Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

MoIB18.1

An Orthogonal Array Optimization for the Economic Dispatch with Nonsmooth Cost Functions

Ying Cai, *Student Member, IEEE*, and Derong Liu, *Fellow, 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 insensitive to initial values of parameters, and is more effective than other previously developed algorithms. In addition, our algorithm is suitable for parallel implementations.

I. INTRODUCTION

THE Taguchi method of experimental design has been widely used in industry for purpose of finding factors that are most important in achieving useful goals in a manufacturing process [5], [6], [8], [14]. Several factors which are related to the goals and are under the user's control are selected. 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 lambda iteration method [1], quadratic programming [3], and gradient method [10]. 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 operations and hence huge revenue losses over time.

The authors are with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA. {ycai,dliu}@ece.uic.edu

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 [9], [12]. A global optimization technique known as the genetic algorithm (GA) has also been successfully applied to solve the ED problem [17]. However, recent research has identified some deficiencies in GA performance. The premature convergence of GA degrades its performance and reduces its search capability that leads to a higher probability toward a local minimum [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 believe that the present work 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 [13], [15], [19], Hopfield neural network [9], [12], hierarchical numerical method [7], and particle swarm optimization [11].

II. PROBLEM STATEMENT

A. The Economic Dispatch Problem With Smooth Cost Functions

To solve the standard economic dispatch (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 losses P_L).

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

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \tag{1}$$

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. The Economic Dispatch Problem With Nonsmooth Cost Functions

In reality, the objective function of an ED problem has nondifferentiable points according 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: Typically, the fuel-cost function considering valve-point loadings of a generating unit is given by

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 + |e_i \times \sin(f_i \times (P_{i,\min} - P_i))| \quad (2)$$

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: Generally, a piecewise quadratic function is used to represent the input-output curve of a generator with multiple fuels [11]. The piecewise quadratic function is given by

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

$$(3)$$

where $a_{i,j}$, $b_{i,j}$, and $c_{i,j}$ are the cost coefficients of generator i for the fuel type j. 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

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

The goal is to determine P_i, $i = 1, 2, \cdots, N$, 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.,

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

Without loss of generality, the transmission loss is not considered in this paper for simplicity (i.e., we can assume that $P_L = 0$) [11].

(2) The operational constraints for unit i is given by

$$P_{i,\min} \le P_i \le P_{i,\max} \quad (6)$$

where $P_{i,\min}$ and $P_{i,\max}$ are the minimum and maximum power generation of unit i.

III. INTRODUCTION TO THE TAGUCHI METHOD

Suppose that an experimental outcome J is a cost function of several variables, $p_1, p_2, \cdots, p_m$, whose values can be controlled. We write $J = J(p_1, p_2, \cdots, p_m)$. The controlled variables p_k, $k = 1, \cdots, m$, are called factors. The goal is to find the optimal values $\hat{p}_k$, $k = 1, \cdots, m$, to minimize the cost function J. This can be done by varying the factors

TABLE I
INITIAL VALUES OF THE FOUR FACTORS

Factor	level 1	level 2	level 3
p_1	0.3	0.5	0.8
p_2	0.3	0.5	0.8
p_3	0.3	0.5	0.8
p_4	0.3	0.5	0.8

simultaneously in a disciplined manner and recording the corresponding values of J until we get the optimal $\hat{p}_k$, $k = 1, \cdots, m$. The Taguchi method involves a disciplined method of varying two or more factors simultaneously.

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 $k = 1, \cdots, 4$. 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 2 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.

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.

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 $J_i, i = 1, 2, \cdots, 9$. 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

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	$V_1^{(3)}$	$V_2^{(3)}$	$V_3^{(3)}$	$V_4^{(3)}$	

Fig. 1. Six different trends of the cost function

are completed, for factor p_3, we calculate

$$V_3^{(1)} = J_1 + J_6 + J_8,$$
$$V_3^{(2)} = J_2 + J_4 + J_9, \qquad (7)$$
$$V_3^{(3)} = J_3 + J_5 + J_7$$

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 Figure 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 trend of the cost function is as shown in Figure 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 Figure 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]

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

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 Figure 1 (a), (b), (d), or (e), we have a very
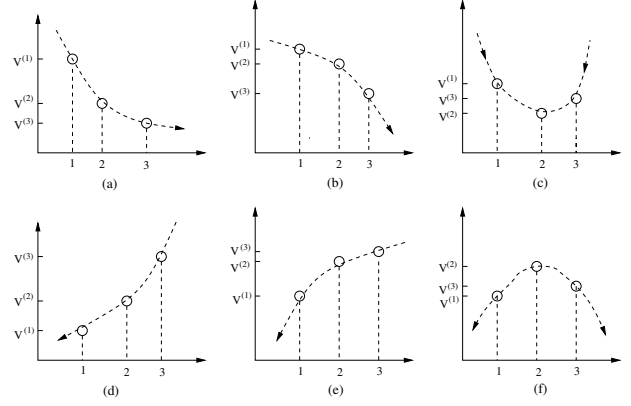
good estimate of the directions of the gradient of the cost function. Note that to minimize a cost function, the key is to determine its gradient with respect to varying parameters without actually using the values of the gradient. If the trend is as shown in Figure 1 (c), that means the parameter value should be set closer to the middle level or the center of the parabolic curve (with 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

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

where M is the number of tests in each cycle according to the orthogonal array, $J_i(k)$ 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. Such a method based on orthogonal arrays is usually referred to as the Taguchi method [5], [8], [14], which we will use for solving the economic dispatch problem with nonsmooth cost functions.

IV. THE TAGUCHI METHOD FOR SOLVING THE ECONOMIC DISPATCH PROBLEM

In this section the economic dispatch problem with non-smooth 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 $P = [P_1, P_2, \cdots, P_N]$ 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, $i = 1, \cdots, N$, 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 $[\alpha_1, \cdots, \alpha_N]$ as the factors to be optimized by the Taguchi method instead of the vector $[P_1, \cdots, P_N]$. Here, $\alpha_i, i = 1, \cdots, N$, 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

$$J_k = F_1(P_1) + F_2(P_2) + \cdots + F_N(P_N) \tag{8}$$

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

$$P_i = \eta_k \alpha_i P_{i,\text{ref}} \quad \text{if} \quad P_{i,\min} \leq \eta_k \alpha_i P_{i,\text{ref}} \leq P_{i,\max}, \tag{9}$$

$$P_i = P_{i,\max} \quad \text{if} \quad \eta_k \alpha_i P_{i,\text{ref}} \geq P_{i,\max}, \tag{10}$$

$$P_i = P_{i,\min} \quad \text{if} \quad \eta_k \alpha_i P_{i,\text{ref}} \leq P_{i,\min}, \tag{11}$$

where $P_{i,\text{ref}} = 0.5 \times (P_{i,\min} + P_{i,\max})$ and η_k is determined such that the constraint (5) is satisfied. Note that here we use $P_{i,\text{ref}}$, 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. 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) Specify the lower and upper bounds of generation powers for each unit. Determine the reference point 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, $i = 1, 2, \cdots, N$. Set $k = 0$.
2) Let $k = k + 1$. For test k, determine α_i according to the orthogonal array. Set $\eta_k = \eta'_k = 1$.
3) Determine P_i according to (9). In case any P_i is determined by (10) or (11), which means that α_i is saturated, go to Step 3.2. Otherwise, go to Step 3.1.
    3.1) Calculate η_k using
    $$\eta_k = \frac{P_D}{\sum\limits_{i=1}^{N} \alpha_i \times P_{i,\text{ref}}}.$$
    If $\eta_k = \eta'_k$, go to Step 4. Otherwise, let $\eta'_k = \eta_k$, and go back to Step 3.
    3.2) Find all α_l, $l \in K$, which are saturated. In order to satisfy the constraint (5), calculate η_k using
    $$\eta_k = \frac{P_D - \sum\limits_{l \in K} P_l}{\sum\limits_{i \in \mathcal{I}} \alpha_i \times P_{i,\text{ref}}}$$

where $\mathcal{I}$ is the index set of nonsaturated factors of α_i. If $\eta_k = \eta'_k$, go to Step 4. Otherwise, let $\eta'_k = \eta_k$, and go back to Step 3.
4) Determine the cost J_k according to (8) with the factors P_i determined from Step 3.
5) 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 6. Otherwise, determine the trends of the cost function, adjust the values of the factors according to the trends, and go back to Step 2 to start a new cycle.
6) Find the factors which gives 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 minimum 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, $\cdots$, 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 minimum. 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.

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

(i) Insensitivity to the choice of initial values of parameters α_i.

(ii) Easy implementation and fast convergence.

(iii) Suitability for parallel implementation.

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 order to show the effectiveness of the proposed algorithm, test results of some typical cases [11], [15] are used in three examples. Specifically, Example 5.1 includes 40 generator units, where valve-point effects are considered. The genetic algorithm and other approaches in [11] are used in Example 5.2 to provide benchmark minimum operating cost. The three initial levels of each factor α_i are chosen as $0.5, 0.8, 1.2$ unless indicated otherwise. In all examples, we use a fixed step size of 0.01 and $\eta = 0.9$ for parameter updates in our examples.

Example 5.1: This case consists of 40 generator units considering the valve-point effects. Exactly the same data of all units as given in [15] will be used in this example.

Here the total demand is 10500 MW. Simulation results are given in Tables III and IV. The orthogonal array used in this example is $L_{81}(3^{40})$ [16]. 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 III 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, 6 times faster than fast evolutionary programming (FEP) [15] and improved fast evolutionary programming (IFEP) [15], and almost 10 times faster than classical evolutionary programming (CEP) [15] and modified fast evolutionary programming (MFEP) [15]. 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 [15]. From Figure 4 of [15], 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 $24000 = 60 \times 400$. 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 6 times more than that required by the Taguchi method. In our table IV, our simulation results show that the Taguchi method is about 7 times faster than IFEP.

Table IV 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 $1.22 \sim 1.24 \times 10^5$ (Table IV). For comparison, 90 runs from IFEP, 86 runs from MFEP, 56 runs from FEP, and 6 runs from CEP are with the final costs in the same range. Table IV reveals that TM has the highest number of runs to achieve the cost values in the smallest bracket $1.22 \sim 1.23 \times 10^5$ which shows the highest probability of attaining minimum solutions.

Example 5.2: The present TM has also been applied to the ED problem with 10 generators where the multiple-fuel effects are considered. In this case, the objective function is represented using a piecewise quadratic cost function. The input data and related constraints of the test system are given in [7], [9], and [13]. In this case, the total demand is 2400 MW.

TABLE III
COMPARISON RESULTS WITH METHODS IN [15] (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 IV
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 hierarchical numerical method (HM) [7], the improved evolutionary programming (IEP) [13], the modified Hopfield neural network (MHNN) [12], the adaptive Hopfield neural network (AHNN) [9], and the modified particle swarm optimization [11] are used to provide the benchmark minimum operating cost against which the Taguchi method is compared. As seen in Table V, the TM always provides better solutions than HM, MHNN, AHNN, IEP, and MPSO. Here in Table V, 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 $L_{27}(3^{13})$ [16].

Recall that in the previous section, we mentioned that the Taguchi method is insensitive 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 cases we used in Tables V and **??**, 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 $P_{i,\mathrm{ref}} = (P_{i,\max} + P_{i,\min})/2$, if the initial value of $\alpha_i > 2$, it results in a saturated P_i, which means $P_{i,\mathrm{ref}} \times \alpha > P_{i,\max}$. 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 original α_i. From Figure 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, which helps to avoid the local minimum. The curves of the

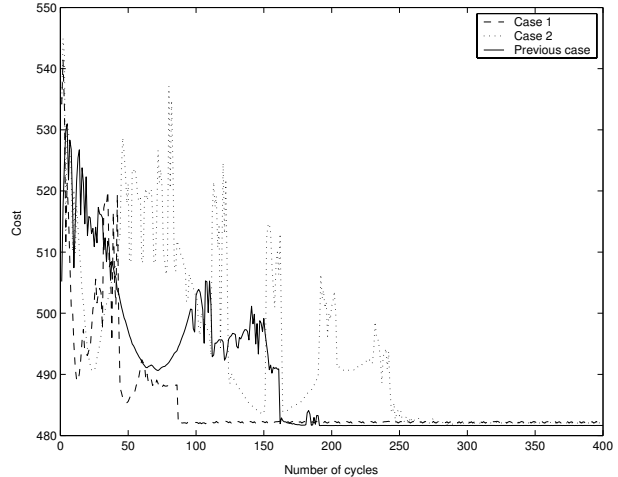| U	F	HM	F	MHNN	F	AHNN
1	1	193.2	1	192.7	1	189.1
2	1	204.1	1	203.8	1	202.0
3	1	259.1	1	259.1	1	254.0
4	3	234.3	2	195.1	3	233.0
5	1	249.0	1	248.7	1	241.7
6	1	195.5	3	234.2	3	233.0
7	1	260.1	1	260.3	1	254.1
8	3	234.3	3	234.2	3	232.9
9	1	325.3	1	324.7	1	320.0
10	1	246.3	1	246.8	1	240.3
TP		2401.2		2399.8		2400.0
TC		488.5		487.9		481.7
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

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.

cost function go up and down during the first 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.

VI. CONCLUSIONS

The Taguchi method is capable of solving the constrained economic dispatch problem for practical power systems. Our analysis and numerical simulation results show that the present Taguchi method is insensitive 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," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 141, pp. 377–382, July 1994.

[2] Y. Cai and D. Liu, "Multiuser detection using the Taguchi method for the DS-CDMA systems," to appear in *IEEE Transactions on Wireless Communications*.

[3] J. Y. Fan and L. Zhang, "Real-time economic dispatch with line flow and emission constraints using quadratic programming," *IEEE Trans. Power Systems*, vol. 13, 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, 1999.

[7] C. E. Lin and G. L. Viviani, "Hierarchical economic dispatch for piecewise quadratic cost functions," *IEEE Trans. Power Apparatus Systems*, vol. PAS-103, pp. 1170–1175, June 1984.

[8] 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 Press, 1990.

[9] K. Y. Lee, A. Sode-Yome, and J. H. Park, "Adaptive Hopfield neural networks for economic load dispatch," *IEEE Trans. Power Systems*, vol. 13, pp. 519–526, May. 1998.

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

[11] J. B. Park, Y. S. Kim, J. R. Shin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions," to apprear in *IEEE Trans. Power Systems*.

[12] 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 Systems*, vol. 8, pp. 1030–1038, Aug. 1993.

[13] Y. M. Park, J. R. Won, and J. B. Park, "New approach to economic load dispatch based on improved evolutionary programming," *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications*, vol. 6, no. 2, pp. 103–110, June 1998.

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

[15] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," *IEEE Trans. Evolutionary Computation*, vol. 7, pp. 83–94, Feb. 2003.

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

[17] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading," *IEEE Trans. Power Systems*, vol. 8, pp. 1325–1332, Aug. 1993.

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

[19] H. T. Yang, P. C. Yang, and C. L. Huang, "Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions," *IEEE Trans. Power Systems*, vol. 11, pp. 112–118, Feb. 1996.