Evolutionary programming approach to reactive power planning

J.T.Ma
L.L.Lai

Abstract: The paper proposes an application of evolutionary programming (EP) to reactive power planning (RPP). RPP is a nonsmooth and nondifferentiable optimisation problem for a multiobjective function. Several techniques to make EP practicable have been developed. The proposed approach is demonstrated with the IEEE 30-bus system. The comprehensive simulation results show that EP is a suitable method to solve the RPP problem. A conventional optimisation method is used as the comparison method. The comparison shows that EP is better than the conventional method in the RPP problem.

List of symbols

N_l = set of numbers of load level durations

N_E = set of branch numbers

N_c = set of numbers of possible reactive power source installation buses

N_i = set of numbers of buses adjacent to bus i, including bus i

N_{PQ} = set of numbers of PQ–buses, which are load buses with constant P and Q injections

N_g = set of generator bus numbers

N_T = set of numbers of tap–setting transformer branches

N_B = set of numbers of total buses

N_{B-1} = set of numbers of total buses, excluding slack bus

h = per unit energy cost (£/p.u.Wh, with S_B = 100MVA)

d_l = duration of load level (h)

g_k = conductance of branch k (p.u.)

V_i = voltage magnitude at bus i (p.u.)

θ_{ij} = voltage angle difference between bus i and bus j (rad)

e_i = fixed reactive power source installation cost at bus i (£)

C_{ci} = per unit reactive power source purchase cost at bus i (£/p.u.VAR, S_B = 100MVA)

Q_{ci} = reactive power source installation at bus i (p.u.)

P_i, Q_i = real and reactive powers, respectively, injected into network at bus i (p.u.)

G_{ij}, B_{ij} = mutual conductance and susceptance, respectively, between bus i and bus j (p.u.)

G_{ii}, B_{ii} = self conductance and susceptance, respectively, of bus i (p.u.)

Q_{gi} = reactive power generation at bus i (p.u.)

T_k = tap–setting of transformer branch k (p.u.)

N_{VPQlim} = set of numbers of PQ-buses at which voltages violate the limits

N_{Qglim} = set of numbers of buses at which reactive power generations violate the limits.

1 Introduction

Reactive power planning (RPP) is one of the most complex problems of power systems as it requires the simultaneous minimisation of two objective functions. The first objective deals with the minimisation of real power losses in reducing the operation cost and improving the voltage profile. The second objective minimises the allocation cost of additional reactive power sources. RPP is a nonlinear optimisation problem for a large-scale system with many uncertainties. During the last decade there has been a growing concern about RPP problems [1–8]. Conventional calculus-based optimisation algorithms have been used in RPP for many years [1–4]. Most conventional optimisation methods are based on successive linearisations and use the first and second derivatives of the objective function and its constraint equations as the search directions. Because the formulae of the RPP problem are hyperquadric functions, such linear and quadratic treatments induce many local minima. Furthermore, the conventional optimisation methods cannot deal with the nondifferentiable factor in the reactive power source installation function in RPP. The conventional optimisation methods can only lead to a local minimum and sometimes result in divergence in solving RPP problems. Recently, new methods based on artificial intelligence have been used in RPP or optimal reactive power control. Abdul-Rahman *et al.* [5] have presented an artificial neural network (ANN) enhanced by fuzzy sets to determine the memberships of VAR control variables to solve the load uncertainties and an expert system to refine the solution with minimum adjustments of control variables. Jwo *et al.* [6] have put

IEE Proc.-Gener. Transm. Distrib., Vol. 143, No. 4, July 1996

365

forward a hybrid expert-system/simulated-annealing in RPP to solve local minimum problems. Genetic algorithms (GAs) have been given [7, 8] for the global optimal solutions of reactive power optimisation problems. However, the ANN trained by linear programming [5] may still have the same problem of being stuck in a local minimum. With only one solution to compare with another to obtain the new solution in its iteration, simulated annealing (SA) would be more likely to either prematurely converge or keep searching without a direction. Mutalik *et al.* [9] have concluded, from their tested cases, that GA consistently performs better than SA. The expert systems based on analysis of sensitivities [5, 6] are in the gradient directions to local minima.

This paper proposes an application of evolutionary programming (EP) to RPP. EP and GAs belong to evolutionary algorithms (EAs), which are search algorithms based on the simulated evolutionary process of natural selection and natural genetics [10–14]. EAs are randomised search algorithms, which, however, do not necessarily mean directionless random walk. EAs are different from other optimisation methods in the following respects:

(1) EAs search from a population of points, not a single point. The population can move over hills and across valleys. EAs can therefore discover a globally or near globally optimal point. Because each individual in the population is computed independently, EAs have an inherent parallel computation ability.

(2) EAs use payoff (fitness or objective functions) information directly for the search direction, neither derivatives nor other auxiliary knowledge. EAs therefore can deal with nonsmooth, noncontinuous and nondifferentiable functions that are the real-life optimisation problems. This property also relieves EAs of the approximate assumptions for many practical optimisation problems, which are quite often required in traditional optimisation methods.

(3) EAs use probabilistic transition rules to select generations, not deterministic rules, so they are a kind of stochastic optimisation algorithm which can search a complicated and uncertain area to find the global optimum. EAs are more flexible and robust than conventional methods.

These features make EAs robust and parallel algorithms which can adaptively search the globally optimal point. EAs offer new tools for the optimisation of complex system problems. EP is different from GAs in these two aspects: EP uses the control parameters, not their codings; the generation selection procedure of EP is mutation and competition, not reproduction, mutation and crossover. GAs emphasise models of genetic operators, while EP emphasises mutational transformations that maintain behavioural linkage. It has been indicated [11, 15] that EP outperforms GAs. The encoding and decoding of each solution and the operations of crossover and mutation on binary-coded variables of GAs use a lot of computing time. The new generation of GAs after mutation and crossover may lose advantages obtained in the last generation, while by competition in the combined old generation and mutated old generation, EP successfully takes such advantages.

The theory of EP has been well established, but some practical problems need to be solved to make EP prac-

ticable. In this paper, some techniques have been developed to solve RPP and other practical problems: first, adaptive mutation scales are introduced to guarantee the global optimum and produce a smooth convergence; secondly, relative fitness values are used to deal with practical problems, where the value of one individual does not differ much from that of the others; and finally, the population size and competition size are carefully studied. These techniques are essential in practical search problems.

The IEEE 30-bus system is used in this paper as the simulation system. The results of EP are compared with those from a conventional method, the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method [16].

2 Problem formulation

The objective function in RPP problem comprises two terms. The first term represents the total cost of energy loss as follows:

$$W_C = h \sum_{l \in N_l} d_l P_{loss}^l$$

$$= h \sum_{l \in N_l} d_l \left[\sum_{\substack{k \in N_E \\ k=(i,j)}} g_k (V_i^2 + V_j^2 - 2V_i V_j \cos \theta_{ij}) \right]^l \tag{1}$$

where $P_{loss}{}^l$ is the network real power loss during the period of load level l.

The second term represents the cost of reactive power source installation which has two components, the fixed installation cost and the purchase cost

$$I_C = \sum_{i \in N_c} (e_i + C_{ci} |Q_{ci}|) \tag{2}$$

where Q_{ci} can be either positive or negative, capacitance or reactance installation, so the absolute variable is used to compute the cost.

The above two objective functions are put into one comprehensive equation which can easily be adjusted by changing the parameters in W_C and I_C according to the practical problem under consideration. The objective function can therefore be expressed as:

$$\min f_C = W_C + I_C$$

$$\text{s.t. } 0 = P_i - V_i \sum_{j \in N_i} V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \quad i \in N_{B-1}$$

$$0 = Q_i - V_i \sum_{j \in N_i} V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \quad i \in N_{PQ}$$

$$Q_{ci}^{min} \le Q_{ci} \le Q_{ci}^{max} \qquad i \in N_c$$

$$Q_{gi}^{min} \le Q_{gi} \le Q_{gi}^{max} \qquad i \in N_g$$

$$T_k^{min} \le T_k \le T_k^{max} \qquad k \in N_T$$

$$V_i^{min} \le V_i \le V_i^{max} \qquad i \in N_B \tag{3}$$

where power flow equations are used as equality constraints. Reactive power source installation restrictions, reactive power generation restrictions, transformer tap-setting restrictions and bus voltage restrictions are used as inequality constraints. The transformer tap-setting T, generator bus voltages V_g and reactive power source installations Q_c are control variables so they are self-restricted. The load bus voltages V_{load} and reactive power generations Q_g are state variables, which are restricted by adding them as the quadratic penalty

366

IEE Proc.-Gener. Transm. Distrib., Vol. 143, No. 4, July 1996

terms to the objective function to form a penalty function. Eqn. 3 is therefore changed to the following generalised objective function:

$$\min F_C = f_C + \sum_{i \in N_{VPQlim}} \lambda_{Vi}(V_i - V_i^{lim})^2 + \sum_{i \in N_{Qglim}} \lambda_{Qgi}(Q_{gi} - Q_{gi}^{lim})^2$$

$$\text{s.t. } 0 = P_i - V_i \sum_{j \in N_i} V_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}) \quad i \in N_{B-1}$$

$$0 = Q_i - V_i \sum_{j \in N_i} V_j (G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}) \quad i \in N_{PQ}$$

$$(4)$$

where λ_{Vi} and λ_{Qgi} are the penalty factors which can be increased in the optimisation procedure; V_i^{lim} and Q_{gi}^{lim} are defined as:

$$V_i^{lim} = \begin{cases} V_i^{min} & \text{if } V_i < V_i^{min} \\ V_i^{max} & \text{if } V_i > V_i^{max} \end{cases}$$

$$Q_{gi}^{lim} = \begin{cases} Q_{gi}^{min} & \text{if } Q_{gi} < Q_{gi}^{min} \\ Q_{gi}^{max} & \text{if } Q_{gi} > Q_{gi}^{max} \end{cases} \quad (5)$$

It can be seen that the generalised objective function F_C is a nonlinear and noncontinuous function. The factor e_i in I_C is nondifferentiable. Gradient-based conventional methods are not good enough to solve this problem.

3 Evolutionary programming

EP is different from conventional optimisation methods. It does not need to differentiate cost function and constraints. It uses probability transition rules to select generations. Each individual competes with other individuals in a combined population of the old generation and the mutated old generation. The competition results are valued using a probabilistic rule. The winners of the same number as the individuals in the old generation constitute the next generation. The EP procedure for RPP is briefly listed as follows:

Step 1. Initialisation: The initial control variable population is selected by randomly selecting $p_i = [V_g^i, Q_c^i, T^i]$, $i = 1, 2, ..., m$, where m is the population size, from the sets of uniform distribution ranging over $[V^{min}, V^{max}]$, $[Q_c^{min}, Q_c^{max}]$ and $[T^{min}, T^{max}]$. The fitness value f_i of each p_i is obtained by running the P-Q decoupled power flow.

Step 2. Statistics: The values of maximum fitness, minimum fitness, sum of fitnesses and average fitness of this generation are calculated as follows:

$$f_{max} = \{f_i | f_i \geq f_j \forall f_j, j = 1, \ldots, m\}$$

$$f_{min} = \{f_i | f_i \leq f_j \forall f_j, j = 1, \ldots, m\}$$

$$f_\Sigma = \sum_{i=1}^{m} f_i \quad (6)$$

$$f_{avg} = \frac{f_\Sigma}{m}$$

Step 3. Outer loop start
Step 4. Inner loop start
Step 5. Mutation: Each p_i is mutated and assigned to p_{i+m} in accordance with the following equation:

$$P_{i+m,j} = p_{i,j} + N(0, \beta(x_{jmax} - x_{jmin})\frac{f_i}{f_{max}}), j = 1, 2, \ldots, n \quad (7)$$

where, $p_{i,j}$ denotes the jth element of the ith individual; $N(\mu, \sigma^2)$ represents a Gaussian random variable with mean μ and variance σ^2; f_{max} is the maximum fitness value of the old generation which is obtained in step 2;

x_{jmax} and x_{jmin} are the maximum and minimum limits, respectively, of the jth element; β is the mutation scale which is given as $0 < \beta \leq 1$. If any $p_{i+m,j}$, $j = 1, 2, ..., n$, where n is the number of control variables, exceeds its limit, $p_{i+m,j}$ will be given the limit value. The corresponding fitness value f_{i+m} is obtained by running power flow with p_{i+m}. A combined population is formed from the old generation and the mutated old generation.

Step 6. Competition: Each individual p_i in the combined population has to compete with some other individuals to have a chance to be transcribed to the next generation. A weight value w_i is assigned to the individual according to the competition as follows:

$$w_i = \sum_{t=1}^{q} \begin{cases} 1, & \text{if } u_1 < \frac{f_t}{f_t + f_i} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where q is the number of competitors; f_t is the fitness value of the tth randomly selected competitor in the combined population; f_i is the fitness value of p_i; u_1 is randomly selected from a uniform distribution set, $U(0, 1)$. When all individuals p_i, $i = 1, 2, ..., 2m$, obtain their competition weights, they will be ranked in descending order of their corresponding value w_i. The first m individuals are transcribed along with their corresponding fitness values f_i to be the basis of the next generation. The values of maximum, minimum and average fitness and sum of fitnesses of this generation are then calculated in step 2.

Step 7. Inner loop convergence criterion: The convergence is achieved when either the maximum fitness value converges to the minimum fitness value or the generations reach the maximum generation number. If the condition is met, the process will go to the next step, otherwise, the process will go back to step 4.

Step 8. Outer loop convergence criterion: The convergence is achieved when either all state variables, voltage magnitudes of load buses and reactive power generations, are within their limits or the outer loops reach the maximum number. If the condition is met, the program will stop. If one or more state variables violate their limits, the penalty factors of these variables will increase, and the process will return to step 3.

To make EP practicable, the following four techniques have been developed:

(1) *Adaptive mutation scale*: In general, EP mutation probability is fixed throughout the whole search processing. However, in practical applications, a small fixed mutation probability can only result in a premature convergence, while the search with a large fixed mutation probability will not converge. An adaptive mutation scale is given to change the mutation probability to solve the problem as follows:

$$\beta(k + 1) = \begin{cases} \beta(k) - \beta_{step}, & \text{if } f_{min}(k) \text{ unchanged} \\ \beta(k), & \text{if } f_{min}(k) \text{ decreased} \\ \beta_{final}, & \text{if } \beta(k) - \beta_{step} < \beta_{final} \end{cases}$$

$$\beta(0) = \beta_{init}$$

$$(9)$$

where k is the generation number; β_{init}, β_{final} and β_{step} are fixed numbers. β_{init} would be around 1 and β_{final} would be 0.005. β_{step} would be 0.001 – 0.01, depending on the maximum generation number. The mutation scale will decrease as the process continues. The decreasing speed of the mutation scale depends on the fitness value, that is, the lower the fitness value, the

IEE Proc.-Gener. Transm. Distrib., Vol. 143, No. 4, July 1996

367

faster the mutation scale decreases. Such an adaptive mutation scale not only prevents premature convergence, but also produces a smooth convergence.

(2) *Relative fitness values*: In practical problems, the fitness value of one individual does not differ significantly from that of the others, especially in the RPP problem, the difference between the minimum point and the original operating point is small. In deterministic transition rules, there may be no problem arising from this situation. However, in probabilistic transition rules, such a small difference will sink into oblivion because of added uncertainties, e.g. u_1 in EP. To deal with the problem, the program trims the fitness value and the maximum fitness value that are used in the mutation and competition procedure. The method is explained as follows:

$$f_{proci} = f_i - \varepsilon f_{min} \quad i = 1, 2, \ldots, m$$
$$f_{proc\ max} = f_{max} - \varepsilon f_{min}$$
(10)

where $0.95 \le \varepsilon < 1$, so f_{proci} and $f_{procmax}$ will be always larger than 0. Only the relative fitness values are used in the process of mutation and competition. The relative values are quite distinct among the fitness values so the better individuals become more competitive. It is the only way for EP to be practicable in real-life systems.

(3) *Adaptive population size*: To ensure the global search area, the population size should increase with the increasing control variable number. We have found that when the control variable number is very small, say less than 10, keeping enough individuals in a population, say 20 individuals, would not use too much computation time, because the generations will decrease with decreasing variable number. However, when the control variables increase, the population size does not need to increase in direct ratio to the increase in variable number. The increase of population size will stop, for the sake of both computation speed and memory, when the size reaches a maximum number, which is given in this paper as 100. The relationship between the population size and control variable number could be represented as

$$Popsize = 20 \times Int(5 - 4e^{\frac{-k}{40}})$$
(11)

where k is the control variable number and Int means that only integer values will be taken. The population size will be 20 for the control variables of 1–11, 40 for 12–27, 60 for 28–55 and 80 for more than 55.

(4) *Competition size*: Since the competition size has very little influence on the computation speed and has no influence on storage, it could be kept high to obtain more information about the generation. However, a very high value of the competition size would result in premature convergence. The competition size is given as 15 for a population size of 20, 20 for 40 and 25 for the larger population size, respectively. It is relatively large for a small population size to speed up the process and small for a large population size to avoid premature convergence.

These techniques are essential in practical search problems. Without these techniques, EP cannot be applied to real-life problems. The population size in eqn. 11 and competition size have been obtained by trial and error. Since the parameters for the four developed techniques depend only on the generation number and the number of control variables, they could also be applied to other systems.

4 Numerical results

In this section, the IEEE 30-bus system [17] is used to show the effectiveness of the algorithm. The network consists of six generator-buses, 21 load-buses and 43 branches, of which four branches, (6, 9), (6, 10), (4, 12) and (28, 27), are under-load-tap-setting transformer branches. The system is shown in Fig. 1. The branch parameters and loads are given in [17]. The possible reactive power source installation buses are buses 6, 17, 18 and 27. The base power and parameters of costs are given in Table 1. Three cases have been studied. Case 1 is of light loads whose loads and initial real power generations, except the generation at the slack bus, are the same as those in [17]. Case 2 is of heavy loads whose loads and initial real power generations are twice as those of case 1. Case 3 has two level load periods, one light load period having the same loads as those in case 1 and one heavy load period having the same loads as those in case 2. The one-year energy loss cost is used to assess the possibility of installing the reactive power sources. The variable limits are given in Table 2. The total loads are:

Case 1: $P_{load} = 2.834\text{p.u.}$ $Q_{load} = 1.262\text{p.u.}$
Case 2: $P_{load} = 5.668\text{p.u.}$ $Q_{load} = 2.524\text{p.u.}$

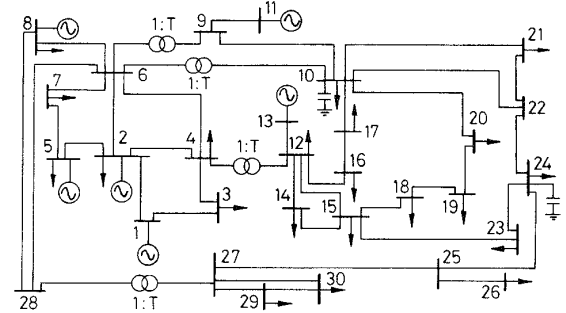Fig. 1 *IEEE 30-bus system*

Table 1: Base power and parameters of costs

S_B (MVA)	h (£/p.u.Wh)	e_i (£)	C_{ci} (£/p.u.VAR)	d_i (h) Case 1	Case 2	Case 3
100	6000	1000	3000,000	8760	8760	4380 for each level

Table 2: Variable limits (p.u.)

Bus	1	2	5	8	11	13
Q_g^{max}	0.596	0.480	0.6	0.53	0.15	0.155
Q_g^{min}	-0.298	-0.24	-0.3	-0.265	-0.075	-0.078
V^{max}	V^{min}	T^{max}	T^{min}	Q_c^{max}	Q_c^{min}	
1.05	0.95	1.1	0.9	0.36	-0.12	

4.1 Initial power flow results

The initial generator bus voltages and transformer taps are set to 1.0p.u. The total generations and power losses are given in Table 3. The limit-violating quantities are given in Table 4. In case 2, because of the heavy loads, all reactive power generations and almost all load bus voltages violate their limits.

Table 3: Initial generations and power losses (p.u.)

	P_g	Q_g	P_{loss}	Q_{loss}
Case 1	2.89388	0.98020	0.05988	-0.28180
Case 2	5.94588	3.26368	0.27788	0.73968

368

IEE Proc.-Gener. Transm. Distrib., Vol. 143, No. 4, July 1996

Table 4: Limit-violating variables (p.u.)

Case 1						
Bus	26	29	30	Bus	8	
V_i	0.932	0.940	0.928	Q_{gi}	0.569	
Case 2						
Bus	9	10	12	14	15	16	17
V_i	0.945	0.909	0.940	0.901	0.893	0.910	0.897
Bus	18	19	20	21	22	23	24
V_i	0.870	0.863	0.872	0.879	0.880	0.866	0.849
Bus	25	26	27	29	30		
V_i	0.855	0.811	0.880	0.829	0.799		
Bus	1	2	5	8	11	13	
Q_{gi}	-0.402	0.496	0.952	1.497	0.281	0.439	

4.2 EP optimal results

The optimal results are given in Table 5. The transformer taps are discrete variables with the change step of 0.025 p.u. All the state variables are regulated back into their limits.

Two sets of control variables are obtained for two different load level periods in cases 3. The P–Q decoupled power flow has to run twice for two sets of control variables, so the computation time almost doubles for the same generation as in cases 1 or 2. In case 3, because the installation cost is only counted in the heavy load period, which has more installation than the light load period for every bus, the reactive power sources used in the light load period do not induce any cost. Therefore, in the light load period, there are some reactive power source installations, which are in fact the reactive power generations from the existing reactive power sources installed for the heavy load period. The real power loss is lower than that in case 1 because of these reactive power sources. The real power savings and annual energy cost savings are given as follows:

Case 1:

$$P_{save}\% = \frac{P_{loss}^{init} - P_{loss}^{opt}}{P_{loss}^{init}} \times 100$$

$$= \frac{0.05988 - 0.05159}{0.05988} \times 100 = 13.84\%$$

$$W_c^{save} = hd_l(P_{loss}^{init} - P_{loss}^{opt})$$

$$= 6000 \times 8760 \times (0.05988 - 0.05159)$$

$$= £435\,722.4$$

Case 2:

$$P_{save}\% = \frac{0.27788 - 0.23311}{0.27788} \times 100 = 16.11\%$$

$$W_c^{save} = 6000 \times 8760 \times (0.27788 - 0.23311)$$

$$= £2\,353\,111.2$$

Case 3:

Light load period :

$$P_{save}\% = \frac{0.05988 - 0.05085}{0.05988} \times 100 = 15.08\%$$

Heavy load period :

$$P_{save}\% = \frac{0.27788 - 0.23396}{0.27788} \times 100 = 15.81\%$$

$$W_c^{save} = 6000 \times 4380$$

$$\times [(0.05988 - 0.05085) + (0.27788 - 0.23396)]$$

$$= £1\,391\,526$$

4.3 Comparison with BFGS method

A nonlinear programming algorithm, the BFGS method, is used as a comparison. After the optimisation of BFGS, the real power losses are 0.05470, 0.23483, 0.05729 and 0.24070 per unit in cases 1 and 2 and the light load period and heavy load period of case 3, respectively. The total reactive power installations are 0.265, 1.071, 0.567 and 0.959 per unit in cases 1 and 2 and the light load period and heavy load period of case 3, respectively. The power savings with the BFGS method are 8.65% in case 1, 15.50% in case 2 and 4.33% in light load period and 12.38% in heavy load period of case 3. The annual energy cost savings are £272 260.8, £2 263 233.6 and £1 045 155.6 in cases 1–3, respectively. The computation times are 0.8, 1.5 and 3.7 min in cases 1–3, respectively. The annual energy cost savings from EP are 160%, 104% and 133% of those from BFGS in the three cases, respectively.

The total cost of energy losses and investments, which the objective of RPP in eqn. 4, are:

Case 1:

EP: $f_C = W_C + I_C = £2\,711\,570 + 0 = £2\,711\,570$
BFGS: $f_C = W_C + I_C = £2\,875\,032 + £799\,000 = £3\,674\,032$

Case 2:

EP: $f_C = W_C + I_C = £12\,252\,262 + £2\,272\,000 = £14\,524\,262$
BFGS: $f_C = W_C + I_C = £12\,342\,665 + £3\,217\,000 = £15\,559\,665$

Case 3:

EP: $f_C = W_C + I_C = £7\,484\,807 + £2\,257\,000 = £9\,741\,807$
BFGS: $f_C = W_C + I_C = £7\,831\,177 + £2\,881\,000 = £10\,712\,177$

In case 3, each installation bus has two reactive power source installations for two load level periods. The larger installation is used to compute the installation cost I_C, while the smaller quantity can be obtained by regulation during the light load period. In both EP and BFGS, the installation at any bus in the heavy load period is larger than the installation in the light load period, so the installation cost of case 3 is the cost for the heavy load period.

The total costs from EP are 74%, 93% and 91% of those from BFGS in the three cases, respectively. From the comparison, it can be seen that in all three cases, EP gives better results. Therefore, in this simulation, EP always goes to the global or near global optimum, while BFGS goes to a local minimum.

5 Conclusions

The use of EP for the RPP of power systems has been reported. RPP is an optimisation problem of a nonlinear, nonsmooth and noncontinuous function. This type of less well behaved function is met with in most engineering problems so the devising, testing and refining of new techniques for finding optimal solutions has become more important with the advent of even more powerful computers. The proposed EP approach has been evaluated on the IEEE 30-bus power network. The simulations show that EP always leads to satisfactory results for multiobjective RPP, especially in noncontinuous and nonsmooth situations. The comparison shows that the proposed EP method is more powerful for global optimisation problems of the nonsmooth,

IEE Proc.-Gener. Transm. Distrib., Vol. 143, No. 4, July 1996

369

Table 5: Optimal control variables

Generator bus voltages, p.u.

Bus		1	2	5	8	11	13
Case 1		1.050	1.044	1.023	1.025	1.050	1.050
Case 2		1.050	1.022	0.973	0.959	1.050	1.050
Case 3	Light load	1.050	1.044	1.023	1.026	1.027	1.050
	Heavy load	1.050	1.022	0.973	0.959	1.045	1.048

Transformer tap-settings, p.u.

Branch		(6, 9)	(6, 10)	(4, 12)	(28, 27)
Case 1		0.95	1.1	1.025	1.05
Case 2		1.05	1.1	1.1	1.1
Case 3	Light load	0.975	1.1	1.05	1.025
	Heavy load	1.05	1.075	1.1	1.1

Reactive power source installations, p.u.

Bus		6	17	18	27
Case 1		0	0	0	0
Case 2		0.198	0.229	0.133	0.196
Case 3	Light load	0.150	0.077	0.068	0.077
	Heavy load	0.198	0.227	0.131	0.195

Power generations and power losses, p.u.

		P_g	Q_g	P_{loss}	Q_{loss}
Case 1		2.88559	0.92579	0.05159	-0.33626
Case 2		5.90110	2.20392	0.23311	0.43591
Case 3	Light load	2.88485	0.52662	0.05085	-0.36343
	Heavy load	5.90195	2.21541	0.23396	0.44257

Iteration and computation time (486/50MHz)

Outer loop		1	2	3	4	5	Time, min
Generations (iterations)	Case 1	85	108				4.2
	Case 2	91	112	134	97		8.5
	Case 3	97	129	106	141	93	20.21

noncontinuous functions. The only disadvantage of EP is that it takes much more computation time than the conventional method. However, with the inherent parallel computation ability and the advance of computer technology, it would not be so difficult to solve such a problem. The comprehensive simulation results show a great potential for applications of EP in economical and secure power system operations, planning and reliability assessment.

6 References

1 KISHORE, A., and HILL, E.F.: 'Static optimization of reactive power sources by use of sensitivity parameters', *IEEE Trans.*, 1971, **PAS–90**, pp. 1166–1173
2 SACHDEVA, S.S., and BILLINTON, R.: 'Optimum network VAR planning by nonlinear programming', *IEEE Trans.*, 1973, **PAS–92**, pp. 1217–1225
3 FERNANDES, R.A., LANGE, F., BURCHETT, R.C., HAPP, H.H., and WIRGAU, K.A.: 'Large scale reactive power planning', *IEEE Trans.*, 1983, **PAS–102**, pp. 1083–1088
4 HONG, Y.Y., SUN, D.I., LIN, S.Y., and LIN, C.J.: 'Multi-year multi-case optimal VAR planning', *IEEE Trans.*, 1990, **PWRS–5**, pp. 1294–1301
5 ABDUL-RAHMAN, K.H., SHAHIDEHPOUR, S.M., and DANESHDOOST, M.: 'AI approach to optimal VAR control with fuzzy reactive loads', *IEEE Trans.*, 1995, **PWRS–10**, pp. 88–97

6 JWO, W.-S., LIU, C.-W., LIU, C.-C., and HSIAO, Y.-T.: 'Hybrid expert system and simulated annealing approach to optimal reactive power planning', *IEE Proc. Gener. Transm. Distrib.*, 1995, **142**, pp. 381–385
7 SWARUP, K.S., YOSHIMI, M., and IZUI, Y.: 'Genetic algorithm approach to reactive power planning in power systems'. Proceedings of the 5th Annual Conference of Power and Energy Society IEE Japan, 1994, pp. 119–124
8 IBA, K.: 'Reactive power optimization by genetic algorithm', *IEEE Trans.*, 1994, **PWRS–9**, pp. 685–692
9 MUTALIK, P.P., KNIGHT, L.R., BLANTON, J.L., and WAINWRIGHT, R.L.: 'Solving combinatorial optimization problems using parallel simulated annealing and parallel genetic algorithms'. Proceedings of the 1992 ACM/SIGAPP symposium on *Applied computing*, USA, 1992, pp. 1031–1038
10 FOGEL, L.J.: 'Autonomous automata', *Ind. Res.*, 1962, **4**, pp. 14–19
11 FOGEL, D.B.: 'System identification through simulated evolution: a machine learning approach to modeling' (Ginn Press, MA, 1991)
12 FOGEL, D.B.: 'An introduction to simulated evolutionary optimization', *IEEE Trans. Neural Netw.*, 1994, **5**, pp. 3–14
13 ANGELINE, P.J.: 'Evolution revolution: an introduction to the special track on genetic and evolutionary programming', *IEEE Expert*, 1995, **10**, pp. 6–10
14 FOGEL, D.B.: 'Evolutionary computation: toward a new philosophy of machine intelligence' (IEEE Press, Piscataway, NJ, 1995)
15 FOGEL, D.B.: 'A comparison of evolutionary programming and genetic algorithms on selected constrained optimization problems', *Simulation*, June 1995, pp. 397–404
16 BAZARAA, M.S., SHERALI, H.D., and SHETTY, C.M., 'Nonlinear programming: theory and algorithms' (John Wiley, New York, 1993), pp. 318–328
17 ALSAC, O., and STOTT, B.: 'Optimal load flow with steady-state security', *IEEE Trans.*, 1974, **PAS–93**, pp. 745–751

370

IEE Proc.-Gener. Transm. Distrib., Vol. 143, No. 4, July 1996