ISTANBUL UNIVERSITY –
JOURNAL OF ELECTRICAL & ELECTRONICS ENGINEERING

YEAR : 2006 (175 - 181)
VOLUME : 6
NUMBER : 2

ECONOMIC POWER DISPATCH USING THE COMBINATION OF TWO GENETIC ALGORITHMS

Mimoun YOUNES Mostafa RAHLI

Département de génie mécanique, Université de Sidi Bel Abbés, BP 89 Sidi DJillali, SBA, Algérie

E-mail: younesmi@yahoo.fr

ABSTRACT

Lately, one notices an increase in the applications of the genetic algorithms (GA) through several fields, and one noted that all the utilisateurs(exploiteurs) these algorithms wonder on the choice of the values of the genetic operators in order to increase the performances of the algorithm. The objective of this article tries to solve this problem, by using two genetic algorithms one to determine the values of the genetic operators and the other to optimize the function cost. Numerical results on a test system consisting of 13 thermal units show that the proposed approach has an ability to find the better solutions than the conventional genetic algorithm.
.

Keywords: *Dispatching, Genetic algorithm (GA).*

1. INTRODUCTION

The theory of natural evolution maintains that adaptive changes to species occur through natural selection, whereby individuals better suited for survival are more likely to reproduce and hence increase the representation of their genetic material in the gene pool.

The combining of genetic material from two such individuals can produce an offspring even better suited to the environment. This is how species evolve, i.e., sex is a mechanism for exchanging genetic material and hence for species to rapidly adapt to their environment.

To address the shortcomings of classical deterministic optimization algorithms, a number of algorithms based on natural selection and embedded randomness have been developed. One such class of algorithms are the population-based methods of Evolutionary Computation (EC). EC is a class of search and optimization techniques based on the Darwinian evolutionary model. The development of EC can be traced back to the 1960s and the Evolutionary Strategies of Rechenberg and Schwefel in Germany and the Genetic Algorithms (GAs) of Holland [6] in the United States (see [8] for further historical background). One of their primary features is that they search through a large population of solutions for the one which optimizes a given evaluation or fitness function.

GA explore better the space of research to the total research of the optimum compared to other methods like the traditional methods (déterministes)[2]. Nevertheless these methods soufrent on the determination and the choice of the values of the genetic operators (operator of crossing, operator of change and size of the population).

Received Date : *23.05.2006*
Accepted Date: *25.06.2006*

In this article one tries to find a solution with this problem, by using the combination of two genetic algorithms the first to determine the genetic operators and the second for the optimization of the objective function (cost).
The method was applied to a network real 62 bus.

This article is organized as follows. Section 2 is developed to the Objective and Mathematical formulation. In section 3 wet expose the genetic algorithm to solve the problem of ED is developed. Section 4 contains the obtained results which show the potential of the genetic algorithm. In conclusion, in section 5, several suitable remarks are given to conclude this paper.

2. OBJECTIVE AND MATHEMATICAL FORMULATION

The probleme of the economic which exist to minmize the cost of production of the real power can generally be stated as folows[1,2,3,4]:

$$Min\left[\sum_{i=1}^{NG} C_i\left(P_{Gi}\right)\right] \quad (1)$$

Under the following constraints:

$$P_{Gi}^{\min} \le P_{Gi} \le P_{Gi}^{\max} \quad (2)$$

$$Q_{Gi}^{\min} \le Q_{Gi} \le Q_{Gi}^{\max} \quad (3)$$

$$\sum_{i=1}^{NG} P_{Gi} = \sum_{j=1}^{ND} P_{Dj} + P_L \quad (4)$$

$$\sum_{i=1}^{NG} Q_{Gi} = \sum_{j=1}^{ND} Q_{Dj} + P_L \quad (5)$$

where generally

$$C_i\left(P_{Gi}\right) = a_i + b_i P_{Gi} + c_i P_{Gi}^2 \quad (6)$$

and a_i ,b_i ,c_i are the known coefficients and NG , is number of generator; ND , is number of loads; P_{Gi} , is real power generation; Q_{Gi} , is reactive power generation; P_{Dj} , is real power load; Q_{Dj} , is reactive power load; P_L and Q_L are respectively real and reactive losses.
The nonlinear programming problem can be formally stat as:

Minimize: $f(x)$

Subject to m linear and /or nonlinear equalty constraints

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

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

$$g_i(x) \ge 0 \quad i = m+1,\cdots,p$$

the nonlinear programming problem is converted into a sequence of unconstrained problems[4] by defining the p function as follows

$$P\left(x^k, r^k\right) = f\left(x^k\right) + \frac{1}{r^k}\sum_{i=1}^{m} h_i^2\left(x^k\right) - r^k \sum_{i=m+1}^{p} \ln g_i\left(x^k\right) \quad (7)$$

Where $r^k > 0$ is called the coefficient of penalty.

3. GENETIC ALGORITHM IN ECONOMIC POWER DISPATCH

3.1. GA Applied to economic power dispatch

A simple Genetic Algorithm is an iterative procedure, which maintains a constant size population P of candidate solutions. During each iteration step (generation) three genetic operators (reproduction, crossover, and mutation) are performing to generate new populations (offspring), and the chromosomes of the new populations are evaluated via the value of the fitness witch is related to cost function. Based on these genetic operators and the evaluations, the better new populations of candidate solution are formed. With the above description, a simple genetic algorithm is given as follow [6,7]:

1. Generate randomly a population of binary string
2. Calculate the fitness for each string in the population
3. Create offspring strings through reproduction, crossover and mutation operation.
4. Evaluate the new strings and calculate the fitness for each string (chromosome).
5. If the search goal is achieved, or an allowable generation is attained, return the best chromosome as the solution; otherwise go to step 3. We now describe the details in employing the simple GA to solve the economic power dispatch.

3.2. Chromosome coding and decoding

GAs work with a population of binary string, not the parameters themselves. For simplicity and convenience,binary coding is used in this paper [9,10]. With the binary coding method, the active generation power set of 14-bus test system (P_{G1} and P_{G2}) would be coded as binary string of '0' and '1' with length L1 and L2 (may be different), respectively. Each parameter P_{Gi} has upper bound b*i* and lower bound ai.The choice of m1 *and* m2 for the parameters is concerned with the resolution specified by the designer in the search space. In the binary coding method, the bit length mi and the corresponding resolution *Ri* is related by

$$R_i = \frac{bi - ai}{2^{mi} - 1} \quad (8)$$

As result, the *Pgi* set can be transformed into a binary string (chromosome) with length ∑m*i* and then the search space is explored. Note that each chromosome present one possible solution to the problem. For example, suppose the parameter domain of (P_{G1} *and* P_{G2}) which is presented in Table 1:

Table1. *Parameters set of P* $_{Gi}$

Bus	P_{Gi}^{min} MW)	P_{Gi}^{max} (MW)	a_i ($/hr)	b_i ($/MW.hr)	c_i ($/MW^2.hr)
1	135	195	100	1.5	0.006
2	70	145	130	2.1	0.009

If the resolution (*R1*, *R2*) is specified as (0.235, 0.294). we have m1=8 and m2=8.Then the parameter set (P_{G1}, P_{G2})can be coded according to the following (Table 2):

Table 2. *Coding of* P_{Gi} *parameter set*

P_{G1} (MW)	code	P_{G2} (MW)	code
135.000	00000000	70.000	00000000
135.235	00000001	70.294	00000001
135.470	00000010	70,588	00000010
135.705	00000011	70,882	00000011
.	.	.	.
.	.	.	.
194.455	11111101	144,382	11111101
194.690	11111110	144,97	11111110
195.000	11111111	145.000	11111111

If the candidate parameters set is (194.455,72.940), then the chromosome is a binary string 1111110100001010. The decoding procedure is the reverse procedure.

The first step of any genetic algorithm is to generate the initial population. A binary string of length L is associated to each member (individual) of the population. The string is usually known as a chromosome and represents a solution of the problem. A sampling of this initial population creates an intermediate population Thus some operators (reproduction, crossover and mutation) are applied to this new intermediate population in order to obtain a new one.

The process, that starts from the present population and leads to the new population, is named a generation when executing a genetic algorithm (Table 3).

Table3. *First generation of GA process for 14 bus example*

Chr	Initial population	P_{G1} (MW)	P_{G2} (MW)	Fcost ($/h)
1	0101011101110001	155.470	103.235	1139.928
2	0101011101110010	155.470	103.529	922.109
3	0110101001100001	159.941	98.529	1627.19
4	1100010100011011	181.117	77.941	925.607
5	0101011001110011	155.235	103.823	931.243
6	1011100100100011	178.529	80.294	994.506
7	1010001000110110	173.117	85.882	916.229
8	0100101001111100	152.411	106.470	958.643

3.3. Crossover

Crossover is the primary genetic operator, which promotes the exploration of new regions in the search space. For a pair of parents selected from the population the recombination operation divides two strings of bits into segments by setting a crossover point at random, i.e. Single Point Crossover. The segments of bits from the parents behind the crossover point are exchanged with each other to generate their offspring[11,12,13]. The mixture is performed by choosing a point of the strings randomly. And switching their segments to the left of this point. The new strings belong to the next generation of possible solutions. The strings to be crossed are selected according to their scores using the roulette wheel [8]. Thus, the strings with larger scores have more chances to be mixed with other strings because all the copies in the roulette have the same probability to be selected.

3.4. Mutation

Mutation is a secondary operator and prevents the premature stopping of the algorithm in a local solution. The mutation operator is defined by a random bit value change in a chosen string with a low probability of such change.
The mutation adds a random search character to the genetic algorithm, and it is necessary to avoid that, after some generations, all possible solutions were very similar ones.
All strings and bits have the same probability of mutation. For example, in the string 0101011101110001, if the mutation affects to time bit number six, the string obtained is 0101011101100001 and the value of P_{G2} change from 103.235 MW to 98.529MW.

3.5. Reproduction

Reproduction is based on the principle of survival of the better fitness. It is an operator that obtains a fixed number of copies of solutions according to their fitness value. If the score increases then the number of copies increases too. A score value is of associated to a given solution according to its distance of the optimal solution (closer distances to the optimal solution mean higher scores).

3.6. Description of Genetic Algorithm

The cost function is defined in [11]
Our objective is to search (P_{G1}, P_{G2})in their admissible limits to achieve the optimisation problem of EPD.
The cost function *F(x)* takes a chromosome (a possible(P_{G1}, P_{G2})) and returns a value. The value of the cost is then mapped into a fitness value *Fit(* P_{G1} *,* P_{G2} *)* so as to fit in the genetic algorithm.
To minimise *F(x)* is equivalent to getting a maximum fitness value in the searching process. A chromosome that has lower cost function should be assigned a larger fitness value.
The objective of EPD has to be changed to the maximisation of fitness to be used in the simulated roulette wheel as follows:

$$fitness = \begin{cases} \text{Fmax} - \text{F}_j(P_{G1}, P_{G2}) & if \quad \text{Fmax} \geq \text{F}_j(P_{G1}, P_{G2}) \\ 0; & otherwise \end{cases}$$

3.7. Other parameters

The operators of the genetic algorithm are guided by a certain number of parameters fixed in advance, whose their values influence the success or not algorithme genetic. These parameters are:

• Size of the population N, and L is the length of the coding of each individual, (in the case of the binary coding): If N is too large the computing time of the algorithm can prove very significant, and if N is too small, it can converge too quickly towards a bad chromosome. This importance of the size is primarily due to the concept of parallelism implicit which implies that the number of individuals treated by the algorithm is at least proportional to the cube of the number of individuals. In general, the value of the size of the population[13] is between 30 and 50 individuals

• Probability of crossing Pc*:* It depends on the form of the selective function. Its choice is in general heuristic (just like for P_m . The higher it is, the more the population sudden of significant changes. The usual rate is selected between 60% and 100%.

• Probability of changes P_m This misses is general E dregs weak (between 0.1% and 5%), since one high misses is likely to lead to has research too alé has toire. Rather than to reduce P_m another way of preventing that the best individuals are faded is to uses the taken back explicit one of the elite in some propo R tion. Thus, very often, best the 5%, for example, of the population are directly reproduced with identical, the operator of reproduction not playing whereas one the 95% remainders. That is called has élitiste strategy.

4. SIMULATION

4.1.Simulation conditions

The GAGA has been developed by the use of Matlab version 6.5. the tested with Pentium 4, 1500 MHz PC with 128 MB RAM.
The proposed algorithms in this paper have been compared to the conventional GA by applying to the test system [14] which consists of 13 thermal generators. The coefficients ai , bi , ci , ei and fi of the heat-rate functions appearing in Eqs. (6) and operation limits of the generators used in the test system are tabulated in Table 4. Total load demand of the system is 2520 (MW), and 13 generators should satisfy this load demand economically. The parameter values used for GA are given in Table 5.

Table 4. *Characteristic of 13 Generators of network 62 – bus*

Type	P_{Gi}^{min} (MW)	P_{Gi}^{max} (MW)	a_i ($/hr)	b_i ($/MW.hr)	c_{i*10}^{-2} ($/MW^2.hr)
1	0	680	550	8.1	0.00028
2	0	360	309	8.1	0.00056
3	0	360	307	8.1	0.00056
4	60	180	240	7.74	0.00324
5	60	180	240	7.74	0.00324
6	60	180	240	7.74	0.00324
7	60	180	240	7.74	0.00324
8	60	180	240	7.74	0.00324
9	60	180	240	7.74	0.00324
10	40	120	126	8.6	0.00284
11	40	120	126	8.6	0.00284
12	55	120	126	8.6	0.00284
13	55	120	126	8.6	0.00284

4.2.Simulation results:

The GAGA, GA1, GA2, GA3 and GA4 have been evaluated on the above power system. The adopted parameters in the algorithms are given in Table 2. The objective function with the active power penalty, equation (7), is used. Owing to the randomness in the GA approach, The best active power dispatch solutions together with the associated power found by the GAGA are tabulated in Table 2. For comparison purposes, the dispatch solutions obtained by the GA1,GA2,GA3 and GA4, are summarized in Table 2.

The results in Tables 2 show that the dispatch solutions determined by the GAGA lead to lower active power than that found by theGA1, GA2, GA3 and GA4, which confirms that the GAGA is well capable of determining the global or near-global optimum dispatch solution. In addition, the results summarized in Table 2 show that the proposed GAGA is about two times faster than GA2 in speed. The optimization search procedures by the GAGAand GA1,GA2,GA3 and GA4 is shown in Figure 3. It can be seen that, by using the adaptive probabilities of crossover and mutation, the iterations for convergence can be reduced greatly.

Table 5. *results of GAGA compared with classical GA for the Networks 62-bus system*

	GAGA	GA 1	GA 2	GA3	GA4
Pc	0.6878	0.6125	0.6094	0.6173	0.6549
Pm	0.0062	0.0022	0.0023	0.0352	0.0060
N	82	100	150	130	100
crossover operator	single point	single point	single point	single point	single point
selection operator	tournament	tournament	tournament	tournament	tournament
1	656.313	680.000	656.000	642.666	666.666
2	340.235	306.352	,289.411	,283.764	340.235
3	354.352	333.176	,333.176	,337.411	341.647
4	122.588	146.117	180.000	136.705	158.823
5	174.352	124.941	121.647	156.000	161.647
6	119.294	179.529	132.470	,168.705	120.235
7	157.411	124.941	175.294	,138.588	124.941
8	168.705	169.176	155.529	134.823	176.235
9	64.705	60.470	61.882	149.411	68.000
10	102.745	58.823	112.470	80.156	102.117
11	45.333	119.058	71.058	98.039	61.333
12	119.235	111.843	119.490	116.176	114.901
13	95.019	105.470	,111.588	77.686	83.294
Cost ($/h)	23681.313	23752.198	23693.211	24474.907	23755.667
Time (S)	33.897	46.27	64.87	57.90	46.42

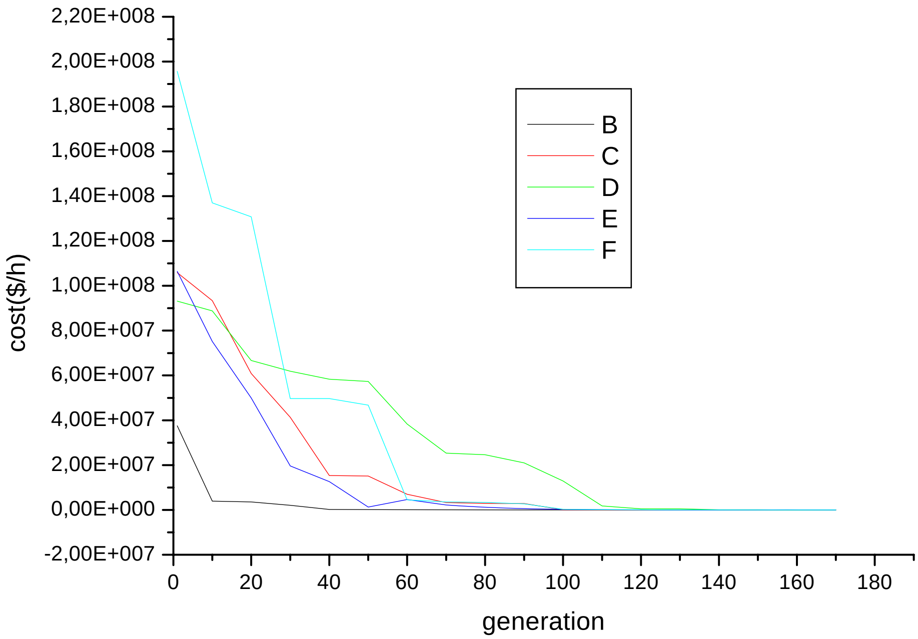

Fig. 8. *Convergence characteristic curves of the GAGA(B),GA1(C),GA2(D),GA3(E) and GA4(F).*

5. CONCLUSIONS

A method composed by two genetic algorithms (GAGA) has been developed for determination of the global or near-global optimum solution for optimal active power dispatch of power systems. In the principal genetic algorithm , the probabilities of crossover (*p*c), mutation (*p*m) and size of the population (N), were chosen by second genetic algorithme. The performance of the proposed method demonstrated through its evaluation on the 62-bus power system shows that the GAGA is able to undertake global search with a fast convergence rate and a feature of robust computation. From the simulation study, it has been found that the GAGA converges to the global optimum.

REFERENCES

[1] R. Fletcher, ” Practical Methods of Optimisation”, John Willey & Sons, 1986.

[2] M. Rahli, “Applied Linear and Nonlinear Programming to Economic Dispatch”, Ph.D. Thesis, Electrical Institute, USTO, Oran, Algeria, 1996.

[3] G.W. Stagg and A.H. E1 Abiadh.” Computer Methods in Power System Analysis” McGraw–Hill Book Company, New York (1968).

[4] D.M. Himmelblau. In: “Applied Non Linear Programming ” McGraw– Hill, New York (1972).

[5] M. Younes, M. Rahli and A. koridak., ”Dispatching Economique par l’intelligence Artificielle ”, Proceeding ICEL’2005,U.S.T. Oran, Algeria, Vol.02, pp.16-21,13-15 November 2005.

[6] Holand, J. H, " Adaptation in natural and Artificial Systems. " The university of Michigan press, Ann Arbor, USA, 1975.

[7] D. E. Goldberg ” Genetic Algorithms in Search, Optimization and Machine Learning”, Addison Wesley Publishing Company, Ind. USA, 1989.

[8] T. B¨ack, F. Ho_meister, and H.-P. Schwefel. A survey of Evolution Strategies. In R. Belew and L. Booker, editors, *Proceedings of the fourth international conference on genetic algorithms*, pages 2-9. Morgan Kaufmann: San Mateo, CA, 1991.

[9] L. Lai, J. T. Ma, R. Yokoma, M. Zhao " Improved genetic algorithms for optimal power flow under both normal and contingent operation states" Electrical Power & Energy System, Vol. 19, N.5, pp. 287-292, 1997.

[10] B. S. Chen, Y. M. Cheng, C. H. Lee, " A Genetic Approach to Mixed H2/H00 Optimal PID Control ", IEEE Control system, pp. 551-59, October 1995

[11]M.Younes, M.Rahli: La Répartition Optimale des puissances en utilisant l'algorithme génétique,ICEEE'2004, Laghouat, April, 24-26, 2004

[12]N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay: Evolutionary programming techniques for economic load dispatch. *IEEE Trans. Evol. Comput.*, vol. 7, no. 1, pp. 83–94, Feb. 2003.

[13]M.Younes, M.Rahli, H Koridak : Genetic/ Evolutionary Algorithms and application to power systems, PCSE'05, O. E. Bouaghi, May 9-11, Univ, Algeria, 2005.

[14] Wong K.P., Wong Y.W., Genetic and genetic/simulated-annealing approaches to economic dispatch, IEE Proc. 141 (5) (1994) 507_ 513.