Taguchi Method for Solving the Economic Dispatch Problem With Nonsmooth Cost Functions

Derong Liu, *Fellow, IEEE,* and Ying Cai, *Student Member, IEEE*

***Abstract*—This paper presents a new algorithm that applies the Taguchi method to solve the economic dispatch problem with nonsmooth cost functions. In our approach, we employ the Taguchi method that involves the use of orthogonal arrays in estimating the gradient of the cost function. The Taguchi method has been widely used in experimental designs for problems with multiple parameters where the optimization of a cost function is required. The use of the Taguchi method for the economic dispatch problem is a novel idea, and it leads to efficient algorithms that can find a satisfactory solution by minimizing the cost function in a few iterations. Simulation results show that the Taguchi method is less sensitive to initial values of parameters and is more effective than other previously developed algorithms. In addition, our algorithm is suitable for parallel implementations.**

***Index Terms*—Economic dispatch (ED) problem, nonsmooth cost function, orthogonal array, Taguchi method.**

NOMENCLATURE

N	Number of generator units.
J	Total generation cost.
P_D	Total load demand.
P_L	Total transmission loss.
F_i	Cost function of unit i.
a_i, b_i, c_i	Fuel cost coefficients of unit i.
e_i, f_i	Fuel cost coefficients of unit i with valve points.
P_i	Power assigned to unit i.
$P_{i,\min}$	Minimum power generation of unit i.
$P_{i,\max}$	Maximum power generation of unit i.
$P_{i,\text{ref}}$	Average of $P_{i,\max}$ and $P_{i,\min}$.
α_i	Relative contribution of unit i to P_D.
$\alpha_i^{(j)}$	jth level of factor α_i; typically, $j = 1, 2, 3$.
$L_M(q^m)$	Orthogonal array that has M rows, m columns, and q levels, e.g., $L_9(3^4)$.
K	Index set of saturated factors.
$\mathcal{I}$	Index set of nonsaturated factors.
J_i	Cost of the ith test.
$V_k^{(j)}$	Total contribution of the jth level of the kth factor to the cost function.
η	Shrinking coefficient.

Manuscript received January 19, 2005; revised April 19, 2005. This work was supported in part by the Open Research Project under Grant ORP-0501 from KLCSIS-IA-CAS. Paper no. TPWRS-00020-2005.

The authors are with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: dliu@ece.uic.edu; ycai@ece.uic.edu).

Digital Object Identifier 10.1109/TPWRS.2005.857939

I. INTRODUCTION

THE Taguchi method of experimental design has been widely used in industry for the purpose of finding factors that are most important in achieving useful goals in a manufacturing process [5], [6], [12], [18]. Several factors that are related to the goals and are under the user's control are selected. These factors are varied over two or more levels in a systematic manner. Experiments are then designed according to an orthogonal array to show the effects of each potential primary factor. The Taguchi method involves an analysis that reveals which of the factors are most effective in reaching the goals and the directions in which these factors should be adjusted to improve the results. The control over achieving the goals will be best obtained by changes in these primary factors in the direction indicated by the analysis. The present paper applies the Taguchi method to solve the economic dispatch (ED) problem with nonsmooth cost functions.

The ED problem is to determine the optimal combination of power outputs of all generating units to minimize the total fuel cost while satisfying the load demand and operational constraints. Over the past few years, a number of approaches have been developed for solving the ED problem using the lambda iteration method [1], quadratic programming [3], and the gradient method [13]. In these numerical methods for solving the ED problem, an essential assumption is that the incremental cost curves of the units are piecewise-linear monotonically increasing functions. Unfortunately, the input-output characteristics of modern power generating units are inherently highly nonlinear because of valve-point loadings, multi-fuel effects, etc. Furthermore, they may lead to multiple local minimum points of the cost function. Classical dispatch algorithms require that these characteristics be approximated, even though such approximations are not desirable as they may lead to suboptimal solutions and hence huge revenue losses over time.

In view of the nonlinear characteristics of the power generating units, there is a demand for techniques that do not have restrictions on the shape of the fuel-cost curves. Classical calculus-based techniques fail to address this type of problems satisfactorily. Dynamic programming is a method that can solve the ED problem without imposing any restrictions on the nature of the cost curves [9]. However, this method suffers from the "curse of dimensionality" leading to high computational cost despite the use of zoom feature [14], [19], or the method may lead to local optimal solutions when avoiding the problem of "curse of dimensionality" [9], [10].

In order to make numerical methods more convenient for solving the ED problem, artificial intelligence techniques, such as the Hopfield neural networks, have been successfully employed to solve the ED problem as a nonsmooth optimization problem [8], [15]. A global optimization technique known as the genetic algorithm (GA), which is a form of probabilistic heuristic algorithms, has also been successfully applied to solve the ED problem [21]. However, recent research has identified some deficiencies in the performance of GA. The premature convergence of GA degrades its performance and reduces its search capability that leads to a higher probability toward local minimum solutions [4].

The goal of the present paper is to develop a method with reduced complexity for solving the ED problem with nonsmooth cost functions by employing the Taguchi method based on orthogonal arrays. We recursively minimize the cost function while satisfying the operating constraints. It turns out that most of the conventional methods involve a large amount of computation and solution time. In our approach, the objective function is optimized using the Taguchi method, in which only a very limited amount of computation is needed, assuming the knowledge of each unit's working range, which is always available. We believe that the present paper is the first systematic investigation into the Taguchi experimental approach for the ED problem. Overall, the present Taguchi algorithm will be shown to provide shorter solution time and has better performance than many existing algorithms including evolutionary programming [17], [19], [23], Hopfield neural networks [8], [15], hierarchical numerical method [11], and particle swarm optimization [16].

This paper is organized as follows. In Section II, the problem statement is presented. In Section III, the Taguchi method is described. In Section IV, an algorithm for solving the ED problem based on the Taguchi method is developed. In Section V, simulation results are presented that demonstrate the potential of the present algorithm. Finally, in Section VI, several pertinent remarks are given to conclude the present paper.

II. Problem Statement

A. ED Problem With Smooth Cost Functions

To solve the standard ED problem, consider the operation of a power system with N units, each loaded to P_i MW, to satisfy a total load demand P_D (including total transmission loss P_L).

The objective function for each unit can be represented by a quadratic cost function given by

$$F_i(P_i) = a_i + b_iP_i + c_iP_i^2 \tag{1}$$

where a_i, b_i, and c_i are the fuel consumption cost coefficients of unit i, and P_i represents the value of the power to be determined for unit i.

B. ED Problem With Nonsmooth Cost Functions

In reality, the objective function of the ED problem has nondifferentiable points due to valve-point effects and change of fuels. Therefore, the objective function is composed of a set of nonsmooth cost functions. In this paper, two cases of nonsmooth cost functions are considered. One is the case with the valve-point loading problem where the objective function is generally described as a superposition of sinusoidal functions and quadratic functions. The other is the case with multiple fuels where the objective function is expressed as a piecewise quadratic cost function. In both cases, the ED problem has multiple minimum points.

1) *Nonsmooth cost functions with valve-point effects:* Approximations using smooth quadratic functions provide the basis for most classical ED problems where the valve point effects are ignored. Such approximations have resulted in inaccuracies in the resulting dispatch. Methods that avoid such approximations of the actual unit data curves without sacrificing computational time would prove very valuable. To model the effects of valve points, a recurring rectified sinusoid term is added to the input-output equation. Typically, the fuel-cost function considering valve-point loadings of a generating unit is given by

$$F_i(P_i) = a_i + b_iP_i + c_iP_i^2 + |e_i \sin(f_i(P_{i,\min} - P_i))| \tag{2}$$

where a_i, b_i, and c_i are the fuel consumption cost coefficients of the ith unit, and e_i and f_i are the fuel cost coefficients of the ith unit with valve-point effects.

2) *Nonsmooth cost functions with multiple fuels:* Traditionally, the cost function of each generator has been approximately represented by a single quadratic cost function. In practice, operating conditions of many generating units require that the generation cost function be segmented into piecewise quadratic functions. Therefore, it is more realistic to represent the generation cost function as a piecewise quadratic cost function [16]. Generally, a piecewise quadratic function is used to represent the input-output curve of a generator with multiple fuels [16]. The piecewise quadratic function is given by

$$F_i(P_i) = \begin{cases} a_{i,1} + b_{i,1}P_i + c_{i,1}P_i^2, & \text{if } P_{i,\min} \le P_i < P_{i,1} \\ a_{i,2} + b_{i,2}P_i + c_{i,2}P_i^2, & \text{if } P_{i,1} \le P_i < P_{i,2} \\ \vdots & \vdots \\ a_{i,n} + b_{i,n}P_i + c_{i,n}P_i^2, & \text{if } P_{i,n-1} \le P_i \le P_{i,\max} \end{cases} \tag{3}$$

where $a_{i,j}$, $b_{i,j}$, and $c_{i,j}$ are the cost coefficients of generator i for the fuel type j, and $P_{i,\min}$ and $P_{i,\max}$ are the minimum and maximum power generation of unit i.

Both cases of nonsmooth cost functions for the ED problem will be considered in the present paper.

The ED problem minimizes the total cost given by

$$J = \sum_{i=1}^{N} F_i(P_i). \tag{4}$$

The goal is to determine P_i, $i = 1, 2, \ldots, N$, so that the cost function in (4) is minimized subject to the following two constraints.

1) The sum of all P_i should be equal to the total load demand plus total transmission loss, i.e.,

$$\sum_{i=1}^{N} P_i = P_D + P_L. \tag{5}$$

TABLE I
INITIAL VALUES OF THE FOUR FACTORS

Factor	level 1	level 2	level 3
p_1	0.2	0.5	0.8
p_2	0.2	0.5	0.8
p_3	0.2	0.5	0.8
p_4	0.2	0.5	0.8

Without loss of generality, the transmission loss is not considered in this paper for simplicity (i.e., we can assume that $P_L = 0$) [16].

2) The operational constraints for unit i is given by

$$P_{i,\min} \leq P_i \leq P_{i,\max}. \tag{6}$$

III. INTRODUCTION TO THE TAGUCHI METHOD

For parameter optimization problems with a given computable objective function, the Taguchi method of experimental design [12] is a suitable method that can rapidly optimize the varying factors to get a desired outcome. Since the goal of this research is to minimize a cost function, it is appropriate to employ the Taguchi method. In the following, the Taguchi method will be described.

Suppose that an experimental outcome J is a cost function of several variables, $p_1, p_2, \ldots, p_m$, whose values can be controlled. We write $J = J(p_1, p_2, \ldots, p_m)$. The controlled variables p_k, $k = 1, \ldots, m$, are called factors. The goal is to find the optimal values $\hat{p}_k$, $k = 1, \ldots, m$, to minimize the cost function J. This can be done by varying the factors simultaneously in a disciplined manner and recording the corresponding values of J until we get the optimal $\hat{p}_k$, $k = 1, \ldots, m$. The Taguchi method involves a disciplined method of varying two or more factors simultaneously.

In a full experimental design, all possible combinations of the values of factors must be tried. In a fractional design, such as the Taguchi method, a subset of the possible value combinations is used. To reduce the time consumed in conducting experiments while taking advantage of the performance of full factorial method, the Taguchi method based on orthogonal arrays is introduced. It is a method of setting up experiments that only require a fraction of the full factorial combinations. The experiment combinations are chosen to provide sufficient information to determine the effects of each factor.

We illustrate next an example of design optimization involving four factors. The four factors are denoted by p_1, p_2, p_3, and p_4, and Table I gives the three initial values (which are called levels) for each of the four factors. Generally speaking, these initial values are selected randomly in an ascending order; that is, level 1 < level 2 < level 3.

We will use the orthogonal array shown in Table II for purpose of demonstration. In the present example, each factor has three different levels, and they are denoted by $p_k^{(1)} = 0.3$, $p_k^{(2)} = 0.5$, and $p_k^{(3)} = 0.8$, for $k = 1, \ldots, 4$. If we use the full factorial method to discover the optimal combination of these factors, we need to conduct $3^4 = 81$ tests, whereas the orthogonal array $L_9(3^4)$ in Table II allows us to set up experiments with only nine tests. The orthogonal array in Table II is in the form of $L_M(q^m)$, where q is the number of levels each factor has, m is the maximum number of factors the table can handle, and M is the total number of tests required using this table. In general, M is much smaller than the value of q^m, which is the total number of combinations for m factors with each having q levels (choices). A cycle in the present Taguchi method is defined as a complete set of tests according to the orthogonal array, consisting of a total of M tests. In an orthogonal array (e.g., Table II), the numbers under each factor in a test indicate the level of that factor to be used in the tests. For example, in test number 4, we would use $p_1^{(2)}$-level 2 of p_1, $p_2^{(1)}$-level 1 of p_2, $p_3^{(2)}$-level 2 of p_3, and $p_4^{(3)}$-level 3 of p_4. Orthogonal arrays are readily composed and are available from many texts (e.g., [5], [6], [20]). The way that they are constructed [6] is to have 1) each level of every factor appear the same number of times in every column of the array (e.g., three times in Table II), and 2) each combination of factors between any two columns, i.e., each $(i, j), i, j = 1, 2, 3$, appears the same number of times (e.g., each pair (i, j) between every two columns in Table II appear one time). For any given values of q (number of levels) and m (number of factors), M (number of tests) is determined as the smallest number that satisfies 1) and 2) above. Typical values of q in the orthogonal arrays found in [5], [6], and [20] are 2 and 3. The number of columns of an orthogonal array indicates how many factors a table can handle. For example, $L_9(3^4)$ has 4 columns and it can handle up to 4 factors, i.e., it can handle 1, 2, 3, or 4 factors. For a given problem with certain number of factors, there may not exist an orthogonal array that has exactly the same number of columns as the number of factors in the problem. In this case, we can choose an orthogonal array that has a few more columns than the number of factors in the problem, and thus not all columns of the chosen orthogonal array will be used. In Example 1 of Section V of the paper where we use $L_9(3^4)$, we have 3 factors and thus we will only use 3 of the 4 columns of the array $L_9(3^4)$. There are also orthogonal arrays that allow different number of levels for different factors in the problem. In the present paper, we use 3 levels for all factors in our illustration and experiments.

TABLE II
ORTHOGONAL ARRAY $L_9(3^4)$

Test number	p_1	p_2	p_3	p_4	Cost
1	1	1	1	1	J_1
2	1	2	2	2	J_2
3	1	3	3	3	J_3
4	2	1	2	3	J_4
5	2	2	3	1	J_5
6	2	3	1	2	J_6
7	3	1	3	2	J_7
8	3	2	1	3	J_8
9	3	3	2	1	J_9
Contributions of level 1	$V_1^{(1)}$	$V_2^{(1)}$	$V_3^{(1)}$	$V_4^{(1)}$	
Contributions of level 2	$V_1^{(2)}$	$V_2^{(2)}$	$V_3^{(2)}$	$V_4^{(2)}$	
Contributions of level 3	$V_1^{(3)}$	$V_2^{(3)}$	$V_3^{(3)}$	$V_4^{(3)}$	

Using the orthogonal array $L_9(3^4)$, each cycle consists of nine individual tests. After each cycle of tests, a minimum cost can be found. While this cost may not be the optimal cost, more cycles of tests are needed until the minimum cost of each cycle converges.

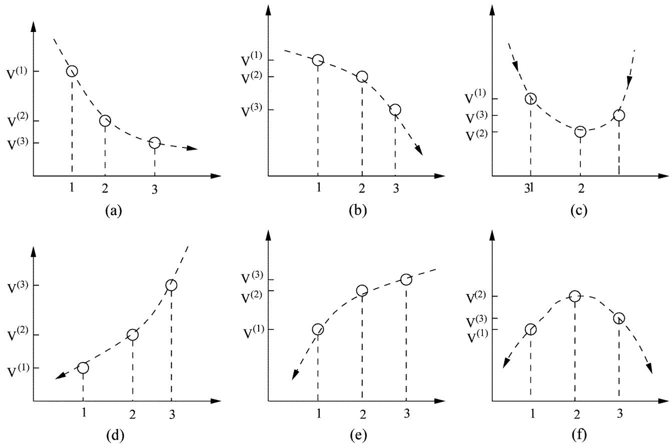


Fig. 1. Six different trends of the cost function.

In the present example with four factors, after each cycle of tests, we perform an analysis to determine the trend of the cost function for each factor. The values of the cost function from the nine tests are calculated and denoted by $J_i, i = 1, 2, \ldots, 9$. For each of the four factors, we calculate the total contribution of each level to the cost function, $V_k^{(j)}$, as the sum of the cost values corresponding to the tests involving that particular level. For example, after nine tests are completed, for factor p_3, we calculate

$$\begin{aligned} V_3^{(1)} &= J_1 + J_6 + J_8 \\ V_3^{(2)} &= J_2 + J_4 + J_9 \\ V_3^{(3)} &= J_3 + J_5 + J_7 \end{aligned} \tag{7}$$

where $V_k^{(j)}$ indicates the total contribution of the jth level of the factor p_k to the cost function. $V_3^{(1)}$ is the summation of J_1, J_6, and J_8 since the three tests involving the first level of p_3 are test numbers 1, 6, and 8. We will then have three total contributions that correspond to the three levels for each factor calculated according to Table II. These three total contributions can be plotted versus the three levels for each factor to determine the trend of the cost function as shown in Fig. 1. In the figure, the numbers "1," "2," "3" along the horizontal axis represent the three levels of each factor, and $V^{(1)}$, $V^{(2)}$, and $V^{(3)}$ are the total contributions of each level of a factor.

From these figures for the trend of the cost function, we know whether we need to increase or decrease the values for each factor. If the trends of the cost function are as shown in Fig. 1(a) or (b), that means the value of this factor should be increased in order to further reduce the value of the cost function. In this case, we can choose a step size (e.g., 0.01) and increase all three levels of every factor by the chosen step size. Likewise, for trends as shown in Fig. 1(d) or (e), the parameter values should be decreased. Alternatively, we can also use the estimated gradient information to determine the direction for each factor to move in and the amount to adjust. The gradient of the cost function J with respect to the factor p_k can be estimated from the experiments using [2]

$$\nabla_k J = \frac{3}{2M}\left(V_k^{(3)} - V_k^{(1)}\right)\frac{1}{\delta} \tag{8}$$

where $M = 9$ and $\delta = 0.3$ in the present example. This indicates that if the trends of cost function from the experiments are given as in Fig. 1(a), (b), (d), or (e), we have a very good estimate of the gradient of the cost function. Note that to minimize a cost function, the key is to determine its gradient with respect to varying parameters. If the trend is as shown in Fig. 1(c), that means the parameter value should be set closer to the middle level or the center of the parabolic curve. In case of Fig. 1(f), we can randomly select a direction, which means either increase or decrease the parameter value. The analysis based on Fig. 1(c) and (f) implies that using three levels for each parameter in the experiments will provide better results than using two levels as in [22]. If two levels for each parameter are used in the experiments, for cases as shown in Fig. 1(c) and (f), the decisions about the next move in the parameter space will often be incorrect. This is especially true for the case as shown in Fig. 1(c). The analysis in Fig. 1(c) of the experimental results indicates that we should stay in the neighborhood of the current value and shrink the interval of search toward the center of the parabolic curve. From the three points shown in Fig. 1(c), we can fit a parabolic function between the cost values and the factor values using

$$V = \alpha x^2 + \beta x + \gamma. \tag{9}$$

The coefficients α, β, and γ can easily be determined by plugging the values of $(x^{(1)}, V^{(1)})$, $(x^{(2)}, V^{(2)})$, and $(x^{(3)}, V^{(3)})$ in the above expression, where $x^{(i)}$ is the value of level i in Fig. 1(c). The center of the fitted parabolic curve will be chosen as the new $x^{(2)}$, i.e., $x_{\text{new}}^{(2)} = -\beta/(2\alpha)$. The new values for level 1 and level 3 are chosen as

$$x_{\text{new}}^{(1)} = x_{\text{new}}^{(2)} - \frac{1}{2}\eta\left(x^{(3)} - x^{(1)}\right) \tag{10}$$

and

$$x_{\text{new}}^{(3)} = x_{\text{new}}^{(2)} + \frac{1}{2}\eta\left(x^{(3)} - x^{(1)}\right) \tag{11}$$

where η is a shrinking coefficient (e.g., $\eta = 0.9$).

According to the trend determined for each factor, we choose a new set of three initial levels for each factor, and we start a new cycle and perform again nine tests according to Table II. The procedure is repeated until the value of the cost function is converged. To determine the convergence of the present iterative algorithm, we compute the minimum cost in each cycle. Define

$$J_{\min}(k) = \min_{i=1,\ldots,M}\{J_i(k)\} \tag{12}$$

where M is the number of tests in each cycle according to the orthogonal array, and $J_i(k)$ is the cost for the ith test in the kth cycle. The difference between the $J_{\min}$ of two consecutive cycles of the Taguchi method will be used to decide whether to stop the algorithm. In particular, the convergence criterion can be chosen as $J_{\min}(k-1) - J_{\min}(k) \leq 10^{-2}$, i.e., when the improvement of the cost function from one cycle to the next is less than 10^{-2}, we stop the algorithm. The convergence criterion 10^{-2} is chosen such that it is smaller than 10^{-4} times the mean value of the cost function. Such a choice of the convergence criterion has been shown to be a good choice in our simulation studies. Clearly, such a procedure will achieve the minimization of the cost function $J(p_1, p_2, p_3, p_4)$ through repeated cycles of tests and analysis according to Table II and Fig. 1. Such a method based on orthogonal arrays is usually referred to as the Taguchi method [5], [12], [18], which we will use for solving the economic dispatch problem with nonsmooth cost functions.

IV. Taguchi Method for Solving the ED Problem

In this section, the economic dispatch problem with nonsmooth cost functions is solved using the Taguchi method.

The cost function is defined in (4), with $F(P_i)$ given in (2) or (3). The objective is to find the power vector $P = [P_1, P_2, \ldots, P_N]$, which minimizes the cost function (4), while satisfying the constraints defined by (5) and (6).

Since the range of each unit's power P_i, $i = 1, \ldots, N$, is known, we can use the Taguchi method to obtain a solution that minimizes the cost function in (4). The function in (4) is therefore chosen as our cost function in the Taguchi method described earlier. The number of factors to be determined depends on the number of units N. We will formulate a new vector $[\alpha_1, \ldots, \alpha_N]$ as the factors to be optimized by the Taguchi method instead of the vector $[P_1, \ldots, P_N]$. Here, α_i, $i = 1, \ldots, N$, stands for the relative contribution with respect to the grand total of load demand P_D. Thus, the cost function for each test k can be written as

$$J_k = F_1(P_1) + F_2(P_2) + \cdots + F_N(P_N) \tag{13}$$

where for each i, $i = 1, 2, \ldots, N$

$$P_i = \xi_k \alpha_i P_{i,\mathrm{ref}}, \text{ if } P_{i,\min} \le \xi_k \alpha_i P_{i,\mathrm{ref}} \le P_{i,\max} \tag{14}$$

$$P_i = P_{i,\max}, \text{ if } \xi_k \alpha_i P_{i,\mathrm{ref}} \ge P_{i,\max} \tag{15}$$

$$P_i = P_{i,\min}, \text{ if } \xi_k \alpha_i P_{i,\mathrm{ref}} \le P_{i,\min} \tag{16}$$

where $P_{i,\mathrm{ref}} = 0.5 \times (P_{i,\min} + P_{i,\max})$ and ξ_k is determined such that the constraint (5) is satisfied. Note that here we use $P_{i,\mathrm{ref}}$, which is the average of $P_{i,\min}$ and $P_{i,\max}$, as the reference point for scaling. In this way, we can control that $\sum_{i=1}^{N} P_i$ is equal to the total demand P_D. Factors that are determined by (15) or (16) are said to be saturated. In order to make sure that the constraints given by (6) are satisfied for each P_i and the constraint for the total power demand given by (5) is also satisfied, some iterations may be involved to obtain the final value of each P_i. We will describe next a procedure to explain how to obtain the final value of each P_i. We then repeat experiments after adjusting the three levels according to the trend of the cost function, until a satisfactory solution is obtained, which corresponds to the case where one of the three levels is close to or equal to the optimal solution.

The search procedure of the Taguchi method for the present economic dispatch problem is summarized as follows.

1) Determine the reference point according to the lower and upper bounds of generation powers for each unit. Without loss of generality, set the same initial values α_i^1, α_i^2, and α_i^3 for each unit, $i = 1, 2, \ldots, N$.

2) For $k = 1, 2, \ldots, M$, loop Steps 2.1–2.5.

2.1) The index of factors α_i, $i = 1, 2, \ldots, N$, will be divided into two sets: The set $\mathcal{I}$ of nonsaturated factors [that are determined by **(14)**] and K of saturated factors [that are determined by **(15) or (16)**]. Initially, we set $\mathcal{I} = \{1, 2, \ldots, N\}$ and $K = \emptyset$. This step will be repeated for each k.

2.2) For $i = 1, 2, \ldots, N$, choose $\alpha_i = \alpha_i^{(j)}$, where j = 1, 2, 3 is determined according to the orthogonal array. Set the initial value of ξ_k as

$$\xi_k = \frac{P_D}{\sum_{i=1}^{N} \alpha_i P_{i,\mathrm{ref}}}. \tag{17}$$

2.3) For $i = 1, 2, \ldots, N$, determine P_i according to **(14), (15) or (16)**, and update the sets $\mathcal{I}$ and K. If no update to $\mathcal{I}$ and K is needed, go to Step 2.5). Otherwise, go to Step 2.4).

2.4) If $\mathcal{I} \neq \emptyset$, calculate ξ_k using

$$\xi_k = \frac{P_D - \sum_{l \in K} P_l}{\sum_{i \in \mathcal{I}} \alpha_i P_{i,\mathrm{ref}}} \tag{18}$$

in order to satisfy the constraint **(5)**. Go back to Step 2.3). If $\mathcal{I} = \emptyset$, set a very large value for J_k (e.g., $J_k = 1.5 \times \max_{1 \le i \le M}\{J_i\}$) since the constraint **(5)** may not be satisfied in this case. Go to Step 2.1) for the next k.

2.5) Determine the cost J_k according to **(13)** with the factors P_i determined from Step 2.3). Go to Step 2.1) for the next k.

3) Check the difference between the minimum cost of the previous cycle and the current cycle. If it is less than 10^{-2}, go to Step 4). Otherwise, determine the trend of the cost function, adjust the values of the factors α_i^j ($i = 1, 2, \ldots, N, j = 1, 2, 3$) according to the trends, and go back to Step 2) to start a new cycle.

4) Find the factors that give the smallest cost from the orthogonal array. The values of these factors will give the optimal powers of all generating units with the minimum total cost.

As with most experimental methods for optimization, there is a concern that the present Taguchi method may also get stuck in local minima in some cases. To reduce the chance of getting stuck in local minimum solutions, we rotate the orthogonal array so that different variations of the array are used in different cycles. In the first cycle, we use the array as it is, e.g., as in Table II.

In the next cycle, we move the second column of the orthogonal array to the first, the third column to the second, etc., and the first column to the last, so that different level combinations are tested for each factor in different cycles. In this way, we can reduce the possibility of getting stuck in local minima. When the orthogonal array is rotated from one cycle to the next, different factors will be used in different dimensions when tests are designed according to the array. We will use an example to show the effect of such rotations in Section V (cf. Example 3).

The Taguchi method provides us with a systematic and efficient method for conducting experimentations to determine near optimal values of the controllable factors. By using orthogonal arrays, the Taguchi method searches in the parameter space with a small number of experiments. The savings will be greater when the number of factors in the problem is larger. For example, in our simulations, by using the orthogonal array $L_{27}(3^{13})$, ten factors each with three levels are optimized by running only 27 tests as opposed to a total of $3^{10} = 59\,049$ tests required by a full experimental design in each cycle. It will be shown in Section V that the algorithm developed in this paper reaches convergence six times faster than fast evolutionary programming (FEP) [19] and improved fast evolutionary programming (IFEP) [19] and ten times faster than classical evolutionary programming (CEP) [19] and modified fast evolutionary programming (MFEP) [19] when applied to solve the economic dispatch problem with 40 units.

There are several important features of the Taguchi method developed in this paper. They are enumerated below.

1) *Less sensitive to the choice of initial values of parameters* α_i. In most algorithms, if the initial values are far away from the optimal values, their complexity increases greatly. Thus, how to choose initial values that are close to the optimal values is a big concern to many existing algorithms. The Taguchi method is less sensitive to the choice of initial values that eases this concern. In Section V, we will show using an example that the present algorithm is less sensitive to the choice of initial values of α_k. Using the present algorithm, the randomly chosen initial levels $\left(\alpha_i^{(1)}, \alpha_i^{(2)}, \alpha_i^{(3)}\right)$ may or may not cover the actual optimal value of α_i in its range. Simulation results reveal that the performance of the present algorithm is less sensitive to the choice of initial levels used in the experiments.
2) *Easy implementation and fast convergence*. The calculations of the values of the cost function J in (4) are straightforward given all the required information. In Section V, we will show that the present algorithm for the economic dispatch problem has a very fast convergence speed. Furthermore, the more complicated the cost function, the more obvious this advantage is.
3) *Suitability for parallel implementation*. In every cycle of tests in the present algorithm, the computations required for all the tests can be done in parallel on different processors. For example, in the case of using the orthogonal array $L_9(3^4)$, the nine tests in every cycle compute the values of the cost function J. In these computations, the procedure remains the same, and we only choose to test different value combinations of parameters α_i. Therefore, these computations can be done in parallel to achieve even faster speed in finding solutions.

V. Simulation Results

In this section, we assess the performance of the Taguchi method developed in the previous section using computer simulations. Throughout this section, TM shall refer to the present Taguchi method. In our simulation studies, we will not compare TM with the traditional Lagrangian relaxation method and its variants, because it is already shown in [8] and [21] that the performance of genetic algorithms and Hopfield neural networks is better than the traditional Lagrangian relaxation method and its variants. We will therefore focus on the comparison of TM with genetic algorithm, Hopfield neural networks, and evolutionary programming instead. In the examples of this section, all algorithms are implemented using Matlab. In order to show the effectiveness of the proposed algorithm, test results of some typical cases [16], [19] are used in three examples. Specifically, Examples 1 and 2 include three and 40 generator units, respectively, where valve-point effects are considered for both problems. The genetic algorithm and other approaches in [16] are used in Examples 1 and 2 to provide benchmark minimum operating cost. Example 3 contains ten units and is with nonsmooth cost functions considering multiple fuels. The three initial levels of each factor α_i are chosen as 0.5, 0.8, and 1.2 unless indicated otherwise. The step size and shrinking coefficient are chosen experimentally. We find that step size of 0.01 and shrinking coefficient $\eta = 0.9$ work very well with all examples in our simulation studies. If smaller step size and shrinking coefficient are used, it takes longer time to converge without much improvement in performance. If larger step size and shrinking coefficient are used, it takes shorter time to converge but with worse results. In all examples, we use a fixed step size of 0.01 and $\eta = 0.9$ for parameter updates.

Example 1: The same three-generator system used in [10] will be used in this example for computer simulation. Here, the total demand for the system is set to 850 MW. In this example, due to the small size of the problem, the global optimal cost of this example is known, and we want to show that the present Taguchi method can also obtain the global minimum value of the cost function. It has been reported in [10] that the global optimal solution found for the present three-generator system is 8234.07. The orthogonal array used in this example is $L_9(3^4)$[20]. The obtained results for the three-generator system using the Taguchi method are given in Table III, and the results are compared with those from GA [21], EP [23], and IEP [17]. GA solutions are obtained with the help of a genetic algorithm optimization toolbox [7], the population size is chosen as 100, the crossover rate is 0.95, and the mutation rate is 0.01. The values of population size and scale factor are chosen as 30 and 0.1, respectively, for EP and IEP. Table III shows that the Taguchi method has succeeded in finding the global optimal solution presented in [10].

Example 2: This case consists of 40 generator units considering the valve-point effects. Exactly the same data of all units as given in [19] will be used in this example. Here the total demand is 10 500 MW. Simulation results are given in Tables IV and V.

TABLE III
COMPARISON OF VARIOUS METHODS CONSIDERING VALVE-POINT EFFECTS (THREE-GENERATOR SYSTEM)

Unit	TM	GA	EP	IEP
1	300.27	300.00	300.26	300.23
2	400.00	400.00	400.00	400.00
3	149.73	150.00	179.74	179.77
Total Power	850.00	850.00	850.00	850.00
Total Cost	8234.07	8237.60	8234.07	8234.09

TABLE IV
COMPARISON RESULTS WITH METHODS IN [19] (40-UNIT SYSTEM WITH VALVE-POINT EFFECTS)

Method	Mean time	Best time	Mean cost	Maximum cost	Minimum cost
CEP	928.36	926.20	124793.48	126902.89	123488.29
FEP	646.16	644.28	124119.37	127245.89	122679.71
MFEP	1056.8	1054.2	123489.74	124356.47	122647.57
IFEP	632.67	630.36	123382.00	125740.63	122624.35
TM	94.28	91.16	123078.21	124693.81	122477.78

TABLE V
RELATIVE FREQUENCY OF CONVERGENCE

Range	CEP	FEP	MFEP	IFEP	TM
$1.265 \sim 1.27 \times 10^5$	10	6	–	–	–
$1.26 \sim 1.265 \times 10^5$	4	–	–	–	–
$1.255 \sim 1.26 \times 10^5$	–	4	–	2	–
$1.25 \sim 1.255 \times 10^5$	16	2	–	–	–
$1.245 \sim 1.25 \times 10^5$	22	10	–	4	2
$1.24 \sim 1.245 \times 10^5$	42	20	14	4	2
$1.235 \sim 1.24 \times 10^5$	4	26	26	18	12
$1.23 \sim 1.235 \times 10^5$	2	24	50	50	52
$1.225 \sim 1.23 \times 10^5$	–	6	10	22	22
$1.22 \sim 1.225 \times 10^5$	–	–	–	–	10

The orthogonal array used in this example is $L_{81}(3^{40})$[20]. The values of population size, scale factor, and penalty multiplier are chosen as 60, 0.05, and 100, respectively, when FEP, IFEP, CEP, and MFEP are implemented. The number of generations for FEP, IFEP, CEP, and MFEP are 600, 400, 800, and 1000, respectively.

Table IV shows the mean and best solution time in seconds, mean cost, maximum cost, and minimum cost achieved by various methods over 100 runs. For each run, before we start the search procedure, we randomly switch two columns of the orthogonal array $L_{81}(3^{40})$ and assign experiments according to the new array. Thus, different initial orthogonal array will be used in each run, and it results in the variation in the final solution. TM requires the least amount of solution time, six times faster than FEP [19] and IFEP [19] and almost ten times faster than CEP [19] and MFEP [19]. We also see that the minimum cost achieved by TM is smaller than that of CEP, FEP, MFEP, and IFEP. Using evolutionary programming, the population size for the 40 units system in this case is 60 [19]. From [19, Fig. 4], we see that the fastest among the four methods is IFEP, which takes about 400 generations. The total number of tests required by IFEP is therefore $24\,000 = 60 \times 400$. On the other hand, using the Taguchi method, we use the orthogonal array $L_{81}(3^{40})$ (which implies an equivalent population size of 81), and it usually converges in 50 cycles, and thus the total number of tests required is $4050 = 81 \times 50$. We can see that the number of tests required by the IFEP is roughly six times more than that required by the Taguchi method. In our Table IV, our simulation results show that the Taguchi method is about seven times faster than IFEP.

Table V shows the frequency of attaining a cost within the specific ranges out of 100 runs for each of the algorithms. From the 100 runs, 96 $(= 10{+}22{+}52{+}12)$ runs of the final costs from TM are in the range of 1.22–1.24 $\times 10^5$ (see Table V). For comparison, 90 runs from IFEP, 86 runs from MFEP, 56 runs from FEP, and six runs from CEP are with the final costs in the same range. Table V reveals that TM has the highest number of runs to achieve the cost values in the smallest bracket 1.22–1.23 $\times 10^5$, which shows the highest probability of attaining minimum solutions.

Example 3: The present TM has also been applied to the ED problem with ten generators where multiple-fuel effects are considered. The hierarchical numerical method (HM) [11], the IEP [17], the modified Hopfield neural network (MHNN) [15], the adaptive Hopfield neural network (AHNN) [8], and the modified particle swarm optimization (MPSO) [16] are used to provide the benchmark minimum operating cost against which the Taguchi method is compared. In this case, the objective function is represented using a piecewise quadratic cost function. In order to prove the usefulness of the Taguchi method, the same system used in [8], [11], and [17] will be used in this example for computer simulations. HM is a numerical method. The population size and scale factor are chosen as 30 and 0.01 for IEP. Continuous neuron model is applied for implementation of MHNN to solve the ED problem. Adaptive learning rate is applied for AHNN. When MPSO is implemented, the dynamic search-space reduction strategy is applied to accelerate the convergence speed. The weight factors are chosen as 2, the step size is 0.05, and the number of particles is 30.

The input data and related constraints of the test system are given in [8], [11], and [17]. In this case, the total demand is varied from 2400 to 2700 MW.

As seen in Table VI, the TM always provides better solutions than HM, MHNN, AHNN, IEP, and MPSO. Here in Table VI, U means unit, F represents fuel type, TP means the sum of the power assigned to each unit, and TC represents the total cost. The orthogonal array used in this example is $L_{27}(3^{13})$ [20].

Recall that in the previous section, we mentioned that the Taguchi method is less sensitive to the choice of initial values. We next consider the system performance with different choices of initial values. Two extreme cases are compared to the case we used for 2400 MW in Table VI, where the three initial levels of each factor are chosen as 0.5, 0.8, and 1.2. Case 1: the three initial levels are all smaller than 1, and Case 2: the three initial levels are all larger that 1 but smaller than 2. Because we choose the reference value as $P_{i,\text{ref}} = (P_{i,\max} + P_{i,\min})/2$, if the initial value of $\alpha_i > 2$, it results in a saturated P_i, which means $P_{i,\text{ref}} \times \alpha > P_{i,\max}$. Some of the choices for initial levels in the two cases could be totally wrong, which may be considered as due to an incorrect guess of the range of the initial α_i. From Fig. 2, we can see that the system performance after convergence is very close to each other, no matter what initial levels we choose. Due to the rotation of the orthogonal array, different level combinations are tested for each factor in different cycles. The curves of the cost function go up and down during the first

TABLE VI
COMPARISON OF OPTIMIZATION METHODS

Demand = 2400 MW						
U	F	HM	F	MHNN	F	AHNN
1	1	193.2	1	192.7	1	189.1
2	1	204.1	1	203.8	1	202.0
3	1	259.1	1	259.1	1	254.0
4	3	234.3	2	195.1	3	233.0
5	1	249.0	1	248.7	1	241.7
6	1	195.5	3	234.2	3	233.0
7	1	260.1	1	260.3	1	254.1
8	3	234.3	3	234.2	3	232.9
9	1	325.3	1	324.7	1	320.0
10	1	246.3	1	246.8	1	240.3
TP		2401.2		2399.8		2400.0
TC		488.5		487.9		481.7
Demand = 2400 MW						
U	F	IEP	F	MPSO	F	TM
1	1	190.9	1	189.7	1	190.3
2	1	202.3	1	202.3	1	203.1
3	1	253.9	1	253.9	1	253.3
4	3	233.9	3	233.0	3	233.1
5	1	243.8	1	241.8	1	241.5
6	3	235.0	3	233.0	3	232.5
7	1	253.2	1	253.3	1	252.8
8	3	232.8	3	233.0	3	233.1
9	1	317.2	1	320.4	1	320.1
10	1	237.0	1	239.4	1	240.2
TP		2400.0		2400.0		2400.0
TC		481.8		481.7		481.6
Demand = 2700 MW						
U	F	HM	F	MHNN	F	AHNN
1	2	218.4	2	224.5	2	225.7
2	1	211.8	1	215.0	1	215.2
3	1	281.0	3	291.8	3	291.8
4	3	239.7	3	242.2	3	242.3
5	1	279.0	1	293.3	1	293.7
6	3	239.7	3	242.2	3	242.3
7	1	289.0	1	303.1	1	302.8
8	3	239.7	3	242.2	3	242.3
9	3	429.2	1	355.7	1	355.1
10	1	275.2	1	289.5	1	288.8
TP		2702.2		2699.7		2700.0
TC		625.2		626.1		626.2
Demand = 2700 MW						
U	F	IEP	F	MPSO	F	TM
1	2	219.5	2	218.3	2	217.0
2	1	211.4	1	211.7	1	213.2
3	1	279.7	1	280.7	1	282.2
4	3	240.3	3	239.6	3	240.6
5	1	276.5	1	278.5	1	278.7
6	3	239.9	3	239.6	3	239.1
7	1	289.0	1	288.6	1	290.5
8	3	241.3	3	239.6	3	241.0
9	3	425.1	3	428.5	3	418.0
10	1	277.2	1	274.9	1	279.7
TP		2700.0		2700.0		2700.0
TC		623.8		623.8		623.7

250 cycles of tests, after which they converge to the same value. The only difference is the time of convergence, which results from different choices of initial levels for each factor.

We will show next the effects of rotating the orthogonal array on the performance of TM. The system is the same as the one used for 2400 MW in Table VI. The orthogonal array will be rotated from one cycle of tests to the next. In the first cycle, we use the orthogonal array as it is. In the next cycle, we move the second column of the orthogonal array to the first, the third column to the second, etc., and the first column to the last. After each cycle of tests, we perform the rotation again in the same way, so that different level combinations are tested for each factor in different cycles. From Fig. 3, we can see that with the help of rotating the orthogonal array, the Taguchi method can reduce the chance of getting stuck in local minima and result in better performance. In the figure, we can see that without rotating the orthogonal array, the cost function converges to a much larger value.

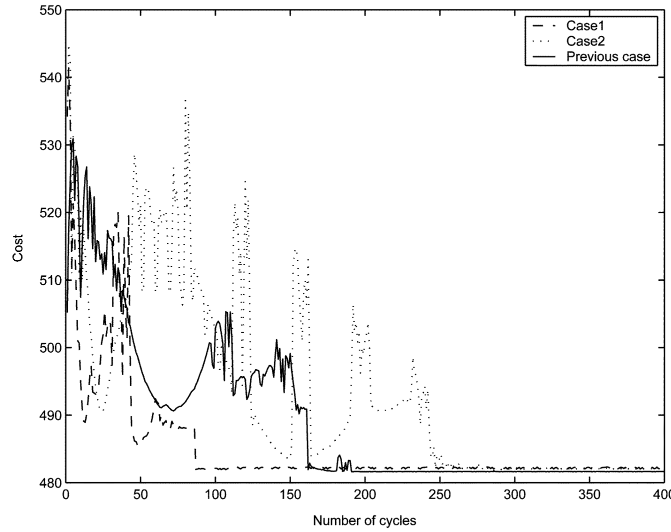


Fig. 2. Performance comparison of different choices of initial values. Case 1: the initial three levels of parameters are all smaller than 1. Case 2: the initial three levels are all larger than 1.

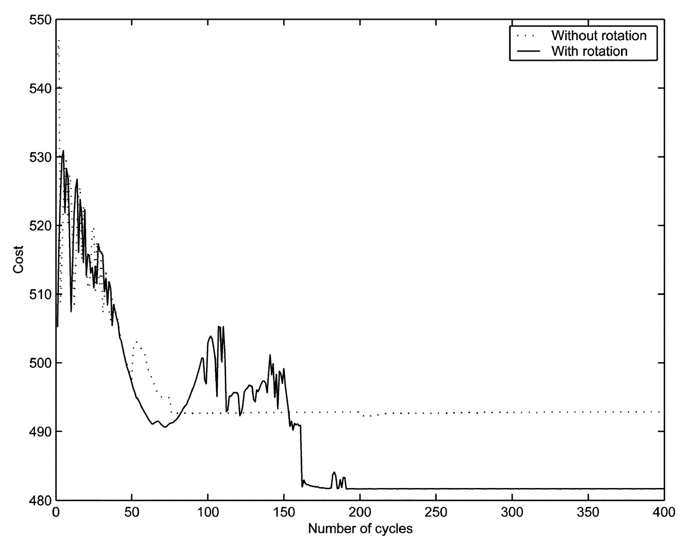


Fig. 3. Effect of rotating the orthogonal array on the performance of the Taguchi method.

VI. CONCLUSIONS

The Taguchi method is capable of solving the constrained ED problem for practical power systems. Our analysis and numerical simulation results show that the present Taguchi method is less sensitive to the choice of initial values for parameters. Combined with their relatively low computational requirements as well as their suitability for parallel implementation, we believe that these features make the present Taguchi method a viable option for solving the economic dispatch problems in real-world applications.

References

[1] A. Bakirtzis, V. Petridis, and S. Kazarlis, "Genetic algorithm solution to the economic dispatch problem," *Proc. Inst. Elect. Eng., Gener., Transm., Distrib.*, vol. 141, no. 4, pp. 377–382, Jul. 1994.

[2] Y. Cai and D. Liu, "Multiuser detection using the Taguchi method for the DS-CDMA systems," *IEEE Trans. Wireless Commun.*, to be published.

[3] J. Y. Fan and L. Zhang, "Real-time economic dispatch with line flow and emission constraints using quadratic programming," *IEEE Trans. Power Syst.*, vol. 13, no. 2, pp. 320–325, May 1998.

[4] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 2nd ed. New York: IEEE Press, 2000.

[5] W. Y. Fowlkes and C. M. Creveling, *Engineering Methods for Robust Product Design: Using Taguchi Methods in Technology and Product Development*. Reading, MA: Addison-Wesley, 1995.

[6] A. S. Hedayat, N. J. A. Sloane, and J. Stufken, *Orthogonal Arrays: Theory and Applications*. New York: Springer-Verlag, 1999.

[7] C. Houck, J. Joines, and M. Kay, "A Genetic Algorithm for Function Optimization: A Matlab Implementation," North Carolina State Univ., Tech. Rep. NCSU-IE TR 95-09, 1995.

[8] K. Y. Lee, A. Sode-Yome, and J. H. Park, "Adaptive Hopfield neural networks for economic load dispatch," *IEEE Trans. Power Syst.*, vol. 13, no. 2, pp. 519–526, May 1998.

[9] Z. X. Liang and J. D. Glover, "A zoom feature for a dynamic programming solution to economic dispatch including transmission losses," *IEEE Trans. Power Syst.*, vol. 7, no. 2, pp. 544–550, May 1992.

[10] W. M. Lin, F. S. Cheng, and M. T. Tsay, "An improved tabu search for economic dispatch with multiple minima," *IEEE Trans. Power Syst.*, vol. 17, no. 1, pp. 108–112, Feb. 2002.

[11] C. E. Lin and G. L. Viviani, "Hierarchical economic dispatch for piecewise quadratic cost functions," *IEEE Trans. Power App. Syst.*, vol. PAS-103, no. 6, pp. 1170–1175, Jun. 1984.

[12] R. H. Lochner and J. E. Matar, *Designing for Quality: An Introduction to the Best of Taguchi and Western Methods of Statistical Experimental Design*. Milwaukee, WI: ASQC Quality, 1990.

[13] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "Practical distribution state estimation using hybrid particle swarm optimization," in *Proc. IEEE Power Engineering Society Winter Meeting*, vol. 2, Columbus, OH, Feb. 2001, pp. 815–820.

[14] W. Ongsakul and J. Tippayachai, "Parallel micro genetic algorithm based on merit order loading solutions for constrained dynamic economic dispatch," *Elect. Power Syst. Res.*, vol. 61, no. 2, pp. 77–88, Mar. 2002.

[15] J. H. Park, Y. S. Kim, I. K. Eom, and K. Y. Lee, "Economic load dispatch for piecewise quadratic cost function using Hopfield neural network," *IEEE Trans. Power Syst.*, vol. 8, no. 3, pp. 1030–1038, Aug. 1993.

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

[17] Y. M. Park, J. R. Won, and J. B. Park, "New approach to economic load dispatch based on improved evolutionary programming," *Int. J. Eng. Intell. Syst. Elect. Eng. Commun.*, vol. 6, no. 2, pp. 103–110, Jun. 1998.

[18] P. J. Ross, *Taguchi Techniques for Quality Engineering: Loss Function, Orthogonal Experiments, Parameter and Tolerance Design*. New York: McGraw-Hill, 1996.

[19] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," *IEEE Trans. Evol. Comput.*, vol. 7, no. 1, pp. 83–94, Feb. 2003.

[20] G. Taguchi and S. Konishi, *Orthogonal Arrays and Linear Graphs: Tools for Quality Engineering*. Allen Park, MI: ASI, 1987.

[21] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading," *IEEE Trans. Power Syst.*, vol. 8, no. 3, pp. 1325–1332, Aug. 1993.

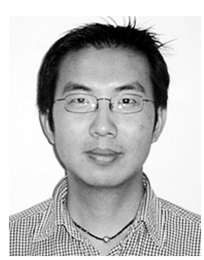
[22] Z. Wang, E. Gao, and E. Zhang, "An orthogonal optimization method" (in Chinese), *ACTA Automatica Sinica*, vol. 15, no. 4, pp. 365–369, Jul. 1989.

[23] H. T. Yang, P. C. Yang, and C. L. Huang, "Evolutionary programming based economic dispatch for units with nonsmooth fuel cost functions," *IEEE Trans. Power Syst.*, vol. 11, no. 1, pp. 112–118, Feb. 1996.

Derong Liu (S'91–M'94–SM'96–F'05) received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1994.

From 1993 to 1995, he was a Staff Fellow at General Motors Research and Development Center, Warren, MI. From 1995 to 1999, he was an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ. He joined the University of Illinois at Chicago in 1999 as an Assistant Professor of electrical engineering and computer science, where he is now an Associate Professor of electrical and computer engineering, of bioengineering, and of computer science. He is coauthor (with A. N. Michel) of the books *Dynamical Systems with Saturation Nonlinearities: Analysis and Design* (New York: Springer-Verlag, 1994) and *Qualitative Analysis and Synthesis of Recurrent Neural Networks* (New York: Marcel Dekker, 2002). He is coeditor (with P. J. Antsaklis) of the book *Stability and Control of Dynamical Systems with Applications* (Boston, MA: Birkhauser, 2003).

Dr. Liu was a Member of the Conference Editorial Board of the IEEE Control Systems Society (1995–2000), served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: FUNDAMENTAL THEORY AND APPLICATIONS (1997–1999), and served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2001–2003). Since 2004, he has been an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS. He has served and is serving as a member of the organizing committee and the program committee of several international conferences. He was recipient of the Michael J. Birck Fellowship from the University of Notre Dame (1990), the Harvey N. Davis Distinguished Teaching Award from Stevens Institute of Technology (1997), and the Faculty Early Career Development (CAREER) award from the National Science Foundation (1999). He is Member of Eta Kappa Nu.



Ying Cai (S'03) received the B.S. and M.S. degrees in electronic and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 1997 and 2000, respectively. Since 2000, he has been working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago.

He is currently a Research Assistant with the Computational Intelligence Laboratory, University of Illinois at Chicago. His research interests include coding and signal processing in wireless communications.