Improved Genetic Algorithm for Power Economic Dispatch of Units With Valve-Point Effects and Multiple Fuels

Article *in* Power Systems, IEEE Transactions on · December 2005

DOI: 10.1109/TPWRS.2005.857924 · Source: IEEE Xplore

1 author:

Chao-Lung Chiang
Nan Kai University of Technology

Improved Genetic Algorithm for Power Economic Dispatch of Units With Valve-Point Effects and Multiple Fuels

Chao-Lung Chiang

***Abstract*—This paper presents an improved genetic algorithm with multiplier updating (IGA_MU) to solve power economic dispatch (PED) problems of units with valve-point effects and multiple fuels. The proposed IGA_MU integrates the improved genetic algorithm (IGA) and the multiplier updating (MU). The IGA equipped with an improved evolutionary direction operator and a migration operation can efficiently search and actively explore solutions, and the MU is employed to handle the equality and inequality constraints of the PED problem. Few PED problem-related studies have seldom addressed both valve-point loadings and change fuels. To show the advantages of the proposed algorithm, which was applied to test PED problems with one example considering valve-point effects, one example considering multiple fuels, and one example addressing both valve-point effects and multiple fuels. Additionally, the proposed algorithm was compared with previous methods and the conventional genetic algorithm (CGA) with the MU (CGA_MU), revealing that the proposed IGA_MU is more effective than previous approaches, and applies the realistic PED problem more efficiently than does the CGA_MU. Especially, the proposed algorithm is highly promising for the large-scale system of the actual PED operation.**

***Index Terms*—Economic dispatch, genetic algorithm, multiple fuels, valve-point effects.**

I. Introduction

THE power economic dispatch (PED) problem is one of the important optimization problems in a power system. Traditionally, in the PED problem, the cost function for each generator has been approximately represented by a single quadratic function, and the valve-point effects [1] were ignored. This would often introduce inaccuracy into the resulting dispatch. However, since the cost curve of a generator is highly nonlinear, containing discontinuities owing to valve-point loadings, the cost function is more realistically denoted as a segmented piecewise nonlinear function [2] rather than a single quadratic function. The PED problem with valve-point effects is represented as a nonsmooth optimization problem having complex and nonconvex characteristics with heavy equality and inequality constraints, which makes the challenge of finding the global optimum hard. Methods, which avoid this approximation of the actual unit curve model without sacrificing computational time would prove very valuable. Moreover, many generating units, particularly those which are supplied with multi-fuel sources (coal, nature gas, or oil), lead to the problem of determining the most economic fuel to burn. Some studies of the PED problem, such as evolutionary programming (EP) [2], [3], dynamic programming (DP) [4], Tabu search [5], genetic algorithm (GA) [1], hybrid EP combined with sequential quadratic programming (SQP) [6], and the particle swarm optimization (PSO) technique with the SQP method (PSO-SQP) [7], consider only valve-point effects, while others, such as a hierarchical method (HM) [8], Hopfield neural network approach (HNN) [9], adaptive Hopfield neural network method (AHNN) [10], and EP [11], have considered only the change fuels. However, few approaches for the PED problem, and none of the studies mentioned above, consider both valve-point loadings and multi-fuel effects, which always exist in real power systems simultaneously. To obtain an accurate and practical economic dispatch solution, the realistic operation of the PED problem should be taken both valve-point effects and multiple fuels into account. In this work, the realistic PED problems are compared to other methods using three examples. The first example consists of a 13-generator system considering only valve-point effects; the second example is a ten-unit system considering only multiple fuels, and the last example addresses both valve-point effects and multiple fuels. Comparative results indicate that the proposed IGA_MU is more effective than the previous methods, and is more efficient than the CGA_MU in the application of the practical PED problem. Especially, the proposed algorithm is highly promising for the large-scale system.

The GA [12] is a parallel and global searching-technique that emulates natural genetic operations. GA can find a global solution after sufficient iterations, but has a high computational burden. To enhance GA's computational efficiency, an improved evolutionary direction operator (IEDO) modified from [13] and a migration operator [14] are embedded in GA to form the IGA. The IGA has been applied successfully to constrained optimization problems such as: nonlinear mixed-integer optimization problems [15]; design of an induction motor controller [16]; prohibited operation zones of power economic dispatch [17], and the combined heat and PED problem [18]. The IGA requires only a small population, and it is more efficient than GA.

Michalewicz *et al.* [19] surveyed and compared several constraint-handling techniques used in evolutionary algorithms, and showed that the penalty function method is among the most popular methods for managing constraints. Powell [20] noted that classical optimization schemes including a penalty function have certain weaknesses that become serious when penalty

Manuscript received February 17, 2005; revised May 11, 2005. Paper no. TPWRS-00101-2005

The author is with the Electronics Engineering Department, Nan Kai Institute of Technology, Nan Tou, Taiwan, R.O.C. (e-mail: t129@nkc.edu.tw).

Digital Object Identifier 10.1109/TPWRS.2005.857924

parameters are large. When large penalty parameters are used, the penalty function becomes "ill-conditioned," making good solutions difficult to obtain. However, if the penalty parameters are too small, the constraint violation does not contribute a high cost to the penalty function. Therefore, choosing appropriate penalty parameters is not trivial. Herein, the MU technique is introduced to handle this constrained optimization problem. Such a technique can eliminate the ill-conditioned property of the objective function.

The proposed IGA_MU integrates the IGA and the MU, making it efficient and effective for the PED problem with both valve-point effects and multiple fuels.

II. Problem Formulation

The PED problem can be described as an optimization (minimization) process with objective [4]

$$\text{Minimize} \quad \sum_{i=1}^{n_p} F_i(P_i) \tag{1}$$

where $F_i(P_i)$ is the fuel cost function of the ith unit, P_i is the power generated by the ith unit, and n_P is the number of dispatchable units. Subject to the equality constraint of the power balance as

$$\sum_{i=1}^{n_p} P_i = P_d + P_L \tag{2}$$

where P_d is the system load demand and P_L is the transmission loss, and generating capacity constraints as

$$P_i^{\min} \le P_i \le P_i^{\max}, \qquad i = 1, \ldots, n_p \tag{3}$$

where $P_i^{\min}$ and $P_i^{\max}$ are the minimum and maximum power outputs of the ith unit.

In reality, the objective function of the PED problem has nondifferentiable points according to valve-point loadings and multiple fuels. Therefore, the objective function should be composed of a set of nonsmooth cost functions. This paper considers three cases of cost functions. The first case considers only the valve-point effect problem where the objective function is normally described as the superposition of a sinusoidal function and a quadratic function. The second case addresses only multiple fuels where the objective function is expressed as a set of piecewise quadratic functions. The third case considers both valve-point effects and multiple fuels for the purpose of realistic PED operation, presenting the objective as a set of piecewise superposition of sinusoidal functions and quadratic functions.

A. PED Problem Considering Valve-Point Effects

The generator cost function is obtained from a data point taken during "heat run" tests, when input and output data are measured as the unit slowly varies through its operating region. Wire drawing effects, which occur as each steam admission valve in a turbine starts to open, produce a rippling effect on the unit curve. To consider the accurate cost curve of each generating unit, the valve-point effects must be included in the cost model. Therefore, the sinusoidal function is incorporated into the quadratic function [1]. The cost function addressing valve-point loadings of generating units is accurately represented as [2]

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 + |e_i \times \sin(f_i \times (P_i^{\min} - P_i))| \tag{4}$$

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

B. PED Problem Considering Multiple Fuels

Any given unit with multiple cost curves needs to operate on the lower contour of the intersecting curves. The resulting cost function is called a "hybrid cost function." Each segment of the hybrid cost function implies some information about the fuel being burned or the unit's operation. Since the dispatching units are practically supplied with multi-fuel sources, each unit should be represented with several piecewise quadratic functions reflecting the effects of fuel type changes, and the generator must identify the most economic fuel to burn. Thus, the fuel cost function should be practically expressed as [8]

$$F_i(P_i) = \begin{cases} a_{i1} + b_{i1} P_i + c_{i1} P_i^2, & \text{fuel 1, } P_i^{\min} \le P_i \le P_{i1} \\ a_{i2} + b_{i2} P_i + c_{i2} P_i^2, & \text{fuel 2, } P_{i1} < P_i \le P_{i2} \\ \vdots & \qquad \vdots \\ a_{ik} + b_{ik} P_i + c_{ik} P_i^2, & \text{fuel } k, P_{ik-1} < P_i \le P_i^{\max} \end{cases} \tag{5}$$

where a_{ik}, b_{ik}, and c_{ik} are cost coefficients of the ith generator using the fuel type k.

C. PED Problem Considering Both Valve-Point Effects and Multiple Fuels

To obtain an accurate and practical economic dispatch solution, the realistic operation of the PED problem should be considered both valve-point effects and multiple fuels. This paper proposed an incorporated cost model, which combines the valve-point loadings and the fuel changes into one frame. Therefore, the cost function should combine (4) with (5), and can be realistically represented as shown in (6), at the bottom of the page.

$$F_i(P_i) = \begin{cases} a_{i1} + b_{i1} P_i + c_{i1} P_i^2 + |e_{i1} \times \sin(f_{i1} \times (P_{i1}^{\min} - P_{i1}))|, & \text{for fuel 1, } P_i^{\min} \le P_i \le P_{i1} \\ a_{i2} + b_{i2} P_i + c_{i2} P_i^2 + |e_{i2} \times \sin(f_{i2} \times (P_{i2}^{\min} - P_{i2}))|, & \text{for fuel 2, } P_{i1} < P_i \le P_{i2} \\ \vdots & \qquad \vdots \\ a_{ik} + b_{ik} P_i + c_{ik} P_i^2 + |e_{ik} \times \sin(f_{ik} \times (P_{ik}^{\min} - P_{ik}))|, & \text{for fuel } k, P_{ik-1} < P_i \le P_i^{\max} \end{cases} \tag{6}$$

Complication of the actual PED problem is due to the incorporated cost model composed of both valve-point effects and multiple fuels. Hence, an algorithm that overcomes these complexities has to be evolved.

III. Proposed Algorithm

This section describes the proposed IGA_MU. First, a brief overview of the IGA is provided, then the MU is presented, and finally the solution procedures of the proposed IGA_MU are stated.

A. IGA

The IGA is a parallel direct search algorithm that uses N_p vectors of variables x in the nonlinear programming problem, namely, $x^G = \{x_i^G, i = 1, \ldots, N_p\}$, as a population in generation G. For convenience, the decision vector (chromosome), x_i, is represented as $(x_{1i} \ldots x_{ji} \ldots x_{n_C i})$. Here, the decision variable (gene), x_{ji} is directly coded as a real value within its bounds.

1) IEDO: The main shortcoming of the evolutionary direction operator (EDO) [13] is that it creates a new chromosome from three arbitrary chromosomes in each generation, making this search operator blind. The improvement of the IEDO is to choose three best solutions in each generation to implement the improved evolutionary direction operation, and then obtain a new solution that is superior to the original best solution. The IEDO is introduced below.

A chromosome which carries a set of solutions with n_c optimizing parameters may be expressed as $x_j = \{C_1, C_2, \ldots, C_p, \ldots, C_{n_C}\}$. Each C_p represents a continuous decision variable, and is limited by its lower and upper bounds (C_p^{MIN} and C_p^{MAX}). Three sets of optimal chromosomes are obtained after a generation. These three preferred chromosomes are ascended according to their fitness and called the "low," "medium," and "high" chromosomes, respectively.

Three inputs (preferred) and the output (created) chromosomes are denoted below.

Inputs:

"low" chromosome, $z_l = \{C_{l1}, C_{l2}, \ldots, C_{in_C}\}$, with fitness F_l

"medium" chromosome, $z_m = \{C_{m1}, C_{m2}, \ldots, C_{mn_C}\}$, with fitness F_m

"high" chromosome, $z_h = \{C_{h1}, C_{h2}, \ldots, C_{hn_C}\}$, with fitness F_h

Output chromosome, $\{C_{o1}, C_{o2}, \ldots, C_{on_C}\}$, with fitness F_{new}

The IEDO can significantly reduce the effort in searching for the optimal solution because it enhances the local searching capability for GA. Fig. 1 shows the flowchart of the minimum optimization for IEDO, and the IEDO procedure is as follows.

Step 1. Set the magnitudes of the two evolution directions to 1 (i.e., $D_1 = 1$, $D_2 = 1$). Then, set the initial index of the IEDO to 1 ($T_s = 1$), and set the number of the IEDO loop to 4 (i.e., $N_L = 4$).

Step 2. Choose three preferred fitness values (F_l, F_m and F_h) and find their associated chromosomes (z_l, z_m, and z_h). Then, obtain three preferred decision variables (C_{lp}, C_{mp}, and C_{hp}, $p = 1, \ldots, n_c$) from these three preferred chromosomes.

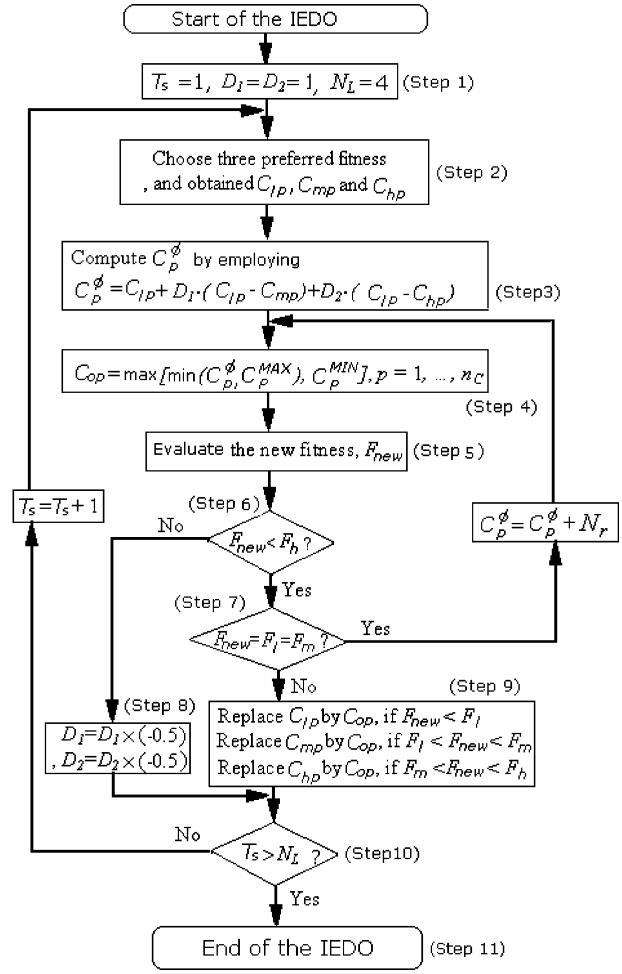

Fig. 1. Flowchart of operation for the IEDO.

Step 3. Compute C_p^{Φ} by

$$C_p^{\Phi} = C_{lp} + D_1 \cdot (C_{lp} - C_{mp}) + D_2 \cdot (C_{lp} - C_{hp}).$$

Starting from the base point C_{lp} and using two difference vectors, $D_1 \cdot (C_{lp} - C_{mp})$ and $D_2 \cdot (C_{lp} - C_{hp})$, the next evolutionary direction and the next evolutionary step-size can be determined by this parallelogram. The point C_p^{Φ} can be then created along the evolutionary direction with the evolutionary step-size.

Step 4. $C_{op} = \max[\min(C_p^{\Phi}, C_p^{\mathrm{MAX}}), 0]$, $p = 1, \ldots, n_c$.

The value of C_{op} must be kept within its set bounds.

Step 5. Evaluate the new fitness (F_{new}) of the newly created output chromosome.

Step 6. If $F_{\mathrm{new}} < F_h$, then go to next step; otherwise, go to Step 8.

Step 7. If $F_{\mathrm{new}} = F_l = F_m$, add a random number ($N_r \in [0,1]$) to C_p^{Φ} and go to Step 4, then recompute F_{new}; otherwise, go to Step 9.

A random number is added to prevent the algorithm from falling into a local optimum.

```
Step 8. Let D1 = D1 × (−0.5), D2 = D2 × (−0.5), then go
to Step 10.
 Use the opposite direction and reduce the half
step-size to search the new solution.
Step 9. Replace Clp by Cop, if Fnew < Fl
 Replace Cmp by Cop, if Fl < Fnew < Fm
 Replace Chp by Cop, if Fm < Fnew < Fh, and go to
Step 10.
 To search a minimum extreme, use three cases of re-
placements mentioned above to choose the best three
individuals, given as zl, zm, and zh for the IEDO op-
eration.
Step 10. If the last iterative loop of the IEDO is
reached, then go to Step 11; otherwise, Ts = Ts + 1,
and go to Step 2.
Step 11. Terminate the IEDO operation.
```

2) Reproduction, Crossover, and Mutation: Three preferred individuals generated by the IEDO are selected for reproduction. Reproduction probabilities of the three chosen individuals are set as follows: the first preferred unit 35%; the second preferred unit 25%, and the third preferred unit 15%. The remainder 25% of population is generated using the randomly created feasible individual. A binomial mutual crossover is adopted to raise the local diversity of individuals. For a small population (e.g., $N_p = 5$), the crossover probability is set to 0.3 which is enough to create new individuals and to avoid high diversity resulting in divergence of the population. The purpose of mutation is to introduce a slight perturbation to increase the diversity of trial individuals after crossover, preventing trial individuals from clustering and causing premature convergence of solution. The probability of mutation is set to 0.03.

3) Migration: A migration is included in the IGA to regenerate a newly diverse population, which prevents individuals from gradually clustering and thus greatly increases the amount of search space explored for a small population. The migrant individuals are generated based on the best individual, $x_b^{G+1} = (x_{1b}^{G+1}, x_{2b}^{G+1}, \ldots, x_{n_{Cb}}^{G+1})$, by nonuniformly random choice. Genes of the ith individual are regenerated according to

$$x_{ki}^{G+1} = \begin{cases} x_{kb}^{G+1} + \rho\left(x_k^L - x_{kb}^{G+1}\right), & \text{if } r_1 < \frac{x_{kb}^{G+1} - x_k^L}{x_k^U - x_k^L} \\ x_{kb}^{G+1} + \rho(x_k^U - x_{kb}^{G+1}), & \text{otherwise} \end{cases} \tag{7}$$

where $k = 1, \ldots, n_C$; $i = 1, \ldots, N_p$, r_1, and ρ are random numbers in the range of [0,1]. The migration may be performed if only the best fitness has not been improved for over 500 generations running, and the migrant population will not only become a set of newly promising solutions but also easily escape the local extreme value trap.

B. MU

Considering the nonlinear programming problem with general constraints as follows:

$$\begin{aligned} &\min_x f(x) \\ \text{subject to}\quad & h_k(x) = 0, \qquad k = 1, \ldots, m_e \\ & g_k(x) \le 0, \qquad k = 1, \ldots, m_i \end{aligned} \tag{8}$$

TABLE I
COMPUTATIONAL PROCEDURES OF THE MU

Step 1. Set the initial iteration $l = 0$. Set initial multiplier, $\nu_k^l = \nu_k^0 = 0, k = 1,\ldots,m_e$, $\upsilon_k^l = \upsilon_k^0 = 0, k = 1,\ldots,m_i$, and the initial penalty parameters, $\alpha_k>0$, $k=1,\ldots,m_e$ and $\beta_k>0$, $k=1,\ldots,m_i$. Set tolerance of the maximum constraint violation, ε_k (e.g. $\varepsilon_k = 10^{32}$), and the scalar factors, $\omega_1 > 1$ and $\omega_2 > 1$.
Step 2. Use a minimization solver, e.g. IGA, to solve $L_a\left(x, \nu^l, \upsilon^l\right)$. Let x_b^l be a minimum solution to the problem $L_a\left(x, \nu^l, \upsilon^l\right)$.
Step 3. Evaluate the maximum constraint violation as $\hat{\varepsilon}_k = \max\{ \max_k \lvert h_k \rvert, \max_k \lvert \max(g_k, -\upsilon_k) \rvert \}$, and establish the following sets of equality and inequality constraints whose violations have not been improved by the factor ω_1: $I_E = \{ k : \lvert h_k \rvert > \frac{\varepsilon_k}{\omega_1}, \quad k = 1, \cdots, m_e \}$ $I_I = \{ k : \lvert \max(g_k, \; -\upsilon_k) \rvert > \frac{\varepsilon_k}{\omega_1}, \quad k = 1, \cdots, m_i \}$
Step 4. If $\hat{\varepsilon}_k \ge \varepsilon_k$, let $\alpha_k = \omega_2 \alpha_k$ and $\nu_k^{l+1} = \nu_k^l / \omega_2$ for all $k \in I_E$, let $\beta_k = \omega_2 \beta_k$ and $\upsilon_k^{l+1} = \upsilon_k^l / \omega_2$ for all $k \in I_I$, and go to step7. Otherwise, go to step 5.
Step 5. Update the multipliers as follows: $\nu_k^{l+1} = h_k\left(x_b^l\right) + \nu_k^l$ $\upsilon_k^{l+1} = \left\langle g_k\left(x_b^l\right) + \upsilon_k^l \right\rangle_+ = \upsilon_k^l + \max\left\{g_k\left(x_b^l\right), -\upsilon_k^l\right\}$
Step 6. If $\hat{\varepsilon}_k \le \varepsilon_k / \omega_1$, let $\varepsilon_k = \hat{\varepsilon}_k$ and go to step 7. Otherwise, let $\alpha_k = \omega_2 \alpha_k$ and $\nu_k^{l+1} = \nu_k^l / \omega_2$ for all $k \in I_E$, and let $\beta_k = \omega_2 \beta_k$ and $\upsilon_k^{l+1} = \upsilon_k^l / \omega_2$ for all $k \in I_I$. Let $\varepsilon_k = \hat{\varepsilon}_k$ and go to step 7.
Step 7. If the maximum iteration reaches, stop. Otherwise, repeat steps 2 to 6.

where x represents a n_C-dimensional variable, and the $h_k(x)$ and $g_k(x)$ stand for equality and inequality constraints, respectively.

The penalty function method is frequently applied to manage constraints in evolutionary algorithms. Such a technique converts the primal constrained problem into an unconstrained problem by penalizing constraint violations. The penalty function method is simple in concept and implementation. However, its primal limitation is the degree to which each constraint is penalized. These penalty terms have certain weaknesses that become fatal when penalty parameters are large. Such a penalty function tends to be ill conditioned near the boundary of the feasible domain where the optimum point is usually located.

The Lagrange method can markedly overcome the drawbacks of the penalty method. The augmented Lagrange function [15] for constrained optimization problems is defined as

$$L_a(x, \nu, \upsilon) = f(x) + \sum_{k=1}^{m_e} \alpha_k \{[h_k(x) + \nu_k]^2 - \nu_k^2\} + \sum_{k=1}^{m_i} \beta_k \{\langle g_k(x) + \upsilon_k \rangle_+^2 - \upsilon_k^2\} \tag{9}$$

where α_k and β_k are the positive penalty parameters, and the corresponding Lagrange multipliers $\nu = (\nu_1, \ldots, \nu_{m_e})$ and $\upsilon = (\upsilon_1, \ldots, \upsilon_{m_i}) \ge 0$ are associated with equality and inequality constraints, respectively.

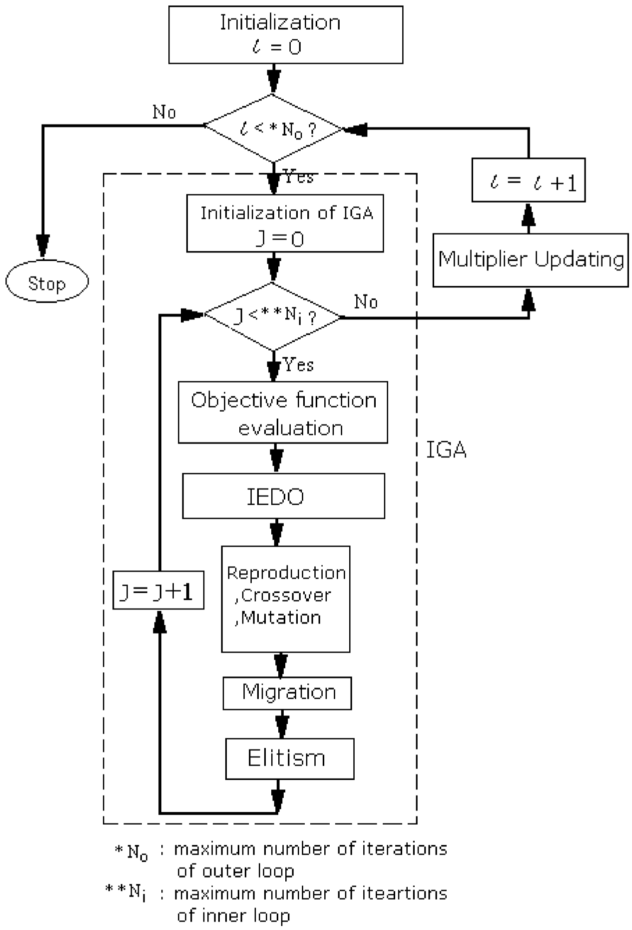

Fig. 2. Flowchart of the IGA_MU. (*N_o : maximum number of iterations of outer loop. **N_i : maximum number of iteartions of inner loop.).

The contour of the augmented Lagrange function does not change shape from generation to generation while constraints are linear. Therefore, the contour of the augmented Lagrange function is simply shifted or biased in relation to that of the original objective function, $f(x)$. Consequently, small penalty parameters can be used in the MU. However, the shape of contour of L_a is changed by penalty parameters while the constraints are nonlinear, demonstrating that the computational difficulties of using large penalty parameters for nonlinear constraints remain. The adaptive penalty parameters are employed to alleviate the above difficulties. Table I presents computational procedures of the MU. The original objective function can be scaled to avoid the ill conditioning by updating penalty parameters and multipliers. Steps 3, 4, and 6 are adopted to reform the constraint violation and update penalty parameters. Step 3 evaluates the maximum constraint violation $\hat{\varepsilon}_k$, which is the maximum value between the maximum equality constraint violation ($|h_k|$, $k = 1, \ldots, m_e$) and the maximum inequality constraint violation ($|\max(g_k, -\upsilon_k)|$, $k = 1, \ldots, m_i$). In Step 4, if constraint violations have not been improved, such that $\hat{\varepsilon}_k \geq \varepsilon_k$, the penalty parameters are increased by a factor ω_2 (e.g., $\omega_2 = 10$, in this paper) and the multipliers are reduced by the same factor, maintaining the product of penalty parameters time multipliers unchanged. Step 6 uses factor ω_1 (e.g., $\omega_1 = 4$, in this paper) to check whether this factor reduces the level of constraint violation. The penalty parameters and multipliers are updated in this step if ω_1 does not reduce the constraint violation.

TABLE II
COMPARED RESULTS OF PREVIOUS METHODS, THE CGA_MU, AND THE PROPOSED IGA_MU FOR EXAMPLE 1

Methods	FEP	MFEP	IFEP	PSO-SQP	CGA_MU	IGA_MU
Min. cost ($)	18018.0	18028.09	17994.07	17969.93	17975.3437	17963.9848

TABLE III
COMPARED RESULTS OF THE CGA_MU AND THE PROPOSED IGA_MU FOR EXAMPLE 1

Methods / Generations	CGA_MU	IGA_MU
Unit1	448.7990	628.3151
Unit2	302.5353	148.1027
Unit3	299.1993	224.2713
Unit4	109.8666	109.8617
Unit5	60.0000	109.8637
Unit6	109.8666	109.8643
Unit7	109.8666	109.8550
Unit8	60.0000	109.8662
Unit9	109.8666	60.0000
Unit10	40.0000	40.0000
Unit11	40.0000	40.0000
Unit12	55.0000	55.0000
Unit13	55.0000	55.0000
P_d(MW)	1800	
TP (MW)	1800.0000	1800.0000
TC ($)	17975.3437	17963.9848
Time (s)	27.91	8.27

C. Solution Procedures of the Proposed IGA_MU

A benefit of the proposed approach is that the augmented Lagrange function can be scaled to avoid the ill-condition leading to difficulty in finding a solution. Fig. 2 depicts the flow chart of the proposed algorithm, which has two iterative loops. The augmented Lagrange function is solved for a minimum value in the inner loop with the given penalty parameters and multipliers, which are then updated in the outer loop toward obtaining an upper bound of L_a. When both inner and outer iterations become sufficiently large, the augmented Lagrange function converges to a saddle-point of the dual problem [15].

IV. SIMULATION RESULTS

This section employs three examples to illustrate the effectiveness of the proposed IGA_MU with respect to the quality of the solution obtained. The first example considers only the valve-point loadings; the second example considers only multi-fuel effects, and the last example considers both valve-point effects and multiple fuels. The MU algorithm was used in the CGA for all examples to manage the system constraints, and the Elitism technique [12] was also adopted in the proposed IGA_MU and the CGA_MU. Both IGA_MU and CGA_MU were directly coded using real values, and were implemented on a personal computer (PIII-700) in FORTRAN-90. [15]–[17], and [18] suggest setting factors for the proposed algorithm. The setting factors used in these examples were as follows: the iteration number of the IEDO operation N_L was set to 4, and the population size N_p was, respectively, set to 5 and 30 for the IGA_MU and CGA_MU in each example. The iteration numbers of the outer loop and inner loop were set to (outer, inner) as

TABLE IV
COMPARED RESULTS OF PREVIOUS METHODS, THE CGA_MU, AND THE PROPOSED IGA_MU FOR EXAMPLE 2

Methods	HM		HNN		AHNN		EP		CGA_MU		IGA_MU	
Unit	Fuel type	Gen	Fuel type	Gen	Fuel type	Gen	Fuel type	Gen	Fuel type	Gen	Fuel type	Gen
1	2	218.4	2	224.5	2	228.2	2	225.2	2	218.4572	2	218.1248
2	1	211.8	1	215.0	1	214.8	1	215.6	1	211.5140	1	211.6826
3	1	281.0	3	291.8	1	291.7	1	291.8	1	280.8987	1	280.8630
4	3	239.7	3	242.2	3	242.3	3	242.1	3	239.6241	3	239.6533
5	1	279.0	1	293.3	1	293.3	1	293.7	1	278.5036	1	278.6304
6	3	239.7	3	242.2	3	242.2	3	241.9	3	239.6390	3	239.6140
7	1	289.0	1	303.1	1	302.3	1	301.6	1	288.6201	1	288.5725
8	3	239.7	3	242.2	3	242.3	3	242.8	3	239.6211	3	239.7057
9	3	429.2	1	335.7	1	354.2	1	356.6	3	428.5760	3	428.4542
10	1	275.2	1	289.5	1	288.9	1	288.7	1	274.5462	1	274.6995
TP (MW)	2702.2		2699.7		2700.0		2700.0		2700.0000		2700.0000	
TC ($)	625.18		626.12		626.24		626.26		623.8095		623.8093	
Time(s)	-		-		-		-		19.42		5.27	

(30, 3000) for all examples, except that the large-scale system test of the Example 3 were set to (50, 3000).

A. Example 1

To address the accurate cost curve of each generating unit, the valve-point effects must be incorporated into the cost model. The first test system consisted of 13 units taking account of only valve-point loadings. The accurate cost model using (4) considers valve-point effects; the sinusoidal function is included in the quadratic cost function, and the system data are the same as those in [2]. For the purpose of comparing previous methods with the same situations, Examples 1–3 do not consider the transmission loss. The proposed algorithm is simple to understand and implement. The implementation of this example can be described as follows:

$$L_a(x,\nu,\upsilon) = f(x) + \alpha_1\{[h_1(x)+\nu_1]^2 - \nu_1^2\} + \sum_{k=1}^{26}\beta_k\{\langle g_k(x)+\upsilon_k\rangle_+^2 - \upsilon_k^2\} \quad (10)$$

$$\text{objective}: \min_{x=(P_1,P_2,\ldots,P_{13})} f(x) = \sum_{i=1}^{13} F_i(P_i) \quad (11)$$

$$\text{subject to} \quad \begin{aligned} h_1 &: \sum_{i=1}^{13} P_i - P_d = 0 \\ g_1 &: P_1 - P_1^{\max} \le 0 \\ &\vdots \\ g_{13} &: P_{13} - P_{13}^{\max} \le 0 \\ g_{14} &: P_1^{\min} - P_1 \le 0 \\ &\vdots \\ g_{26} &: P_{13}^{\min} - P_{13} \le 0. \end{aligned} \quad (12)$$

This minimum cost problem includes one objective function with 13 variable parameters ($P_1, P_2, \ldots, P_{13}$), one equality constraint (h_1), and 26 inequality constraints (g_1 to g_{26}). This example compares the proposed IGA_MU with three EPs (FEP: a fast EP; MFEP: a FEP using the weighted mean of Gaussian and Cauchy mutations, and IFEP: an improved FEP) [2], PSO-SQP [7], and the CGA_MU in terms of the best dispatch solution for a load demand of 1800 MW. The compared results are listed in Table II, where TP and TC are the total power and total cost. The proposed IGA_MU has the lowest cost of all methods tested, demonstrating that the proposed algorithm is more effective than other methods for the PED problem. The best solutions of the CGA_MU and IGA_MU are provided in Table III, which shows that the proposed IGA_MU has lower cost and simulation time than the CGA_MU. Furthermore, considering the accurate cost model, the proposed algorithm not only has the best economic dispatch of all methods, but also completely satisfies the system constraints. Therefore, the proposed IGA_MU is clearly more efficient and effective than the CGA_MU.

B. Example 2

To address the practical operation of units with multiple fuel options, the optimal economic dispatch is obtained only if each unit uses the most economic fuel to burn. This example comprised ten generating units considering only multiple fuels. The cost function using (5) considers multiple fuels. The system data and related constraints of this example are given in [8]. The implementations of the proposed algorithm for the second example are similar to the first example, but this economic cost problem has one objective function with ten variable parameters ($P_1, P_2, \ldots, P_{10}$), one equality constraint (h_1), and 20 inequality constraints (g_1 to g_{20}). This example compares IGA_MU with HM [8], HNN [9], AHNN [10], EP [11], and the CGA_MU in terms of production cost for a load demand of 2700 MW. Table IV provides the comparison, showing that the fuel types and dispatch levels from the proposed IGA_MU are quite different from those of previous approaches. The proposed algorithm also yields better solution quality than other methods, and is more efficient and effective than the CGA_MU in the PED problem. The comparisons in Tables II and IV, clearly reveal that the proposed IGA_MU is more effective than previous methods in applying the practical PED problem.

C. Example 3

1) Incorporated Model Test: Few attempts have been made at considering both valve-point effects and multiple fuels for the realistic PED problem, which always exist in real power systems simultaneously. Herein, the proposed algorithm was employed to solve the accurate and practical PED problem with the incorporated cost model (6). This test system contained ten dispatching units addressing both valve-point effects and multiple

TABLE V
SYSTEM DATA OF UNITS CONSIDERING BOTH VALVE-POINT EFFECTS AND MULTIPLE FUELS FOR EXAMPLE 3

Unit	Generation Min P_1 P_2 Max F1 F2 F3	Fuel type	Cost coefficients a_i	b_i	c_i	e_i	f_i
1	100 196 250 1 2	1 2	.2697e2 .2113e2	-.3975e0 -.3059e0	.2176e-2 .1861e-2	.2697e-1 .2113e-1	-.3975e1 -.3059e1
2	50 114 157 230 2 3 1	1 2 3	.1184e3 .1865e1 .1365e2	-.1269e1 -.3988e-1 -.1980e0	.4194e-2 .1138e-2 .1620e-2	.1184e0 .1865e-2 .1365e-1	-.1269e2 -.3988e0 -.1980e1
3	200 332 388 500 1 3 2	1 2 3	.3979e2 -.5914e2 -.2875e1	-.3116e0 .4864e0 .3389e-1	.1457e-2 .1176e-4 .8035e-3	.3979e-1 -.5914e-1 -.2876e-2	-.3116e1 .4864e1 .3389e0
4	99 138 200 265 1 2 3	1 2 3	.1983e1 .5285e2 .2668e3	-.3114e-1 -.6348e0 -.2338e1	.1049e-2 .2758e-2 .5935e-2	.1983e-2 .5285e-1 .2668e0	-.3114e0 -.6348e1 -.2338e2
5	190 338 407 490 1 2 3	1 2 3	.1392e2 .9976e2 -.5399e2	-.8733e-1 -.5206e0 .4462e0	.1066e-2 .1597e-2 .1498e-3	.1392e-1 .9976e-1 -.5399e-1	-.8733e0 -.5206e1 .4462e1
6	85 138 200 265 2 1 3	1 2 3	.5285e2 .1983e1 .2668e3	-.6348e0 -.3114e-1 -.2338e1	.2758e-2 .1049e-2 .5935e-2	.5285e-1 .1983e-2 .2668e0	-.6348e1 -.3114e0 -.2338e2
7	200 331 391 500 1 2 3	1 2 3	.1893e2 .4377e2 -.4335e2	-.1325e0 -.2267e0 .3559e0	.1107e-2 .1165e-2 .2454e-3	.1893e-1 .4377e-1 -.4335e-1	-.1325e1 -.2267e1 .3559e1
8	99 138 200 265 1 2 3	1 2 3	.1983e1 .5285e2 .2668e3	-.3114e-1 -.6348e0 -.2338e1	.1049e-2 .2758e-2 .5935e-2	.1983e-2 .5285e-1 .2668e0	-.3114e0 -.6348e1 -.2338e2
9	130 213 370 440 3 1 3	1 2 3	.8853e2 .1530e2 .1423e2	-.5675e0 -.4514e-1 -.1817e-1	.1554e-2 .7033e-2 .6121e-3	.8853e-1 .1423e-1 .1423e-1	-.5675e1 -.1817e0 -.1817e0
10	200 362 407 490 1 3 2	1 2 3	.1397e2 -.6113e2 .4671e2	-.9938e-1 .5084e0 -.2024e0	.1102e-2 .4164e-4 .1137e-2	.1397e-1 -.6113e-1 .4671e-1	-.9938e0 .5084e1 -.2024e1

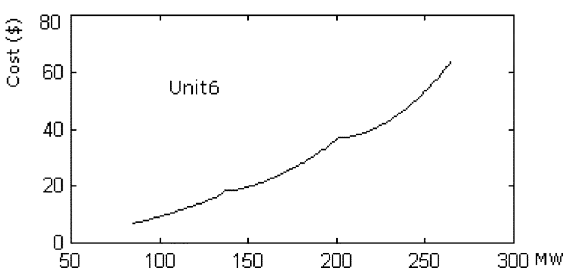

Fig. 3. Curve of unit 6 has both valve-point effects and multiple fuel changes.

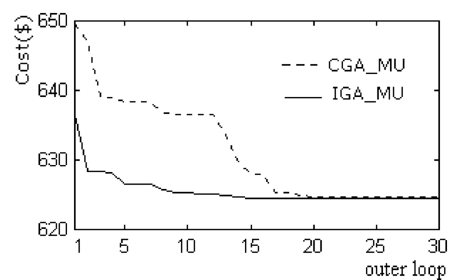

Fig. 4. Convergence natures of the CGA_MU and the proposed IGA_MU.

TABLE VI
COMPARED RESULTS OF THE CGA_MU AND THE PROPOSED IGA_MU FOR EXAMPLE 3

Methods	CGA_MU		IGA_MU	
Unit	Fuel type	Gen.	Fuel type	Gen.
1	2	222.0108	2	219.1261
2	1	211.6352	1	211.1645
3	1	283.9455	1	280.6572
4	3	237.8052	3	238.4770
5	1	280.4480	1	276.4179
6	3	236.0330	3	240.4672
7	1	292.0499	1	287.7399
8	3	241.9708	3	240.7614
9	3	424.2011	3	429.3370
10	1	269.9005	1	275.8518
P_d (MW)	2700			
TP (MW)	2700.0000		2700.0000	
TC ($)	624.7193		624.5178	
Time (s)	26.17		7.25	

TABLE VII
COMPARISONS OF THE CGA_MU AND THE PROPOSED IGA_MU ON RELATIVE FREQUENCY OF CONVERGENCE IN THE RANGES OF COST

Methods	Range of cost ($) 634.5 - 633.5	633.5 - 632.5	632.5 - 631.5	631.5 - 630.5	630.5 - 629.5	629.5 - 628.5	628.5 - 627.5	627.5 - 626.5	626.5 - 625.5	625.5 - 624.5
CGA_MU	1	0	2	3	7	10	21	31	20	5
IGA_MU	0	0	0	1	0	2	2	11	45	39

TABLE VIII
SUMMARIZED RESULTS OF THE COMPARISON BETWEEN THE CGA_MU AND THE PROPOSED IGA_MU

Methods	Mean time (s)	Best time (s)	Mean cost ($)	Max. cost ($)	Min. cost ($)
CGA_MU	26.64	25.65	627.6087	633.8652	624.7193
IGA_MU	7.32	7.14	625.8692	630.8705	624.5178

fuels for a load demand of 2700 MW. The system data of this example derived mainly from [8] and replenished the cost coefficients (e_i and f_i) of sinusoidal functions for each unit, which are listed in Table V. The characteristic unit curves have nondifferentiable points according to both valve-point loadings and fuel changes. For example, the generating unit 6 clearly appears as a

TABLE IX
RESULTS OF THE PROPOSED IGA_MU WITH VARIOUS INITIAL PENALTY PARAMETERS FOR EXAMPLE 3

Initial penalty	Constraints	Final penalty and multiplier of L_a				IGA_MU			
	h_1 and $g_1 \sim g_{20}$	α_1	ν_1	$\beta_1 \sim \beta_{20}$	$\upsilon_1 \sim \upsilon_{20}$	Time(s)	*SCV*	TP (MW)	TC ($)
10	h_1= 0.0000	1.0E8	0.0000	-	-	7.25	0.0000	2700.0000	624.5178
	$g_1 \sim g_{20}$=0.0000	-	-	1.0E8	0.0000				
10^3	h_1=0.0000	1.0E14	0.0000	-	-	7.34	0.0000	2700.0000	624.5178
	$g_1 \sim g_{20}$=0.0000	-	-	1.0E12	0.0000				
10^6	h_1=0.0000	1.0E16	0.0000	-	-	7.31	0.0000	2700.0000	624.5178
	$g_1 \sim g_{20}$=0.0000	-	-	1.0E14	0.0000				

piecewise nonsmooth cost function due to both the valve-point effects and multiple fuels, which is shown in Fig. 3.

The implementations of the proposed algorithm of this example are identical to the second example. The aim of this example is to demonstrate the effectiveness of the proposed IGA_MU for the realistic PED problem. The comparisons between the proposed algorithm and the CGA_MU are illustrated in Table VI, and the proposed IGA_MU exhibits not only better solution quality but also less computational time than the CGA_MU. Fig. 4 presents the convergence natures of the CGA_MU and the proposed IGA_MU. Although the MU has efficaciously tackled system constraints, the CGA_MU with a large population ($N_p = 30$) did not discover the global solution, whereas the proposed IGA_MU discovered the global solution only using a small population ($N_p = 5$) within 15 outer loops. These findings prove clearly that the proposed IGA_MU is more efficient and effective than the CGA_MU for the actual PED problem.

2) Robustness Analysis: To show the robustness of the proposed IGA_MU for the realistic PED problem, the relative frequency of convergence for the proposed algorithm and the CGA_MU are listed in Table VII for each cost range among 100 randomly initiated trials. The compared results are also itemized in Table VIII. Table VII reveals that the proposed IGA_MU has provided the global solution with a high probability to demonstrate its effectiveness and efficiency. Moreover, as shown in Table VIII, the proposed IGA_MU has both a better economic cost and lower mean time than the CGA_MU, even if the proposed algorithm uses a small population. Therefore, Tables VII and VIII clearly show that the proposed IGA_MU is more robust than the CGA_MU.

3) Penalty Setting Test: This test compared the proposed IGA_MU with IGA using the fixed penalty (IGA_FP) in terms of penalty setting, illustrates that the IGA_MU has the advantage of automatically adjusting the randomly given penalty to a proper value. Herein, the IGA_MU with initial penalty parameters of 10, 10^3 and 10^6, was employed to solve Example 3 again, and the sum of constraint violation was defined as $SCV = |h_1| + \sum_{k=1}^{20} \max\{\, g_k\,,\ \ 0.0\,\}$, to elucidate the effect of constraint feasibility on the final solution. Table IX provides three cases of the best solutions obtained among 100 randomly initiated trials in each case. The adaptive penalty parameters and multipliers (α_k, ν_k, β_k, and υ_k) were automatically updated by the MU to avoid ill conditioning. Experimental results of these three sets all yielded the same total cost and the same total power, and SCVs less than 10^{-12}, confirming that the IGA_MU is effective in handling the realistic PED operation by automatically adjusting the randomly given penalty to a proper value and requiring only a small population.

TABLE X
RESULTS OF THE IGA_FP WITH VARIOUS FIXED PENALTY PARAMETERS FOR EXAMPLE 3

Fixed penalty	Constraints	IGA_FP			
	h_1 and $g_1 \sim g_{20}$	Time(s)	*SCV*	TP (MW)	TC ($)
10	h_1= 0.021872 $g_1 \sim g_{20}$=0.0000	7.03	0.021872	2699.9781	623.9232
10^3	h_1=0.000812 $g_1 \sim g_{20}$=0.0000	6.92	0.000812	2699.99992	624.4557
10^6	h_1=0.000000 $g_1 \sim g_{20}$=0.0000	6.97	0.0000	2700.0000	624.5960

The IGA_FP was also applied to solve this example. The IGA_FP, like the IGA_MU, uses 90 000 iterations. Table X lists three cases of the best solutions obtained for various fixed penalty parameters among 100 randomly initiated trials in each case. For fixed penalty parameters of 10 and 10^3, the total costs obtained by the IGA_FP are below those obtained by the IGA_MU. Nevertheless, such comparison is meaningless, since all SCVs of the IGA_FP are greater than those of the IGA_MU. The two IGA_PF cases violated the constraints of the system, and produced infeasible solutions. This test clearly indicates that the IGA_MU can automatically adjust the randomly given penalty to a proper value to obtain a global solution. However, the IGA_FP can generate a satisfactory result only using a large penalty parameter. Because the IGA_MU actively adopts a flexible multiplier updating strategy to probe for the new solution, while the IGA_FP only uses a fixed penalty function to test passively feasible and infeasible solutions.

4) Large-Scale System Test: This test shows that the proposed approach is also applicable to the large-scale system of the actual PED problem. Example 3 was expanded to build four systems with 20, 40, 80, and 160 units. To avoid confusion owing to the stochastic nature of the solution methods, computational results of 50 runs beginning with random initial populations were averaged, and are given in Table XI. This table reveals that the proposed approach is superior to the CGA_MU in terms of average cost (Cost_av) and average CPU time (CPU_av) in all cases. The advantage of the proposed algorithm is mostly in the solution quality and calculation speed, and its superiority becomes clearer as the system size is enlarged. Moreover, the proposed method requires a smaller population size than the CGA_MU. Significantly, the computational burden of the proposed algorithm grows almost linearly with respect to the increase in system size, showing that proposed approach is applicable to large-scale PED problems with the accurate and realistic cost model.

TABLE XI
COMPUTATIONAL RESULTS OF THE CGA_MU AND THE PROPOSED IGA_MU FOR THE LARGE-SCALE PED PROBLEM OF EXAMPLE 3

Methods	CGA_MU, (N_p =30)				IGA_MU, (N_p =5)			
No. of units	20	40	80	160	20	40	80	160
Cost_av ($)	1249.3893	2500.9220	5008.1426	10143.7263	1249.1179	2499.8243	5003.8832	10042.4742
CPU_av (s)	80.48	157.39	309.41	621.30	21.64	43.71	85.67	174.62
#ΔCost_av(%)	0.0217	0.0439	0.0851	1.0082	-	-	-	-

$$\#\Delta\text{Cost\_av}(\%) = \frac{\text{Cost\_av of the CGA\_MU} - \text{Cost\_av of the proposed IGA\_MU}}{\text{Cost\_av of the proposed IGA\_MU}} \times 100\%$$

V. DISCUSSIONS

A. Discussions

The aim of this investigation is to explore further the PED problem with the accurate and practical unit cost model. The valve-point loadings and multiple fuel changes are integrated into a frame, an algorithm for solving the PED operation is introduced. Previous papers have not considered valve-point effects, probably because they focused on their algorithms and lost the accuracy by approximating the realistic cost curve. However, many approaches [1]–[7] have addressed the valve-point effects to solve the accurate PED problem.

For the large-scale PED problem, EP-based methods might become trapped in a suboptimal state, where the variation operators cannot produce any offspring outperforming its parents; involved a large number of iterations, and was susceptible to the related control parameters. Nevertheless, the proposed algorithm has been tested and compared to demonstrate its effectiveness in solving the large-scale PED problem.

The proposed algorithm was applied successfully to solve the nonconvex PED problem [17] of units with prohibited operation zones, and can manage complex constraints composed of both equalities and inequalities of the nonconvex system.

Even if the system has only one equality constraint, a lower penalty value may not be enough to make the problem feasible. In Table X, the result obtained by using the lower penalty 10 is infeasible, because of the SCV associated with this case is 0.021 872, which is far greater than that of the other two cases. Such a large SCV implies that the result seriously violates the equality constraint. The fixed penalty method is selective in the penalty parameter. However, the MU algorithm can automatically adjust the penalties to handle the system constrains.

B. Discussion Case

This study utilizes a discussion case to compare the DP approach with the proposed algorithm. This test system comprised three generating units with third-order cost functions for a load demand of 1400 MW. The system data are in [21], and the transmission loss P_L is denoted by the B-coefficient method [4]. Table XII lists dispatching results obtained from DP, two simulated annealing (SA) methods (SA1 and SA2) [22], the CGA_MU, the proposed IGA_MU and the IGA_FP using the fixed penalty of 10^6. Obviously, this system has two local optima (about 6639 and 6642), and DP and SA1 are likely to become stuck in a locally solution. Conversely, the proposed algorithm using both the IGA and MU can effectively obtain a better solution than previous methods (DP, SA1, and SA2), the CGA_MU and the IGA_FP.

TABLE XII
COMPARED RESULTS OF PREVIOUS METHODS, THE CGA_MU, THE PROPOSED IGA_MU, AND THE IGA_FP FOR A DISCUSSION CASE

Methods	DP	SA1	SA2	CGA_MU	IGA_MU	IGA_FP
Unit1	360.2	359.546	376.123	365.8716	365.4085	365.3819
Unit2	406.4	406.734	100.052	100.0013	100.0000	100.0000
Unit3	676.8	677.152	986.273	996.8656	997.3436	997.3712
TP(MW)	1443.4	1443.434	1462.448	1462.7385	1462.7521	1462.7531
P_L(MW)	43.4	43.434	62.448	62.7385	62.7521	62.7530
TC($)	6642.459	6642.657	6639.504	6639.1894	6639.1849	6639.1857

VI. CONCLUSION

Few reports have been made at units considering both valve-point effects and multiple fuels for the realistic PED problem, which always exist in real power systems simultaneously. Herein, this paper proposes the IGA_MU to solve the practical PED operation. The realistic PED problem is complicated because both valve-point loadings and change fuels must be considered. The IGA helps the proposed algorithm to efficiently search and actively explore the solution, and the MU helps the proposed method to efficaciously handle constraints of the system. One of the contributions of this research is the inclusion of valve-point loadings and multi-fuel sources into a frame. The other contribution of this paper is the combination of the IGA and the MU to solve the PED problem.

Moreover, the proposed approach has the following merits: simple concept; easy implementation; better effectiveness than previous methods; better efficiency than the CGA_MU; robustness of algorithm; applicable to the larger-scale system; automatic adjustment of the randomly assigned penalty to a proper value, and the requirement for only a small population in the accurate and practical PED problem. The comparative results demonstrate that the proposed algorithm has the advantages mentioned above for solving the real-world PED problem.

REFERENCES

[1] 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.

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

[3] 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.

[4] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed. New York: Wiley, 1996.

[5] 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.

[6] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, "A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function," *IEEE Trans. Power Syst.*, vol. 17, no. 2, pp. 411–416, May 2002.

[7] T. A. A. Victoire and A. E. Jeyakumar, “Hybrid PSO-SQP for economic dispatch with valve-point effect,” *Elect. Power Syst. Res.*, vol. 71, pp. 51–59, 2004.
[8] C. E. Lin and G. L. Viviani, “Hierarchical economic dispatch for piecewise quadratic cost functions,” *IEEE Trans. Power App. Syst.*, vol. PAS-103, pp. 1170–1175, 1984.
[9] 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.
[10] K. Y. Lee, A. Sode-Yome, and J. H. Park, “Adaptive Hopfield neural network for economic load dispatch,” *IEEE Trans. Power Syst.*, vol. 13, no. 2, pp. 519–526, May 1998.
[11] T. Jayabarathi and G. Sadasivam, “Evolutionary programming-based economic dispatch for units with multiple fuel options,” *Eur. Trans. Elect. Power*, vol. 10, no. 3, pp. 167–170, 2000.
[12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[13] K. Yamamoto and O. Inoue, “New evolutionary direction operator for genetic algorithms,” *AIAA J. Tech. Notes*, vol. 33, no. 10, pp. 1990–1993, 1995.
[14] J. P. Chiou and F. S. Wang, “A hybrid method of differential evolution with application to optimal control problems of a bioprocess system,” in *Conf. Rec. IEEE Int. Conf. Evolutionary Computation*, 1998, pp. 627–632.
[15] C. L. Chiang, C. T. Su, and F. S. Wang, “Augmented Lagrangian method for evolutionary optimization of mixed-integer nonlinear constrained problems,” *Int. Math. J.*, vol. 2, no. 2, pp. 119–154, 2002.
[16] C. T. Su and C. L. Chiang, “Optimal position /speed control of induction motor using improved genetic algorithm and fuzzy phase plane controller,” *Control Intell. Syst.*, vol. 32, no. 2, pp. 104–115, 2004.
[17] ——, “Nonconvex power economic dispatch by improved genetic algorithm with multiplier updating method,” *Elect. Power Compon. Syst.*, vol. 32, no. 3, pp. 257–273, 2004.
[18] ——, “An incorporated algorithm for combined heat and power economic dispatch,” *Elect. Power Syst. Res.*, vol. 69, pp. 187–195, 2004.
[19] Z. Michalewicz and M. Schoenauer, “Evolutionary algorithms for constrained parameter optimization problems,” *Evolut. Comput.*, vol. 4, no. 1, pp. 1–32, 1996.
[20] M. J. D. Powell, “Algorithms for nonlinear constraints that use Lagrangian function,” *Math. Program.*, vol. 14, pp. 224–248, 1978.
[21] 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.
[22] K. P. Wong and C. C. Fung, “Simulated annealing based economic dispatch algorithm,” *Proc. Inst. Elect. Eng. C*, vol. 140, no. 6, pp. 509–515, 1993.

Chao-Lung Chiang received the M.S. degree from the Department of Automatic Control Engineering, Feng Chia University, Taichung, Taiwan, R.O.C., in 1991 and the Ph.D. degree from the Institute of Electrical Engineering, National Chung Cheng University, Chia-Yi, Taiwan, in 2004.

He is currently an Associate Professor at the Nan Kai Institute of Technology, Nan-Tou, Taiwan. His research interests include control theory, applications of optimization, evolutionary algorithms, and power economic dispatch.