

Comparative study of the Genetic Algorithms and Hessians Method for Minimization of the Electric Power Production Cost

L. Abdelmalek, M. Zerikat, and M. Rahli

Abstract—In this paper, we present a comparative study of the genetic algorithms and Hessian's methods for optimal research of the active powers in an electric network of power. The objective function which is the performance index of production of electrical energy is minimized by satisfying the constraints of the equality type and inequality type initially by the Hessian's methods and in the second time by the genetic Algorithms. The results found by the application of AG for the minimization of the electric production costs of power are very encouraging. The algorithms seem to be an effective technique to solve a great number of problems and which are in constant evolution. Nevertheless it should be specified that the traditional binary representation used for the genetic algorithms creates problems of optimization of management of the large-sized networks with high numerical precision.

Keywords—Genetic algorithm, Flow of optimum load impedances, Hessians method, Optimal distribution.

I. INTRODUCTION

THE technological development contributed to an increase in the consumption of energy which had as a consequence an increase in powers to generate and to transport. Consequently, the networks become increasingly large and complicated from where the interest of any organization charged to produce the electric power is to ensure in any time and any place the cover of the active and reactive powers required by the customers with a fuel cost as weak as possible. The contribution of this work is to use a method of optimization to minimize the objective function which is the performance index of production of electrical energy, by satisfying the constraints of the equality type and inequality of this function. The contribution of this work is to use a method of optimization to minimize the objective function which is the performance index of production of electrical energy, by satisfying the constraints of the equality type and inequality of this function. That, we chose to present a comparative study of AG under Matlab and the Hessians method (Zoutendijk method, Pearson method, DFP method, and Broyden method) [3,4]. The genetic algorithms currently constitute an important subject of research within the framework of optimization and operations research in the management of the electrical supply networks.

II. MATHEMATICAL MODELLING

Let us consider an energy electro network which one at any moment knows the active powers requested in order to satisfy

the many consumers related to this network. The fuel expenses necessary for the production of the electric outputs are a monotonous function. The supervisor in charge for the control system has infinity of solutions to distribute these powers with the consumers. But among all the existing solutions, it is necessary to guarantee the optimal distribution in a time very reduced and consisting in minimizing the total production cost of the electric power. The function of the total cost of energy production known as objective function (strongly depending on the active powers to generate) is given by the following mathematical form:

$$F = \sum_{i=1}^{n_g} F_i(P_{gi}) \quad (1)$$

Where:

F : denotes the total performance index

F_i : performance index of the th generator

n_g : generator numbers.

P_{gi} : active power produced by the th generator.

The goal consists in minimizing the total performance index of production.

$$\min \left[\sum_{i=1}^{n_g} F_i(P_{gi}) \right] \quad (2)$$

with :

$$P_{gi}^m \leq P_{gi} \leq P_{gi}^M \quad (3)$$

$$Q_{gi}^m \leq Q_{gi} \leq Q_{gi}^M \quad (4)$$

$$\sum_{i=1}^{n_g} P_{gi} = \sum_{j=1}^n P_{chj} + P_L \quad (5)$$

$$E_i \leq E_i^M \quad (6)$$

Generally $F_i(P_{gi})$ is a quadratic function of the form:

$$F_i(P_{gi}) = a_i + b_i P_{gi} + c_i P_{gi}^2 \quad (7) \quad \text{penalty).}$$

a_i, b_i, c_i Represent known constants.

n_g generator numbers

P_{gi} active power produced by generator i

Q_{gi} reactive power produced by generator i

P_{ch} power activates of total load

Q_{ch} power reactivates of total load

P_L active losses total

E_i (internal excitation) nodal tension of node i .

III. MATHEMATICAL FORMULATION

The problem of minimization of the function cost of the powers generated [3, 4] by the nonlinear programming can be raised in the following way:

To minimize $f(x)$

Subject to the equality and inequality constraints.

$$H_i(x) = 0 \quad \text{for } i = 1, \dots, m \quad (8)$$

$$G_i(x) \geq 0 \quad \text{for } i = m+1, \dots, p \quad (9)$$

IV. METHOD OF PENALTY

The mathematical methods that we will compare are methods with constraints (genetic Algorithm and Hessian's methods). For this reason one goes used a method based on the transformation of the original problem constrained into an unconstrained auxiliary problem and where the minimum is the same one in both cases. The guiding principle of these methods consists in modifying the criterion by adding a function of penalization to him which allows the passage of the programming with constraints in programming without constraints.

$$F(P_{gi}, R_k) = F(P_{gi}) + R_k \sum_{i=1}^k G_i^2(P_{gi}) + \frac{1}{R_k} \sum_{i=1}^m H_i^2(P_{gi}) \quad (10)$$

$$G_i(P_{gi}) \geq 0 \quad \text{for } i = 1, \dots, k \quad (11)$$

(Constraints of the inequality type)

$$H_i(P_{gi}) = 0 \quad \text{for } i = 1, \dots, m \quad (12)$$

(Constraints of the equality type)

R_k is a constant of adjustment of calculation (Coefficient of

V. HESSIANS METHOD DESCRIPTION

The principle of these methods consist primarily of a generalization of the iterative formula of Newton[5] original by the approximation of the quadratic formula of $F_i(P_{gi})$ by neglecting the third term as well as the terms of a nature higher starting from the development in Taylor series. At the iteration k , the transition from item $x^{(k)}$ to another successive point $x^{(k+1)}$ is given by the formula below:

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)} = x^{(k)} + \lambda^{(k)} s^{(k)} \quad (13)$$

where:

$\Delta x^{(k)}$ the vector of $x^{(k)}$ to $x^{(k+1)}$

$s^{(k)}$ the vector in the direction $\Delta x^{(k)}$

$\lambda^{(k)}$ the scalar with $\Delta x^{(k)} = \lambda^{(k)} \cdot s^{(k)}$

The minimum of the function $f(x)$ in the direction of $\Delta x^{(k)}$ is obtained by the differentiation of $f(x)$ by respecting each component of Δx and by canceling the resulting equation.

$$\Delta x^{(k)} = -[\nabla^2 f(x^{(k)})]^{-1} \cdot \nabla f(x^{(k)}) \quad (14)$$

where $[\nabla^2 f(x^{(k)})]^{-1}$ is the reverse Hessian matrix $H(x^{(k)})$.

Substituting the expression of the formula (14) into the formula (13) item $x^{(k)}$ as in point $x^{(k+1)}$ of Newton method, we obtain the following expression (15).

$$x^{(k+1)} = x^{(k)} - [\nabla^2 f(x^{(k)})]^{-1} \cdot \nabla f(x^{(k)}) \quad (15)$$

But in general for a nonlinear objective function, the minimum of $f(x)$ is not obtained only for one position as equation (15) indicates it but is usually modified by introducing the parameter λ called the parameter calculation.

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} \cdot H^{-1}(x^{(k)}) \cdot \nabla f(x^{(k)}) \quad (16)$$

The expression (16) is called Newton iterative formula. A direction research is given as follows:

$$s^{(k)} = -H^{-1}(x^{(k)}) \cdot \nabla f(x^{(k)}) \quad (17)$$

In the Quasi-Newtonians method, the matrix of Hessian is replaced by its reverse.

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} \cdot \eta(x^{(k)}) \cdot \nabla f(x^{(k)}) \quad (18)$$

where $\eta(x^{(k)})$ an approximation of $H^{-1}(x)$ and often it is called matrix of direction. In the majority of the methods of

mathematical resolution $H^{-1}(x^{(k+1)})$ is approximated various information collected to k^{th} rank.

$$H^{-1}(x^{(k+1)}) \approx \omega \cdot \eta^{(k+1)} = \omega \cdot (\eta^{(k)} + \Delta \eta^{(k)}) \quad (19)$$

where:

η the approximate matrix de $H^{-1}(x)$.

$\eta(x^{(k)})$ a specific matrix.

ω a constant.

At the iteration $x^{(k+1)}$, we know $x^{(k)}$, $\nabla f(x^{(k)})$, $\eta(x^{(k)})$

and $\nabla f(x^{(k+1)})$, and we must seek the matrix of direction $\eta(x^{(k+1)})$ of the equation (15) while defining:

$$x^{(k+1)} - x^{(k)} = \Delta x^{(k)}$$

$$H^{-1}(x^{(k+1)}) \cong \omega \cdot \eta^{(k+1)} \cong \omega \cdot (\eta^{(k)} + \Delta \eta^{(k)})$$

$$\nabla f(x^{(k+1)}) - \nabla f(x^{(k)}) = \Delta g^{(k)}$$

We deduce the following relation:

$$\Delta \eta^{(k)} = \frac{1}{\omega} \frac{\Delta x^{(k)} y^T}{y^T \Delta g^{(k)}} - \frac{\eta^{(k)} \Delta g^{(k)} z^T}{z^T \Delta g^{(k)}} \quad (20)$$

y and z are arbitrary vectors of dimensions (N x 1).

VI. PROCEDURE OF RESOLUTION

A. Method of Broyden

$$\omega = 1, y = z = \Delta x^{(k)} - \eta^{(k)} \cdot \Delta g^{(k)}$$

The matrix of direction is:

$$\eta^{(k+1)} = \eta^{(k)} + \frac{\left[\Delta x^{(k)} - \eta^{(k)} \Delta g^{(k)} \right] \left[\Delta x^{(k)} - \eta^{(k)} \Delta g^{(k)} \right]^T}{\left[\Delta x^{(k)} - \eta^{(k)} \Delta g^{(k)} \right]^T \Delta g^{(k)}} \quad (21)$$

B. Method of Davidon-fletcher-Powell

$$\omega = 1, y = \Delta x^{(k)} \text{ et } z = \eta^{(k)} \cdot \Delta g^{(k)}$$

The matrix of direction is:

$$\eta^{(k+1)} = \eta^{(k)} + \frac{\Delta x^{(k)} (\Delta x^{(k)})^T}{(\Delta x^{(k)})^T \Delta g^{(k)}} - \frac{\eta^{(k)} \Delta g^{(k)} [\eta^{(k)} \Delta g^{(k)}]^T}{(\Delta g^{(k)})^T \eta^{(k)} \Delta g^{(k)}} \quad (22)$$

C. Method of Pearson

$$\omega = 1, y = z = \eta^{(k)} \Delta g^{(k)}$$

The matrix of direction is:

$$\eta^{(k+1)} = \eta^{(k)} + \frac{\left[\Delta x^{(k)} - \eta^{(k)} \Delta g^{(k)} \right] \left[\Delta x^{(k)} \right]^T}{(\Delta g^{(k)})^T \eta^{(k)} \Delta g^{(k)}} \quad (23)$$

D. Method of Zoutendijk

$$\omega \rightarrow \infty, y = z = \eta^{(k)} \Delta g^{(k)}$$

The matrix of direction is:

$$\eta^{(k+1)} = \eta^{(k)} - \frac{\left[\eta^{(k)} \Delta g^{(k)} \right] \left[\eta^{(k)} \Delta g^{(k)} \right]^T}{(\Delta g^{(k)})^T \eta^{(k)} \Delta g^{(k)}} \quad (24)$$

VII. HESSIANS METHOD ALGORITHM

Defining $x^{(k)} = P_G^k$ and $H^{-1}(x^{(k)}) = [\nabla^2 f(x^{(k)})]^{-1} = \eta(x^{(k)})$

Step 1: To choose P_G^0 , and η^0 stamps approximate Hessian definite positive.

(We take $\eta^0 = I$, stamp unit $k = 0$)

Step 2: Determination of the direction of research

$$s^{(k)} = -H^{-1}(x^{(k)}) \nabla f(x^{(k)})$$

Step 3: Determination of $P_G^{(k+1)} = P_G^{(k)} + \lambda^{(k)} \cdot s^{(k)}$

λ will be selected such as $P_G^{(k+1)}$ is a value acceptable for the next iteration. One uses for that a linear research.

Step 4: Determination of $\Delta P_G^{(k)} = P_G^{(k+1)} - P_G^{(k)}$ and to calculate $\Delta g^{(k)} = \nabla f(P_G^{(k+1)}) - \nabla f(P_G^{(k)})$

Step 5: Calculation of the matrix of direction

$$\eta^{(k+1)} = \eta^{(k)} + \frac{1}{\omega} \frac{\Delta x^{(k)} y^T}{y^T \Delta g^{(k)}} - \frac{\eta^{(k)} \Delta g^{(k)} z^T}{z^T \Delta g^{(k)}}$$

Step 6: Test of stop $|f(P_G^{(k+1)}) - f(P_G^k)| \leq \varepsilon$ so not to go at step 2.

VIII. GENETIC ALGORITHMS (GAS)

The Genetic Algorithms (GAs) [1, 2, 12] are algorithms of stochastic optimization founded on the mechanisms of the natural selection and the genetics. Their operation is extremely simple. One leaves with a population potential solutions (chromosomes) initial arbitrarily selected. One evaluates their performance (fitness) relative. On the basis of these performances one creates a new population of potential solutions by using simple operator's evolutionnaires: **selection**, **crossover** and **the mutation**. One starts again this cycle until one finds satisfactory a solution. The simplicity of

their mechanisms, the facility of their application and their effectiveness even for complex problems led to an increasing number of works these last years [7, 8]. Such algorithms were developed since 1950 by the biologists who used already computers to simulate the evolution of the organizations and were adapted by John Holland and al. (1975) and Goldberg (1994) for the search for solutions to problems optimization, by developing an analogy between an individual in a population and the solution of a problem in a whole of solutions.

IX. GUIDING PRINCIPLE OF STANDARD GA

The principle of AG is simple; it acts to simulate the evolution of a population of individuals until a criterion of stop. One starts by generating initial population individuals (solutions) in a random way. Then, with each generation, of the individuals are selected, this selection is carried out starting from a function objective called function of adaptation. Then, the operators of crossing and mutation are applied and a new population is created. This process is reiterated until a criterion of stop. The criterion most usually used is the maximum number of generations which one wishes to carry out. Fig. 1 presents the principle of standard AG [12].

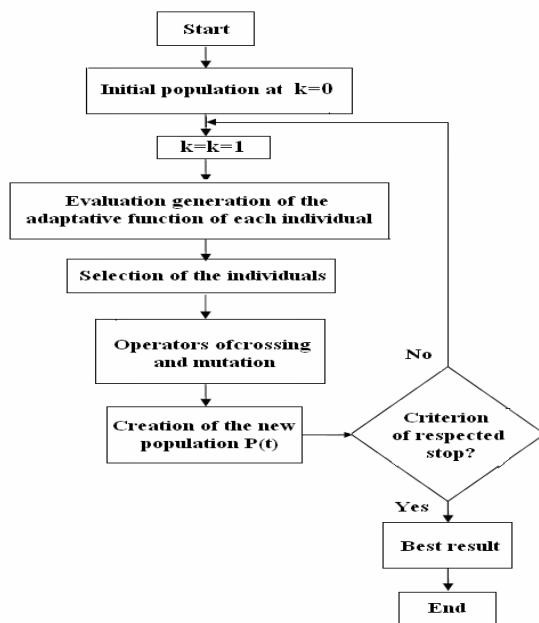


Fig. 1 Flow chart of standard GA

AG begins with the generation from an initial population and the evaluation from the function from adaptation from all the individuals who make this first population. Then, of the individuals are selected by chance for the reproduction according to the principle of the survival of more adapted. Then, of the individuals “children” (or descendants) are generated by applying the two following genetic operators: the crossing and the change. These children are placed in a new population P (T) and will substitute themselves, in all or partly, with the population of the preceding generation. New populations of individuals then will follow one another, of a

generation (k) to the generation (k+1), each generation representing an iteration until the attack of the criterion of stop. AG presented above is known as generational because all the individuals’ generated children are placed in a population and will entirely replace the population of the individuals’ parents.

X. APPLICATION EXAMPLE

The application was made on a network of 14 nodes [12] with two generating nodes G_1 and G_2 as shown in Fig. 2.

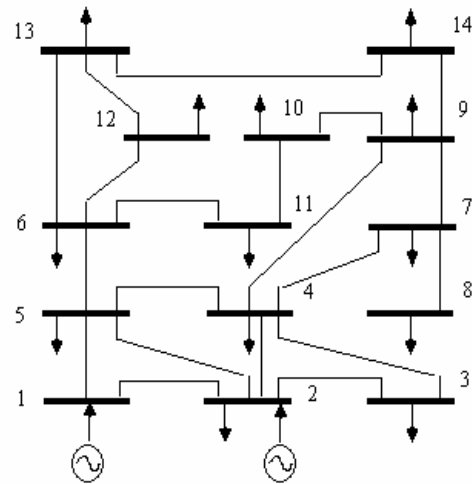


Fig. 2 Network with two generating nodes G_1 and G_2

The objective functions of the two generators are written in the following way:

$$F_1(P_{G1}) = 0.006P_{G1}^2 + 1.5P_{G1} + 100$$

$$F_2(P_{G2}) = 0.009P_{G2}^2 + 2.1P_{G2} + 130$$

Under the constraints of the equality type:

$$P_{G1} + P_{G2} - P_{ch} - P_L = 0$$

Under the constraints of the inequality type:

$$135 \leq P_{G1} \text{ (MW)} \leq 195$$

$$70 \leq P_{G2} \text{ (MW)} \leq 145$$

The active total losses are calculated by the method of Gauss-Seidel [6, 8] and are maintained constant. They are equal to 18.15 (MW). The total powers of load are fixed as follows.

$$P_{ch} = 259 \text{ (MW)}$$

The constraint of the equality type is written as.

$$P_{G1} + P_{G2} - 277.15 = 0$$

The values of bases are:

$$U_b = 220 \text{ (MW)} \quad S_b = 100 \text{ (MVA)}$$

XI. RESULTS AND DISCUSSION

A. Hessian's Methods

For the Hessian's methods took account of the constraints of inequalities of the active powers and we varied the initial values $P_{gi}^{(0)}$ (Table I) to observe their influences on the optimal results. The results are gathered in Tables II, III, IV, V for the optimal generated active powers, the costs of minimum fuel and the computing time (iteration numbers) for the various methods.

TABLE I
INITIAL POWER

$P_{G1}^{(0)}$ MW	$P_{G2}^{(0)}$ MW
150	125
180	85
155	120
170	100
140	135
135	70

TABLE II
METHOD OF BROYDEN

P_{G1}^{opt} MW	P_{G2}^{opt} MW	$F^{(opt)}$ (\$/h)	No. of iterations
153.41	124.94	1001.40	13
187.68	96.79	1010.46	13
157.69	119.92	997.03	13
174.74	106.13	999.59	13
144.86	132.96	1011.53	13
194.94	130.75	1178.88	13

TABLE III
METHOD OF DFP

P_{G1}^{opt} MW	P_{G2}^{opt} MW	$F^{(opt)}$ (\$/h)	No. of iterations
159.66	116.87	990.80	15
181.31	89.92	960.83	15
185.77	90.52	979.59	15
171.33	101.91	970.62	15
151.79	125.66	1001.97	15
152.99	87.75	853.51	15

TABLE IV
METHOD OF PEARSON

P_{G1}^{opt} MW	P_{G2}^{opt} MW	$F^{(opt)}$ (\$/h)	No. of iterations
164.97	107.91	972.17	15
172.85	104.66	986.94	15
162.93	110.14	974.17	15
167.93	108.46	984.76	15
164.07	108.59	971.80	15
167.77	108.51	984.44	15

TABLE V
METHOD OF ZOUTENDIJK

P_{G1}^{opt} MW	P_{G2}^{opt} MW	$F^{(opt)}$ (\$/h)	No. of iterations
154.52	123.97	1003.74	13
172.88	102.74	979.44	13
157.53	119.74	995.72	13
172.69	103.33	981.07	13
150.93	130.53	1020.55	13
170.36	105.57	981.73	13

B. Genetic Algorithm

The operators of the genetic algorithm are guided by a certain parameter numbers fixed in advance. The value of these parameters influences the success or not genetic algorithm. These parameters are shown:

- Size of the population, N_R , and the length of the coding of each individual L (in the case of the binary coding). If N_R is too large; the computing time of the algorithm can prove very important, and if N_R 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.

- Probability of crossing P_C .

It depends on the form of the function of fitness. Its choice is in general heuristic (just like for P_m).

The higher it is, the more the population sudden of important changes. The generally allowed values lie between 0.5 and 0.9.

- Probability of mutation P_m .

This rate is generally weak since a high rate is likely to lead to a solution under optimal.

We fixed the parameters of the genetic algorithm [4]:

- A number of generations = 100
- Size of population = 70

- Parameter $P_C = 0.6$
- Parameter $P_m = 0.2$

The initial generated powers are fixed by the genetic algorithm [5]:

$$P_{G1} = 135.00 \text{ (MW)}$$

$$P_{G2} = 70.00 \text{ (MW)}$$

We gathered the generated powers and the performance indexes according to the generation numbers in Table VI.

TABLE IV

No. of iterations	performance index (\$/hour)	P_{G1}	P_{G2}
0	1005.0056	135.0000	70.0000
1	983.9025	146.6750	130.4750
2	982.9234	171.1950	105.9550
3	982.8538	168.8450	108.3050
4	982.8059	164.3440	112.8060
5	982.7687	186.3660	90.7839
8	982.7648	186.3221	90.8277
10	982.7648	186.2901	90.8599
12	982.7646	186.2900	90.8600
13	982.7646	186.2900	90.8600

The generated powers and the optimal performance index are:

$$P_{G1} = 186.2900 \text{ (MW)}$$

$$P_{G2} = 90.8600 \text{ (MW)}$$

$$F_{Cost} = 982.7646 \text{ (\$/hour)}$$

The convergence of the performance index according to the number of generation is illustrated by the Fig. 3.

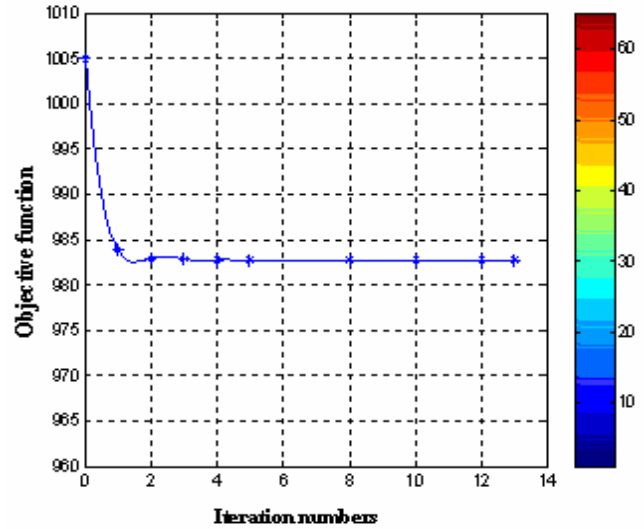


Fig. 3 Convergence of the performance index

In the same way we represented the convergence of the powers generated of the 1st and the 2nd generator are illustrated by the Figs. 4 and 5.

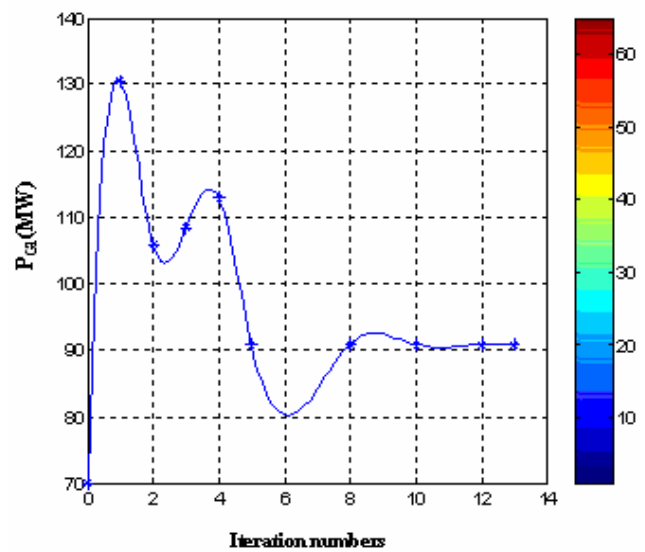


Fig. 4 Convergence of the generated power of the 1st generator

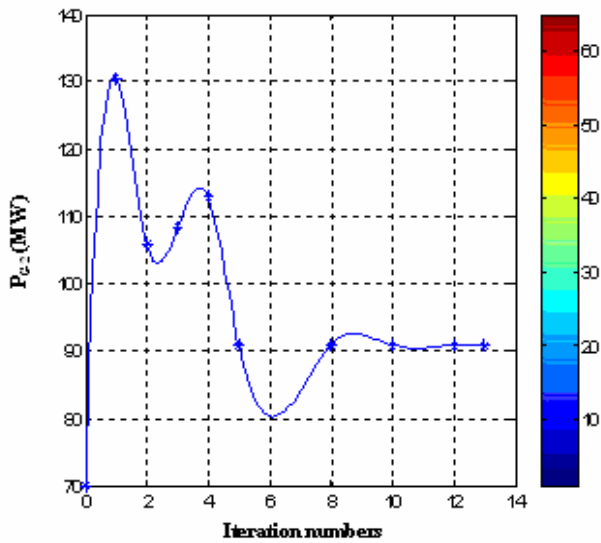


Fig.5. Convergence of the generated power of the 2nd generator

The envelope of variation of generated powers P_{G1} and P_{G2} is given in the Fig. 6.

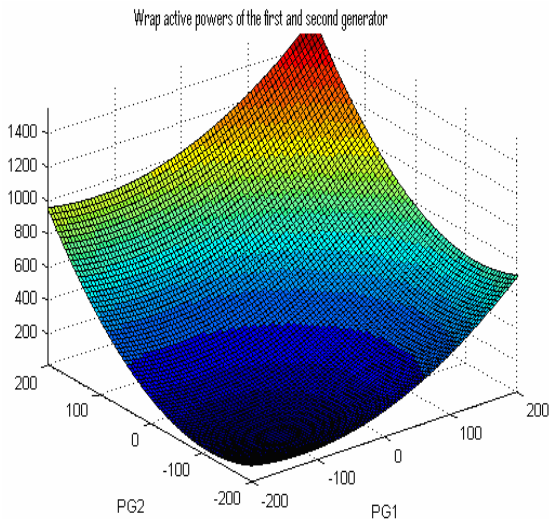


Fig. 6 Wrap variation of the generated powers

XII. CONCLUDING REMARKS

The comparative study of the Algorithms genetics and the Hessian's methods rest on two aspects:

- Mathematical approach.
- Found results.

For the mathematical aspect and concerning the Hessian's methods, the optimization of the electric production cost of power (in our case seeks minimum) is obtained under two conditions:

-The function of the cost must be continuous and two (2) derivable order.

-The matrix Hessian's (matrix of the direction of research) must be definite positive. What is not the case of the genetic Algorithms which hold account much more in optimal research, of the probabilities of Pm change and of crossing PC which influence the success or not in the search for optimal solutions. For the found results and concerning the Hessian's methods we can affirm immediately that the production cost is well minimized. The low value of the production cost (960.83 \$/hour) is given the method of Davidon-Fletcher-Powell while the other methods generate satisfactory costs. The highest cost is given by the method of Broyden. The computing time which interests us more (iteration count) is very small as well for the genetic algorithms for the Hessian's methods. It varies from 13 iterations for the method of Broyden with 15 iterations for the method of DFP and Pearson. This is very interesting taking into account the fact that we make a real-time control. In the same way, we noticed (concerning the values of the generated powers), that the difference between the optimal end values is a function of the initial values chosen by the person in charge for the economic control system. One can thus affirm that the person in charge for the economic control system will have to make a judicious choice for the initial values.

The results found by the application of AG for the minimization of the electric production costs of power are very encouraging. The algorithms seem to be an effective technique to solve a great number of problems and which are in constant evolution. The curves plotted by simulation (on PC Pentium III under Matlab environment) illustrate perfectly the optimal convergence of the performance index as well as the active generated powers. We did not represent the curves of illustration of convergence of the reactive powers owing to the fact that they do not intervene (or partially) in the performance index (objective function). The optimal values of the generated powers and the production cost differ slightly from those found by the Hessian's methods.

$$P_{G1} = 186.2900 \text{ (MW)}$$

$$P_{G2} = 90.8600 \text{ (MW)}$$

$$F_{Cost} = 982.7646 \text{ (\$/hour)}$$

Genetic algorithm is been applied of the electric power production. A computer program as a chromosome is a subject of genetic operators such as recombination, crossover and mutation. It gives possibility to represent knowledge that is specific to the problem in more intelligent way than for the data structure. That is, we process the potential warps of founding solution not the possible solution.

REFERENCES

- [1] Holland J.H., "Adaptation in natural and artificial system", Ann Arbor, The University of Michigan Press, 1975.
- [2] D.E Goldberg, "Genetic Algorithm in Search Optimization and Machine Learning", Addison Wesley 1994.
- [3] D.M Himmelblau, "Applied non linear programming □Edition Mc Graw-Hill, 1972
- [4] M. Minoux: "Programmation mathématique: Théorie et algorithmes Tome 1" Edition Dunod, 1983.

- [5] J.C Dodu et P. Huard: "Méthodes quasi-newtoniennes sous contraintes non linéaires "Bulletin de la direction des études de recherche, Electricité de France, série C, N°2, 1988.
- [6] M. Rahli et P. Pirotte, "Dispatching économique par une nouvelle méthode de programmation non linéaire à la répartition économique des puissances actives dans un réseau d'énergie électrique", CIMASI'96 Casablanca, Maroc, 14-16 Novembre 1996, pp325-330.
- [7] T. Vallée et M. Yildizoglu, "Présentation des algorithmes génétiques et de leurs applications en économie. " 7 septembre 2001, v. 1.2
- [8] Z. Michalewicz, et N.F. Attia, "Evolutionary optimization of constrained problems", Proceedings of the 3rd Annual Conference on Evolutionary Programming, World Scientific, pp. 98–108.
- [9] M. Rahli, "Contribution à l'Etude de la Répartition Optimale des Puissances Actives dans un Réseau d'Energie Electrique", thèse de doctorat, 06 janvier 1996, USTO. Algérie
- [10] L. Abdelmalek, "Répartition Optimale des Puissances Actives et Réactives par les méthodes Hessiennes", CIMASI'2002, Casablanca 22-25 Octobre 2002. Maroc.
- [11] M. Rahli et P. Pirotte, "Optimal load flow using sequential unconstrained minimization technique SUMT method under power transmission losses minimization", Electric Power Research, 1999 Elsevier Science.
- [12] R. Ouiddir et M. Rahli, "Dispatching Economique Actif dans un Réseau d'Energie Electrique par un Algorithme Génétique", 2nd International Conférence on Electrotechnics, 13-15 Novembre2000, ICEL2000, USTOran, Algérie.
- [13] L. Drdi, Extrait du 1er chapitre de la thèse de doctorat INRS-ETE, 2005.