Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm

S.-K. Wang, J.-P. Chiou and C.-W. Liu

Abstract: This paper presents a novel stochastic optimisation approach to determining the feasible optimal solution of the economic dispatch (ED) problem considering various generator constraints. Many practical constraints of generators, such as ramp rate limits, prohibited operating zones and the valve point effect, are considered. These constraints make the ED problem a non-smooth/non-convex minimisation problem with constraints. The proposed optimisation algorithm is called self-tuning hybrid differential evolution (self-tuning HDE). The self-tuning HDE utilises the concept of the 1/5 success rule of evolution strategies (ESs) in the original HDE to accelerate the search for the global optimum. Three test power systems, including 3-, 13- and 40-unit power systems, are applied to compare the performance of the proposed algorithm with genetic algorithms, the differential evolution algorithm and the HDE algorithm. Numerical results indicate that the entire performance of the proposed self-tuning HDE algorithm outperforms the other three algorithms.

1 Introduction

Economic dispatch (ED) is an important optimisation task in power system operation for allocating generation among the committed units. Its objective is to minimise the total generation cost of units, while satisfying the various physical constraints. Because of physical limitations of the power generators, a generating unit may have prohibited operating zones between the minimum and maximum power outputs. Generators that operate in these zones may experience amplification of vibrations in their shaft bearings, which should be avoided in practical application. On the other hand, due to the fact that unit generation output cannot be changed instantaneously, the unit in the actual operating processes is restricted by its ramp rate limit [1, 2]. Moreover, the units of real input–output characteristics include higher order nonlinearities and discontinuities owing to the valve point effect, which has been modelled as a circulating commutated sinusoidal function in [3, 4]. The ED problem with the above considerations is usually a non-smooth/non-convex optimisation problem [1]. This kind of optimisation problem is very hard, if not impossible, to solve using traditionally deterministic optimisation algorithms. Therefore we propose a stochastic optimisation approach. A brief survey of proposed algorithms for the ED problem in the open literature is given as follows, but it is far from complete.

The Lagrangian multiplier method [5], which is generally used in the ED problem, is no longer directly applicable. To solve such a non-smooth/non-convex ED problem, Lee and Breipohi [1] decomposed the non-convex decision space into a number of convex sub-regions and then used the Lagrangian multiplier method to solve the problem. Nevertheless, this may require a large computational burden to obtain an optimal solution when a system has several units with prohibited zones. Fan and McDonald [6] proposed an algorithm based on conventional $\lambda - \delta$ iterative dispatch to obtain the solution. Su and Chiou [7] applied the Hopfield network approach to solve the ED with prohibited operating zones, but the Hopfield network method requires two-phase computations and cannot consider power loss. Lin et al. [8] presented integrated evolutionary programming, tabu search (TS) and quadratic programming (QP) methods to solve non-convex ED problems. This integrated artificial intelligence method also requires two-phase computations. Lin et al. [9] developed an improved TS algorithm for ED with non-continuous and non-smooth cost functions, but the prohibited zones and system spinning reserve are relaxed in this work. Sewtohul et al. [10] proposed genetic algorithms (GAs) to solve the ED problem in light of the valve point effect. Wong and Fung [11] developed a simulated annealing-based ED algorithm to solve ED considering the transmission loss. Sinha et al. [12] used an evolutionary programming (EP) method to solve ED problems. However, the last two studies did not consider the prohibited zones. Gaing [13, 14] proposed a particle swarm optimisation (PSO) method for solving the ED problems in power systems.

Hybrid differential evolution (HDE) [15, 16] is a stochastic optimisation method. The fittest of an offspring competes one by one with that of the corresponding parent, which is different from the other evolutionary algorithms (EAs). This competition implies that the parent is replaced by its offspring if the fitness of the offspring is better than that of its parent. On the other hand, the parent is retained in the next generation if the fitness of the offspring is worse than that of its parent. This one by one competition gives rise to a faster convergence rate. However, this faster convergence also leads to a higher probability of obtaining a local optimum because the diversity of the population descends faster during the solution process. To overcome this drawback, the migrating operator and the accelerated operator act as trade-off operators for the diversity of population and convergence properties in HDE. The migrating operator

IET Gener. Transm. Distrib., 2007, **1**, (5), pp. 793–803

793

maintains the diversity of the population, which guarantees a high probability of obtaining the global optimum. The accelerated operator is used to accelerate convergence. However, a fixed scaling factor is used in HDE. Using a smaller scaling factor, HDE becomes increasingly robust, but much computational time must be expanded to evaluate the objective function. HDE with a larger scaling factor generally produces a local solution or divergence. Lin et al. [17] used a random number with a value between zero and one as a scaling factor. However, a random scaling factor could not guarantee the fast convergence.

In this study, the self-tuning HDE for solving the non-smooth/non-convex ED problems considering the various physical constraints is proposed. Different from the HDE, Np scaling factors based on the 1/5 success rule of evolution strategies (ESs) [18, 19] are used in the self-tuning HDE method to accelerate the search for the global solution. That is, every individual has a corresponding scaling factor. According to the convergence property of every individual, the corresponding scaling factor is adjusted based on the 1/5 success rule of ESs to accelerate the search for the global solution. The 3-, 13-, and 40-unit ED systems from the literature are used to compare the performance of the proposed method with other stochastic optimisation methods, such as GAs, differential evolution (DE) and HDE.

2 Problem formulation

The objective of the ED is to minimise the total generation cost of a power system over some appropriate period while satisfying various constraints. The power system balance of conditions for system demand, power losses and entire generator power, as well as the generating power constraints, including prohibited zones, ramp rate limit, spinning reserve and valve point effect for all units, should be satisfied. In this paper, some unit of the maximum spinning reserve capacity is set at 15–30% of the maximum power [20]. The valve point constraint has been modelled on either the piece-wise quadratic cost function [8] or a circulating commutated sinusoidal function [4] to represent the valve point loading in the cost function. Therefore the ED problem considering generator constraints can be mathematically described as follows

$$\min_{P_i} \sum_{i \in \Psi} F_i(P_i) = \min_{P_i} \sum_{i \in \Psi} (a_i + b_i P_i + c_i P_i^2 \qquad (1)$$
$$+ |e_i \sin(f_i(P_{i\min} - P_i))|)$$

where i denotes index of units, F_i, fuel cost function of unit i; P_i, power generation of unit i; $P_{i\min}$, minimum generation limit of unit i; Ψ, set of all units; a_i, b_i, c_i, e_i, f_i, fuel cost coefficients of unit i.

Subject to the following constraints

(i) Power balance constraint

$$\sum_{i \in \Psi} P_i = P_D + P_L \qquad (2)$$

$$P_L = \sum_i \sum_j P_i B_{ij} P_j + \sum_i B_{0i} P_i + B_{00} \qquad (3)$$

where P_D denotes total load demand; P_L, power losses; B_{ij}, power loss coefficient.

(ii) System spinning reserve constraints

$$\sum_{i \in \Psi} S_i \geq S_R \qquad (4)$$

$$S_i = \min\{(P_{i\max} - P_i), S_{i\max}\} \quad \forall i \in (\Psi - \psi) \qquad (5)$$

$$S_i = 0 \quad \forall i \in \psi \qquad (6)$$

where S_i, spinning reserve of unit i; S_R, system spinning reserve requirement; $P_{i\max}$, maximum generation limit of unit i; $S_{i\max}$, maximum spinning reserve of unit i; ψ, set of all units with prohibited zones. In a unit with prohibited operating zones, these zones strictly limit the unit's ability to regulate system load because load regulation may result in its falling into certain prohibited operating zones. Therefore the system spinning reserve requirement must be supplied by way of regulating the units without prohibited zones.

(iii) Generation limits of units

Ramp rate limits constraints

$$\max(P_{i\min}, P_{0i} - DR_i) \leq P_i$$
$$\leq \min(P_{i\max}, P_{0i} + UR_i) \qquad (7)$$

Units with prohibited operating zones

$$\begin{cases} P_{i\min} \leq P_i \leq P_{i,1}^l \\ \text{or } P_{i,j-1}^u \leq P_i \leq P_{i,j}^l, \quad j = 2, \ldots, n_i \\ \text{or } P_{i,n_i}^u \leq P_i \leq P_{i\max}, \qquad \forall i \in \psi \end{cases} \qquad (8)$$

Units without prohibited operating zones

$$P_{i\min} \leq P_i \leq P_{i\max} \qquad (9)$$

where P_{0i}, previous output power of unit i; UR_i, up ramp limit of unit i; DR_i, down ramp limit of unit i; $P_{i,j}^l$, lower bound of the jth prohibited zone of unit i; $P_{i,j}^u$, upper bound of the jth prohibited zone of unit i; n_i, number of prohibited zones in unit i.

In this study, the treatment of constraints is performed with the penalty function methods. The penalty function methods are among the most popular techniques used to handle constraints, are easy to implement and are considered efficient. The penalty method is usually a close degree to the nearest solution in a reasonable region, and it can allow an objective function effort to arrive at the optimum solution. The penalty method is implemented in this paper as follows.

First, if the ED takes account of the prohibited zone constraints, the delimitation point divides the prohibited zone into two sub-zones, that is the left and right prohibited sub-zones. The delimitation point is set in the middle point of each prohibited zone in this work. When a unit operates in one of its prohibited zones, the strategy is to force the unit to move either towards the lower bound of that zone from the left sub-zone or towards the upper bound of that zone from the right sub-zone. The unit power must conform to the constraint of (8) when the unit does not include prohibited zones. In addition to prohibited zone constraints, the computation results also must conform to ramp rate limits in (7), the system spinning reserve requirement in (4) and the power balance condition in (2). The fuel cost function with these constrains is rewritten from (1) as

794

IET Gener. Transm. Distrib., Vol. 1, No. 5, September 2007

follows

$$\min_{P_i} M = \sum_{i \in \Psi} F_i(P_i) + q_1 |\sum_{i \in \Psi} P_i - P_D - P_L|$$
$$+ q_2 [\max(0, S_R - \sum_{i \in \Psi} S_i)]$$
$$+ q_3 [\max(0, \max(P_{i\min}, P_{0i} - DR_i) - P_i)$$
$$+ \max(0, P_i - \min(P_{i\max}, P_{0i} + UR_i))]$$
(10)

where q_1, q_2 and q_3 are penalty factors when these terms are zero in (10), and no constraints are violated; otherwise, these terms are positive values. To solve the above mentioned system, the self-tuning HDE is described as follows.

3 Self-tuning HDE algorithm

The main idea of the self-tuning HDE is to use the variable scaling factor based on the 1/5 success rule of the ESs [18, 19] to overcome the drawback of the fixed and random scaling factor used in the HDE. Different from the HDE, every individual has the corresponding scaling factor in the self-tuning HDE. The rule of updating the scaling factor based on the 1/5 success rule of ESs is used to adjust the scaling factor. The 1/5 success rule emerged as a conclusion of the process of optimising the convergence rate of two functions (the so-called corridor mode and sphere model [18, 19, 21]). The rule of updating the scaling factor is as follows

$$SF_i^{t+1} = \begin{cases} c_d * SF_i^t, & \text{if } p_i^t < 1/5 \\ c_j * SF_i^t, & \text{if } p_i^t > 1/5 \quad i = 1, \dots, N_p \\ SF_i^t, & \text{if } p_i^t = 1/5 \end{cases}$$
(11)

where p_i^t and N_p are the frequency of successful mutations measured of the ith individual and population size, respectively. The successful mutation of the ith individual defines that the fitness value of the ith individual is better than that of its parent. The initial value of the scaling factor, SF_i, is set to 1.2 [22, 23]. The factors of $c_d = 0.82$ and $c_j = 1/0.82$ [18, 19] are used for adjustment, which should take place for every y iterations. The iteration index y suggested by [19] is equal to $10 \times b$, where b is a constant. When the migration operator is performed, the value of the scaling factor is defined as follows.

$$SF_i = 1 - \frac{iter}{itermax} \quad i = 1, \dots, N_p$$
(12)

Where iter and itermax are the numbers of the current iteration and the maximum iteration, respectively. The scaling factor can be reset as (12) when the scaling factor is too small to find a better solution in the solution process.

Formally, the self-tuning HDE algorithm is briefly described in the following.

Step 1: Initialisation

The system data is input and the initial population generated. The initial population is chosen randomly in order to cover the entire parameter space uniformly. A uniform probability distribution for all random variables is assumed in the following

$$X_i^0 = X_i^{\min} + \sigma_i \times (X_i^{\max} - X_i^{\min}) \quad i = 1, \dots, N_p$$
(13)

where $\sigma_i \in (0, 1]$ is a random number. The initial process can produce N_P individuals of X_i^0 randomly.

Step 2: Mutation operation

Five strategies of mutation operator have been introduced by [22]. The essential ingredient in the mutation operation is the difference vector. Each individual pair in a population at the gth generation defines a difference vector $\boldsymbol{D}_{jk}$ as

$$\boldsymbol{D}_{jk} = X_j^g - X_k^g$$
(14)

The mutation process at the gth generation begins by randomly selecting either two or four population individuals X_j^g, X_k^g, X_l^g, and X_m^g for any j, k, l and m. These four individuals are then combined to form a difference vector $\boldsymbol{D}_{jklm}$ as

$$\boldsymbol{D}_{jklm} = \boldsymbol{D}_{jk} + \boldsymbol{D}_{lm} = X_j^g + X_l^g - (X_k^g + X_m^g)$$
(15)

A mutant vector is then generated based on the present individual in the mutation process by

$$\bar{\boldsymbol{X}}_i^{g+1} = X_p^g + F_i \times \boldsymbol{D}_{jklm} \quad i = 1, \dots, N_p$$
(16)

where the scaling factor, SF_i, is a constant. In addition, j, k, l and m are randomly selected.

The perturbed individual in (16) is essentially a noisy replica of X_p^g. Herein, the parent individual X_p^g depends on the circumstance in which the type of the mutation operation is employed.

Step 3: Crossover operation

In order to extend the diversity of further individuals at the next generation, the perturbed individual of $\bar{X}_i^{g+1}$ and the present individual of X_i^g are chosen by a binomial distribution to progress the crossover operation to generate the offspring. Each gene of i-th individual is reproduced from the mutant vectors $\bar{\boldsymbol{X}}_i^{g+1} = [\bar{X}_{1i}^{g+1}, \bar{X}_{2i}^{g+1}, \dots, \bar{X}_{ni}^{g+1}]$ and the present individual $X_i^g = [X_{1i}^g, X_{2i}^g, \dots, X_{ni}^g]$.

$$\bar{\boldsymbol{X}}_{qi}^{g+1} = \begin{cases} X_{qi}^g, & \text{if a random number} > C_r \\ \bar{X}_{qi}^{g+1}, & \text{otherwise} \end{cases}$$
(17)

where $i = 1, \dots, N_P; q = 1, \dots, n$; and the crossover factor $C_r \in [0, 1]$ is assigned by the user.

Step 4: Estimation and selection

The evaluation function of a child is one-to-one competed to that of its parent. This competition means that the parent is replaced by its child if the fitness of the child is better than that of its parent. On the other hand, the parent is retained in the next generation if the fitness of the child is worse than that of its parent, that is

$$X_i^{g+1} = \arg\min(f(X_i^g), (\bar{\boldsymbol{X}}_i^{g+1}))$$
(18)

$$X_b^{g+1} = \arg\min(f(X_i^g))$$
(19)

where 'arg min' means the argument of the minimum.

Step 5: Migrating operation if necessary

In order to effectively enhance the investigation of the search space and reduce the choice pressure of a small population, a migration phase is introduced to regenerate a new diverse population of individuals. The new population is yielded based on the best individual X_b^{g+1}. The

IET Gener. Transm. Distrib., Vol. 1, No. 5, September 2007

795

q-th gene of the ith individual is as follows

$$X_{qi}^{g+1} = \begin{cases} X_{bq}^{g+1} + \sigma_i \\ (X_{q\min} - X_{bq}^{g+1}), & \text{if } \delta < \dfrac{X_{bq}^{g+1} - X_{q\min}}{X_{q\max} - X_{q\min}} \\ X_{bq}^{g+1} + \sigma_i \\ (X_{q\max} - X_{bq}^{g+1}), & \text{otherwise} \end{cases} \quad (20)$$

where σ_i and δ are randomly generated numbers uniformly distributed in the range of $[0,1]$; $i = 1, \ldots, N_P$; and $q = 1, \ldots, n$.

The migrating operation is executed only if a measure fails to match the desired tolerance of population diversity. The measure is defined as follows

$$\varepsilon = \sum_{\substack{i=1 \\ i \neq b}}^{N_p} \sum_{g=1}^{n} \frac{\eta_{Xqi}}{n(N_p - 1)} < \varepsilon_1 \quad (21)$$

where

$$\eta_{Xqi} = \begin{cases} 0, & \text{if } \varepsilon_2 \geq \left| \dfrac{X_{qi}^{g+1} - X_{bi}^{g+1}}{X_{bi}^{g+1}} \right| \\ 1, & \text{otherwise} \end{cases} \quad (22)$$

Parameter $\varepsilon_1, \varepsilon_2 \in [0, 1]$ expresses the desired tolerance for the population diversity and the gene diversity with respect to the best individual. The η_X is the scale index. From (21) and (22), it can be seen that the value ε is in the range of $[0,1]$. If ε is smaller than ε_1, then the migrating operation is executed to generate a new population to escape the local point; otherwise, the migrating operation is turned off.

The convergence rate of the proposed self-tuning HDE can be improved by the use of the 1/5 success rule of ESs. However, this faster descent typically results in a local minimum or premature convergence. That is, the candidate individuals gradually cluster around the best individual such that the population diversity is quickly decreased. These closely clustered individuals cannot reproduce to the next better individuals via the mutation and crossover operations as obtained from (15) and (17). Thus, the migration operation in the self-tuning HDE must be performed to regenerate a new diversified population if a measure of population diversity fails to satisfy the desired tolerance, (21) and (22).

Step 6: Accelerated operation if necessary

When the best individual at the present generation is not improved any longer by the mutation and crossover operations, a decent method is then employed to push the present best individual towards attaining a better point. Thus, the accelerated phase is expressed as follows

$$X_b^{g+1} = \begin{cases} X_b^{g+1}, & \text{if } J(X_b^{g+1}) < J(X_b^g) \\ X_b^{g+1} - \alpha \nabla J, & \text{otherwise} \end{cases} \quad (23)$$

where X_b^g denotes the best individual, as obtained from (19). The gradient of the objective function, ∇J, can be approximately calculated by finite difference. The step size $\alpha \in (0, 1]$ in (23) is determined by the descent property. Initially, α is set to one to obtain the new individual.

Step 7: Updating the scaling factor if necessary

The scaling factor should be updated as (11) in every y iterations. When the migrating operation that is performed or

the scaling factor is too small to find the better solution, the scaling factor is reset as (12).

Step 8: Repeat step 2 to step 7 until the maximum iteration quantity or the desired fitness is accomplished

There are seven parameters employed in the proposed self-tuning HDE method, including population size N_P, crossover factor C_r, mutation operator strategy, iteration index q, maximum iterations itermax, tolerance of population diversity ε_1 and tolerance of gene diversity ε_2. The first four parameters are easy to select, as these parameters do not affect the computational results significantly. According to the author's experience, appropriate values for these parameters are $N_P = 5$, $C_r = 0.5$ or 0.8, mutation operator strategy $= 2$ and iteration index $q = 10$ or 20. However, slight changes to these four parameters do not obviously affect the computational results. If the number of decision parameters is smaller, then the maximum iteration number should be given a smaller value. Otherwise, the maximum iteration number is given a greater value. The last two parameters, ε_1 and ε_2, are relatively hard to handle. According to the author's experience, if the objective function is sensitive to these two parameters, these parameters are given smaller values. On the other hand, if the objective function is not sensitive to these two parameters, they are given greater values. Typically, both ε_1 and ε_2 are set to 0.1 and 0.01, respectively.

This computational process of the self-tuning HDE algorithm for solving the optimal ED problem is stated using a flowchart, shown in Fig. 1.
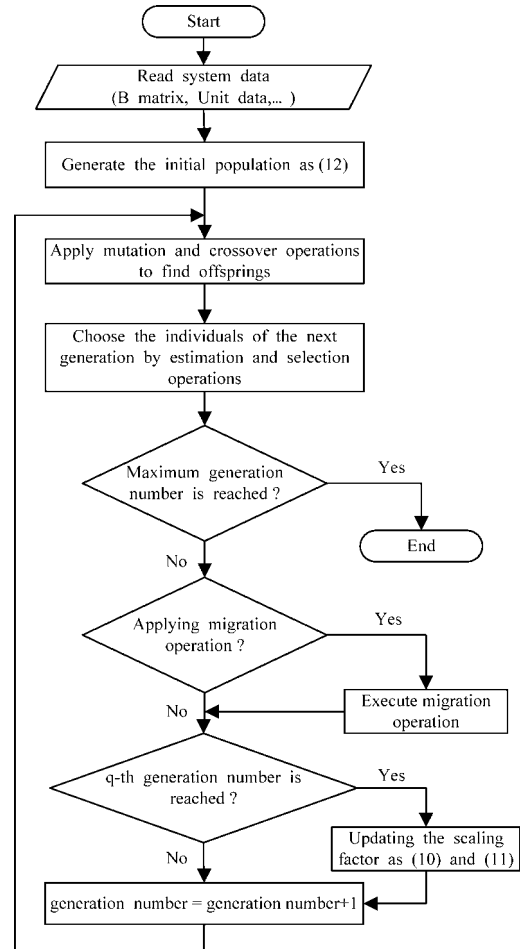

Fig. 1 *Main calculation procedures of the proposed algorithm*

Table 1: Comparison of case study results for 13 and 40 unit systems presented in the literature

Algorithm	13-unit			40-unit		
	Best cost ($/h)	average cost ($/h)	Average time (s)	Best cost ($/h)	average cost ($/h)	Average time (s)
PSO [24]	18 030.72	18 205.78	77.37	122 930.45	124 154.49	933.39
MPSO [25]	–	–	–	122 252.27	–	–
PSO-SQP [24]	17 969.93	18 029.99	33.97	122 094.67	122 245.25	733.97
IFEP [12]	17 994.07	18 127.06	157.43	122 624.35	123 382.00	1167.35
DEC-SQP [26],[27]	17 963.94	17 973.13	0.5	121 741.98	122 295.13	14.26
HDE	17 975.73	18 134.80	1.65	121 813.26	122 705.66	6.92
Self-tuning HDE	17 963.89	18 046.38	1.41	121 698.51	122 304.30	6.07

4 Numerical examples and results

This section presents the computation results on three test systems which were performed to evaluate the performance of the proposed algorithm. An ED problem including 3-, 13-, and 40-unit power systems from the literature have been investigated. In order to simulate the valve point effects of the generating units, a recurring sinusoid component is added with the objective function of fuel cost. However, many practical constraints of generators, such as ramp rate limits, prohibited operating zones, spinning reserve and power loss are also considered in the optimisation process. To verify the performance of the proposed algorithm, these three systems are repeatedly tested a hundred times by the self-tuning HDE, HDE, DE and GA methods. The software was written in Matlab and executed on a Pentium 1.5 GHz with 768 MB of RAM. The results of fuel costs and average CPU times are used to compare the performance of the proposed self-tuning HDE method with power balance constraints only for those obtained in recent studies presented in the literature and with, as mentioned above, all constraints for that of the GA, DE and HDE methods. The C code for the real-coded GA [21] algorithm was built by Denis Cormier (North Carolina State University) and modified by Sita S. Raghavan (University of North Carolina at Charlotte). The GA packages were rewritten in Matlab software. The Matlab code of the DE algorithm was obtained from Kenneth Price and Rainer Storn's website in [22], and the HDE algorithm, also written in Matlab code, was obtained from [15].

4.1 ED problem with power balance constraint only

In the study, the power loss, ramp rate limit and prohibited zones are disregarded to conform to the system conditions in the literature [12]. Therefore the penalty factors in (10) are set at 20 for q_1, and zero for q_2 and q_3. The 3-, 13- and 40-unit power systems, which have respective load demands of 850, 1800 and 10 500 MW, are computed by both self-tuning HDE and HDE methods. The common setting factors used in the self-tuning HDE to solve the three systems are as follows. The crossover factor (C_r), is set at 0.8, and population size (N_P) is 5. The scaling factor is updated as (11) and (12) in every 20 iterations. The second mutation operator is used in the self-tuning HDE. Two tolerances, ε_1 and ε_2, used in the migrating operation are set to 0.1 and 0.01, respectively. These initial setting factors for the HDE method are the same as that of the self-tuning HDE, except that the HDE uses a scaling factor fixed at 0.5, according to the author's suggestion [16]. For the 3-unit system, the maximum iteration is set at 50, and the best fuel costs of computation results with self-tuning HDE and HDE methods are all the same,

Table 2: Best result obtained by self-tuning HDE for the 13-unit system with power balance constrain only

Unit power, MW	Generation
P_1	628.3172
P_2	149.5986
P_3	222.7987
P_4	109.8673
P_5	109.8418
P_6	60.0000
P_7	109.8641
P_8	109.8547
P_9	109.8576
P_{10}	40.0000
P_{11}	40.0000
P_{12}	55.0000
P_{13}	55.0000

Table 3: Best result obtained by self-tuning HDE for the 40-unit system with power balance constrain only

Unit power, MW	Generation	Unit power, MW	Generation
P_1	113.3515	P_{21}	522.7598
P_2	110.6763	P_{22}	523.4137
P_3	120.0000	P_{23}	523.2754
P_4	180.5395	P_{24}	522.6744
P_5	95.47723	P_{25}	523.3112
P_6	140.0000	P_{26}	524.1180
P_7	300.0000	P_{27}	10.00000
P_8	284.1949	P_{28}	10.00000
P_9	297.0010	P_{29}	10.00000
P_{10}	130.0030	P_{30}	97.00000
P_{11}	168.5498	P_{31}	189.9364
P_{12}	94.00000	P_{32}	189.9161
P_{13}	125.0000	P_{33}	189.9226
P_{14}	394.1231	P_{34}	200.0000
P_{15}	304.6147	P_{35}	198.0817
P_{16}	394.3282	P_{36}	165.7141
P_{17}	495.4328	P_{37}	109.3638
P_{18}	489.7258	P_{38}	109.8876
P_{19}	511.2027	P_{39}	110.0000
P_{20}	511.1898	P_{40}	511.2148

IET Gener. Transm. Distrib., Vol. 1, No. 5, September 2007

797

Table 4: Comparison of the four methods for the Example 1 system

Unit power	GA	DE	HDE	Self-tuning HDE
P_1 (MW)	402.4026	399.1984	399.1993	399.1993
P_2 (MW)	324.5534	324.4048	324.3994	324.3995
P_3 (MW)	110.0535	113.4125	113.4171	113.4170
Power loss (MW)	2.0097	2.0158	2.0159	2.0159
Total generation (MW)	837.0097	837.0158	837.0159	837.0159
Best cost ($/h)	8197.87	8189.29	8189.26	8189.26
Average cost ($/h)	8438.18	8232.15	8208.10	8201.78
Worst cost ($/h)	8802.65	8405.68	8353.38	8262.20
Average CPU time (s)	0.03959	0.04356	0.11706	0.11826

3234.07 $/h, and the average CPU times are 0.053 and 0.056, respectively. The results show that self-tuning HDE and HDE methods are both able to quickly search out the global solution. For the 13- and 40-unit power systems, the maximum iterations are set at 2500 and 8500, respectively, by the self-tuning HDE and HDE algorithms. The computational results of the self-tuning HDE and HDE methods are compared with particle swarm optimisation (PSO) [24], modified particle swarm optimisation (MPSO) [25], hybrid particle swarm optimisation and sequential quadratic programming (PSO-SQP) [24], improved fast evolutionary programming (IFEP) [12] and a combination of chaotic differential evolution and sequential quadratic programming (DEC-SQP) [26, 27] in Table 1. The best results obtained for units solution by the self-tuning HDE method with a minimum fuel cost of 17 963.89 $/h and 121 698.50$/h for 13- and 40-unit are displayed in Tables 2 and 3, respectively. Nevertheless, the results of 17 963.89$/h and 121 698.50 $/h attained by the self-tuning HDE method reported here are comparatively

lower than those obtained in recent studies presented in the literature.

In (11), the comparison is made with different factor values of cd and cj for observation of the convergence situation of the self-tuning HDE algorithm. These values of [0.5, 0.6, 0.7, 0.8, 0.82, 0.9] are individually substituted for c_d, and the c_j is designated as the inverse of the c_d, as suggested by Schwefel [19], for solving ED problems in the case of 40-unit power system. The computational results of best fuel costs by different c_d and c_j values are [121 986.95, 121 895.46, 121 874.20, 121 793.63, 121 698.51, 121 829.11]$/h, respectively. However, these computational results indicate that $c_d = 0.82$ can obtain the best result, 121 698.51$/h.

4.2 ED problem with all constraints

In this work, the penalty factors q_1 and q_3 in (10) are set at 20 and 10 000, respectively, and q_2 is set at 10 000 when the test system has a system spinning reserve constraint; otherwise, q_2 is set at zero. The fuel cost function is solved in the 3-, 13- and 40-unit power systems by the self-tuning HDE, HDE, DE and GA methods.

Example 1: The data of the test system is obtained from [4], which includes three generating units cost function and loss coefficients **B** matrix, with modification in the fuel cost functions to incorporate the prohibited zones and ramp rate limits. The total load demand of the system is 835 MW. The input data including prohibited zones and ramp rate limits of the three units are listed in the Appendix, except for the fuel cost coefficients of the units in [4]. To verify the performance of the proposed self-tuning HDE method, the GA and DE methods are also applied to solve this system. These initial setting factors for the HDE and self-tuning HDE methods are the same as those mentioned in the above section, and the maximum iteration is set at 50. The GA and DE methods seem to be sensitive to
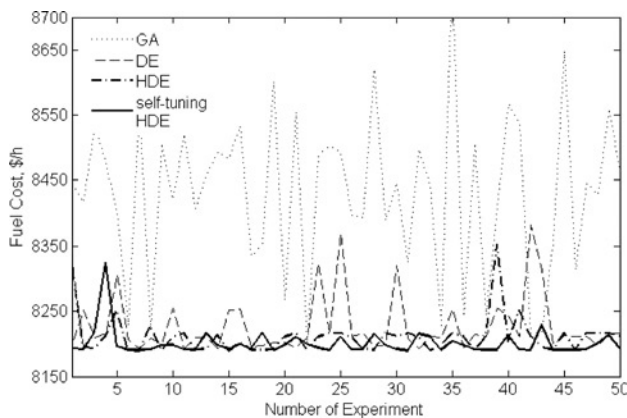
Fig. 2 *Distribution of fuel cost for four algorithms in Example 1*

Table 5: Computational results of the 3-unit system for different maximum iteration and fixed population size

Algorithm	itermax	Best cost ($/h)	Average cost ($/h)	Worst cost ($/h)	Average CPU time (s)	Standard deviation ($/h)
GA	100	8197.48	8405.34	8727.39	0.07775	113.7490
	150	8194.35	8378.73	8622.57	0.11717	99.8292
DE	100	8189.26	8218.84	8432.13	0.08591	37.6564
	150	8189.26	8214.38	8372.17	0.12688	30.2567
HDE	50	8189.26	8206.93	8353.38	0.11706	22.6934
Self-tuning HDE	50	8189.26	8201.77	8262.20	0.11526	13.8948

Table 6: Comparison of the four methods for the Example 2 system

Unit power	GA	DE	HDE	Self-tuning HDE
P_1 (MW)	628.4311	628.0117	628.3290	628.3185
P_2 (MW)	305.0000	300.2498	299.3286	299.2092
P_3 (MW)	302.6497	348.2995	304.5139	299.2018
P_4 (MW)	158.9094	159.0591	159.7930	159.7416
P_5 (MW)	160.4743	159.7318	159.8114	159.7433
P_6 (MW)	159.7312	159.7324	159.8572	159.7335
P_7 (MW)	160.1004	159.7330	159.9505	159.7384
P_8 (MW)	159.6400	147.6877	109.8658	159.7331
P_9 (MW)	109.6715	160.7340	159.7405	159.7338
P_{10} (MW)	114.5156	77.29379	114.8171	114.8027
P_{11} (MW)	116.2229	115.6040	115.7702	116.7061
P_{12} (MW)	92.08722	55.01118	94.97113	55.25513
P_{13} (MW)	92.43267	91.19282	92.40933	92.41379
Power loss (MW)	39.8664	42.3412	39.15823	44.3314
Total generation (MW)	2559.87	2562.34	2559.16	2564.33
Total reserve (MW)	198.3218	198.3218	190.8761	186.5761
Best cost ($/h)	24 632.42	24 819.32	24 591.76	24 560.08
Average cost ($/h)	24 874.93	25 217.64	24 739.53	24 706.63
Worst cost ($/h)	25 188.59	25 656.40	25 074.90	24 872.44
Average CPU time (s)	2.25174	2.58151	3.57327	2.97826

tuning of some parameters; thus, these parameters are set according to the experiences of many experiments conducted on these three ED problems. The GA parameters could be selected as follows. The crossover ratio is 0.5, the population size is 5, the maximum iteration is 50 and the mutation ratio is 0.01. The DE parameters were selected as follows. The crossover factor is 0.6, the population size is

5, the maximum iteration is 50 and the mutation operation is 2. These two methods obtain the best solution under these conditions. The computational results of the best, average, and the worst fuel costs, unit generations, power loss and average CPU time among the 100 runs of solutions which satisfy the system constraints are listed in Table 4. In Table 4, the best solution of the Example 1 system is displayed in boldface. From the computation results shown in Table 1, we find that all four algorithms have the potential to find the global solution. Among these approaches, the self-tuning HDE algorithm results in the longest average CPU time, but the average fuel cost is the minimum. Fig. 2 displays the computed fuel cost of the first 50 of the 100 runs. As shown in Fig. 2, the self-tuning HDE algorithm has the most stable and minimum fuel cost. Thus, the self-tuning HDE algorithm is more reliable to find out the global minimum fuel cost in this example. For the sake of further comparison with the approach of GA and DE algorithms, the maximum iteration for GA and DE is set to be 100 and 150 to run the Example 1 system 100 times again. The results of the best, average and the worst fuel costs and average CPU time from this experiment are listed in Table 5. It is observed that fuel costs obtained by the GA, DE and HDE approaches are slightly higher than or equal to that of the self-tuning HDE approach. Under the condition of itermax = 150, the average fuel cost for

Fig. 3 *Convergence characteristics for four algorithms in Example 2*

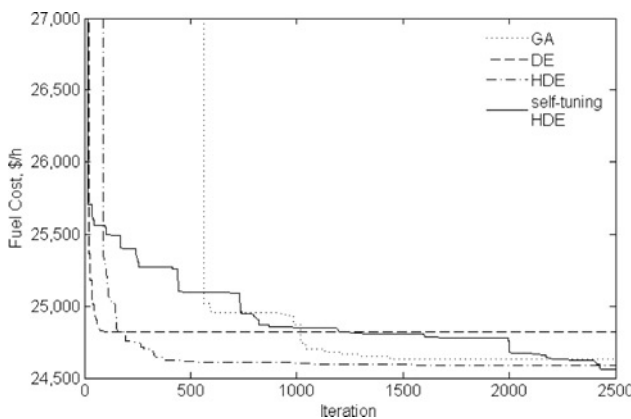**Table 7: Computational results of the 13-unit system for different population size and fixed maximum iteration**

Algorithm	N_p	Best cost ($/h)	Average cost ($/h)	Worst cost ($/h)	Average CPU time (s)	Standard deviation ($/h)
GA	10	24 610.75	24 786.99	25 108.45	4.34874	110.8468
	15	24 597.23	24 750.97	25 070.53	6.54261	95.6446
DE	10	24 592.75	24 826.03	25 178.27	5.77439	119.7034
	15	24 560.10	24 710.30	24 965.41	7.88623	77.3527
HDE	5	24 591.76	24 739.53	25 074.90	3.57327	86.4049
Self-tuning HDE	5	24560.08	24706.63	24 872.44	2.97826	58.0620

IET Gener. Transm. Distrib., Vol. 1, No. 5, September 2007

799

GA and DE is closer to the computation result of the self-tuning HDE with itermax = 50 and the average CPU time for GA and DE is slightly greater than that of the self-tuning HDE; the maximum iteration of GA and DE is three times that of self-tuning HDE.

Example 2: The data of the test system is obtained from [28], which has 13 generating units, and the loss coefficients

B matrix is obtained from [13, 14] with modification in the fuel cost functions to incorporate the prohibited zones and ramp rate limits. The spinning reserve constraint is also considered in this application system. Input data of the 13-unit system are listed in the Appendix, except for the fuel cost coefficients of units in [28]. In this system, five units have prohibited operating zones, and the remaining eight units contribute the required spinning reserve to the system.

Table 8: Comparison of the four methods for the Example 3 system

Unit power	GA ($N_P = 10$)	DE ($N_P = 10$)	HDE ($N_P = 5$)	Self-tuning HDE ($N_P = 5$)
P_1 (MW)	111.7213	113.5764	113.9783	110.84697
P_2 (MW)	110.6113	113.7926	114.0000	112.1565
P_3 (MW)	99.16071	97.40588	98.53020	120.0000
P_4 (MW)	179.7968	180.0051	182.8223	179.4657
P_5 (MW)	90.21361	89.61925	91.99572	93.17119
P_6 (MW)	139.7402	139.9999	139.1252	139.6558
P_7 (MW)	261.0782	299.9999	299.2042	299.9904
P_8 (MW)	285.7480	284.7046	285.6743	292.9315
P_9 (MW)	285.4551	284.6111	296.1914	284.5741
P_{10} (MW)	130.0000	130.0000	200.0000	130.0000
P_{11} (MW)	169.3225	94.21490	94.66729	94.00994
P_{12} (MW)	169.8604	168.7940	169.3861	94.00000
P_{13} (MW)	300.0000	304.4362	125.0000	125.0000
P_{14} (MW)	392.8375	394.2793	394.5668	393.6505
P_{15} (MW)	305.4536	394.2793	484.7173	478.7679
P_{16} (MW)	383.7715	304.5195	304.5127	393.1025
P_{17} (MW)	489.5959	489.2794	491.3617	489.1105
P_{18} (MW)	489.1406	489.2793	489.4712	489.0735
P_{19} (MW)	510.4080	511.2793	511.4723	511.2153
P_{20} (MW)	511.4157	511.2982	511.5457	511.0469
P_{21} (MW)	524.1014	523.2794	524.4637	523.1869
P_{22} (MW)	523.6807	523.9855	526.8726	530.5903
P_{23} (MW)	522.6112	523.7207	524.7330	523.1463
P_{24} (MW)	526.8539	523.2843	523.2879	522.9946
P_{25} (MW)	522.9654	523.3065	524.1151	523.6739
P_{26} (MW)	522.9853	523.2793	523.2267	525.2442
P_{27} (MW)	10.62509	10.00001	10.17141	10.00000
P_{28} (MW)	10.58456	10.16713	10.00000	10.02717
P_{29} (MW)	11.36923	10.00006	10.63973	10.00000
P_{30} (MW)	88.39085	91.16536	91.94259	89.01253
P_{31} (MW)	189.9861	189.9883	190.0000	188.8546
P_{32} (MW)	189.0402	189.9996	187.2378	189.9981
P_{33} (MW)	185.8470	189.9956	190.0000	189.6139
P_{34} (MW)	166.8364	199.9996	165.6872	200.0000
P_{35} (MW)	199.7900	199.9997	200.0000	199.9993
P_{36} (MW)	199.9763	165.8468	174.8976	199.9172
P_{37} (MW)	92.01623	109.9999	110.0000	108.4430
P_{38} (MW)	107.1946	99.30869	109.9254	110.0000
P_{39} (MW)	99.50363	105.9249	109.8258	110.0000
P_{40} (MW)	511.3090	511.2793	512.1575	511.2681
Power loss (MW)	120.9994	119.9064	117.4074	117.74044
Total generation (MW)	10 621.00	10 619.90	10 617.40	10 617.74
Best cost ($/h)	123 652.24	125 074.40	123 598.76	123 496.02
Average cost ($/h)	124 076.43	127 399.36	124 210.34	124 007.10
Worst cost ($/h)	124 668.69	129 639.79	124 855.80	124 570.74
Average CPU time (s)	27.11811	29.31566	17.94273	16.86025

800

IET Gener. Transm. Distrib., Vol. 1, No. 5, September 2007

The total load demand of the system is 2520 MW, and the system provides a required spinning reserve of 180 MW at least. These initial setting factors for the GA, DE, HDE and self-tuning HDE methods are the same as those in Example 1, except that the maximum iteration is set at 2500. The computation results are listed in Table 6. Table 6 shows that both the best fuel cost and average fuel cost obtained by GA, DE and HDE algorithms are relatively higher than those obtained by the proposed self-tuning HDE. In Table 6, the best solution of the Example 2 system is displayed in boldface. Fig. 3 represents the convergence properties of the GA, DE, HDE and self-tuning HDE in the process of searching for the minimum fuel cost. The figure shows that the convergence property of the self-tuning HDE algorithm is superior to that of the other three algorithms. For the sake of further comparison with the approach of GA and DE algorithms, the N_P is set to be 10 and 15 to run the Example 2 system 100 times again. The results of computations from these experiments are listed in Table 7. It is observed that fuel costs obtained by the GA, DE and HDE algorithms are slightly higher than or equal to that of the self-tuning HDE algorithm. In addition, the average CPU times of the self-tuning HDE is lower than those of the GA, DE and HDE methods.

Example 3: This test system consisting of a practical 40-unit Taiwan power (Taipower) system [2], with modifications to incorporate the valve point effect [12]. In this system, the total load demand of the system is 10 500 MW. The input data of the system are given in the Appendix, except for the fuel cost coefficients of units in [12]. Generator constraints of prohibited zone and ramp rate limit from number 10 to 14 units of prohibited zones are listed in the Appendix. These prohibited zones result in four disjoint feasible sub-regions for each of the units. Hence, those zones result in a non-convex decision space which consists of 1024 convex sub-spaces for the example system. In order to obtain the minimum fuel cost, the maximum iteration parameter setting for the four methods is set to be 8500; the population size is modified to be 10 for the GA and DE methods. The remaining parameters are the same as those of Example 1. The computational results are shown in Table 8. In Table 8, the best solution of the Example 3 system is displayed in boldface, and the lowest of the best fuel cost among the four algorithms is observed from the self-tuning HDE; it is 123 496.02 $/h. The worst fuel cost of these 100 runs obtained by the self-tuning HDE method is 124 570.74 $/h. The best and worst fuel costs compared with the other three methods are too close.
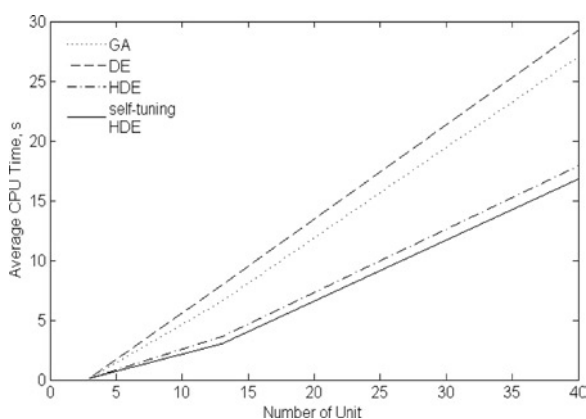

Fig. 4 *Comparison of computation complexity*

From the computation results in the three test systems, it appears that the self-tuning HDE could search for the minimum fuel cost under various generator constraints via the fixed population size. The comparison of computation complexity of four algorithms is shown in Fig. 4, which reveals the relation of the numbers of units against the average CPU times. We find that the self-tuning HDE algorithm has better computation complexity.

5 Conclusions

In this paper, we suggest employing the self-tuning HDE algorithm to solve the non-smooth/non-convex ED problem with the valve point effect and with/without constraints of the prohibited zones, ramp rate limit and spinning reserve. Three stochastic optimisation algorithms, including GA, DE and HDE, are also employed to solve the ED problem for comparison with the self-tuning HDE. Three test systems have been employed to illustrate the application of the proposed method. Computational results show that the proposed self-tuning HDE algorithm is superior to the other three (GA, DE and HDE) algorithms in terms of computed minimum fuel cost and computational complexity.

6 References

1 Lee, F.N., and Breipohl, A.M.: 'Reserve constrained economic dispatch with prohibited operating zones', *IEEE Trans. Power Syst.*, 1993, **8**, (1), pp. 246–254
2 Chen, P.H., and Chang, H.C.: 'Large-scale economic dispatch by genetic algorithm', *IEEE Trans. Power Syst.*, 1995, **10**, (4), pp. 1919–1926
3 Kim, J.O., Shina, D.J., Parka, J.N., and Singh, C.: 'Atavistic genetic algorithm for economic dispatch with valve point effect', *Electr. Power Syst. Res.*, 2002, **62**, (3), pp. 201–207
4 Walters, D.C., and Sheble, G.B.: 'Genetic algorithm solution of economic dispatch with valve point loading', *IEEE Trans. Power Syst.*, 1993, **8**, (3), pp. 1325–1332
5 Wood, A.J., and Wollenberg, B.F.: 'Power generation, operation and control' (John Wiley & Sons, New York, 1996)
6 Fan, J.Y., and McDonald, J.D.: 'A practical approach to real time economic dispatch considering unit's prohibited operating zones', *IEEE Trans. Power Syst.*, 1994, **9**, (4), pp. 1737–1743
7 Su, C.T., and Chiou, G.J.: 'A Hopfield network approach to economic dispatch with prohibited operating zones'. IEEE Proc. Int. Conf. Energy Management and Power Delivery, Singapore, November 1995, pp. 382–387
8 Lin, W.M., Cheng, F.S., and Tsay, M.T.: 'Nonconvex economic dispatch by integrated artificial intelligence', *IEEE Trans. Power Syst.*, 2001, **16**, (2), pp. 307–311
9 Lin, W.M., Cheng, F.S., and Tsay, M.T.: 'An improved tabu search for economic dispatch with multiple minima', *IEEE Trans. Power Syst.*, 2002, **17**, (1), pp. 108–112
10 Sewtohul, L.G., Ah King, R.T.F., and Rughooputh, C.S.: 'Genetic algorithms for economic dispatch with valve point effect'. IEEE Int. Conf. Networking, Sensing and Control, Taipei, Taiwan, March 2004, Vol. 2, pp. 1358–1363
11 Wong, K.P., and Fung, C.C.: 'Simulated annealing based economic dispatch algorithm', *IEE Proc., Gener. Transm. Distrib.*, 1993, **140**, (6), pp. 509–515
12 Sinha, N., Chakrabarti, R., and Chattopadhyay, P.K.: 'Evolutionary programming techniques for economic load dispatch', *IEEE Trans. Evol. Comput.*, 2003, **7**, (1), pp. 83–94
13 Gaing, Z.L.: 'Particle swarm optimisation to solving the economic dispatch considering the generator constraints', *IEEE Trans. Power Syst.*, 2003, **18**, (3), pp. 1187–1195
14 Gaing, Z.L.: 'Closure to discussion of particle swarm optimisation to solving the economic dispatch considering the generator constraints', *IEEE Trans. Power Syst.*, 2004, **19**, (4), pp. 2121–2123
15 Chiou, J.P., and Wang, F.S.: 'Hybrid method of evolutionary algorithms for static and dynamic optimisation problems with application to a fed-batch fermentation process', *J. Comput. Chem. Eng.*, 1999, **23**, pp. 1277–1291
16 Chiou, J.P., and Wang, F.S.: 'Estimation of monod model parameters by hybrid differential evolution', *J. Bioprocess Biosyst. Eng.*, 2001, pp. 109–113
17 Lin, Y.C., Hwang, K.S., and Wang, F.S.: 'Plant scheduling and planning using mixed-integer hybrid differential evolution with

multiplier updating'. Proc. Congress on Evolutionary Computation, San Diego, CA, July 2000, vol. 1, pp. 593–600

18 Back, T., Hoffmeister, F., and Schwefel, H.-P.: 'A survey of evolution strategies'. Proc. Int. Conf. Genetic Algorithms, San Diego, CA, July 1991, pp. 2–9

19 Back, T., and Schwefel, H.P.: 'An overview of evolutionary algorithms for parameter optimisation', *Evolution. Comput.*, 1993, **1**, (1), pp. 1–23

20 Turgeno, A.: 'Optimal scheduling of thermal generating units', *IEEE Trans. Autom. Control*, 1978, **23**, (6), pp. 1000–1005

21 Michalewicz, Z.: 'Genetic algorithms + data structures = evolution programs' (New York, 1999, 3rd edn.)

22 Storn, R., and Price, K.: 'Minimizing the real functions of the ICEC '96 contest by differential evolution'. IEEE Proc. Int. Conf. Evolutionary Computation, Nagoya, Japan, May 1996, pp. 842–844

23 Price, K.V.: 'Differential evolution vs. functions of the 2nd ICEC'. IEEE Conf. Evolutionary Computation, Indianapolis, USA, 1997, pp. 153–157

24 Victoire, T.A.A., and Jeyakumar, A.E.: 'Hybrid PSO-SQP for economic dispatch with valve-point effect', *Electr. Power Syst. Res.*, 2004, **71**, (1), pp. 51–59

25 Park, J.-B., Lee, K.-S., Shin, J.-R., and Lee, K.Y.: 'A particle swarm optimization for economic dispatch with nonsmooth cost function', *IEEE Trans. Power Syst.*, 2005, **20**, (1), pp. 34–42

26 Coelho, Ld.S., and Mariani, V.C.: 'Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect', *IEEE Trans. Power Syst.*, 2006, **21**, (2), pp. 989–996

27 Coelho, Ld.S., and Mariani, V.C.: 'Correction to "combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect', *IEEE Trans. Power Syst.*, 2006, **21**, (3), pp. 1465–1465

28 Wong, K.P., and Wong, Y.W.: 'Thermal generator scheduling using hybrid genetic/simulated-annealing approach', *IEE Proc., Gener. Transm. Distrib.*, 1995, **142**, (4), pp. 372–380

7 Appendix

Three application systems, including 3-, 13- and 40-unit parameters, are listed in Table 9.

Table 9: Units data of the three example systems

Unit	$P_{i\min}$ (MW)	$P_{i\max}$ (MW)	P_{0i} (MW)	UR_i (MW)	DR_i (MW)	Prohibited zones (MW)	$S_{i\max}$ (MW)
3-unit							
1	100	600	350	280	320	[120,200][320,350]	–
2	100	400	370	150	190	[120,170][250,320]	–
3	50	200	144	100	150	[80,110]	–
13-unit							
1	0	680	420	335	360	[180,200][260,335][390,420]	–
2	0	360	280	250	290	[30,45][185,225][305,335]	–
3	0	360	280	250	290	[30,40][180,220][305,335]	–
4	60	180	120	80	130	–	50
5	60	180	150	80	130	–	40
6	60	180	130	80	130	–	30
7	60	180	160	80	130	–	30
8	60	180	140	80	130	–	40
9	60	180	140	80	130	–	50
10	40	120	100	120	120	[45,55][65,75]	–
11	40	120	80	120	120	[45,55][65,75]	–
12	55	120	80	120	120	–	30
13	55	120	75	120	120	–	25
40-unit							
1	36	114	100	114	114	–	–
2	36	114	100	114	114	–	–
3	60	120	90	120	120	–	–
4	80	190	150	100	150	–	–
5	47	97	80	97	97	–	–
6	68	140	120	80	125	–	–
7	110	300	280	165	200	–	–
8	135	300	200	165	200	–	–
9	135	300	230	165	200	–	–
10	130	300	240	155	190	[130,150][200,230][270,299]	–
11	94	375	210	150	185	[100,140][230,280][300,350]	–
12	94	375	210	150	185	[100,140][230,280][300,350]	–
13	125	500	230	206	235	[150,200][250,300][400,450]	–
14	125	500	355	260	290	[200,250][300,350][450,490]	–
15	125	500	350	186	215	–	–
16	125	500	350	186	215	–	–
17	220	500	460	240	270	–	–
18	220	500	470	240	268	–	–

(*Table continued*)

802

IET Gener. Transm. Distrib., Vol. 1, No. 5, September 2007

Table 9:   Continued

Unit	$P_{i\min}$ (MW)	$P_{i\max}$ (MW)	P_{0i} (MW)	UR_i (MW)	DR_i (MW)	Prohibited zones (MW)	$S_{i\max}$ (MW)
19	242	550	500	290	315	–	–
20	242	550	500	290	315	–	–
21	254	550	510	335	360	–	–
22	254	550	520	335	360	–	–
23	254	550	520	335	362	–	–
24	254	550	450	350	378	–	–
25	254	550	400	350	380	–	–
26	254	550	520	350	380	–	–
27	10	150	20	95	145	–	–
28	10	150	20	95	145	–	–
29	10	150	25	98	145	–	–
30	47	97	90	97	97	–	–
31	60	190	170	90	145	–	–
32	60	190	150	90	145	–	–
33	60	190	190	90	145	–	–
34	90	200	190	105	150	–	–
35	90	200	150	105	150	–	–
36	90	200	180	105	150	–	–
37	25	110	60	110	110	–	–
38	25	110	40	110	110	–	–
39	25	110	50	110	110	–	–
40	242	550	512	290	315	–	–

IET Gener. Transm. Distrib., Vol. 1, No. 5, September 2007

803