Optimal Reactive Power Planning Using Evolutionary Algorithms: A Comparative Study for Evolutionary Programming, Evolutionary Strategy, Genetic Algorithm, and Linear Programming

Kwang. Y. Lee, Senior Member, IEEE, and Frank F. Yang
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802

Abstract: This paper presents a comparative study for three evolutionary algorithms (EAs) to the Optimal Reactive Power Planning (ORPP) problem: evolutionary programming, evolutionary strategy, and genetic algorithm. The ORPP problem is decomposed into P- and Q-optimization modules, and each module is optimized by the EAs in an iterative manner to obtain the global solution. The EA methods for the ORPP problem are evaluated against the IEEE 30-bus system as a common testbed, and the results are compared against each other and with those of linear programming.

Keywords: Optimal reactive power planning, evolutionary algorithms, genetic algorithm.

I. INTRODUCTION

In general, the problem of optimal reactive power planning (ORPP) can be defined as to determine the amount and location of shunt reactive power compensation devices needed for minimum cost while keeping an adequate voltage profile. The ORPP is one of the most challenging problems since both objective functions, the operation cost and the investment cost of new reactive power sources, should be minimized simultaneously. The ORPP is a large-scale nonlinear optimization problem with a large number of variables and uncertain parameters. Various mathematical optimization algorithms have been developed for the ORPP, which in most cases, use nonlinear [1], linear [2], or mixed integer programming [3], and decomposition methods [4-7]. However, these conventional techniques are known to converge to a local optimal solution rather than the global one for problems sush as ORPP which have many local minima.

PE-958-PWRS-1-04-1997 A paper recommended and approved by the IEEE Power System Analysis, Computing and Economics Committee of the IEEE Power Engineering Society for publication in the IEEE Transactions on Power Systems. Manuscript submitted January 2, 1997; made available for printing April 11, 1997.

Recently, evolutionary algorithms (EAs) [8-15] have been used for optimization; in particular both the genetic algorithm and evolutionary programming have been used in the ORPP problem. The EA is a powerful optimization technique analogous to the natural selection process in genetics. Theoretically, this technique converges to the global optimum solution with probability one. It is useful especially when other optimization methods fail in finding the optimal solution. Evolutionary algorithm is an inherently parallel process. Recent advances in distributed processing architectures could result in dramatically reduced execution times, and it is now possible to do a large amount of computation in order to obtain the global instead of a local optimal solution.

This paper investigates the applicability of the following three different EAs in the ORPP problem: evolutionary programming (EP), evolutionary strategy (ES), and genetic algorithm (GA). Rather than the usual approach of loss minimization [11], the fuel cost minimization approach [4-7,10] is adopted as a direct measure of operation cost since the loss minimization does not guarantee the optimal operation with minimum fuel. The ORPP problem is then decomposed into the real power (P) and the reactive power (Q) optimization problem. The P-optimization is to minimize the operation cost by adjusting real power generation; while the Q-optimization is to adjust reactive power generation, transformer tap-settings, and the amount of Var source investment. The EA methods are evaluated against the IEEE 30-bus system [4-7] as a common testbed for comparison with each other and with linear programming.

II. EVOLUTIONARY ALGORITHMS

The EAs, including Evolutionary Programming (EP), Evolutionary Strategy (ES), and Genetic Algorithm (GA), are artificial intelligence methods for optimization based on the mechanics of natural selection, such as mutation, recombination, reproduction, selection, etc. Mutation randomly perturbs a candidate solution; recombination randomly mixes their parts to form a novel solution; reproduction replicates the most successful solutions found in a population; whereas selection purges poor solutions from a population. Starting from an initial generation of candidate solutions, this process produces advanced generations with candidates that are successively better suited to their environment.

These methods share many similarities. The EP is introduced first, and followed by ES and GA.

A. *Evolutionary Programming* [11]

1) Initialization: The initial population of control variables is selected randomly from the set of uniformly distributed control variables ranging over their upper and lower limits. The fitness score f_i is obtained according to the objective function and the environment.

2) Statistics: The maximum fitness f_{max}, minimum fitness f_{min}, the sum of fitness Σf, and average fitness f_{avg} of this generation are calculated.

3) Mutation: Each selected parent, for example P_i, is mutated and added to its population following the rule:

$$P_{i+m,j} = P_{i,j} + N(0, \beta\ (\overline{x}_j - \underline{x}_j)\frac{f_i}{f_{\max}}), \quad j = 1, 2, \ldots, n, \quad (1)$$

where n is the number of decision variables in an individual, $P_{i,j}$ denotes the j^{th} element of the i^{th} individual; $N(\mu, \sigma^2)$ represents a Gaussian random variable with mean μ and variance σ^2; f_{max} is the maximum fitness of the old generation which is obtained in *statistics*; $\overline{x}_j$ and $\underline{x}_j$ are, respectively, maximum and minimum limits of the j^{th} element; and β is the mutation scale, $0 < \beta \le 1$, that could be adaptively decreased during generations.

If any mutated value exceeds its limit, it will be given the limit value. The mutation process (1) allows an individual with larger fitness to produce more offspring for the next generation [11].

4) Competition: Several individuals (k) which have the best fitness are kept as the parents for the next generation. Other individuals in the combined population of size ($2m$ - k) have to compete with each other to get their chances for the next generation. A weight value W_i of the i^{th} individual is calculated by the following competition:

$$W_i = \sum_{t=1}^{N} W_{i,t}\ , \quad (2)$$

where N is the competition number generated randomly; $W_{i,t}$ is either 0 for loss or 1 for win as the i^{th} individual competes with a randomly selected (r^{th}) individual in the combined population. The value of $W_{i,t}$ is given in the following equation:

$$W_{i,t} = \begin{cases} 1 & \text{if } U_l < \dfrac{f_r}{f_r + f_i} \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where f_r is the fitness of the randomly selected r^{th} individual, and the f_i is the fitness of the i^{th} individual; and U_l is randomly selected from a uniformly distributed set, $U(0,1)$. When all $2m$ individuals, get their competition weights, they will be ranked in a descending order according to their corresponding value W_i. The first m individuals are selected along with their corresponding fitness f_i to be the bases for the next generation. The maximum, minimum and the average fitness and the sum of fitness of current generation are then calculated in the *statistics*.

5) Convergence test: If the convergence condition is not met, the *mutation* and the *competition* processes will run again. The maximum generation number can be used for convergence condition. Other criteria, such as the ratio of the average and the maximum fitness of the population is computed and generations are repeated until

$$\{f_{avg} / f_{\max}\} \ge \delta \quad (4)$$

where δ should be very close to 1, which represents the degree of satisfaction. If the convergence has reached a given accuracy, an optimal solution has been found for an optimization problem.

B. *Evolutionary Strategy* [9,12]

The evolutionary strategy is very similar to the evolutionary programming, and the difference is as follows:

In the *mutation* process, each selected parent, for example P_i, is mutated and added to its population following the rule,

$$P_{i+m,j} = P_{i,j} + N(0, \beta\ \nabla_{dev}), \quad j = 1, 2, \cdots, n, \quad (5)$$

where n is the number of decision variables, ∇_{dev} is fixed, and its value depends on the size of decision variables.

In the *competition* process, the fitness of individuals of population size $2m$ are sorted in a descending order. The first m individuals are kept as the parents for the next *mutation* process.

C. *Genetic Algorithm* [8,9,10-15]

Genetic algorithm (GA) emphasizes models of DNA selection as observed in nature, such as crossover and mutation, and are applied to abstracted chromosomes. This will be easily realized by a string representation, which costs additional encoding and decoding time. This is contrast in to ES and EP, which emphasize mutational transformations that maintain behavioral linkage between each parent and its offspring. The GA used in this paper is very similar to the algorithm that can be found in the standard literature on the topic [8,9,10-15], also known as the *simple genetic algorithm*. We use the three-operator GA with only minor deviations from the original.

1) Initial population generation: In this scheme, an initial population of binary strings is created randomly. Each of these strings represents one feasible solution to the search problem, i.e., a point in the search space or a domain satisfying constraints.

2) Fitness evaluation: Next the solution strings are converted into their decimal equivalents and each candidate solution is tested in its environment. The fitness of each candidate is evaluated through some appropriate measure, such as the inverse of the cost function:

$$f = 1/(\alpha + C), \tag{6}$$

where C is the cost function to be minimized and α is the fitness function parameter.

The algorithm is driven towards maximizing this fitness measure. After the fitness of the entire population has been determined, it must be determined whether or not the termination criterion has been satisfied. This criterion can be any number of things. One possibility is to stop the algorithm at some finite number of generations and designate the result as the best fit from the population. Another possibility is to test if the average fitness of the population exceeds some fraction of the best fit in the population. If the criterion is not satisfied then we continue with the three genetic operations of *reproduction, crossover,* and *mutation.*

3) Selection and reproduction: Fitness-proportionate *reproduction* is effected through the simulated spin of a weighted roulette wheel. The roulette wheel is biased with the fitness of each of the solution candidates. The wheel is spun N times where N is the number of strings in the population. This operation yields a new population of strings that reflect the fitness of the previous generation's fit candidates.

4) Crossover: The next operation, crossover, is performed on two strings at a time that are selected from the population at random. Crossover involves choosing a random position in the two strings and swapping the bits that occur after this position. In one generation the crossover operation is performed on a specified percentage of the population. Crossover can occur at a single position (single crossover), or at number of different positions (multiple crossover). Crossover can also be performed in two different means: Tail-tail and head-tail crossovers [10,13]. The tail-tail crossover is the usual crossover, where the tail ends of the two strings are swapped. In the head-tail crossover, on the other hand, the tail end of one string becomes the head of another string and vice-versa. The tail-tail crossover tends to change the less significant bits; while the head-tail crossover gives more chance of changes by changing the more significant bits. The two crossover methods can be changed during iterations: the head-tail crossover can be used in the earlier generations and then switched to tail-tail crossover in the later generations for fine tuning.

5) Mutation: The final genetic operator in the algorithm is mutation. Mutation is performed sparingly, typically after every 100-1000 bit transfers from crossover, and it involves selecting a string at random as well as a bit position at random and changing it from a 1 to a 0 or vice-versa. It is used to escape from a local minimum. After mutation, the new generation is complete and the procedure begins again with the *fitness evaluation* of the population.

III. PROBLEM FORMULATION

The optimal reactive power planning (ORPP) problem is to determine the optimal investment of Var sources over a planning horizon [5]. The cost function to be minimized is the sum of the operation cost and the investment cost. The long-term ORPP is often decomposed into a three-level hierarchical optimization problem using the maximum principle, the Bender's decomposition, and the P-Q decomposition methods [7]. To highlight the use of EAs however, this paper considers a short-term ORPP, where the investment is to be performed only once [10,11].

A. Objective Functions

The operation cost is often assumed to have single quadratic cost functions. In reality, the cost function has discontinuities corresponding to change of fuels [12]. Therefore, it is more appropriate to represent the cost function with piecewise quadratic functions. When using piecewise quadratic cost functions, the operation cost is defined as follows:

$$C_F = \sum_{i \in N_g} C_i\ (P_i) \tag{7}$$

$$C_i = \begin{cases} a_{i1} + b_{i1}P_i + c_{i1}P_i^2 & \text{if } \underline{P}_i \le P_i < P_{i1} \\ a_{i2} + b_{i2}P_i + c_{i2}P_i^2 & \text{if } P_{i1} \le P_i < P_{i2} \\ \cdots & \\ \cdots & \\ \cdots & \\ a_{im} + b_{im}P_i + c_{im}P_i^2 & \text{if } P_{im-1} \le P_i < \overline{P}_i, \end{cases}$$

where

N_g: the set of generators,

$C_i\,(P_i)$: cost of the i^{th} generator,

a_{ij}, b_{ij}, c_{ij}: cost coefficients of the i^{th} generator at the j^{th} power level,

P_i: the generated power of the i^{th} generator [MW],

$\underline{P}_i, \overline{P}_i$: minimum and maximum real power generation of the i^{th} generator.

Since EAs use the objective function directly, rather than its derivatives, more realistic cost functions can be used for the ORPP problem. Most papers consider the transmission loss in the objective function. However, the minimization of loss does not guarantee the minimization of the operation cost unless all units have the same efficiency. Therefore the fuel cost has been used for both real and reactive power dispatch [4-7].

The investment cost is simply the installation cost of the Var sources:

$$C_I = \sum_{i \in N_C} \left\{ C_{fi} + C_{ci} \left| \overline{Q}_{ci} \right| \right\}, \tag{8}$$

where N_C is the set of compensators, C_{fi} and C_{ci} are, respectively, the fixed and the unit costs for investment, and $\overline{Q}_{ci}$ is the amount of Var source investment in discrete steps. This cost function assumes the capacitive compensators; however, the reactive compensator can also be included by replacing $\overline{Q}_{ci}$ with $\underline{Q}_{ci}$.

B. P-Q Decomposition

The ORPP problem is decomposed into two subproblems, the Q-optimization module, and the P-optimization module.

1) The P-module: In this module, the objective is to minimize the sum of the operation cost:

$$C_P = \sum_{l \in N_l} d_l C_F^l, \tag{9}$$

where C_F^l is the operation cost (6) for load level l, d_l is the duration of load level l, and N_l is the set of load levels. The minimization is with respect to the real power generations P_g for each load level subject to the real and reactive power balance of the power system which can be solved by calling the *load flow* program, and the following inequality constraints:

$$\begin{aligned} &\underline{P}_{gi} \le P_{gi} \le \overline{P}_{gi} \\ &\underline{Q}_{gi} \le Q_{gi} \le \overline{Q}_{gi} \\ &\underline{T}_k \le T_k \le \overline{T}_k \\ &\underline{V}_i \le V_i \le \overline{V}_i \\ &\underline{Q}_{ci} \le Q_{ci} \le \overline{Q}_{ci} \end{aligned} \tag{10}$$

where P_g and Q_g are generator real and reactive powers, respectively, T are transformer tap-settings, V are bus voltages, and Q_c are the reactive power output of compensators.

The real power generations P_g are the optimization or *decision variables* for the P-module, which are self-constrained. Since the generator bus voltages, transformer tap-settings and capacitor investments are the optimization variables in the Q-module, they are fixed in the P-module, and thus the constraints are automatically satisfied. The load bus voltages and the generator reactive powers are state variables, which can be constrained by augmenting them as the quadratic penalty terms to the objective function. The P-optimization module is therefore changed to minimize the following generalized objective function:

$$\begin{aligned} C_P = \sum_{l \in N_l} d_l \{ C_F^l + \sum_{i \in N_l} \lambda_{vi} (V_i - Sat(V_i))^2 \\ + \sum_{i \in N_g} \lambda_{gi} (Q_{gi} - Sat(Q_{gi}))^2 \}, \end{aligned} \tag{11}$$

where N_l is the set of load buses and λ_{vi} and λ_{gi} are the penalty weights and $Sat(x)$ is the saturation function defined by

$$Sat(x) = \begin{cases} \underline{x} & \text{if } x < \underline{x} \\ x & \text{if } \underline{x} \le x \le \overline{x} \\ \overline{x} & \text{if } x > \overline{x} \end{cases} \tag{12}$$

2) The Q-module: In this module, the objective function is to minimize the sum of the operation cost and the investment cost,

$$C_Q = C_P + C_I, \tag{13}$$

with respect to transformer tap-settings T, generator bus voltages and the Var source investment $\overline{Q}_c$; subject to the real and reactive power balance equations which can be solved by calling the *load flow* program, and the inequality constraints (10).

The transformer tap-settings, generator bus voltages and the Var source investment are the *optimization or decision variables* for the Q-module, which are self-constrained. Since the real power generations are obtained from the last P-module, they should satisfy the constraint. The voltages of load buses and the reactive powers of generators are state variables, which may be out of limits, and penalties are also added to the objective function. The formulation of the penalty parts are the same as the procedure in the P-module. Then the Q-module is to minimize the generalized cost function as follows:

$$\begin{aligned} C_Q = C_I + \sum_{l \in N_l} d_l \{ C_F^l + \sum_{i \in N_l} \lambda_{vi} (V_i - Sat(V_i))^2 \\ + \sum_{i \in N_g} \lambda_{gi} (Q_{gi} - Sat(Q_{gi}))^2 \}. \end{aligned} \tag{14}$$

The P-Q decomposition is now complete. The optimization variables for the P-optimization subproblem are generator real power outputs, and those for the Q-optimization subproblem are generator voltage magnitudes, Var source investments, and transformer tap-settings. The security constraints are the operating limits of these control variables, line flows, and the state variables.

The P- and Q- modules were solved sequentially to obtain local optimal values of the optimization variables for each module. The computation diagram of the ORPP problem is shown in Fig. 1.

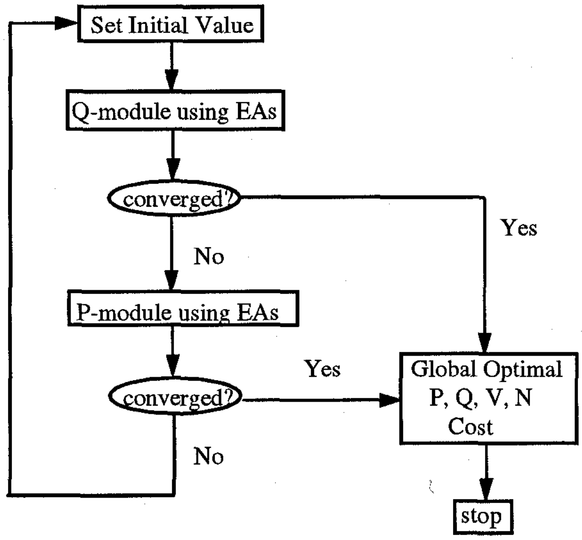

Fig. 1. Flow chart for ORPP by using EAs.

IV. SIMULATION RESULTS

A comparative study is performed for the three evolutionary algorithms (EAs) and the linear programming (LP) by solving the optimal reactive power planning (ORPP) problem for the IEEE 30-bus system as a testbed [4-7]. String representations and simulation parameters for EAs are discussed, and the modification of fuel costs with piecewise quadratic cost functions and the simulation results and the convergence speeds of EAs are given.

1) String representation is an important factor in solving the ORPP problem using the GA. In order to accommodate different decision variables, i.e., the investment or operation variables, the following representation method is used.

A string consists of substrings; the number of substrings is equal to the number of decision or control variables. In the P-optimization problem, it is the total number of generators and each substring represents the generator real power output. In the Q-optimization problem, on the other hand, the decision variables are generator voltage magnitudes, the maximum Var limits of compensators, and transformer tap settings. The simulation parameters for the three EAs are given in Table I and Table II.

TABLE I. SIMULATION PARAMETERS IN EP, ES, EP+ES AND GA

coefficients \ method	ES	EP	EP+ES	GA
number of parents	25	25	25	25
number of offspring	25	25	25	25
standard deviation	0.03	NA	EP(NA), ES(0.03)	NA

NA: Not Applicable

TABLE II. SIMULATION PARAMETERS IN GA

parameters \ value	GA
mutation rate	0.01
crossover rate	0.8
abandoning rate	0.9
maximum generations	100
population size	25
control variables	6
parameter resolution	8 bits per substring
chromosome length	48
fitness function parameter α	0.01

2) The fuel cost coefficients of the piecewise quadratic cost functions are given in Table III, which are generated from the original single quadratic cost functions [4]. Small units (at buses 5, 8, 11, and 13) are represented by two segments, while large units (at buses 1 and 3) are by three segments of fuel curves.

As shown in the P- and Q-optimization modules, the formulation is general to include a number of different load levels to optimally balance the savings in the operation cost against the cost of capital investment. However for simplicity, only the peak load condition [4] is considered for the comparative study. This scenario, obviously, will give an optimistic savings in the operation cost, and therefore could suggest over-investment.

TABLE III
COST COEFFICIENTS OF PIECEWISE QUADRATIC FUNCTION

U	GENERATION							F	cost coefficients		
	Min	F1	P1	F2	P2	F3	Max		a	b	c
1	50		100		190		200	1	0.000	1.900	0.00355
		1		2		3		2	0.000	2.000	0.00375
								3	0.000	2.200	0.00415
2	20		35		50		80	1	0.000	1.700	0.01700
		1		2		3		2	0.000	1.750	0.01750
								3	0.000	2.050	0.02350
3	15		30		50			1	0.000	1.000	0.06250
		1		2				2	0.000	1.200	0.08250
4	10		25		35			1	0.000	3.250	0.00834
		1		2				2	0.000	3.650	0.01234
5	10		20		30			1	0.000	3.000	0.02500
		1		2				2	0.000	3.300	0.03500
6	12		25		40			1	0.000	3.000	0.02500
		1		2				2	0.000	3.300	0.03500

3) The initial load flow, with generator voltages set to 1.0 p.u., showed that the voltage magnitudes at the load buses 14-30 were all below the lower operating limit of 0.95 p.u., and the operation cost was as high as 903.31 \$/hr. First for optimal power flow, the P-optimization module and the Q-optimization module <u>without the investment</u> were run by using both LP [6] and ES. Real power distribution, voltage magnitudes and transformer tap-settings were optimized to minimize the <u>operation cost</u>. The results of LP and ES showed that, all the control variables and state variables were within their hard limits, but for some load buses, such as buses 18-26, 29, and 30, the voltages were very close to their lower limits. Thus, more reactive power from other Var sources were needed to improve the voltage profile. As we compared the optimal power flow results, ES gave slightly higher voltage profile at slightly lower operation cost and power loss.

4) For the ORPP problem, the candidate buses for reactive power compensation are 15, 17, 20, 23, 24 and 29. All are shunt capacitors with the investment cost of 0.02 \$/Mvar per unit hour. The fixed cost for installation is neglected for simplicity. However, the site-dependent fixed cost can be incorporated in additional substrings in EAs; the number of additional substrings being equal to the number of candidate buses. The investments are made in discrete steps: 2.5, 5, 7.5, ... , 30 Mvars, which become the upper limits $\overline{Q}_{ci}$ in the operational constraints (10). In both LP and EAs, when the constraints are violated in any iteration, additional steps of the investments are made in the candidate buses in the Q-optimization module and the iteration is repeated.

TABLE IV. OPTIMAL OPERATION WITH INVESTMENT

(A) REAL POWER AND REACTIVE POWER DISTRIBUTION

Variable	LP	ES	EP	EP+ES
P_1	184.19	179.838	178.67	179.162
P_2	47.77	48.110	49.335	48.345
P_5	20.00	19.822	19.583	19.944
P_8	15.75	21.571	20.733	21.790
P_{11}	13.46	11.402	12.422	11.423
P_{13}	12.00	12.000	12.000	12.071
Q_1	27.71	8.861	13.027	12.409
Q_2	13.35	30.379	25.606	25.827
Q_5	30.12	30.136	30.060	30.753
Q_8	22.49	33.747	23.885	25.890
Q_{11}	29.58	13.826	18.934	17.064
Q_{12}	20.03	9.283	12.845	12.199
Loss	9.8+j44	9.3+j38.3	9.3+j38.7	9.3+j38.5

(B) BUS VOLTAGES

Variable(p.u.)	LP	ES	EP	EP+ES
V_1	1.0903	1.084	1.084	1.083
V_2	1.0616	1.064	1.063	1.062
V_3	1.0508	1.050	1.050	1.049
V_4	1.0425	1.043	1.042	1.042
V_5	1.0289	1.032	1.030	1.030
V_6	1.0340	1.037	1.036	1.035
V_7	1.0234	1.027	1.025	1.025
V_8	1.0294	1.038	1.033	1.033
V_9	1.0006	1.028	1.019	1.022
V_{10}	1.0032	1.014	1.010	1.013
V_{11}	1.0584	1.055	1.056	1.056
V_{12}	1.0111	1.019	1.018	1.019
V_{13}	1.0380	1.032	1.035	1.035
V_{14}	1.0000	1.010	1.008	1.010
V_{15}	0.9992	1.011	1.009	1.011
V_{16}	1.0013	1.011	1.009	1.011
V_{17}	0.9988	1.011	1.007	1.010
V_{18}	0.9901	1.003	1.001	1.004
V_{19}	0.9879	1.002	1.000	1.003
V_{20}	0.9924	1.007	1.005	1.008
V_{21}	0.9958	1.006	1.002	1.005
V_{22}	0.9970	1.007	1.002	1.006
V_{23}	0.9981	1.010	1.006	1.009
V_{24}	0.9972	1.002	0.997	1.001
V_{25}	1.0114	1.006	0.996	1.002
V_{26}	0.9936	0.998	0.978	0.984
V_{27}	1.0289	1.018	1.005	1.012
V_{28}	1.0264	1.033	1.032	1.031
V_{29}	1.0183	1.014	1.002	1.009
V_{30}	1.0030	0.995	0.983	0.990

(C) TAP-SETTINGS, CAPACITIVE VARS, AND COSTS

Variable	LP	ES	EP	EP+ES
N_{11}	1.0155	1.0290	1.0223	1.0165
N_{12}	0.9940	1.0337	1.0199	1.0261
N_{15}	1.0320	1.0298	1.0286	1.0277
N_{36}	0.9930	1.0274	1.0176	1.0152
Q_{c15}(Mvar)	3.05	6.5218	4.7253	4.7541
Q_{c17}(Mvar)	2.89	6.4387	4.4645	4.4418
Q_{c20}(Mvar)	2.79	6.8332	5.0405	5.0644
Q_{c21}(Mvar)	4.73	6.6059	4.7493	4.7540
Q_{c23}(Mvar)	2.96	6.4141	4.4747	4.4762
Q_{c24}(Mvar)	7.47	6.3411	4.2969	4.2865
Q_{c29}(Mvar)	3.09	6.6167	4.7567	4.7358
Gen. cost (\$/hr)	801.87	801.125	801.58	801.41
Inv. cost (\$/hr)	0.75	1.05	0.75	0.75
Total cost (\$/hr)	802.62	802.175	802.33	802.16

5) The results of the ORPP by using LP and EAs are given in Table IV. The results show that all the control variables and state variables, such as voltage magnitudes, transformer tap-settings, real power and reactive power generations are within their hard limits. The results of ES, EP, and EP+ES are very close and comparable with those of LP, and the average differences of voltage magnitudes and real power generations are within 5/1000. The power loss and the operation cost of LP is slightly higher than those of EAs. It is noted that ES showed the least operation cost at the expense of the highest investment cost; while the combined EP and ES (EP+ES) showed the least total cost (sum of the generation and the investment costs). Table IV(C) showed that two of the four tap-settings are reversed in direction in LP; while they all are in the same direction in all EAs. This resulted in the fairly even investment of Var sources in all seven candidate buses for EAs, while it is not the case for LP. It should be noted that the capacitive Vars, Q_{ci}, shown in the table are the *operational* values after the

investment, which are in general less than the maximum limits, $\overline{Q}_{ci}$, given by the investment in discrete steps.

The total cost of each P- and Q-modules in each iteration of ORPP are shown in the Table V. The iteration process is alternating as Q-P-Q-P-, etc. It is shown in the table that *only 4 or 5 iterations* of P- and Q-modules are required for convergence.

TABLE V. TOTAL COST DURING ITERATIONS OF ES FOR ORPP

Iter	1	2	3	4	5
Q	961.07	802.12	802.00	802.19	801.90
P	802.56	801.95	802.30	801.90	801.93

Each P- or Q-module converges in at least 30 generations. The CPU time of VAX for each generation is about 1.85 seconds, and in the ORPP problem, the CPU time for EAs to reach the *global* minimum is therefore at least $1.85 \times 30 \times 5$ seconds. However, for LP, it needs about 13 iterations to reach a *local* minimum, and the total CPU time is only around 7 seconds in VAX. Although the EAs take longer than LP, the formulation is straight forward and can handle arbitrary cost functions which may not be convex; and moreover, they can find the global minimum, while LP can only find a local minimum.

6) Within the EA family, the characteristics of ES, EP and GA were compared. For fair comparison, the same conditions are set for all decision variables of the Q-module, and only the P-module was run. Sample results are given in Table VI. The optimal real power distribution and the operation cost are very close for all methods. However, the number of generations to converge are different for different methods, and also different for different runs in the same method. Thus these programs were run in batch file for 5 times, and the average of the best fitness in each generation was calculated, which are shown in Fig. 2 and Fig. 3. In the EP+ES method, at first the EP process is called, and the best fitness of these generations decreases rapidly; then after 10 generations, ES process is called to continue the calculation. It is shown that, the ES and ES+EP methods need nearly the same number of generations (on average) to converge, while the EP and GA methods need almost twice as much time to converge.

TABLE VI. COMPARISON OF EA'S FOR P-OPTIMIZATION

Method	EP	ES	EP+ES	GA
P1[MW]	177.92	177.93	177.13	178.46
P2[MW]	48.587	48.887	49.664	47.891
P3[MW]	19.975	19.996	19.996	20.879
P4[MW]	22.739	22.467	22.775	23.086
P5[MW]	12.613	12.355	12.103	10.703
P6[MW]	11.745	11.776	11.753	12.328
Loss[MW]	9.995	10.013	9.985	9.944
Total cost	803.87	803.86	803.87	803.64
Generations to converge	60	37	36	63

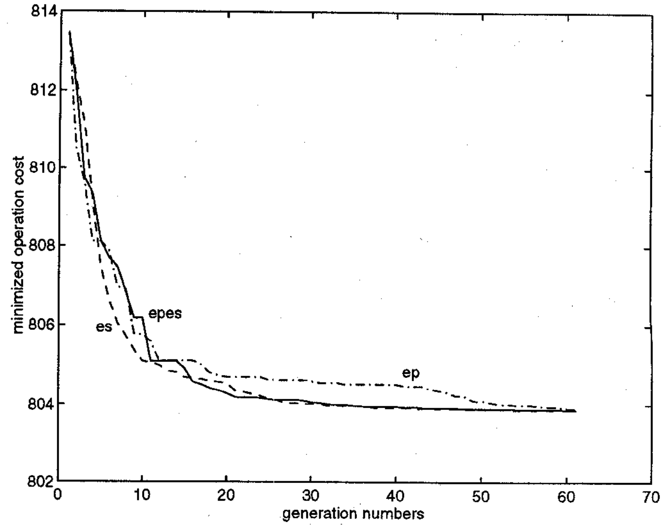

Fig. 2. Convergence comparison for EAs in the P-module

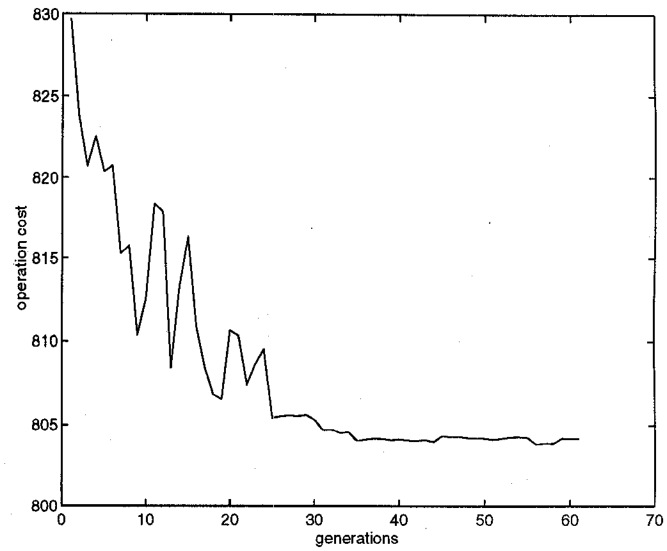

Fig. 3. Convergence characteristics of GA in the P-module

V. CONCLUSIONS

In this paper, the ORPP problem was solved by minimizing the total cost which includes the operation cost and the investment cost. The IEEE 30-bus system with piecewise quadratic cost functions is selected as a testbed. The results of ORPP by using different EA methods are almost identical. When the results are compared with the LP's, the EAs seem to be better; the total cost and the power loss are slightly lower, while all the hard limits are satisfied. Moreover, the P- and Q-modules of EAs can be easily formulated for general piecewise cost functions, not necessarily convex, while for LP, it is quite difficult. The characteristics of EAs are also compared. The ES needs less generations to converge in either P- or Q-modules, but it has a higher probability to fall into a local minimum. The EP needs more generations to converge, however, it is less likely to fall into a local minimum. When the EP is combined with ES, it only needs nearly the same number of generations to converge as ES, but with the improved robustness in finding the global minimum.

VI. ACKNOWLEDGMENT

This work is supported in part by the Allengeny Power Company under grant "Optimal Reactive Power Planning" and the National Science Foundation under grant "Research and Curriculum Development for Power Plant Intelligent Distributed Control" (EID-9212132).

VII REFERENCES

[1] R. Billington and S. S. Sachdev, "Optimum network VAR planning by nonlinear programming", *IEEE Trans. on Power Appar. and Syst.*, Vol. PAS-92, pp. 1217-1225, July/August 1973.

[2] G. T. Heydt and W. M. Grady, "Optimal Var siting using linear load flow formulation", *IEEE Trans. on Power Appar. and Syst.*, pp. 1214-1222. Vol. PAS-102, No. 5, May 1983.

[3] K. Aoki, M. Fan, and A. Nishikori, "Optimal var planning by approximation method for recursive mixed integer linear planning", *IEEE Trans. on Power Syst.*, Vol. PWRS-3, No. 4, pp. 1741-1747, November 1988.

[4] K. Y. Lee, Y. M. Park, and J. L. Oritz, "A united approach to optimal real and reactive power dispatch", *IEEE Trans. on Power Appar. and Syst.*, Vol. PAS-104, pp. 1147-1153, May 1985.

[5] K. Y. Lee, J. L. Ortiz, Y. M. Park, and L. G. Pond, "An optimization technique for reactive power planning of subtransmission network under normal operation", *IEEE Trans. on Power Syst.*, Vol. PWRS-1, pp. 153-159, May 1986.

[6] M. K. Mangoli, K. Y. Lee, and Y. M. Park, "Operational real and reactive power control using linear programming", *Electric Power Systems Research*, Vol. 26, pp. 1-10, 1993.

[7] M. K. Mangoli, K. Y. Lee, and Y. M. Park, "Optimal long-term reactive power planning using decomposition techniques", *Electric Power System Research*, Vol. 26, pp. 41-52, 1993.

[8] D. E. Goldberg, "*Genetic algorithms in search, optimization, and machine learning*", Addison-Wesley, 1989.

[9] D. B. Fogal, "An introduction to simulated evolutionary optimization", *IEEE Trans. on Neural Network*, Vol. 5, No. 1, pp. 3-14, January 1994.

[10] K. Y. Lee, X. Bai, and Y. M. Park, "Optimization method for reactive power planning by using a modified simple genetic algorithm", *IEEE Trans. on Power Systems*, Vol. 10, No. 4, pp. 1843-1850, November 1995.

[11] L. L. Lai and J. T. Ma, "Application of evolutionary programming to reactive power planning-comparison with nonlinear programming approach", *IEEE/PES* Winter Meeting, #96 WM 248-5 PWRS, January 1996.

[12] J. H. Park, S. O. Yang, H. S. Lee, and Y. M. Park, "Economic load dispatch using evolutionary algorithm", *Proc. of International Conference on Intelligent Systems Applications to Power Systems, Orlando, FL*, pp. 441-445, January 1996.

[13] R. Dimeo and K. Y. Lee, "Boiler-turbine control system design using a genetic algorithm", *IEEE Trans. on Energy Conversion*, Vol. 10, pp. 752-759, December 1995.

[14] V. Miranda, J. V. Ranito, and L. M. Proenica, "Genetic algorithm in optimal multistage distribution network planning", *IEEE PES Winter Meeting*, #94 WM 229-5-PWRS.

[15] K. Iba, "Reactive power optimization by genetic algorithm", *IEEE Trans. on Power Systems*, Vol. 9, No. 2, pp. 685-692, May 1994.

Kwang Y. Lee was born in Pusan, Korea on March 6, 1942. He received the B. S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1964, the M.S. degree in electrical engineering from North Dakota State University, Fargo, ND, in 1967, and the Ph.D. degree in systems science from Michigan State University, East Lansing, MI, in 1971. He has been on the faculties of Michigan State, Oregon State, University of Houston, and the Pennsylvania State University, University Park, PA, where he is currently a Professor of Electrical Engineering. His current research interests include control theory, intelligence systems, and their applications to power systems control, operation and planning.

Dr. Lee is currently the Director of Power Systems Control Laboratory and co-director of Intelligent Distributed Controls Research Laboratory at Penn State. He is a Senior Member of IEEE.

Frank F. Yang was born in Taixing, China on March 13, 1969. He received the B. S. degree in electrical engineering from Tsinghua University, Beijing, China, in 1992, the M.S. degree in electrical engineering from Pennsylvania State University, PA, in 1996.

Mr. Yang's interests include control theory, power systems, computer architecture, computer network, and computational intelligence.