Evolutionary programming techniques for economic load dispatch

Article *in* IEEE Transactions on Evolutionary Computation · March 2003

DOI: 10.1109/TEVC.2002.806788 · Source: IEEE Xplore

Nidul Sinha
National Institute Of Technology Silchar

Ritashree Chakrabarti
IHS Markit

Brief Papers

Evolutionary Programming Techniques for Economic Load Dispatch

Nidul Sinha, R. Chakrabarti, and P. K. Chattopadhyay

***Abstract*—Evolutionary programming has emerged as a useful optimization tool for handling nonlinear programming problems. Various modifications to the basic method have been proposed with a view to enhance speed and robustness and these have been applied successfully on some benchmark mathematical problems. But few applications have been reported on real-world problems such as economic load dispatch (ELD). The performance of evolutionary programs on ELD problems is examined and presented in this paper in two parts. In Part I, modifications to the basic technique are proposed, where adaptation is based on scaled cost. In Part II, evolutionary programs are developed with adaptation based on an empirical learning rate. Absolute, as well as relative, performance of the algorithms are investigated on ELD problems of different size and complexity having nonconvex cost curves where conventional gradient-based methods are inapplicable.**

***Index Terms*—Cauchy mutation, classical evolutionary programming, economic load dispatch, fast evolutionary programming, Gaussian mutation, self-adaptation.**

I. Introduction

ECONOMIC load dispatch (ELD) is an important optimization task in power system operation for allocating generation among the committed units such that the constraints imposed are satisfied and the energy requirements in terms of British thermal units per hour (Btu/h) or dollar per hour ($/h) are minimized. Improvements in scheduling the unit outputs can lead to significant cost savings. Traditional dispatch algorithms employ Lagrangian multipliers and require monotonically increasing incremental cost curves. Unfortunately, the input–output characteristics of modern units are inherently highly nonlinear because of valve-point loadings, rate limits, etc., and furthermore they may generate multiple local minimum points in the cost function. Classical dispatch algorithms require that these characteristics be approximated; however, such approximations are not desirable as they may lead to suboptimal operation and hence huge revenue loss over time.

In light of the nonlinear characteristics of the units, there is a demand for techniques that do not have restrictions on the shape of the fuel-cost curves. Classical calculus-based techniques fail to address these types of problems satisfactorily. Unlike some traditional algorithms, dynamic programming (DP) [1] imposes no restrictions on the nature of the cost curves and therefore it can solve ELD problems with inherently nonlinear and discontinuous cost curves. This method, however, suffers from the "curse of dimensionality" or local optimality.

In this respect, stochastic search algorithms such as genetic algorithms (GAs) [2]–[13], evolution strategies (ESs) [14], [15], evolutionary programming (EP) [16]–[28], and simulated annealing (SA) [11], [29] may prove to be very effective in solving nonlinear ELD problems without any restrictions on the shape of the cost curves. Although these heuristic methods do not always guarantee discovering the globally optimal solution in finite time, they often provide a fast and reasonable solution (suboptimal near-globally optimal). In practice, the annealing schedule of SA should be tuned carefully; otherwise the solution achieved will still be only locally optimal. Nevertheless, an appropriate annealing schedule often requires tremendous computational time. Recent research endeavors, therefore, have been directed toward application of efficient and near-optimal GA, ES, and EP. These evolutionary algorithms (EAs) are search algorithms based on the simulated evolutionary process of natural selection, variation, and genetics. EAs are more flexible and robust than conventional calculus-based methods. EP differs from traditional GAs in two aspects: EP uses the control parameters (real values), but not their codings as in traditional GAs, and EP relies primarily on mutation and selection, but not crossover, as in traditional GAs. Hence, considerable computation time may be saved in EP. Many studies have indicated situations where EP outperformed GAs (e.g., [21], [23], [26], [30], [32]–[34], and also [35]–[39]).

A. Brief Review of EP Techniques

Roughly 40 years ago, EP was proposed for the evolution of the finite-state machines in order to solve prediction tasks [20]. Since then, modifications, enhancements, and implementations have been proposed and investigated. EP has been extended to treat real-valued object variables or any other feasible data structure.

Mutation is often implemented by adding a random number or a vector from a certain distribution [e.g., a Gaussian distribution in the case of classical EP (CEP)] to a parent. The degree of variation of the Gaussian mutation is controlled by its standard deviation, which is also known as a 'strategy parameter" in evolutionary search. In the self-adaptation scheme of EP, this parameter is not prefixed; rather, it is evolved along with the objective variables. Experiments with self-adaptive EP have indicated efficient convergence to quality solutions [19], [23], [27], [28], [30].

Manuscript received February 9, 2000; revised March 7, 2002 and July 22, 2002.

The authors are with the Department of Electrical Engineering, Jadavpur University, Calcutta 700032, India.

Digital Object Identifier 10.1109/TEVC.2002.806788

Recently, Yao *et al.* [27] proposed a Cauchy-mutation-based EP, called fast EP (FEP), which demonstrated much better performance than CEP in converging to a near-global optimum point on some benchmark mathematical functions, but not on all. FEP's success can be attributed to its greater ability to escape local minima by using Cauchy mutation. At the same time, this will lead to difficulties in coping with some other functions. Larger jumps are beneficial when the current solution is far away from the global optimum or a better optimum, but such large jumps near the global optimum point are undesirable. Chellapilla and Fogel [16] proposed a fast EP using the weighted mean of Gaussian and Cauchy mutations, called henceforth MFEP, with an objective of having a step size greater than Gaussian mutation and smaller than Cauchy mutation so that advantages of both Gaussian as well as Cauchy mutations can be exploited. Yao *et al.* [28] proposed an improved FEP (IFEP) that uses both Gaussian and Cauchy mutations to create offspring from the same parent and better ones are chosen for next generation. This method outperforms the other EP methods in almost all the benchmark functions they studied.

In general, with little knowledge about the global optimum, it is difficult to constrain the search space to a sufficiently small region. Initial solutions are usually far from the global optimum and hence larger step sizes may prove to be beneficial. But as the evolution progresses, later solutions may be nearer to the global point and the step size should be reduced gradually to enable quick convergence. Unfortunately, there is no way of predicting the transition point for the change of step size from a larger to smaller one. As no unique step size can give desired results for all types of functions, IFEP appears to be most promising simply because it uses both smaller as well as larger step sizes, depending on the immediate necessity. The only drawback is its requirement of more mutation time. In most efforts on self-adaptive EPs, the step size (standard deviation) has been adapted using empirical learning rates $(\sqrt{(2\sqrt{n})})^{-1}$ and $(\sqrt{2n})^{-1}$, where n is the number of objective variables. Many efforts on EP have been demonstrated on purely numerical benchmark optimization problems. Relatively fewer data are available on the performances of CEP, FEP, MFEP, and IFEP on real-world problems, such as power system ELD.

Though there are some reports [26] on the performance of CEP using scaled-cost based adaptation of strategy parameter in solving power system ELD, the performances of FEP, MFEP, and IFEP with this type of adaptation have not yet been tested on power system ELD. Moreover, there are no known reports on the performances of CEP, FEP, MFEP, and IFEP on power system ELD with adaptation based on empirical learning rates. Also, there is no explicit mathematical relationship between the adaptation of step size with empirical learning rate and the rate of convergence toward the global optimum point. Moreover, there are mixed reports regarding the performance of all EPs. Wolpert and Macready [24] have shown that, under certain assumptions, no search algorithm is best for all problems. Thus, the success and quality of a solution achieved by an EP algorithm is dependent on the step size, the method of updating, and the nature and number of suboptimal points, as well as their distances with respect to the globally optimum point and other factors.

All these facts demand investigation and comparison of the performances of CEP, FEP, MFEP, and IFEP with suitable adaptation of strategy parameters using: 1) empirical learning rates and 2) a scaled cost on power system ELD problems. While Part I of this paper studies the performances of FEP, MFEP, and IFEP with scaled-cost-based adaptation on a power system ELD, Part II investigates the absolute as well as relative performances of EPs using empirical learning rates for adaptation with those using scaled-cost-based adaptation in solving a power system ELD. The performances of EP algorithms have been investigated on three test systems of increasing size and complexity on a power system ELD.

The main objectives of the present work are to develop and study the absolute as well as relative performances of the following EP techniques applied to power system ELD problems.

i) CEP with Gaussian mutation with scaled cost as proposed by Yang *et al.* [26];
ii) FEP with Cauchy mutation with scaled cost;
iii) MFEP with mean of Gaussian and Cauchy mutations with scaled cost;
iv) IFEP with better of Gaussian and Cauchy mutations with scaled cost;
v) CEP$''$ with Gaussian mutation with empirical learning rate as described by Bäck *et al.* [23];
vi) FEP$''$ with Cauchy mutation with empirical learning rate as proposed by Yao *et al.* [27];
vii) MFEP$''$ with mean of Gaussian and Cauchy mutations with empirical learning rate as proposed by Chellapilla and Fogel [16];
viii) IFEP$''$ with better of Gaussian and Cauchy mutations with empirical learning rate as proposed by Yao *et al.* [28];

B. ELD Problem Formulation

The ELD load dispatch problem can be described as an optimization (minimization) process with the objective:

$$Min \sum_{j=1}^{n} FC_j(P_j) \tag{1}$$

where $FC_j(P_j)$ is the fuel cost function of the j^{th} unit and P_j is the power generated by the j^{th} unit, subject to power balance constraints:

$$D = \sum_{j=1}^{n} P_j - P_L \tag{2}$$

where D is the system load demand and P_L is the transmission loss, and generating capacity constraints:

$$P_{\mathrm{jmin}} \leq P_j \leq P_{\mathrm{jmax}}, \qquad \text{for} \qquad j = 1, 2, \ldots n \tag{3}$$

where P_{jmin} and P_{jmax} are the minimum and maximum power outputs of the j^{th} unit.

The fuel-cost function without valve-point loadings of the generating units is given by

$$f(P_j) = a_i P_j^2 + b_j P_j + c_j. \tag{4}$$

The fuel-cost function considering valve-point loadings of the generating units is given as

$$f(P_j)=a_jP_j^2+b_jP_j+c_j+|e_j\times\sin(f_j\times(P_{i\,\min}-P_j))| \quad (5)$$

where a_j, b_j, and c_j are the fuel-cost coefficients of the j^{th} unit, and e_j and f_j are the fuel cost-coefficients of the j^{th} unit with valve-point effects.

The generating units with multivalve steam turbines exhibit a greater variation in the fuel-cost functions. The valve-point effects introduce ripples in the heat-rate curves.

II. Part I—EP-Based ELD With Adaptation of Strategy Parameter Using Scaled Cost

Let $p_i = P_1, P_2, \ldots, P_n$ be a trial vector denoting the individual of a population to be evolved. The elements of p_i are the real power outputs of the committed n generating units subject to their respective capacity constraints in (3). To meet the load demand in (2) exactly, a dependent unit is selected arbitrarily from among the committed n units. P_i, the power output of the dependent unit, is calculated by

$$P_d = D + P_L - \sum_{\substack{j=1\\ \neq d}}^{n} P_j. \quad (6)$$

In this work, the power loss is not considered. However, it may be calculated by an iterative algorithm or by using the B-loss matrix of the power system directly.

Each initial parent trial vector p_i, $i = 1, 2, \ldots n$ is determined by setting the j^{th} component $P_j \sim U(P_{\text{jmin}}, P_{\text{jmax}})$, where $j = 1, 2, \ldots, n$. $U(P_{\text{jmin}}, P_{\text{jmax}})$ denotes a uniform random variable ranging over $[P_{\text{jmin}}, P_{\text{jmax}}]$. There are four alternative methods for creating offspring.

i) Using Gaussian mutation (in CEP) as in [26]. An offspring vector p_i' is created from each parent by adding to each component of p_i, a Gaussian random variable with a zero mean and a standard deviation proportional to the scaled cost values of the parent trial solution, i.e.,

$$p_i' = [P_1', P_2', \ldots\ldots, P_n'] \quad (7)$$

$$P_j' = P_j + N(0, \sigma_j^2), \quad \text{for} \quad j = 1, 2, \ldots, n \quad (8)$$

where $N(0, \sigma_j^2)$ represents a Gaussian random variable with mean 0 and standard deviation σ_j.

ii) Using Cauchy mutation (in FEP), an offspring is created by

$$P_j' = P_j + \sigma_j.\, C_j(0, 1), \quad \text{for} \quad j = 1, 2, \ldots, n \quad (9)$$

where $C_j(0, 1)$ is a Cauchy random variable with scale parameter $t = 1$ centered at zero that is generated anew for each value of j.

iii) Using the mean of the Gaussian and Cauchy mutations (in MFEP), an offspring is created in this method as

$$P_j' = P_j + (\sigma_j/2)\{N_j(0, 1) + C_j(0, 1)\} \quad (10)$$

where N_j and C_j are Gaussian and Cauchy random variables, respectively, to be generated anew for each value of j.

iv) Using the method of choosing the better from two offspring generated from each parent, one by Gaussian mutation and the other by Cauchy mutation (in IFEP). Let P_{1j}' and P_{2j}' be the two offspring generated from the parent P_j

$$P_{1j}' = P_j + \sigma_j \cdot N_j(0, 1) \quad (11)$$

$$P_{2j}' = P_j + \sigma_j \cdot C_j(0, 1). \quad (12)$$

The values of the objective function value of both offspring are evaluated, compared, and better individuals are chosen as parents for next generation.

In each method mentioned above, the standard deviation (or scaling factor) σ_j is given by the expression

$$\sigma_j = \beta \cdot f_i / f_{\min}(P_{\text{jmax}} - P_{\text{jmin}}) \quad (13)$$

where $f_{\min}$ is the minimum cost value among the n trial solutions, β is a scaling factor, and $f_i \cong f(p_i)$ is the value of the objective function associated with the trial vector p_i. Here, σ_j is adapted based on its scaled values of the parent trial solution; if f_i is relatively low, the step size will be small and the offspring trial solution will be created near the current solution on average, and if f_i is relatively high, the step size will be larger and next trial solution will be searched within a wider range on average thereby providing an adaptive effect based on the fitness (cost) value. Also, as $f_{\min}$ keeps updating toward a globally optimum point, adaptation will also be toward the global point.

Each individual in the combined population of n parent trial vectors and their corresponding n offspring must compete to have a chance to survive to the next generation. A weight value w_i is assigned to the individual as follows [26]:

$$w_i = \sum_{r=1}^{R} w_r$$

$$w_r = 1, \qquad \text{if} \quad u_1 > \frac{f_i}{f_r + f_i}$$

$$\quad = 0, \qquad \text{otherwise} \quad (14)$$

where R is the number of competitors, f_r is the fitness value of rth randomly selected competitor from 2 N trial solutions based on $r = [2N \cdot u_2 + 1]$, f_i is the fitness value of p_i, and u_1 and u_2 are u(0,1). Individuals are ranked in descending order of their corresponding score, w_r. The first n individuals are selected and transcribed along with their corresponding fitness values f_i to be parents of the next generation.

Each proposed EP algorithm has been implemented in command line in Matlab 5.1 for solution of three test cases of economic load dispatch. Only binary GA and floating-point GA solutions are obtained with the help of the genetic toolbox (GAOT) developed by [40]. The floating-point GA was implemented with heuristic crossover and nonuniform mutation. Heuristic crossover takes two parents and performs an extrapolation along the line formed by the two parents outward in the direction of the better parent. Non-uniform mutation changes one of the parameters of the parent based on a nonuniform probability distribution. This Gaussian distribution starts wide, and narrows to a point distribution as the current generation approaches the maximum generation. All the programs were run on a 350 MHz,

TABLE I
UNITS DATA FOR TEST CASE I (THREE-UNIT CASE) WITH VALVE-POINT LOADING. a, b, c, e, AND f ARE COST COEFFICIENTS IN THE FUEL COST FUNCTION $f(P_j) = a_j P_j^2 + b_j P_j + c_j + |e_j \times \sin(f_j \times (P_{i\,\min} - P_j))|$

Generator	P_{min}(MW)	P_{max}(MW)	a	b	c	e	f
1	100	600	0.001562	7.92	561	300	0.0315
2	50	200	0.004820	7.97	78	150	0.063
3	100	400	0.001940	7.85	310	200	0.042

TABLE II
RESULTS OF TEST CASE I (THREE-UNIT SYSTEM WITH VALVE-POINT LOADING) WITH LOAD DEMAND 850 MW. THE EVALUATION METHODS USED ARE GAB, GAF, CEP, FEP, MFEP, AND IFEP

Evolution method	Mean time (sec.)	Best time (sec.)	Mean cost [$]	Maximum cost ($)	Minimum cost ($)
GAB	35.80	32.46	-------	-------	8234.08
GAF	24.65	23.03	-------	-------	8234.07
CEP	20.46	18.35	8235.97	8241.83	8234.07
FEP	4.54	3.79	8234.24	8241.78	8234.07
MFEP	8.00	6.31	8234.71	8241.80	8234.08
IFEP	6.78	6.11	8234.16	8234.54	8234.07

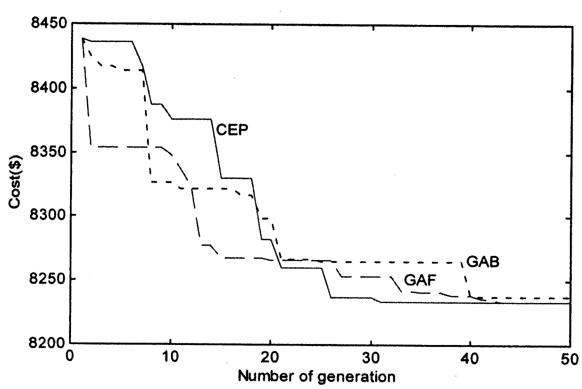

Fig. 1. Convergence nature of GAB, GAF, and CEP in test case I (three-unit case).

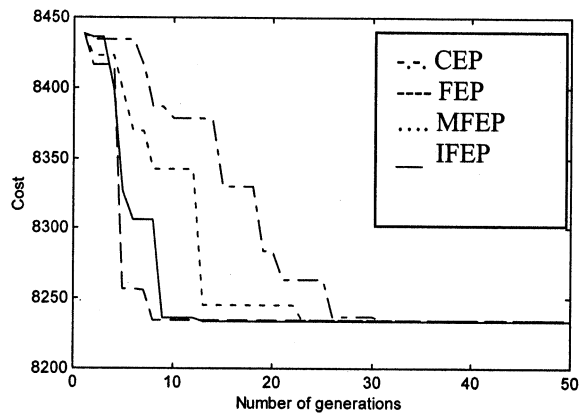

Fig. 2. Convergence nature of CEP, FEP, MFEP, and IFEP in test case I (three-unit case).

TABLE III
RELATIVE FREQUENCY OF CONVERGENCE IN PERCENT FOR TEST CASE I USING CEP, FEP, MFEP, AND IFEP

Evaluation method	Ranges of cost ($)				
	8244 - 8242	8242 - 8240	8240 - 8238	8238 - 8236	8236 - 8234
CEP	-----	24	-----	-----	76
FEP	1	-----	-----	-----	99
MFEP	-----	7	-----	-----	93
IFEP	-----	-----	-----	-----	100

Pentium-II, with 128MB RAM PC. The crossover and mutation rates for binary GA were chosen as 60% and 0.05%, respectively, following common literature.

A. Test Case I

This test case, adapted from [10], comprises three generating units with quadratic cost functions together with the effects of valve-point loadings as given in Table I.

The value of scale factor β was taken as 0.01 after experimental verification. The dependent-unit active power operating limits were enforced through a quadratic penalty function; after experimentation, a penalty multiplier of 1000 was chosen. The value of population size (N) was taken as 20 for all the EPs in this test case after experimentation with population sizes of 10, 20, and 30. To compare the performances of CEP, FEP, MFEP, and IFEP with respect to instances of a GA, test case I was solved with binary as well as a floating-point GA. The results are shown in Table II. The binary GA took more time than the floating-point GA as it converged toward the global optimum. Both GAs converged slowly near the globally optimum point (see Fig. 1). The EP algorithms performed faster than both the GAs in terms of solution time. However, the prime emphasis in this work was to have a comparative performance of FEPs, MFEPs, and IFEPs with respect to CEP, as the performance of CEP on ELD is well reported already. Hence, comparison was made with respect to CEP instead of GAs. Among the EPs, FEP

TABLE IV
UNITS DATA FOR TEST CASE II (13-UNIT CASE) WITH VALVE-POINT LOADING. a, b, c, e, AND f ARE COST COEFFICIENTS IN THE FUEL COST FUNCTION $f(P_j) = a_j P_j^2 + b_j P_j + c_j + |e_j \times \sin(f_j \times (P_{i\,\min} - P_j))|$

Generator	P_{min}(MW)	P_{max}(MW)	a	b	c	e	f
1	00	680	0.00028	8.10	550	300	0.035
2	00	360	0.00056	8.10	309	200	0.042
3	00	360	0.00056	8.10	307	200	0.042
4	60	180	0.00324	7.74	240	150	0.063
5	60	180	0.00324	7.74	240	150	0.063
6	60	180	0.00324	7.74	240	150	0.063
7	60	180	0.00324	7.74	240	150	0.063
8	60	180	0.00324	7.74	240	150	0.063
9	60	180	0.00324	7.74	240	150	0.063
10	40	120	0.00284	8.6	126	100	0.084
11	40	120	0.00284	8.6	126	100	0.084
12	55	120	0.00284	8.6	126	100	0.084
13	55	120	0.00284	8.6	126	100	0.084

TABLE V
RESULTS OF TEST CASE II (13-UNIT SYSTEM WITH VALVE-POINT LOADING) WITH 1800-MW LOAD. THE EVALUATION METHODS USED ARE CEP, FEP, MFEP, and IFEP

Evolution method	Mean time (sec.)	Best time (sec.)	Mean cost ($)	Maximum cost ($)	Minimum cost ($)
CEP	294.96	293.41	18190.32	18404.04	18048.21
FEP	168.11	166.43	18200.79	18453.82	18018.00
MFEP	317.12	315.98	18192.00	18416.89	18028.09
IFEP	157.43	156.81	18127.06	18267.42	17994.07

outperformed all other EPs in terms of the convergence rate, as well as solution time (see Fig. 2 and Tables II and III). The convergence rate at a particular generation was calculated as the difference of the costs at the initial generation and the particular generation divided by the number of generations.

Table III shows the frequency of attaining a cost within specific ranges out of 100 runs for each of the four EP algorithms with 100 different initial trial solutions. IFEP has the highest probability of achieving better solutions followed by FEP; however, the convergence rate and solution time of IFEP in achieving the minimum cost is slightly slower than FEP because of increased mutation time. CEP showed the poorest performance amongst the four EPs in terms of convergence rate and observed likelihood of attaining the minimum cost.

B. Test Case II

Test case II [11] consisted of 13 generating units with the valve-point loadings as given in Table IV. This is a larger system with more nonlinearities.

A population size of 30 was used after experimentation with population sizes of 20, 30, 40, and 50. The value of the scale factor β and the penalty factor, 0.01 and 1000, respectively, were found to give better and satisfactory performance in this test case. The simulation results of the EPs are given in Tables V and VI.

Table VI depicts the frequency of attaining a cost within the specific ranges out of 50 runs for each of the four EP algorithms with 50 different initial trial solutions. Fig. 3 and Tables V and VI reveal that CEP performed worst in terms of convergence rate. FEP converged faster and attained lower minimum cost than either of CEP or MFEP, but it had higher average cost. Higher average cost implies that, on average, the quality of solutions discovered by FEP was worse than either of CEP or MFEP. This might reflect FEP's tendency to spend time competing between different nearby hills rather than improving along a single hill that contains the globally optimum point. MFEP possessed the highest solution time. IFEP appeared to be the best amongst the four in this test case in terms of convergence rate, solution time, probability of attaining better solutions, mean cost, and minimum cost achieved.

C. Test Case III

Test case III consisted of 40 generating units, adapted from [3], with modifications to incorporate the valve-point loading. Table VII shows the units' data. The values of population size, scale factor, and penalty multiplier, 60, 0.05, and 100, respectively, were found to give better and satisfactory performance in this test case by experimentation. Simulation results are given in Tables VIII and IX.

Table VIII shows the best solution time, maximum cost, mean cost, and minimum cost achieved by the EP algorithms. FEP required the least solution time, followed by IFEP. The minimum cost achieved by IFEP was the best, followed by FEP. MFEP was the slowest amongst the four.

Table IX shows the frequency of attaining a cost within the specific ranges out of 50 runs for each of the four EP algorithms with 50 different initial trial solutions. IFEP achieved the best mean cost followed by that of MFEP. The average cost, as well as the minimum cost achieved by CEP was the highest (worst), followed by FEP. MFEP performed better than FEP in terms of mean cost as well as minimum cost but its solution time in this test case was the highest. Fig. 4 and Tables VIII and IX reveal that IFEP had the fastest convergence rate, the best probability

TABLE VI
RELATIVE FREQUENCY OF CONVERGENCE IN PERCENT FOR TEST CASE II USING CEP, FEP, MFEP, AND IFEP

Evaluation method	Ranges of cost ($)										
	18500 - 18450	18450 - 18400	18400 - 18350	18350 - 18300	18300 - 18250	18250 - 18200	18200 - 18150	18150 - 18100	18100 - 18050	18050 - 18000	18000 - 17950
CEP	----	2	4	10	8	18	16	34	6	2	----
FEP	2	4	4	2	16	14	26	18	12	2	----
MFEP	-----	6	----	8	12	12	26	32	2	2	-----
IFEP	----	----	----	----	2	14	22	26	22	12	02

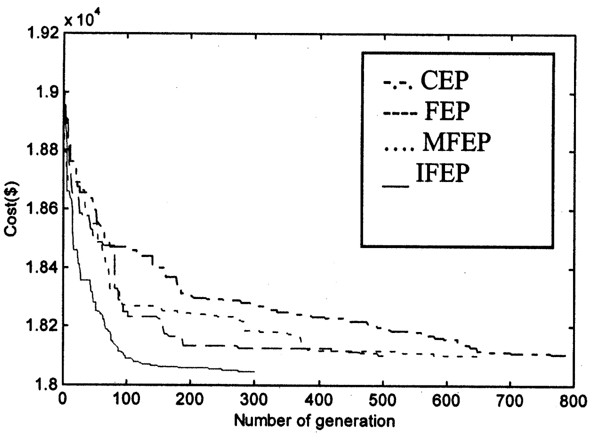

Fig. 3. Convergence nature of CEP, FEP, MFEP, and IFEP in test case II (13–unit case).

of attaining better solutions, and the best average, as well as minimum cost amongst the four EPs in this test case. Hence, for power system ELD problems of greater size with more nonlinearities, IFEP proved to be the best algorithm amongst the four EPs.

III. PART II— EP-BASED ELD WITH ADAPTATION OF STRATEGY PARAMETER USING EMPIRICAL LEARNING RATE

The EP method employed is based on the technique offered in [23] by Bäck *et al.*

An initial population of N individuals is generated. Each individual is taken as a pair of real-valued vectors $(p_i,\, s_i)$, $\forall_i \in \{1, 2, \ldots N\}$, where p_i's are the objective variables and determined by setting the jth component $P_j \sim U(P_{\text{jmin}}, P_{\text{jmax}})$, for $j = 1, 2, \ldots n$, and $U(P_{\text{jmin}}, P_{\text{jmax}})$ denotes a uniform random variable ranging over $[P_{\text{jmin}}, P_{\text{jmax}}]$, and s_i's are standard deviations for Gaussian mutation (also known as 'strategy parameters' in the self-adaptive evolutionary algorithms). Here, s_i is initialized to a suitable value, chosen experimentally.

TABLE VII
UNITS DATA FOR TEST CASE III (40-UNIT CASE) WITH VALVE-POINT LOADING. HERE, a, b, c, d, AND f ARE COST COEFFICIENTS IN THE FUEL COST FUNCTION $f(P_j) = a_j P_j^2 + b_j P_j + c_j + |e_j \times \sin(f_j \times (P_{i\,\min} - P_j))|$

Generator	P_{min}(MW)	P_{max}(MW)	a	b	c	e	f
1	36	114	0.00690	6.73	94.705	100	0.084
2	36	114	0.00690	6.73	94.705	100	0.084
3	60	120	0.02028	7.07	309.54	100	0.084
4	80	190	0.00942	8.18	369.03	150	0.063
5	47	97	0.0114	5.35	148.89	120	0.077
6	68	140	0.01142	8.05	222.33	100	0.084
7	110	300	0.00357	8.03	287.71	200	0.042
8	135	300	0.00492	6.99	391.98	200	0.042
9	135	300	0.00573	6.60	455.76	200	0.042
10	130	300	0.00605	12.9	722.82	200	0.042
11	94	375	0.00515	12.9	635.20	200	0.042
12	94	375	0.00569	12.8	654.69	200	0.042
13	125	500	0.00421	12.5	913.40	300	0.035
14	125	500	0.00752	8.84	1760.4	300	0.035
15	125	500	0.00708	9.15	1728.3	300	0.035
16	125	500	0.00708	9.15	1728.3	300	0.035
17	220	500	0.00313	7.97	647.85	300	0.035
18	220	500	0.00313	7.95	649.69	300	0.035
19	242	550	0.00313	7.97	647.83	300	0.035
20	242	550	0.00313	7.97	647.81	300	0.035
21	254	550	0.00298	6.63	785.96	300	0.035
22	254	550	0.00298	6.63	785.96	300	0.035
23	254	550	0.00284	6.66	794.53	300	0.035
24	254	550	0.00284	6.66	794.53	300	0.035
25	254	550	0.00277	7.10	801.32	300	0.035
26	254	550	0.00277	7.10	801.32	300	0.035
27	10	150	0.52124	3.33	1055.1	120	0.077
28	10	150	0.52124	3.33	1055.1	120	0.077
29	10	150	0.52124	3.33	1055.1	120	0.077
30	47	97	0.01140	5.35	148.89	120	0.077
31	60	190	0.00160	6.43	222.92	150	0.063
32	60	190	0.00160	6.43	222.92	150	0.063
33	60	190	0.00160	6.43	222.92	150	0.063
34	90	200	0.0001	8.95	107.87	200	0.042
35	90	200	0.0001	8.62	116.58	200	0.042
36	90	200	0.0001	8.62	116.58	200	0.042
37	25	110	0.0161	5.88	307.45	80	0.098
38	25	110	0.0161	5.88	307.45	80	0.098
39	25	110	0.0161	5.88	307.45	80	0.098
40	242	550	0.00313	7.97	647.83	300	0.035

The fitness scores for each individual $(U(p_i, s_i)$, $\forall_i \in \{1, 2, \ldots, N\})$ of the population are evaluated based on the objective function $f_i \cong f(p_i)$.

TABLE VIII
RESULTS OF TEST CASE III (40-UNIT SYSTEM) WITH VALVE-POINT LOADING FOR LOAD DEMAND 10500 MW. THE EVALUATION METHODS USED ARE CEP, FEP, MFEP, AND IFEP

Evolution method	Mean time (sec.)	Best time (sec.)	Mean cost ($)	Maximum cost ($)	Minimum cost ($)
CEP	1956.93	1955.20	124793.48	126902.89	123488.29
FEP	1039.16	1037.90	124119.37	127245.59	122679.71
MFEP	2196.10	2194.70	123489.74	124356.47	122647.57
IFEP	1167.35	1165.70	123382.00	125740.63	122624.35

TABLE IX
RELATIVE FREQUENCY OF CONVERGENCE IN PERCENT FOR TEST CASE II USING CEP, FEP, MFEP, AND IFEP

Evaluation method	Ranges of cost ($)								
	127000 –126500	126500 – 126000	126000 – 125500	125500 – 125000	125000 – 124500	124500 – 124000	124000 – 123500	123500 - 123000	123000 - 122500
CEP	10	4	----	16	22	42	4	2	----
FEP	6	----	4	2	10	20	26	24	6
MFEP	----	----	----	----	----	14	26	50	10
IFEP	----	----	2	----	4	4	18	50	22

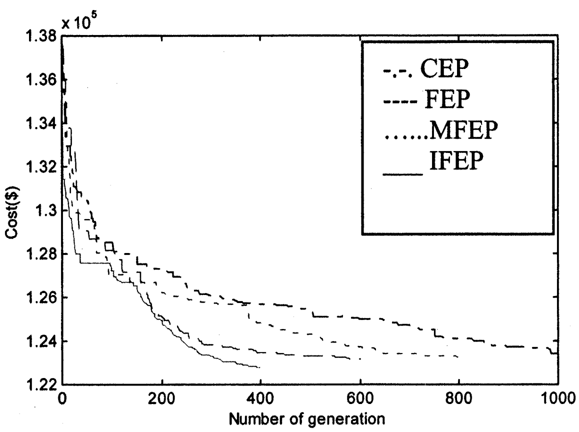

Fig. 4. Convergence nature of CEP, FEP, MFEP, and IFEP in test case III (40–unit case).

Four mutation schemes, explained below, are employed for creation of offspring.

i) Using Gaussian mutation (in CEP″)

$$S'_j = S_j \exp\{\tau' N(0,1) + \tau N_j(0,1)\} \tag{15}$$
$$P'_j = P_j + S'_j N_j(0,1) \tag{16}$$

where S_j, S'_j, P_j, and P'_j denote the jth component of vectors s_j, s'_j, p_j, and p'_j, respectively. $N(0,1)$ denotes a normally distributed random number with mean 0 and standard deviation 1, and $N_j(0,1)$ denotes the random number generated anew for each value of j. The factors"τ" and "τ'" are called learning rates and are commonly set to $(\sqrt{(2\sqrt{n})})^{-1}$ and $(\sqrt{(2n)})^{-1}$, respectively, where n is the number of objective variables.

ii) Using Cauchy mutation (in FEP″), an offspring from each parent is created by

$$P'_j = P_j + S'_j C_j(0,1) \tag{17}$$

where C_j is a Cauchy random variable with scale parameter $t = 1$ and generated anew for each value of j.

iii) Using the mean of Gaussian and Cauchy mutations (in MFEP″): an offspring from each parent is generated by

$$P'_j = P_j + \frac{1}{2} \cdot S'_j\{N_j(0,1) + C_j(0,1)\}. \tag{18}$$

iv) Using the better of the two offspring generated from each parent, one by Gaussian mutation and the other by Cauchy mutation (in IFEP″)

$$P'_{1j} = P_j + S'_j N_j(0,1) \tag{19}$$
$$P'_{2j} = P_j + S'_j C_j(0,1) \tag{20}$$

where P'_{1j} and P'_{2j} are the two offspring generated by Gaussian and Cauchy mutations, respectively. The objective function value of both the offspring are evaluated, compared, and better individuals are chosen as parents for the next generation. S'_j is updated in schemes ii), iii), and iv) as in scheme v) above.

The fitness of each offspring ($U(p'_i, s'_i)$, $\forall_i \in \{1, 2, \ldots, n\}$) is calculated.

Competition and selection is the same as in Part I.

IV. ANALYSIS OF RESULTS

A. Test Case I

Different initial values of strategy parameters were tried and 3.0 was found to give better performance for this test case. The same population size of $N = 20$ and the same initial populations as in test case I in Part I were used to compare performances of EPs. The simulation results of EPs (CEP″, FEP″, MFEP″, and IFEP″) with adaptation using empirical learning rates are shown in Table X and in Fig. 5.

Table X shows the best solution time, as well as the mean, maximum, and minimum cost achieved in this test case. It has been observed that the frequency of attaining a cost within specific ranges out of 100 runs for each of the four algorithms with 100 different initial trial solutions is 100. Fig. 5 and Table X show that the best convergence rate as well as the best solution time amongst the four was achieved by FEP″, followed by IFEP″. CEP″ was the worst performer, followed by MFEP″. All the four EP algorithms had almost similar frequency of attaining a cost in this test case.

TABLE X
RESULTS OF TEST CASE I (THREE-UNIT SYSTEM WITH VALVE-POINT LOADING) WITH LOAD DEMAND 850 MW. THE EVALUATION METHODS USED ARE CEP, FEP, MFEP, AND IFEP″ WITH EMPIRICAL LEARNING RATE. CEP, FEP, MFEP, IFEP⇒ EVOLUTIONARY PROGRAMS WITH SCALED COST-BASED ADAPTATION IN PART I

Evolution method	Mean time (sec.)	Best time (sec.)	Mean cost [$]	Maximum cost ($)	Minimum cost ($)
CEP″	16.17	14.61	8234.17	8234.44	8234.08
FEP″	5.22	5.00	8234.12	8234.22	8234.07
MFEP″	9.38	8.63	8234.16	8234.24	8234.07
IFEP″	6.77	6.37	8234.11	8234.20	8234.07
CEP	20.46	18.35	8235.97	8241.83	8234.07
FEP	4.54	3.79	8234.24	8241.78	8234.07
MFEP	8.00	6.31	8234.71	8241.80	8234.08
IFEP	6.78	6.11	8234.16	8234.54	8234.07

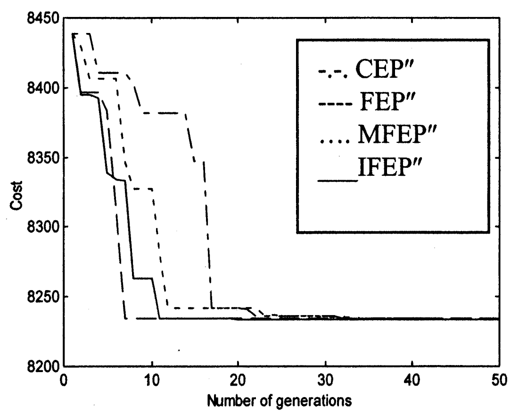

Fig. 5. Convergence nature of CEP″, FEP″, MFEP″, and IFEP″ in test case I (three–unit case).

1) *Comparative Performance With Their Counterparts in Part I:* The comparative responses of CEP and CEP″, and FEP and FEP″, are given in Fig. 6. The comparative responses of MFEP and MFEP″, and IFEP and IFEP″, are given Fig. 7. Tables III, and X and Figs. 6 and 7 reveal that FEP performed best amongst all EPs in Part I and Part II in terms of convergence rate and solution time followed by FEP″. All the EPs with empirical learning rates in Part II had almost similar empirical likelihood (1.00) of attaining the minimum cost. There was no significant superiority of EPs with empirical learning rates over their counterparts in Part I in this test case. Both IFEP and IFEP″ had almost the same competence as FEP in terms of convergence rate and probability of attainment of minimum cost. The solution time of both IFEP and IFEP″, however, were greater as compared to FEP and FEP″ because of their requirement of more mutation time.

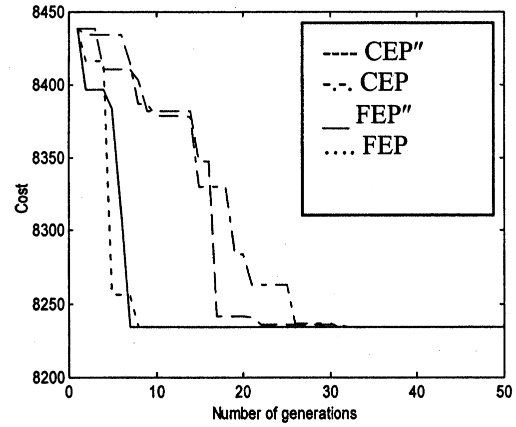

Fig. 6. Comparative convergence nature of CEP″, CEP, FEP″, and FEP in test case I.

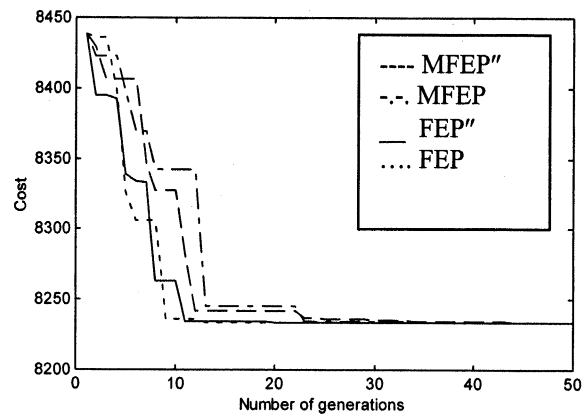

Fig. 7. Comparative convergence nature of MFEP″, MFEP, IFEP″, and IFEP in test case I.

TABLE XI
RESULTS OF TEST CASE II (13-UNIT CASE) WITH VALVE-POINT LOADING FOR LOAD DEMAND 1800 MW. THE EVALAUATION METHODS USED ARE CEP″, FEP″, MFEP″, AND IFEP″ WITH EMPIRICAL LEARNING RATE. CEP, FEP, MFEP, IFEP⇒ EVOLUTIONARY PROGRAMS WITH SCALED COST-BASED ADAPTATION IN PART I

Evolution method	Mean time (Sec.)	Best time (sec.)	Mean cost ($.)	Maximum cost ($)	Minimum cost ($)
CEP″	305.23	303.03	18184.49	18298.93	18016.88
FEP″	174.17	172.46	18229.99	18349.38	18028.57
MFEP″	323.89	322.52	18208.12	18305.47	18021.34
IFEP″	164.87	163.13	18174.91	18303.03	18012.97
CEP	294.96	293.41	18190.32	18404.04	18048.21
FEP	168.11	166.43	18200.79	18453.82	18018.00
MFEP	317.12	315.98	18192.00	18416.89	18028.09
IFEP	157.43	156.81	18127.06	18267.42	17994.07

TABLE XII
RELATIVE FREQUENCY OF CONVERGENCE IN PERCENT FOR TEST CASE II USING CEP″ FEP″, MFEP″, AND IFEP″ WITH EMPIRICAL LEARNING RATE

Evaluation method	Ranges of cost ($)										
	18500 - 18450	18450 - 18400	18400 - 18350	18350 - 18300	18300 - 18250	18250 - 18200	18200 - 18150	18150 - 18100	18100 - 18050	18050 - 18000	18000 - 17950
CEP″	----	----	----	2	12	28	26	28	2	2	----
FEP″	----	----	2	14	32	30	10	10	----	2	----
MFEP″	----	----	----	6	34	14	28	16	----	2	-----
IFEP″	----	----	----	2	12	18	38	18	10	2	----

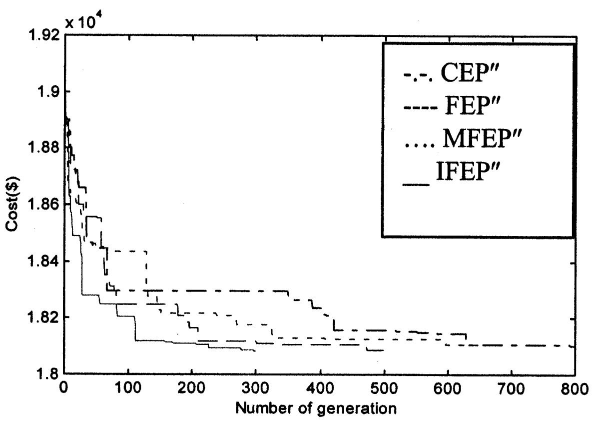

Fig. 8. Convergence nature of CEP″, FEP″, MFEP″, and IFEP″ in test case II (13–unit case).

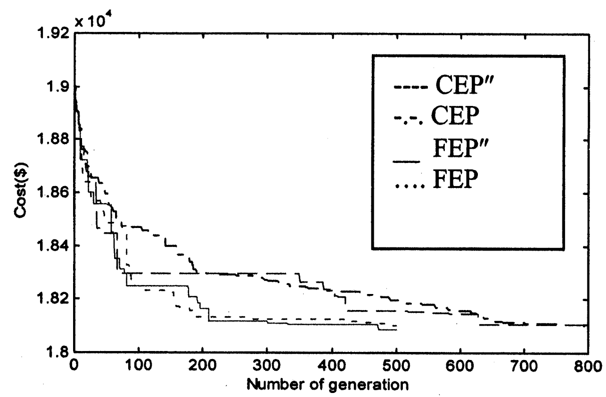

Fig. 9. Comparative convergence nature of CEP″, CEP, FEP″, and FEP in test case II.

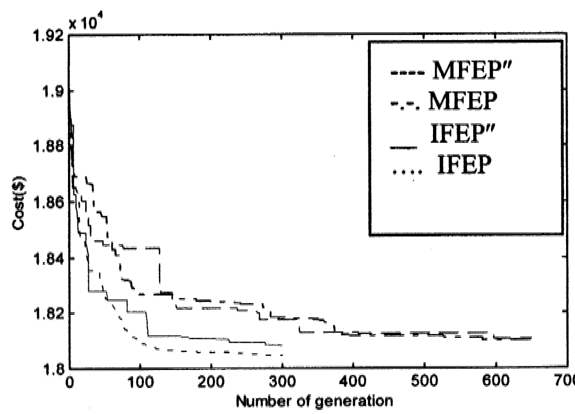

Fig. 10. Comparative convergence nature of MFEP″, MFEP, IFEP″, and IFEP in test case II.

B. Test Case II

In this test case, the same population of 30 and the same initial populations as in test case II in Part I were used for all the four EP algorithms for comparison. The initial strategy parameter of 3.0 and penalty of 1000 were used in this test case after experimentation. The simulation results are shown in Tables XI and XII and in Fig. 8.

Table XI shows the best solution time, and the mean, maximum, and minimum cost achieved by the algorithms, while Table XII shows the probability of attainment of a cost within specific ranges out of 50 runs for each of the four algorithms with 50 different initial trial solutions.

Table XI shows that IFEP″ possessed the best solution time amongst the four EPs (CEP″, FEP″, MFEP″, and IFEP″), followed by FEP″. MFEP″ proved to be the slowest. Observation of Table XI reveals that IFEP″ performed the best amongst the four in terms of solution time, the mean, and minimum cost, followed by CEP″. Though FEP″ had faster convergence rate than either of CEP″ or MFEP″, it has higher average cost, implying that the quality of solutions achieved by it on average was below that achieved by either CEP″ or MFEP″. IFEP″ was the best performer amongst the four in terms of convergence rate, solution time, mean cost, minimum cost achieved, and probability of attaining better solutions.

1) Comparative Performance With Their Counterparts in Part I: Tables VI, XI, and XII and Figs. 9 and Figs. 10 show that IFEP outperformed all EPs in both Part I and Part II in terms of all measured parameters. MFEP″ was the worst performer amongst all in terms of solution time. The performances

TABLE XIII
RESULTS OF TEST CASE III (40-UNIT CASE) WITH VALVE-POINT LOADING FOR LOAD DEMAND 10500 MW. THE EVALUTION METHODS USED ARE CEP″, FFEP″, MFEP″, AND IFEP″ WITH EMPIRICAL LEARNING RATE. CEP, FEP, MFEP, IFEP⇒ EVOLUTIONARY PROGRAMS WITH SCALED COST-BASED ADAPTATION IN PART I

Evolution method	Mean time (sec.)	Best time (sec.)	Mean cost ($)	Maximum cost ($)	Minimum cost ($)
CEP″	2029.55	2027.52	125066.11	126702.60	123983.53
FEP″	1070.88	1069.79	125504.08	127026.64	124518.59
MFEP″	2237.13	2235.85	125021.73	126321.64	123743.73
IFEP″	1211.81	1210.40	124862.42	126180.47	123292.23
CEP	1956.93	1955.20	124793.48	126902.89	123488.29
FEP	1037.90	1037.90	124119.37	127245.59	122679.71
MFEP	2196.10	2194.70	123489.74	124356.47	122647.57
IFEP	1167.35	1165.70	123382.00	125740.63	122624.35

TABLE XIV
RELATIVE FREQUENCY OF CONVERGENCE IN PERCENT FOR TEST CASE III USING CEP″, FEP″, MFEP″, AND IIFEP″ WITH EMPIRICAL LEARNING RATE

Evaluation method	Ranges of cost ($)								
	127000 – 126500	126500 – 126000	126000 – 125500	125500 – 125000	125000 – 124500	124500 – 124000	124000 - 123500	123500 - 123000	123000 - 122500
CEP″	6	6	16	20	24	26	2	----	----
FEP″	----	10	40	36	14	----	----	----	----
MFEP″	----	4	14	36	32	14	----	----	----
IFEP″	----	4	12	24	32	18	8	2	----

of all other EPs in Part II had no significant superiority over their counterparts in Part I.

C. Test Case III

The same population size of 60 and the same initial populations as in test case III in Part I were also used here. A strategy parameter of 1.5 and a penalty of 100 were found to give better and satisfactory performance in this test case. The simulation results are produced in Tables XIII and XIV and in Fig. 11.

Table XIII shows the best time, and the mean, maximum, and minimum cost achieved by the algorithms and Table XIV provides the frequency of attainment of a cost within specific ranges out of 50 runs for each of the four algorithms with 50 different

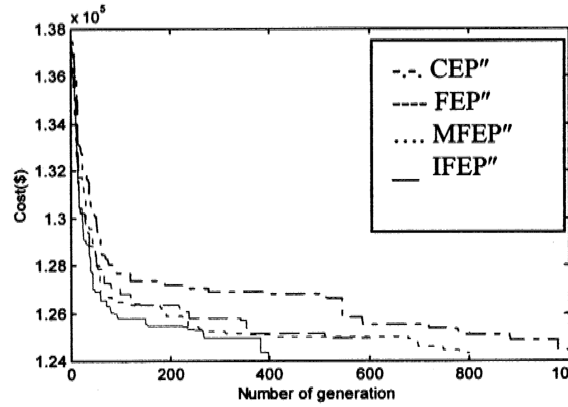

Fig. 11. Convergence nature of CEP″, FEP″, MFEP″, and IFEP″ in test case III (40–unit case).

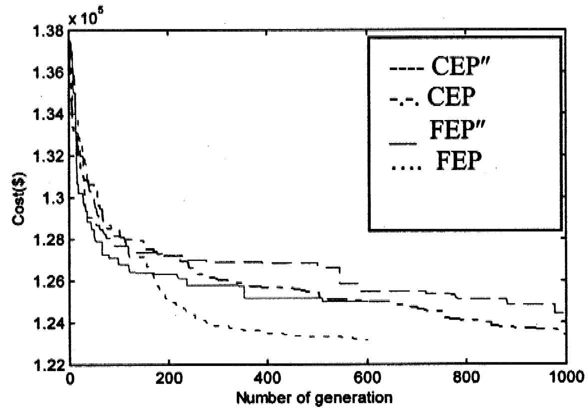

Fig. 12. Comparative convergence nature of CEP″, CEP, FEP″, and FEP in test case III (40–unit case).

initial trial solutions. Fig. 11 shows the convergence nature of all the four algorithms in solving the test case.

The tables and the figure reveal that IFEP″ performed best amongst the four (CEP″, FEP″, MFEP″, and IFEP″) in terms of convergence rate, the mean, and minimum cost achieved. Though FEP″ outperformed IFEP″ in terms of solution time, its average cost achieved was the highest amongst the four. That is, on average, the quality of solution achieved by FEP″ was the worst amongst the four. MFEP″ performed better than either of CEP″ or FEP″ in terms of average cost and minimum cost. However, it possessed the highest solution time. IFEP″ had a higher probability of attaining better solutions than any of other three algorithms in Part II, followed by CEP″.

1) Comparative Performance With Their Counterparts in Part I: Tables IX, XIII, and XIV and Figs. 12 and 13 show that all EPs with mutations based on scaled cost as in Part I performed better than their respective counterparts in Part II in terms of solution time and minimum cost. Though the convergence rates of all EPs with empirical learning rates were initially faster but near the global optimum, their rates fell bellow their counterparts. IFEP outperformed all EPs, both in Part I and Part II, in terms of convergence rate and the mean, minimum cost, and the probability of attaining better solutions.

Though FEP had the least solution time amongst all, its average cost was higher than that of MFEP. The performance of

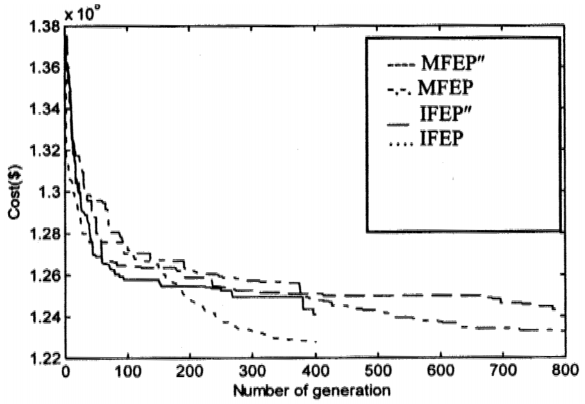

Fig. 13. Comparative convergence nature of MFEP$''$, MFEP, IFEP$''$, and IFEP in test case III (40–unit case).

MFEP came next to IFEP amongst all EPs in both parts in terms of convergence rate, mean cost, minimum cost, and probability of attainment of better solutions. However, MFEP had the highest solution time amongst all EPs with the exception of MFEP$''$.

V. Conclusion

Evolutionary programs with adaptations based on scaled cost, as well as empirical learning rate, have been developed and their performances examined on three test cases of a power system ELD. In smaller problems, all fast EPs performed much better than EP with Gaussian mutation (CEP, CEP$''$) in terms of convergence rate, solution time, minimum cost, and probability of attaining better solutions. In such cases, FEP performed better than all other EPs with both types of adaptations and required comparatively less solution time. This was true even if nonlinearities such as the valve-point loadings were present. But as the size of the problem increases, as in large-scale modern power systems, IFEP offered superior performance over all other EPs with both types of adaptations as revealed in Part I and II. Though the solution time of IFEP in case of larger systems with more nonlinearities is slightly higher than FEP, it offered a higher convergence rate and better solution quality as compared to FEP.

Little difference in performance of EPs was seen between Part I and II in test case I. However, as the system size increased, the difference in their performance became prominent. Both in test case I and II, the performance of EPs with adaptation based on scaled cost were superior to their respective counterparts with empirical learning rate. IFEP performed the best in all respects in comparison to any of the EP techniques considered in Part I and II especially in large power systems with more nonlinearities. As power systems are usually large-scale systems, IFEP may be suggested for the solution of ELD problems. In the problems studied here, it ensured better convergence rate, quality of average solutions, minimum cost to be achieved, and higher probability of attainment of better solutions.

Acknowledgment

The authors would like to thank Dr. D. B. Fogel, Prof. E. Stadlober, and K.-H. Liang for their helpful suggestions and comments during the research for this paper. The authors are also thankful to Jadavpur University for providing necessary facilities.

References

[1] 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, pp. 544–549, May 1992.

[2] A. Bakirtzis, V. Petridis, and S. Kazarlis, "A genetic algorithm solution to the economic dispatch problem," in *Proc. Inst. Elect. Eng.—Generation, Transmission, Distribution*, vol. 141, July 1994, pp. 377–382.

[3] P.-H. Chen and H.-C. Chang, "Large-scale economic dispatch by genetic algorithm," *IEEE Trans. Power Syst.*, vol. 10, pp. 1919–1926, Nov. 1995.

[4] G.B. Sheble and K. Brittig, "Refined genetic algorithm-economic dispatch example," *IEEE Trans. Power Syst.*, vol. 10, pp. 117–124, Feb. 1995.

[5] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison–Wesley, 1989.

[6] H. Mori and T. Horiguchi, "A genetic algorithm based approach to economic load dispatching," in *Proc. 2nd Int. Forum Applic. Neural Networks to Power Systems (ANNPS '93)* , Yokohama, Japan, Apr. 1993, pp. 145–150.

[7] J. T. Alander, "An indexed bibliography of genetic algorithms in power engineering," in *Practical Handbook of Genetic Algorithms: New Frontiers*, L. Chambers, Ed. Boca Raton, FL: CRC Press, 1995, vol. 2.

[8] M. Yoshimi, K. S. Swarup, and Y. Izui, "Optimal economic power dispatch using genetic algorithms," in *Proc. 2nd Int. Forum Applic. Neural Networks to Power Systems (ANNPS '93)* , Yokohama, Japan, Apr. 1993, pp. 157–162.

[9] Y. H. Song, G. S. Wang, P. Y. Wang, and A. T. Johns, "Environmental/economic dispatch using fuzzy logic controlled genetic algorithms," in *Proc. Inst. Elect. Eng. C*, vol. 144, July 1993, pp. 377–381.

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

[11] K. P. Wong and Y. W. Wong, "Thermal generator scheduling using hybrid genetic/simulated annealing approach," in *IEE Proc. C*, vol. 142, July 1995, pp. 372–380.

[12] K. S. Swarup and S. Yamashiro, "Unit commitment solution methodologies using genetic algorithm," *IEEE Trans. Power Syst.*, vol. 17, pp. 87–91, Feb. 2002.

[13] A. G. Bakirtzis, P. N. Biskas, C. E. Zoumas, and V. Petridis, "Optimal power flow by enhance genetic algorithm," *IEEE Trans. Power Systems*, vol. 17, pp. 229–236, May 2002.

[14] D. B. Fogel, "An introduction to simulated evolutionary optimization," *IEEE Trans. Neural Networks*, vol. 5, pp. 3–14, 1994.

[15] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ: IEEE Press, 1995.

[16] K. Chellapilla and D. B. Fogel, "Two new mutation operators for enhanced search and optimization in evolutionary programming," in *SPIE Int. Symp. Optical Science and Engineering Instrum. Conf., 3165: Applic. Soft Comput.*. Bellingham, WA: SPIE Press, pp. 260–269.

[17] K. Chellapilla, "Combining mutation operators in evolutionary programming," *IEEE Trans. Evol. Comput.*, vol. 2, pp. 91–96, 1998.

[18] D. B. Fogel, "An analysis of evolutionary programming," in *Proc. 1st Annu. Conf. Evolutionary Programming*, D. B. Fogel and W. Atmar, Eds. La Jolla, CA, 1992, pp. 43–51.

[19] L. J. Fogel, D. B. Fogel, and P. J. Angeline, "A preliminary investigation on extending evolutionary programming to include self-adaptation on finite state machines," *Informatica*, vol. 18, pp. 387–398, 1994.

[20] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence Through Simulated Evolution*. New York: Wiley, 1966.

[21] L. L. Lai, "Evolutionary programming for economic load dispatch of units with nonsmooth input-output characteristic functions," in *Intelligent System Applications in Power Engineering—Evolutionary Programming and Neutral Networks*. Chichester, U.K.: Wiley, 1998, ch. 8.

[22] J. T. Ma and L. L. Lai, "Evolutionary programming approach to reactive power planning," in *Proc. Inst. Elect. Eng. C*, vol. 143, July 1996, pp. 365–370.

[23] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evol. Comput.*, vol. 1, pp. 2–23, 1993.

[24] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 67–82, 1997.

[25] K. P. Wong and J. Yuryevich, "Evolutionary programming based algorithm for environmentally-constrained economic dispatch," *IEEE Trans. Power Syst.*, vol. 13, pp. 301–306, May 1998.

[26] 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, pp. 112–118, Feb. 1996.

[27] X. Yao and Y. Liu, "Fast evolutionary programming," in *Proc. 5th Annu. Conf. Evolutionary Programming*, L. J. Fogel, T. Bäck, and P. J. Angeline, Eds. Cambridge, MA, 1996, pp. 451–460.

[28] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 82–102, July 1999.

[29] K. P. Wong and C. C. Fung, "Simulated annealing based economic dispatch algorithm," in *Proc. Inst. Elect. Eng. C*, vol. 140, 1992, pp. 544–550.

[30] D. B. Fogel, "A comparison of evolutionary programming and genetic algorithms on selected constrained optimization problems," *Simulation*, pp. 397–404, June 1995.

[31] L. Devroye, *Non-Uniform Random Variate Generation*. New York: Springer-Verlag, 1986.

[32] D. B. Fogel, "An evolutionary approach to the traveling salesman problem," *Biol. Cybern.*, vol. 60, no. 2, pp. 139–144, 1988.

[33] D. B. Fogel and J. W. Atmar, "Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems," *Biol. Cybern.*, vol. 63, no. 2, pp. 111–114, 1990.

[34] D. B. Fogel and L. C. Stayton, "On the effectiveness of crossover in simulated evolutionary optimization," *BioSystems*, vol. 32, no. 3, pp. 171–182, 1994.

[35] T. Jones, "Crossover, macromutation, and population-based search," in *Proc. 5th Int. Conf. Genetic Algorithms*, L. Eshelman, Ed. San Mateo, CA: Morgan Kaufman, 1995, pp. 73–80.

[36] L. Eshelman and J. D. Schaffer, "Crossover's niche," in *Proc. 5th Int. Conf. Genetic Algorithms*, S. Forrest, Ed. San Mateo, CA: Morgan Kaufman, 1993, pp. 9–14.

[37] L. D. Davis, "Bit-climbing, representational bias, and test suite design," in *Proc. 4th Int. Conf. Genetic Algorithms*, R. K. Belew and L. B. Booker, Eds. San Mateo, CA: Morgan Kaufman, 1991, pp. 18–23.

[38] P. J. Angeline, "Subtree crossover: building block engine or macromutation?," in *Proc. Genetic Programming 1997*, J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba, and R. L. Riolo, Eds. San Francisco, CA, 1997, pp. 9–17.

[39] K. Chellapilla, "Evolving computer programs without subtree crossover," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 209–216, 1997.

[40] C. R. Houck, J. A. Joines, and M. Kay, . Cambridge, MA: Free Software Foundation, Inc.