

Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms for power economic dispatch

A. Laoufi¹, S. Hadjeri² and A. Hazzab¹

¹*University Center of Bechar B.P 417 BECHAR (08000) ALGERIA*

²*University of Sidi Bel Abbes, Sidi Bel Abbes (22000) ALGERIA*

E-mails: laoufi_ab@yahoo.fr, shadjeri2@yahoo.fr and a_hazzab@yahoo.fr

Abstract

In this paper we describe an efficient approach for solving the economic dispatch problem using Genetic Algorithms (GAs). We recommend the use of adaptive probabilities crossover and mutation to realize the twin goals of maintaining diversity in the population and sustaining the convergence capacity of the GA. In the Adaptive Genetic Algorithm (AGA), the probabilities of crossover and mutation, p_c and p_m , are varied depending on the fitness values of the solutions. By using adaptively varying, p_c and p_m , we also provide a solution to the problem of deciding the optimal values of p_c and p_m , i.e., p_c and p_m need not be specified at all. We compare the performance of the AGA with that of the Standard GA (SGA) in optimizing the penalty function by using a sequential unconstrained minimization technique (SUMT). In this work, the AGA has been applied to a practical 14-bus system to show its feasibility and capabilities. The Numerical and graphical results show that the proposed approach is faster and more robust than the simple static Genetic Algorithm.

Keywords: Economic Dispatch, Sequential unconstrained minimization technique, Genetic Algorithm, Adaptation.

Introduction

In an electrical power system, a continuous balance must be maintained between electrical generation and varying load demand, while system frequency, voltage levels and security have to be maintained. Furthermore, it is desirable that the cost of such generation be minimal [1]. Moreover, the division of load in the generating plant

becomes an important operation, as well as an economic aspect, which must be solved every changing load (1%) or every 30 min [2].

Genetic Algorithms [2, 7, 10, 17] are robust search and optimization techniques which are finding application in a number of practical problems. The robustness of Genetic Algorithms (hereafter referred to as GAs) is due to their capacity to locate the global optimum in a multimodal landscape. A plethora of such multimodal functions exist in engineering problems. Optimisation of neural network structure and learning neural network weights, solving optimal control problems, designing structures and solving flow problems are a few examples. It is for the above reason that considerable attention has been paid to the design of GAs for optimising multimodal functions. GAs employs a random, yet directed, search for locating the globally optimal solution. They are superior to 'gradient descent' techniques as the search is not biased towards the locally optimal solution. On the other hand, they differ from random sampling algorithms due to their ability to direct the search towards relatively 'prospective' regions in the search space.

Typically a GA is characterized by the following Components:

1. genetic representation (or an encoding) for the feasible solutions to the optimization problem
2. a population of encoded solutions
3. a fitness function that evaluates the optimality of each solution
4. genetic operators that generate a new population from the existing population
5. control parameters.

The GA may be viewed as an evolutionary process wherein a population of solutions evolves over a sequence of generations. During each generation, the fitness of each solution is evaluated, and solutions are selected for reproduction based on their fitness. Selection embodies the principle of 'Survival of the fittest. Good solutions are selected for reproduction while bad solutions are eliminated. The selected solutions then undergo recombination under the action of the crossover and mutation operators. It has to be noted that the genetic representation may differ considerably from the natural form of the parameters of the solutions. Fixed-length and binary encoded strings for representing solutions have dominated GA research since they provide the maximum number of schemata and as they are amenable to simple implementation.

The power of GAs arises from crossover. Crossover causes a structured, yet randomized exchange of genetic material between solutions, with the possibility that 'good' solutions can generate 'better' ones. The paper [10] aptly summarizes the working of GAs.

Crossover occurs only with some probability p_c (the crossover probability or crossover rate). When the solutions are not subjected to crossover, they remain unmodified. Notable crossover techniques include the single-point, the two-point, and the uniform types [23].

Mutation involves the modification of the value of each 'gene' of a solution with some probability p_m , (the mutation probability). The role of mutation in GAs has been

that of restoring lost or unexplored genetic material into the population to prevent premature convergence of the GA suboptimal solutions.

Apart from selection, crossover, and mutation, various other auxiliary operations are common in GAs. Of these, scaling mechanisms [16] are widely used. Scaling involves a readjustment of fitness values of solutions to sustain a steady selective pressure in the population and to prevent the premature convergence of the population to suboptimal solutions.

In this paper we describe an efficient technique for solving the economic dispatch problem by using a sequential unconstrained minimization technique (SUMT) under some equality and inequality constraints detailed in equations (10), (11). The equality constraints reflect a real and reactive power balance and the inequality constraints reflect the limits of real and reactive generation. We assume that voltage levels and security are maintained. We recommend the use of adaptive probabilities of crossover and mutation to realize the twin goals of maintaining diversity in the population and sustaining the convergence capacity of the GA. With the approach of adaptive probabilities of crossover and mutation, we also provide a solution to the problem of choosing the optimal values of the probabilities of crossover and mutation (hereafter referred to as p_c and p_m , respectively) for the GA. The choice of p_c and p_m , is known to critically affect the behaviour and performance of the GA. A number of guidelines exist in the literature for choosing p_c and p_m [6, 10, 16, 22]. These generalized guidelines are inadequate as the choice of the optimal p_c and p_m , becomes specific to the problem under consideration. Grefenstette [16] has formulated the problem of selecting p_c and p_m , as an optimization problem in itself, and has recommended the use of a second-level GA to determine the parameters of the GA. The disadvantage of Grefenstette's method is that it could prove to be computationally expensive. In our approach, p_c and p_m are determined adaptively by the GA itself, and the user is relieved of the burden of specifying the values of p_c and p_m .

In the following sections, we firstly, present the problem of the economic dispatch based on the sequential unconstrained minimization technique (SUMT) for its solution by GAs. Section 3 presents the application of genetic algorithms in the multimodal function optimisation. In section 4, we outline the Dynamic Parametric AGA approach; of using adaptively varying probabilities of crossover and mutation for economic dispatch optimisation. In Section 5, we present the measures to quantify the performance of the GA's. Section 6 evaluates a Dynamic Parametric AGA designed in the previous section on the task of solving the (economic dispatch) ED problem and compares AGA performance results with results from a SGA.

Mathematical Formulation of The Economic Dispatch (ED)

The objective of the ED problem is to minimize the total fuel cost at thermal plants:

$$\text{OBJ} = \left[\sum_{i=1}^n F_i(P_i) \right] \quad (1)$$

Subject to the constraint of equality in real and reactive power balance:

$$\sum_{i=1}^n P_i - P_{ch} - P_L = 0 \quad (2)$$

$$\sum_{i=1}^n Q_i - Q_{ch} - Q_L = 0 \quad (3)$$

The inequality of real and reactive power limits on the generator outputs are:

$$P_i^{\min} \leq P_i \leq P_i^{\max} \quad (4)$$

$$Q_i^{\min} \leq Q_i \leq Q_i^{\max} \quad (5)$$

Where $F_i(P_i)$ is the individual generation production cost in terms of its real power generation P_i .

P_i, Q_i = real and reactive power generation for unit i respectively.

n = number of generators in the system.

P_{ch}, Q_{ch} = total real and reactive current system load demand respectively.

P_L, Q_L = total real and reactive system transmission losses respectively.

The thermal plant can be expressed as input-output models (cost function), where the input is the fuel cost and the output is the power output of each unit. In practice, the cost function could be represented by a quadratic function:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \quad (6)$$

The nonlinear programming problem can be formally stated as:

$$\text{Minimize: } f(x) \quad (7)$$

Subject to m linear and/or nonlinear equality constraints:

$$h_i(x) = 0 \quad i = 1, \dots, m \quad (8)$$

and $(p - m)$ linear and/or nonlinear inequality constraints:

$$g_i(x) \geq 0 \quad i = m + 1, \dots, p \quad (9)$$

The SUMT algorithm has been developed to solve the nonlinear programming problem stated by Eqs. (7)-(9), in which the objective function $f(x)$ and inequality constraints $g_i(x)$, can be nonlinear functions of the independent variables, but the equality constraints $h_i(x)$ must be a linear function of the independent variables if convergence to the solution of the nonlinear programming problem is to be guaranteed [1].

The GAs can alleviate this convergence condition; indeed, by using the GAs hence convergence to the solution in this case is always guaranteed.

The constraint problem can be converted as unconstrained problem by defining the penalty function (P) as follows [1]:

$$P(x^{(k)}, r^{(k)}) = f(x^{(k)}) + \frac{1}{r^{(k)}} \sum_{i=m}^m h_i^2(x^{(k)}) + r^{(k)} \sum_{i=m+1}^p \frac{1}{g_i(x^{(k)})} \tag{10}$$

Where the weighting factors r are positive and form a monotonically decreasing sequence of values $\{r / r^\circ > r^1 > \dots > 0\}$.

For the use of the GA's in this problem, this sequence may be expressed as follows:

$$r^k = \begin{cases} r_0 & \text{if } \text{mod}(k, PS) = 0 \\ r_0 / k & \text{Otherwise} \end{cases}, \quad k = 1, \dots, N_g$$

Where:

$r_0 = 1$;

N_g = Maximum number of generation of the GA's;

PS = the population size of the GA's which must have a suitable value.

Note that Fiacco and McCormick [1] originally chose to make the function of the inequality constraints in the form of an added 'barrier':

$$G(g(x^{(k)})) = \sum_{i=m+1}^p \frac{1}{g_i(x^{(k)})}$$

for as one or more $g_i(x^{(k)}) \rightarrow 0$ from the feasible region, $G(g(x^{(k)})) \rightarrow \infty$; hence the concept of a barrier. As $r^{(k)}$ is reduced, the effect of the barrier is reduced, and x may move closer to an inequality constraint boundary. As mentioned before, other possible choices exist for $G(g(x^{(k)}))$, such as:

$$G(g(x^{(k)})) = \sum_{i=m+1}^p \min\{0, g_i(x^{(k)})\}^2$$

or

$$G(g(x^{(k)})) = - \sum_{i=m+1}^p \log(g_i(x^{(k)})) = \sum_{i=m+1}^p \log\left(\frac{1}{g_i(x^{(k)})}\right)$$

The final form of the fitness or the penalty function becomes:

$$P(x^{(k)}, r^{(k)}) = f(x^{(k)}) + \frac{1}{r^{(k)}} \sum_{i=1}^m h_i^2(x^{(k)}) - r^{(k)} \sum_{i=m+1}^p \log(g_i(x^{(k)})) \tag{11}$$

Where 'log' indicate the natural logarithm function.

Adaptive Probabilities Crossover and Mutation

It is essential to have two characteristics in GAs for optimizing the penalty function. The first characteristic is the capacity to converge near global optimum after locating

the region containing the optimum. The second characteristic is the capacity to explore new regions of the solution space in search of the global optimum. The balance between these characteristics of the GA is dictated by the values of p_c and p_m , and the type of crossover employed [23]. Increasing values of p_c and p_m , promote exploration at the expense of exploitation. Moderately large values of p_c (0.5-1.0) and small values of p_m (0.001-0.05) are commonly employed in GA practice [21]. In our approach, we aim at achieving this trade-off between exploration and exploitation in a different manner, by varying p_c and p_m , adaptively in response to the fitness values of the solutions; p_c and p_m , are increased when the population tends to get stuck at a local optimum and are decreased when the population is scattered in the solution space.

To vary p_c and p_m , adaptively, for preventing premature convergence of the GA to local optimum, it is essential to be able to identify whether the GA is converging to an optimum.

One possible way for detecting convergence is to observe the average fitness value \bar{f} of the population in relation to the minimum fitness value f_{\min} of the population. $\bar{f} - f_{\min}$ is likely to be less for a population that has converged to an optimum solution than that for a population scattered in the solution space. We have observed the above property in all our experiments with GAs, and Fig. 1 illustrates this property for a typical case. In Fig. 1, we notice that $\bar{f} - f_{\min}$ decreases when the GA converges to a local optimum. We use the difference in the average and minimum fitness values, $\bar{f} - f_{\min}$, as a yardstick for detecting the convergence of the GA. The values of the p_c and p_m are varied depending on the value of $\bar{f} - f_{\min}$. Since p_c and p_m have to be increased when the GA converged to local optimum, i.e., when $\bar{f} - f_{\min}$ decreases, p_c and p_m will have to be varied inversely with $\bar{f} - f_{\min}$. The expressions that we have chosen for p_c and p_m are of the following form:

$$p_c = k_1 / (\bar{f} - f_{\min}) \quad (12)$$

$$p_m = k_2 / (\bar{f} - f_{\min}) \quad (13)$$

It has to be observed in the above expressions that p_c and p_m do not depend on the fitness value of any particular solution, and have the same values for all the solutions of the population. When a population converges to a globally optimal solution (or even a locally optimal solution), p_c and p_m increase and may cause the disruption of the near-optimal solutions. The population may never converge to the global optimum. Though we may prevent the GA from getting stuck at a local optimum, the performance of the GA (in terms of the generations required for convergence) will certainly deteriorate.

To overcome this problem, we need to preserve 'good' solutions of the population. This can be achieved by having lower values of p_c and p_m , for high fitness solutions and higher values of p_c and p_m for low fitness solutions. While the high fitness

solutions aid in the convergence of the GA, the low fitness solutions prevent the GA from getting stuck at a local optimum. The value of p_m should depend not only on $\bar{f} - f_{\min}$ but also on the fitness value f of the solution. Similarly, p_c should depend on the fitness values of both the parent solutions. The closer f is to f_{\min} , the smaller p_m should be. Similarly, p_c should vary directly as $f_{\min} - f'$. These expressions for p_c and p_m now take the following forms

$$p_c = k_1(f_{\min} - f') / (\bar{f} - f_{\min}), \quad k_1 \leq 1.0 \tag{14}$$

and

$$p_m = k_2(f - f_{\min}) / (\bar{f} - f_{\min}), \quad k_2 \leq 1.0 \tag{15}$$

to constrain p_c and p_m to the range (0.0-1.0).

To prevent the overshooting of p_c and p_m beyond 1.0, we also have the following constraints,

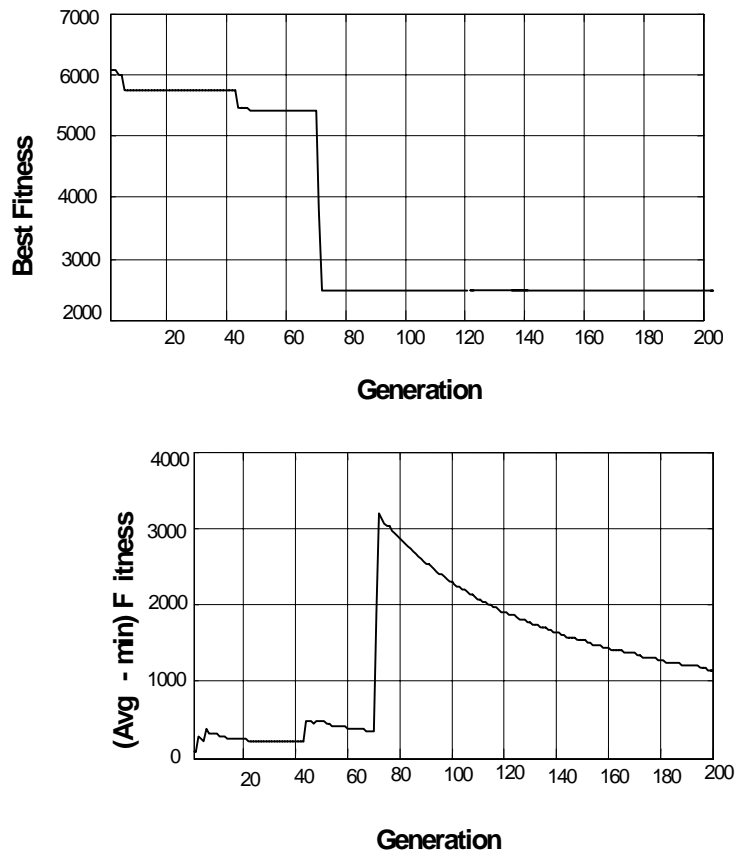


Figure 1: Variation of $\bar{f} - f_{\min}$ and f' (best fitness).

$p_c = k_3$, when $f_{\min} < f'$

and

$p_m = k_4$, when $f < f_{\min}$

where

$k_3, k_4 \leq 1.0$

We saw that for a solution with the minimum fitness value, p_c and p_m are both zero. The best solution in a population is transferred undisrupted into the next generation (elitism). Together with the selection mechanism, this may lead to an exponential growth of the solution in the population and may cause premature convergence.

We now discuss the choice of values for k_1 , k_2 , k_3 and k_4 . For convenience, the expressions for p_c and p_m are given as:

$$p_c = \begin{cases} k_1(f_{\min} - f')/(\bar{f} - f_{\min}) & f' \leq f_{\min} \\ k_3 & f' > f_{\min} \end{cases} \quad (16)$$

and

$$p_m = \begin{cases} k_2(f - f_{\min})/(\bar{f} - f_{\min}) & f \geq f_{\min} \\ k_4 & f < f_{\min} \end{cases} \quad (17)$$

where k_1 , k_2 , k_3 and $k_4 < 1.0$.

It has been well established in GA literature [6] [10] that moderately large values of p_c ($0.5 < p_c < 1.0$), and small values of p_m ($0.001 < p_m < 0.05$) are essential for the successful working of GAs. The moderately large values of p_c promote the extensive recombination of schemata, while small values of p_m are necessary to prevent the disruption of the solutions. These guidelines, however, are useful and relevant when the values of p_c and p_m vary.

For this purpose we use arbitrarily a value of 0.5 for k_4 and k_2 . We assign k_1 and k_3 a value of 1.0.

GA Performance Indices

The objective of our approach is to improve the performance of the GA controls. In this paper we use two measures to quantify the performance of the GAs: online performance to measure ongoing performance and offline performance to measure convergence. Online performance is the running average of all evaluations performed until a given generation (see Eq. 18). Offline performance is the running average of the best performance value until a given generation (see Eq. 19).

$$x_{online}(i) = \frac{1}{PS} \sum_{k=1}^{PS} f_e(i, k) \quad (18)$$

$$x_{offline}(i) = \frac{1}{i} \sum_{j=1}^i f_e^*(j) \quad (19)$$

Where:

$f_e(i, k)$ are the objective function values obtained at generation i for $k = 1, \dots, PS$.

$f_e^*(j)$ are the best function values obtained until a given generation i for $j = 1, \dots, i$.

System studies

To validate our findings we compared the results of the proposed AGA with the results of the simple static GA, on the economic dispatch problem for a 30-bus test system [1, 16]. The single-line data are given in table 1 and 2.

The fuel cost equations (in US \$/h) for the two generators are:

$$\begin{cases} F_1(P_{g1}) = 0.00375 P_{g1}^2 + 2.0 P_{g1} \\ F_2(P_{g2}) = 0.00175 P_{g2}^2 + 1.5 P_{g2} \\ F_5(P_{g5}) = 0.0625 P_{g5}^2 + 1.8 P_{g5} \\ F_8(P_{g8}) = 0.00834 P_{g8}^2 + 2.0 P_{g8} \\ F_{11}(P_{g11}) = 0.02500 P_{g11}^2 + 1.5 P_{g11} \\ F_{13}(P_{g13}) = 0.02500 P_{g13}^2 + 1.8 P_{g13} \end{cases}$$

and the constraints are:

$$50 \leq P_{g1} \leq 200 \quad 20 \leq P_{g2} \leq 80$$

$$15 \leq P_{g5} \leq 50 \quad 10 \leq P_{g8} \leq 40$$

$$10 \leq P_{g11} \leq 30 \quad 12 \leq P_{g13} \leq 40$$

$$-20 \leq Q_{g1} \leq 250 \quad -20 \leq Q_{g2} \leq 100$$

$$-15 \leq Q_{g5} \leq 80 \quad -15 \leq Q_{g8} \leq 60$$

$$-15 \leq Q_{g8} \leq 60 \quad -15 \leq Q_{g13} \leq 60$$

$$P_{ch} = 283 \text{ MW}$$

$$Q_{ch} = 126 \text{ MVAR}$$

By using Newton-Raphson's method [1, 20], the obtained real (P_L) and reactive (Q_L) transmission line losses are respectively 18.84 MW and 26 MVAR. The transmission line data in p.u are given in table 1 and table 2 [1, 20].

Table 1: Bus data in p.u.

Bus	Active	Reactive
1	0.000	0.000
2	0.217	0.127
3	0.024	0.012
4	0.076	0.016
5	0.942	0.190
6	0.000	0.000
7	0.228	0.109
8	0.300	0.300
9	0.000	0.000
10	0.058	0.020
11	0.012	0.010
12	0.112	0.075
13	0.013	0.012
14	0.062	0.016
15	0.082	0.025
16	0.035	0.0018
17	0.090	0.058
18	0.032	0.009
19	0.095	0.034
20	0.022	0.007
21	0.175	0.112
22	0.000	0.000
23	0.032	0.016
24	0.087	0.067
25	0.000	0.000
26	0.035	0.023
27	0.000	0.000
28	0.000	0.000
29	0.024	0.009
30	0.106	0.019

Table 2: Transmission line data in p.u

line	R	X	B(total)
1-2	0.0192	0.0575	0.0264
1-3	0.0452	0.1852	0.0204
2-4	0.0570	0.1737	0.0184
3-4	0.0132	0.0379	0.0042
2-5	0.0472	0.1983	0.0209
2-6	0.0581	0.1763	0.0187
4-6	0.0119	0.0414	0.0045
5-7	0.0460	0.1160	0.0102
6-7	0.0267	0.0820	0.0085
6-8	0.0120	0.0420	0.0045
6-9	0.0000	0.2080	0.0000
6-10	0.0000	0.5560	0.0000
9-11	0.0000	0.2080	0.0000
9-10	0.0000	0.1100	0.0000
4-12	0.0000	0.2560	0.0000
12-13	0.0000	0.1400	0.0000
12-14	0.1231	0.2559	0.0000
12-15	0.0662	0.1304	0.0000
12-16	0.0945	0.1987	0.0000
14-15	0.2210	0.1997	0.0000
16-17	0.0824	0.1923	0.0000
15-18	0.1070	0.2185	0.0000
18-19	0.639	0.1292	0.0000
19-20	0.0340	0.0680	0.0000
10-20	0.0936	0.2090	0.0000
10-17	0.0324	0.0845	0.0000
10-21	0.0348	0.0749	0.0000
10-22	0.0727	0.1499	0.0000
21-22	0.0116	0.0236	0.0000
15-23	0.1000	0.2020	0.0000
22-24	0.1150	0.1790	0.0000
23-24	0.1320	0.2700	0.0000
24-25	0.1885	0.3292	0.0000
25-26	0.2544	0.3800	0.0000
25-27	0.1093	0.2087	0.0000
27-28	0.0000	0.3960	0.0000
27-29	0.2198	0.4153	0.0000
27-30	0.3202	0.6027	0.0000
29-30	0.2399	0.4533	0.0000
8-28	0.0636	0.2000	0.0214
6-28	0.0169	0.0599	0.0065

Simulation GAs Results

First Case

We will take into account only real constraints.

First variant

The transmission line losses are calculated and maintained constant ($P_L = 17.51$ MW)

The power balance equation will become:

$$P_{G1} + P_{G2} + P_{G5} + P_{G8} + P_{G11} + P_{G13} = 300.51 \text{ MW}$$

The initial points are:

$$P_{G10} = 120 \text{ MW}, \quad P_{G20} = 60 \text{ MW}, \quad P_{G50} = 40 \text{ MW}$$

$$P_{G80} = 30 \text{ MW}, \quad P_{G11} = 20 \text{ MW}, \quad P_{G10} = 30 \text{ MW}$$

The results of the real generated optimal power and minimum fuel cost are given in table 3.

Results for online and offline comparisons of the simple static GA (static DeJong parameter settings) and the Dynamic Parametric GA are given in fig. 2 and 3.

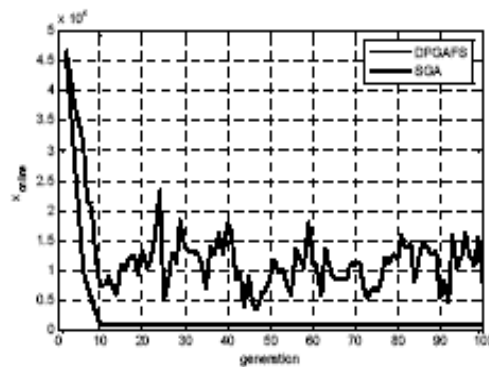


Figure 2: Comparison of online performance of the DPGA-FLC and the SGA (case 1, 1st variant)

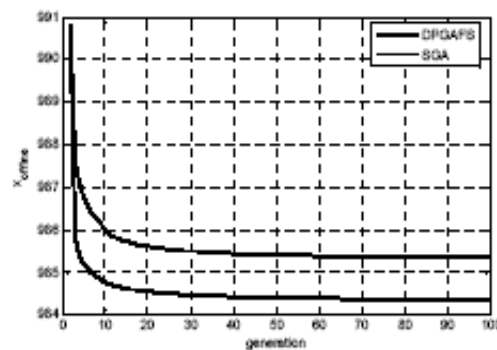


Figure 3: Comparison of offline performance of the DPGA-FLC and the SGA (case 1, 1st variant)

Second variant

The transmission line losses are considered as a linear function of real generated power. The coefficients were calculated by the Gauss-seidel's method:

$$P_L = 0.0119P_{G1} + 0.0341P_{G2} + 0.1004P_{G5} \\ + 0.1625P_{G8} + 0.0275P_{G11} + 0.0390P_{G13}$$

The power balance equation will become therefore:

$$0.9881 P_{G1} + 0.9656 P_{G2} + 0.8996 P_{G5} + 0.8375 P_{G8} \\ + 0.9725 P_{G11} + 0.961 P_{G13} \\ = 283.4 MW$$

We take the same initial points as the first variant. The results of the real generated optimal power minimum fuel cost transmission line losses are given in table 4.

Table 3: Real constraints only

	Adaptive GA		Static GA	
	1 st variant	2 nd variant	1 st variant	2 nd variant
P_{G1}^{opt}	138.29	120.77	131.43	131.54
P_{G2}^{opt}	68.14	59.62	55.05	51.81
P_{G5}^{opt}	38.41	39.64	35.36	41.19
P_{G8}^{opt}	18.73	29.81	26.05	21.96
P_{G11}^{opt}	18.66	20.33	23.53	26.45
P_{G13}^{opt}	18.81	30.01	28.70	26.82
P_L	12.12	17.51	13.00	17.51
F_{opt}	855.0	865.79	857.05	876.7

Where P_{Gi}^{Opt} and Q_{Gi}^{Opt} are the optimal value of the real and reactive power of i^{th} generator in MW and MVAR respectively. And:

$$F_{opt} = \sum_{i=1}^{N_g} F_i(P_i)$$

Second Case

We will take into account real and reactive constraints with the same considerations in the first case; In the two variants we will take the same initial points::

$$P_{G10} = 150 \text{ MW and } P_{G20} = 120 \text{ MW}$$

$$Q_{G10} = 30 \text{ MVAR and } Q_{G20} = 20 \text{ MVAR}$$

The results of real and reactive generated optimal power, total real transmission losses, minimum fuel cost and computing time are given in table 5.

Table 4: Real and reactive constraints

	Adaptive GA		Static GA	
	1 st variant	2 nd variant	1 st variant	2 nd variant
P_{G1}^{opt}	137.69	120.77	130.85	31.54
P_{G2}^{opt}	67.55	59.62	50.99	51.81
P_{G5}^{opt}	37.80	39.64	40.66	41.19
P_{G8}^{opt}	18.09	29.81	21.07	21.96
P_{G13}^{opt}	18.15	30.01	26.00	26.82
Q_{G1}^{opt}	40.18	49.18	39.00	21.63
Q_{G2}^{opt}	31.93	39.50	18.72	8.49
Q_{G5}^{opt}	21.97	19.79	28.50	27.59
Q_{G8}^{opt}	2.18	9.79	8.63	17.58
Q_{G11}^{opt}	11.93	-0.04	8.59	17.51
Q_{G13}^{opt}	-1.55	-8.90	8.63	12.87
P_L	11.85	17.51	13.00	17.51
F_{opt}	865.47	873.18	871.96	883.50

Conclusion

Recent research on GAs has witnessed the emergence of new trends that break the traditional mold of 'neat' GAs that are characterized by static crossover and mutation rates, fixed length encodings of solutions, and populations of fixed size.

In this paper, we adopt a new approach to determine p_c and p_m the probabilities of crossover and mutation. The approach is different from the previous techniques for adapting operator probabilities as p_c and p_m . The values of p_c and p_m range from 0.0 to 1.0 and 0.0 to 0.05 respectively. However, it is the manner in which p_c and p_m are adapted to the fitness values of the solutions, in the way that not only improves the

convergence rate of the GA, but also prevents the GA from getting stuck at a local optimum. In the adaptive GA, low values of p_c and p_m are assigned to high fitness solutions, while low fitness solutions have very high values of p_c and p_m . The best solution of every population is 'protected', i.e., it is subjected to crossover, and when a problem occurs, it receives only a minimal amount of mutation. On the other hand, all solutions with a fitness value less than the average fitness value of the population have p_c . This means that all subaverage solutions are completely disrupted and totally new solutions are created. The GA can, thus, rarely get stuck at a local optimum.

In this work, we have chosen one particular way of adapting p_c and p_m based on the various fitnesses of the population. The results are encouraging, and future work should be directed at developing other such adaptive models for the probabilities of crossover and mutation.

References

- [1] M. Rahli and P. Pirotte, "Optimal load flow using sequential unconstrained minimization technique (SUMT) method under power transmission losses minimization" *Electric Power Systems Research* 52 (1999), pp. 61-64.
- [2] J. D. Schaffer and A. Morishima, "An adaptive crossover mechanism for genetic algorithms," in *Proc. Second Int. Conf. Genetic Algorithms*, 1987, pp. 36-40.
- [3] L. Davis (Ed.), *Genetic Algorithms and Simulated Annealing*, London: Pitman, 1987.
- [4] L. Davis, "Adapting operator Probabilities in genetic algorithms," *Proc. Third Int. Genetic Algorithms*, 1989, pp. 61-69.
- [5] ——— (Ed), "Handbook of Genetic Algorithms," Van Nostrand Reinhold, 1991.
- [6] ———, "Bit climbing, representational bias, and test suite design" in *Proc. Fourth Int. Conf Genetic Algorithms*, pp. 18-23, 1991.
- [7] K. A. DeJong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, University of Michigan (1975).
- [8] K. A. DeJong, "Genetic algorithms: A 10 year perspective" in *Proceedings of an International Conference of Genetic Algorithms and their Applications*, (J Greffenstette, editor), Pittsburgh, July 24-26, 1985. pp. 169-177.
- [9] K. A. Ddong, "Adaptive system design: a genetic approach," *IEEE Trans Syst, Man, and Cybernetics*, Vol. 10(9), pp. 566-574, Sep. 1980.
- [10] T. C. Fogarty, "Varying the probability of mutation in genetic algorithms," *Proc. Third Int. Conf. Genetic Algorithms*, 1989, pp. 104-109.
- [11] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Reading MA: Addison Wesley 1989.
- [12] D. E. Goldberg, "A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing," *Complex Syst.*, Vol. 4, pp. 445-460, 1990.

- [13] D. E. Goldberg, "Genetic algorithms and Walsh functions: Part II, deception and its analysis," *Complex Syst.*, Vol. 3, pp. 153-171, 1989.
- [14] D. E. Goldberg and al., "Messy genetic algorithms: motivation, analysis and first results," *Complex Syst.*, Vol. 3, pp. 493-530, 1989.
- [15] D. E. Goldberg and al., "Messy genetic algorithms revisited: Studies in mixed size and scale," *Complex Syst.*, Vol. 4, pp. 415-444, 90.
- [16] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," *Proc. Second 2nt Conf Genetic Algorithms*, 1987, pp. 41-49.
- [17] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Trans Syst. Man, and Cybernetics*, Vol. SMC-16, N°1, pp. 122-128, Jan./Feb. 1986.
- [18] J. H. Holland, "Adaptation in Natural and Artificial Systems," Ann Arbor. Univ. Michigan Press, 1975.
- [19] O. H. Ibarra & S. K. Sahni, "Polynomially complete fault detection problems," *IEEE Trans. Computers*, Vol. C-24, pp. 242-247, Mar. 1975.
- [20] ———, "Special session: Recent algorithms for gate-level atpg with fault simulation and their performance assessment," *Proc.1985 IEEE Int. Symp. Circuits and Syst. (ISCAS)*, June 1985, pp. 663-698.
- [21] Jurgen Hesser & Reinhard Manner, "Towards an optimal mutation probability for genetic algorithms," *Proceedings of the First Workshop*,
- [22] M. Oliver, D. J. Smith and J. R. C. Holland, "A study of permutation crossover operators on the travelling salesman problem," *Proc. Second Int. Con. Genetic Algorithms*, 87, pp. 224-230.
- [23] J. D. Schaffer and al., "A study of control parameters affecting online performance of genetic algorithms for function optimisation" *Proc. Third Int. Conf Genetic Algorithms*, 1989, pp. 51-60.
- [24] W. M. Spears and K. A. DeJong, "An analysis of multipoint crossover," in the *Proc. 1990 Workshop of the Foundations of Genetic Algorithms*, 1991 pp. 301-315.
- [25] D. Whitley and D. Starkweather, "Genitor-II A distributed genetic algorithm," *J. Expt. Theor. Artif Int.*, Vol. 2, pp. 189-214, 1990.
- [26] D. Whitley and al., "Genetic algorithms and neural networks: Optimizing connections and connectivity," *Parallel Computing*, Vol. 14, pp. 347-361, 1990.