-UC for solving the unit commitment problem. The effectiveness and validity of the QEA-UC algorithm have been demonstrated through its applications to test systems with the number of units from 10 to 100. It has been found that the QEA-UC algorithm is very powerful and efficient and it outperforms many other methods. The QEA-UC algorithm can perform well even with a small population size, and has been found to have a linear relationship between the scale of the UC problem and computational time. Moreover, the proposed algorithm has been successfully applied to solve a large-scale UC problem in which 100 units have been scheduled over a seven-day horizon with unit ramp-rate limits considered. The QEA-UC algorithm is therefore very promising and has a large potential to be applied to large-scale UC problems.

REFERENCES

K. P. Wong (M’87–SM’90–F’02) received the M.Sc. and Ph.D. degrees and the higher doctorate D.Eng. degree from the University of Manchester Institute of Science and Technology, Manchester, U.K., in 1972, 1974, and 2001, respectively.

He was with The University of Western Australia, Perth, Australia, from 1974 to 2002, where he is now an Adjunct Professor. Since 2002, he has been with the Department of Electrical Engineering, The Hong Kong Polytechnic University, as a Chair Professor. He has published numerous research papers in power systems and in the applications of artificial intelligence and evolutionary computation to power system planning and operations. His current research interests include evolutionary optimization in power, power market analysis, power system planning and operation in deregulated environment, and power quality.

Prof. Wong received three Sir John Madsen Medals (1981, 1982, and 1988) from the Institution of Engineers Australia, the 1999 Outstanding Engineer Award from IEEE Power Chapter Western Australia, and the 2000 IEEE Third Millennium Award. He was Editor-in-Chief of *IEEE Proceedings GTD* and was General Chairman of IEEE/CSEE PowerCon2000 and IEE APSCOM2003 conferences. He is a Fellow of IET, HKIE, and IEAust.

T. S. Chung received the B.Eng. (Hons.) degree in electrical engineering from The University of Hong Kong, Hong Kong, China, the M.Sc. degree from Imperial College of Science and Technology, London, U.K., and the Ph.D. degree from the University of Strathclyde, Glasgow, U.K.

Currently, he is a Professor and Associate Head of the Department of Electrical Engineering, The Hong Kong Polytechnic University. His research interests are in power system security analysis, stability control, state estimation, artificial intelligence, and optimization.

S. L. Ho received the B.Sc. and Ph.D. degrees in electrical engineering from the University of Warwick, Warwick, U.K., in 1976 and 1979, respectively. He is now a Chair Professor in electrical utilization and the Head of the Department of Electrical Engineering, Hong Kong Polytechnic University, Hong Kong, China. His main research interests include traction engineering, the application of finite elements in electrical machines, phantom loading of machines, and optimization of electromagnetic devices.