

Optimal Power Flow Based on Hybrid Genetic Algorithm

M. YOUNES, M. RAHLI AND L. ABDELHAKEM-KORIDAK

*Power Systems Optimization Laboratory
Faculty of Electrical Engineering
University of Science and Technology of Oran
Oran El M'naouer, Oran, 31000 Algeria
E-mail: {younesmi; rahlim; hkoridak}@yahoo.fr*

Optimal Power Flow (OPF) is one of the main functions of Power Generation Operation and Control. It determines the optimal settings of generating units, bus voltage, and transformer tap and shunt elements in Power System with the objective of minimizing total production costs or losses while the system is operating within its security limits. The aim of this paper is to propose a methodology (GA-Matpower-OPF) that solves OPF including both active and reactive power dispatches. It is based on combining the Genetic Algorithm (GA) to obtain a near global solution, and the package of Matlab m-files for solving power flow and optimal power flow problems (Matpower) to determine the optimal global solution. This method was tested on the modified IEEE 57 bus test system. The results obtained by this method are compared with those obtained with GA or Matpower separately.

Keywords: optimal power flow, genetic algorithm, power systems, economic dispatch, Matpower software

1. INTRODUCTION

OPF [1] is a nonlinear programming problem, and is used to determine optimal outputs of generators, bus voltage and transformer tap, setting in power system, with an objective to minimize total production cost. While the system is operating within its security limit. Since OPF was introduced in 1968 [2], several methods have been employed to solve this problem, *e.g.* Gradient base[2], Linear programming method [3] and Quadratic programming [4]. However all of these methods suffer from three main problems. Firstly, they may not be able to provide optimal solution and usually getting stuck at a local optimal. Secondly, all these methods are based on assumption of continuity and differentiability of objective function which is not actually allowed in a practical system. Finally, all these methods can not be applied with discrete variables, which are transformer taps.

It seems that GA is an appropriate method to solve this problem, which eliminates the above drawbacks. GA, invented by Holland [5] in the early 1970s, is a stochastic global search method that mimics the metaphor of natural biological evaluation.

Genetic Algorithms (GAs) operates on a population of candidate solutions encoded to finite bit string called chromosome. In order to obtain optimality, each chromosome exchanges the information using operators borrowed from natural genetic to produce the

Received September 23, 2005; revised November 2, 2006; accepted January 9, 2007.
Communicated by John Yen and H. Y. Mark Liao.

better solution. GAs differs from other optimization and search procedures in four ways [6]:

- (1) GAs work with a coding of the parameter set, not the parameters themselves. Therefore GAs can easily handle the integer or discrete variables.
- (2) GAs search within a population of points, not a single point. Therefore GAs can provide a globally optimal solution.
- (3) GAs use only objective function information, not derivatives or other auxiliary knowledge. Therefore GAs can deal with the non-smooth, non-continuous and non-differentiable functions which are actually exist in a practical optimization problem.
- (4) GAs use probabilistic transition rules, not deterministic rules, Although GAs seem to be a good method to solve optimization problem, sometimes the solution obtained from GAs is only a near global optimum solution.

Matpower is a package of Matlab files for solving power flow and optimal power flow problems. It is a simulation tool for researchers and educators and it is easy to use and modify. Matpower is designed to give the best performance possible while keeping the code simple to understand and modify [9]. Matpower was developed by Ray Zimmerman and Deqiang Gan of PSERC at Cornell University under the direction of Robert Thomas. Therefore this paper employs Matpower applied with GAs to obtain the global solution. Basically, this method can be divided into two parts. The first part employs GA to obtain a near global solution, while the other part employs Matpower to determine the solution closer to the global solution. This method was tested on the modified IEEE 57 bus test system. The result of the study of this method is compared with those obtained from GA or Matpower separately.

The remainder of the paper is organized as follows. In section 2, we present the optimal power flow formulation and the optimisation under equality and inequality constraints. This is followed by an explanation of the proposed methodology in section 3. Simulation results are shown and discussed in section 4. Finally, we conclude in section 5.

2. OPTIMAL POWER FLOW FORMULATION

The OPF problem is to minimize the objective function, fuel cost, while satisfying several equality and inequality constraints. Generally the problem is formulated as follows [14].

$$\text{Min} \left\{ f(P_G) = \sum_{i=1}^{NG} f_i(P_{Gi}) \right\} \quad (1)$$

In Eq. (1), the generation cost function $f_i(P_{Gi})$ in \$/h is usually expressed as a quadratic polynomial [15]

$$f_i(P_{Gi}) = a_i P_{Gi}^2 + b_i P_{Gi} + c_i \quad (2)$$

where

$f_i(P_G)$: Total production cost (\$/h); $f_i(P_{Gi})$ is the cost of the i th generator in US\$/h;

P_{Gi} : The power output of generator i in MW; a_i, b_i, c_i the cost coefficients of the i th generator.

In minimizing the cost, the equality constraint (power balance) and inequality constraint (power limits) should be satisfied.

2.1 Equality Constraint

$$\sum_{i=1}^{NG} P_{Gi} - \sum_{j=1}^{ND} P_{Dj} - P_L = 0 \quad (3)$$

$$\sum_{i=1}^{NG} Q_{Gi} - \sum_{j=1}^{ND} Q_{Dj} - Q_L = 0 \quad (4)$$

where

P_{Dj}, Q_{Dj} : active and reactive power generation at bus j ;

P_{Gi}, Q_{Gi} : active and reactive power generation at bus i ;

P_L, Q_L : real and reactive losses.

The transmission loss can be represented by the B-coefficient method [16] as

$$P_L = \sum_i \sum_j P_{Gi} B_{ij} P_{Gj} \quad (5)$$

where B_{ij} is the transmission loss coefficient, P_{Gi}, P_{Gj} the power generation of i th and j th units. The B-coefficients are found through the Z-bus calculation technique.

2.2 Inequality Constraint

The generation capacity of each generator has some limits and it can be expressed as

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \quad (6)$$

$$Q_{Gi}^{\min} \leq Q_{Gi} \leq Q_{Gi}^{\max} \quad (7)$$

$$V_{Gi}^{\min} \leq V_{Gi} \leq V_{Gi}^{\max} \quad (8)$$

where

$P_{Gi}^{\min}, P_{Gi}^{\max}$: Lower and upper limit of active power generation at bus i ;
 $Q_{Gi}^{\min}, Q_{Gi}^{\max}$: lower and upper limit of reactive power generation at bus i ;
 $V_{Gi}^{\min}, V_{Gi}^{\max}$: lower and upper limit of bus voltage at bus i ;
 N : number of bus; NG : number of generating buses; ND : number of demand buses.

3. METHODOLOGY

Traditionally, GA is a stochastic optimization method which starts from multiple points to obtain a solution, but it provides only a near global solution. On the other hand, Matpower searches from a single point and using gradient information to obtain a solution. However, the solution obtained from Matpower is normally a local optimum solution. Therefore, in order to obtain a high quality solution, the two parts method, comprising both GA and Matpower is proposed in this paper. Basic theory for the GA and Matpower will be explained in the subsequent sections. In the proposed method, after the specified termination criteria for the GA is reached, Matpower is applied in the second part by using the solution from GA as an initial point and search using gradient information to obtain a solution which is closer to the global solution.

3.1 Genetic Algorithm Subproblem

3.1.1 Chromosome coding and decoding

GAs work with a population of binary string instead the parameters themselves [19]. For simplicity and convenience, binary coding is used in this paper. With the binary coding method, the active generation power set of 57-bus test system; $P_{G1}, P_{G2}, P_{G3}, P_{G4}, P_{G5}, P_{G6}, P_{G7}$ (Table 6 and Fig. 2) would be coded as binary string (0 and 1) with length $L_1, L_2, L_3, L_4, L_5, L_6$ and L_7 (may be different), respectively. Each parameter P_{Gi} have upper bound $b_i (p_{Gi}^{\max})$ and lower bound $a_i (p_{Gi}^{\min})$. The choice of $L_1, L_2, L_3, L_4, L_5, L_6$ and L_7 for the parameters is concerned with the resolution specified by the designer in the search space. In the binary coding method, the bit length L_i and the corresponding resolution R_i are related by

$$R_i = \frac{b_i - a_i}{2^{L_i} - 1} \quad (9)$$

$$P_{Gi} = a_i + decimal \cdot (string_2) \cdot R_i. \quad (10)$$

As result, the P_{Gi} set can be transformed into a binary string (chromosome) with length $\sum L_i$ and then the search space is explored. Note that each chromosome presents one possible solution to the problem. For example, suppose the parameter domain of $((P_{G1}, P_{G2}, P_{G3}, P_{G4}, P_{G5}, P_{G6}, P_{G7}))$ which is presented in Table 1. If the resolution $(R_1, R_2, R_3, R_4, R_5, R_6, R_7)$ is specified as (38.392, 6.66, 9.33, 6.66, 36.66, 6.66, 27.33) and we have $L_1 = 4, L_2 = 4, L_3 = 4, L_4 = 4, L_5 = 4, L_6 = 4$ and $L_7 = 4$ (The number of the bits per gene is dependent on the required precision), Then the parameter set $((P_{G1}, P_{G2}, P_{G3}, P_{G4}, P_{G5}, P_{G6}, P_{G7})$ and $P_{G7})$ can be coded according to the following (Table 1):

Table 1. Coding of P_{Gi} parameter set.

	P_{G1}	P_{G2}	P_{G3}	P_{G4}	P_{G5}	P_{G6}	P_{G7}	code
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0000
1	38,392	6,66	9,33	6,66	36,66	6,66	27,33	0001
2	76,784	13,32	18,66	13,32	73,32	13,32	54,66	0010
3	115,176	19,98	27,99	19,98	109,98	19,98	81,99	0011
4	153,568	26,64	37,32	26,64	146,64	26,64	109,32	0100
5	191,96	33,3	46,65	33,3	183,3	33,3	136,65	0101
6	230,352	39,96	55,98	39,96	219,96	39,96	163,98	0110
7	268,744	46,62	65,31	46,62	256,62	46,62	191,31	0111
8	307,136	53,28	74,64	53,28	293,28	53,28	218,64	1000
9	345,528	59,94	83,97	59,94	329,94	59,94	245,97	1001
10	383,92	66,6	93,3	66,6	366,6	66,6	273,3	1010
11	422,312	73,26	102,63	73,26	403,26	73,26	300,63	1011
12	460,704	79,92	111,96	79,92	439,92	79,92	327,96	1100
13	499,096	86,58	121,29	86,58	476,58	86,58	355,29	1101
14	537,488	93,24	130,62	93,24	513,24	93,24	382,62	1110
15	575.88	100.00	140.00	100.00	550.00	100.00	410.00	1111

If the candidate parameters set is (460.70, 53.33, 102.66, 6.66, 403.33, 13.33, 246.00), then the chromosome is a binary string 1100100010110001101100101001. The decoding procedure is the reverse procedure. The first step of any genetic algorithm is to create an initial population of GA by randomly generating a set of feasible solutions [17]. 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 an intermediate population in order to obtain a new one, this process is called.

The process, that starts from the present population and leads to the new population, is called a generation process (Table 2).

Table 2. First generation of GA process for 57 bus example.

Initial Population	P_{G1}	P_{G2}	P_{G3}	P_{G4}	P_{G5}	P_{G6}	P_{G7}	$f(P_{Gi})$	$f_{max}-f(P_{Gi})$
001000110010110011011010000	76,784	19,98	18,66	79,92	476,58	86,58	0.00	1067300	293200
011110001011111000010110111	268,744	53,28	102,63	100.00	0.00	73,26	191,31	915730	444770
101001111111010000100100000	383,92	46,62	140.00	66,6	36,66	13,32	0.00	1360500	0.00
0110010101100100110000111110	230,352	33,3	55,98	26,64	439,92	19,98	382,62	30509.80	1329991
Max									1360500
Min									30509.80

3.1.2 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

Table 3. Single point crossovers.

Locus = 3			
Parent1	011 0010101100100110000111110	Child 1	0111100010111111000010110111
Parent2	011 1100010111111000010110111	Child 2	0110010101100100110000111110

Table 4. The result after crossover and mutation of the first population.

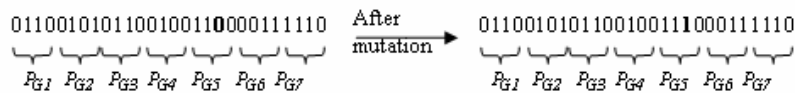
Chromosome	the result after crossover of the first population	the result after mutation of the first population
1	0110010101100100110000111110	0110010101100100110000111110
2	0111100010111111000010110111	0111100011111111000010110111
3	0110010101100100110000111110	0110010101010100110010111111
4	0111100010111111000010110111	0111100010111111111010110111

at random locus, *i.e.* Single Point Crossover (Table 3) [15]. The segments of bits from the parents behind the crossover point are exchanged with each other to generate their off-spring. The mixture is performed by choosing a point of the strings randomly and switching the left segments of this point. The new strings belong to the next generation of possible solutions (Table 4). The strings to be crossed are selected according to their scores using the roulette wheel [6]. 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.

3.1.3 Mutation

Mutation is a secondary operator; it prevents the premature stopping of the algorithm in a local solution. This operator is defined by a random bit value change in a chosen string with a low probability. The mutation adds a random search character to the genetic algorithm (Table 4).

All strings and bits have the same probability of mutation. For example, in the string 0110010101100100110000111110, if the mutation affects bit number six, the string obtained is 0110010101100100111000111110 and the value of P_{GS} change from 439.92 to 513,24.



3.1.4 Reproduction

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

Table 5. Second generation of GA process for 57 bus example.

second generation	P_{G1}	P_{G2}	P_{G3}	P_{G4}	P_{G5}	P_{G6}	P_{G7}	$f(P_{Gi})$	$fmax - f(P_{Gi})$
01100101011001001110001111110	230,35	33,3	55,98	26,6	513,2	19,98	382,62	5503.1	882116,9
0111100011111111000010110111	268,74	53,28	140,00	100,0	0,00	73,26	163,98	887620	0,00
0110010101010100110010111111	230,35	33,3	46,65	26,64	439,92	73,26	410,00	4985.56	882634,44
011110001011111111010110111	268,74	53,28	102,63	100,0	513,24	73,26	191,31	8015.6	879604,4
Max									887620
Min									4985.56

3.1.5 Candidate solutions fitness and cost function

The cost function is defined as:

$$f_i(P_{Gi}) = a_i P_{Gi}^2 + b_i P_{Gi} + c_i; \quad P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}. \quad (11)$$

Our objective is to search ($P_{G1}, P_{G2}, P_{G3}, P_{G4}, P_{G5}, P_{G6}, P_{G7}$) in their admissible limits to achieve the optimisation problem of OPF. The cost function $f(P_{Gi})$ takes a chromosome (a possible ($P_{G1}, P_{G2}, P_{G3}, P_{G4}, P_{G5}, P_{G6}, P_{G7}$)) and returns a value. The value of the cost is then mapped into a fitness value $fit(P_{G1}, P_{G2}, P_{G3}, P_{G4}, P_{G5}, P_{G6}, P_{G7})$ so as to fit in the genetic algorithm.

To minimise $f(P_{Gi})$ 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 OPF should change to the maximisation of fitness used in the simulated roulette wheel as follows:

$$fitness_i = \begin{cases} fmax - fi(P_{Gi}), & \text{if } fmax \geq fi(P_{Gi}); i = 1, NG, \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

It should be given by the slack generator with considering different reactive constraints. Examples of reactive constraints are the min and the max reactive rate of the generators buses and the min and max of the voltage levels of all buses [15]. All these require a fast and robust load flow program with best convergence properties. The developed load flow process is based upon the full Newton-Raphson algorithm using the optimal multiplier technique [18].

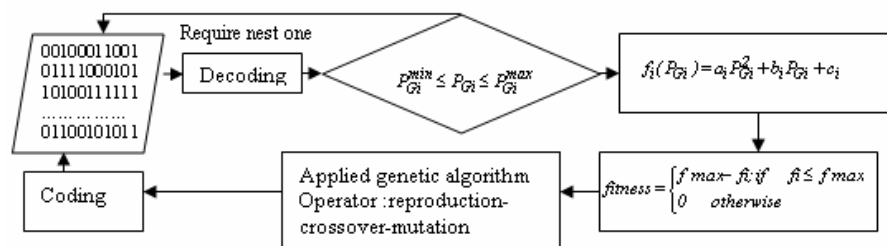


Fig. 1. A Simple flow chart of the GAOPF.

3.1.6 Termination of the GA

Because GA is a stochastic search method, it is difficult to formally specify convergence criteria. As the fitness of a population may remain static for a number of generations before a superior individual is found, the application of convergence termination criteria becomes problematic, a common practice is to terminate GA after a prespecified number of generations (in our case the number of generations is 300) and then test the fitness of best members in the last population. If no acceptable solutions are found, the GA may be restarted or fresh search initiated [19].

3.2 Matpower Subproblem

3.2.1 Introduction

Matpower is a package of Matlab *m*-files for solving power flow and optimal power flow problems [9].

3.2.2 Power flow

Matpower has three power flow solvers. The default power flow solver is based on a standard Newton's method [7] using a full Jacobian, updated at each iteration. This method is described in detail in many textbooks. The other two power flow solvers are variations of the fast-decoupled method [8]. Matpower implements the XB and BX variations as described in [10]. Currently, Matpower's power flow solvers do not include any transformer tap changing or feasibility checking capabilities.

Performance of the power flow solvers should be excellent even on very large-scale power systems, since the algorithms and implementation take advantage of Matlab's built-in sparse matrix handling. On a Sun Ultra 2200, Matpower solves a 9600-bus test case in about 10 seconds and a 38400 bus case in about 50 seconds.

3.2.3 Optimal power flow

Matpower includes two solvers for the optimal power flow (OPF) problem. The first is based on the *constr* function included in Matlab's Optimization Toolbox, which uses a successive quadratic programming technique with a quasi-Newton approximation for the Hessian matrix. The second approach is based on linear programming. It can use the LP solver in the Optimization Toolbox or other Matlab LP solvers available from third parties.

The performance of Matpower's OPF solvers depends on several factors. First, the *constr* function uses an algorithm which does not exploit or preserve sparsity, so it is inherently limited to small power systems. The LP-based algorithm, on the other hand, preserve sparsity. However, the LP-solver included in the Optimization Toolbox does not exploit this sparsity. In fact, the LP-based method with the default LP solver performs worse than the *constr*-based method, even on small systems. Fortunately, there are LP-solvers available from third parties which do exploit sparsity. In general, these yield *much* higher performance. One in particular, called *bpmpd* [11] (actually a QP-solver),

has proven to be robust and efficient.

It should be noted, however, that even with a good LP-solver, Matpower's LP-based OPF solver, unlike its power flow solver, is not suitable for very-large scale problems. Substantial improvements in performance may still be possible, though they may require significantly more complicated coding and possibly a custom LP-solver. On a Sun Ultra 2200, the LP-based OPF solver using *bpmpd* solves a 30-bus system in less than 4 seconds and a 118-bus case in less than 25 seconds.

3.2.4 OPF formulation

The OPF problem solved by Matpower is a "smooth" OPF with no discrete variables or controls. The objective function is the total cost of real and/or reactive generation. These costs may be defined as polynomials or as piecewise-linear functions of generator output. The problem is formulated as follows:

$$\min \sum f_{1i}(P_{Gi}) + f_{2i}(Q_{Gi}) \quad (13)$$

Such that

$$P_{Gi} - P_{Li} - P(V, \theta) = 0 \quad (\text{Active power balance equations}) \quad (14)$$

$$Q_{Gi} - Q_{Li} - Q(V, \theta) = 0 \quad (\text{Reactive power balance equations}) \quad (15)$$

$$S_{ij}^f \leq S_{ij}^{\max} \quad (\text{Apparent power flow limit of lines, from side}) \quad (16)$$

$$S_{ij}^t \leq S_{ij}^{\max} \quad (\text{Apparent power flow limit of lines, to side}) \quad (17)$$

$$V_i^{\min} \leq V_i \leq V_i^{\max} \quad (\text{Bus voltage limits}) \quad (18)$$

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \quad (\text{Active power generation limits}) \quad (19)$$

$$Q_{Gi}^{\min} \leq Q_{Gi} \leq Q_{Gi}^{\max} \quad (\text{Reactive power generation limits}) \quad (20)$$

Here f_{1i} and f_{2i} are the costs of active and reactive power generation, respectively, for generator i at a given dispatch point. Both f_{1i} and f_{2i} are assumed to be a polynomial or piecewise-linear functions. The problem can be written more compactly in the following form:

$$\min f(x). \quad (21)$$

Such that

$$g(x) \leq 0 \quad (22)$$

where f and g are non-linear functions.

3.2.5 Optimization toolbox based OPF solver (constr)

The first of the two OPF solvers in Matpower is based on the `constr` non-linear constrained optimization function in Matlab's Optimization Toolbox. The `constr` function and the algorithms it uses are covered in the Optimization Toolbox manual [12]. Matpower provides `constr` with two *m*-files which it uses during for the optimization. One computes the objective function, f , and the constraint violations, g , at a given point, x , and the other computes their gradients $\frac{\partial f}{\partial x}$ and $\frac{\partial g}{\partial x}$.

Matpower has two versions of these *m*-files. One set is used to solve systems with polynomial cost functions. In this formulation, the cost functions are included in a straightforward way into the objective function. The other set is used to solve systems with piecewise-linear costs. Piecewise linear cost functions are handled by introducing a cost variable for each piecewise-linear cost function.

The objective function is simply the sum of these cost variables which are then constrained to lie above each of the linear functions which make up the piecewise-linear cost function. Clearly, this method works only for convex cost functions. In the Matpower documentation this will be referred to as a constrained cost variable (CCV) formulation.

The algorithm codes 100 and 200, respectively, are used to identify the `constr`-based solver for polynomial and piecewise-linear cost functions. If algorithm 200 is chosen for a system with polynomial cost function, the cost function will be approximated by a piecewise-linear function by evaluating the polynomial at a fixed number of points determined by the options vector.

3.2.6 Unit decommitment algorithm

Matpower includes a unit decommitment algorithm which allows it to shut down these expensive units. The algorithm is based on a simplified version of the decommitment technique proposed in [13].

The algorithm proceeds as follows:

Step 0: Assume all generators are on-line with all generator limits in place.

Step 1: Solve a normal OPF.

Step 2: If the OPF converged to a feasible solution and the objective function decreased from the previous iteration (or if this is the first iteration), go to step 3, otherwise go to step 4.

Step 3: Compute a decommitment index for each generator i as follows:

$$d_i = f_i(P_i) - \lambda_i P_i \quad (23)$$

where P_i is generator i 's dispatch computed by the OPF, f_i is the cost of operating at P_i , and λ_i is the Lagrange multiplier on the real power equality constraint at the bus where generator i is located. Continue with step 5.

Step 4: Return to the previous commitment and set d_k to zero (to eliminate it from consideration).

Step 5: Find the generator k with the smallest decommitment index. If d_k is negative, shut down generator k and return to step 1. If d_k is positive, stop.

3.3 Computational Procedure

The software was developed using the Matlab GA toolbox and Optimization toolbox to implement GA and Matpower respectively. This software can run in Matlab5 environment:

Step 1: Read system data.

Step 2: Solve OPF problem using GA.

Step 3: Use answer from step 2 as a starting point and solve OPF problem using Matpower.

Step 4: Getting the final result and quitting program.

4. SIMULATION RESULTS

In this study, the standard IEEE 57-bus 7-generator test system is considered to investigate the effectiveness of the proposed approach. The IEEE 57 bus system has 80 transmission circuits. The single-line diagram of this system is shown in Fig. 2 and the detailed data are given in [9]. The values of fuel cost coefficients are given in Table 6, Total load demand of the system is 1250.8 (MW), and 7 generators should satisfy this load demand economically. The result obtained from Matpower method, GA method and the proposed method are shown in Tables 8-10. This method has been tested 30 times. Three test cases are considered, specifically, the first test case ignores the transmission losses. The second test case differs from the first in that it incorporates the transmission losses. The transmission line losses are calculated and maintained constant ($P_L = 19.06$ MW).

Finally, in the third test case, we consider the variable losses according to each method. The parameter values used for GA are in Table 7. One string represents 7 generators outputs that should be dispatch. This string is coded in binary format of 149 bits length (23 bits + 20 bits + 21 bits + 20 bits + 23 bits + 20 bits + 22 bits = 149 bits).

Table 6. Generator operating limits and quadratic cost function coefficients data (IEEE-57 bus).

Bus	Pmin	Pmax	Qmin	Qmax	Vmin	Vmax	a	b	c
1	0.00	575.88	-200	300	0.94	1.06	0.01	0.30	0.20
2	0.00	100.00	-17	50	0.94	1.06	0.01	0.30	0.20
3	0.00	140.00	-10	60	0.94	1.06	0.01	0.30	0.20
6	0.00	100.00	-8	25	0.94	1.06	0.01	0.30	0.20
8	0.00	550.00	-140	200	0.94	1.06	0.01	0.30	0.20
9	0.00	100.00	-3	9	0.94	1.06	0.01	0.30	0.20
12	0.00	410.00	-150	155	0.94	1.06	0.01	0.30	0.20

Table 7. Parameter values for GA & Matpower.

Population size	100	Mutation probability	0.01
Number of generations	300	Crossover probability	0.8

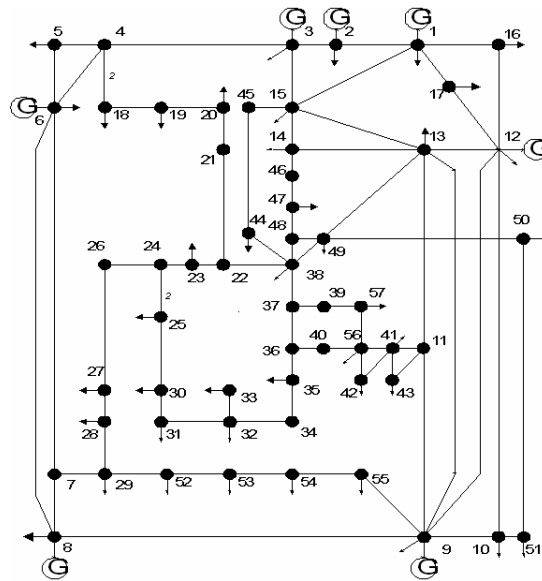


Fig. 2. One-line diagram of the IEEE 57 bus test system.

Table 8. Results of GAOPF compared with Classical methods and GAMatpowerOPF for the IEEE 57-bus system.

Variable	MatpowerOPF	GA OPF	GAMatpowerOPF
P_{G1} (MW)	270.27	262.075	262.352
P_{G2} (MW)	100.00	99.9985	100.000
P_{G3} (MW)	140.00	139.980	140.000
P_{G6} (MW)	100.00	99.9969	100.000
P_{G8} (MW)	270.27	275.828	275.537
P_{G9} (MW)	100.00	99.9999	99.9999
P_{G12} (MW)	270.27	272.278	272.265
Q_{G1} (Mvar)	23.520	219.055	219.057
Q_{G2} (Mvar)	47.980	- 9.7219	- 9.7308
Q_{G3} (Mvar)	20.490	13.1918	13.124
Q_{G6} (Mvar)	22.230	0.93780	0.9328
Q_{G8} (Mvar)	49.390	- 87.349	- 87.359
Q_{G9} (Mvar)	9.0000	1.83880	1.8389
Q_{G12} (Mvar)	116.98	151.740	151.736
V_{G1} (pu)	0.9740	0.95301	1.01062
V_{G2} (pu)	0.9680	1.05084	0.94682
V_{G3} (pu)	0.9780	1.02944	1.02175
V_{G6} (pu)	0.9990	0.98493	0.9697
V_{G8} (pu)	0.9980	1.05306	1.02238
V_{G9} (pu)	0.9870	0.94.83	1.04713
V_{G12} (pu)	1.0080	0.98800	0.94425
Cost (\$/hr)	3063.96	3063.07	3062.89
P_L (MW)	0.0000	0.0000	0.0000
Time (S)	5.75	86.25	45.225

4.1 First Variant

Ignores the transmission losses ($P_L = 0.00$ MW).

4.2 Second Variant

Transmission line losses are calculated and maintained constant ($P_L = 19.06$ MW).

4.3 Third Variant

We consider the variable losses according to each method. From Tables 8, 9, 10, although Matpower can give a solution within a short time, the solution is only a local optimal which depends on a starting point. The result obtained from GA is better, but it takes longer computational time.

Table 9. Results of GAOPF compared with classical methods and GAMatpowerOPF for the IEEE 57-bus system with the constant losses = 19.06 kW.

Variable	MatpowerOPF	GAOPF	GAMatpowerOPF
P_{G1} (MW)	265.33	277.127	277.149
P_{G2} (MW)	100.00	100.000	100.000
P_{G3} (MW)	140.00	140.000	140.000
P_{G6} (MW)	100.00	100.000	100.000
P_{G8} (MW)	276.97	277.260	277.282
P_{G9} (MW)	100.00	100.000	100.000
P_{G12} (MW)	287.56	275.403	275.403
Q_{G1} (Mvar)	72.730	255.725	93.9932
Q_{G2} (Mvar)	50.000	3.94830	38.0285
Q_{G3} (Mvar)	36.740	10.5590	22.3758
Q_{G6} (Mvar)	6.3700	- 7.3619	- 4.5756
Q_{G8} (Mvar)	55.780	30.8469	31.9729
Q_{G9} (Mvar)	9.0000	5.86470	- 0.4368
Q_{G12} (Mvar)	48.250	- 20.499	97.5420
V_{G1} (pu)	1.0320	0.97112	0.98338
V_{G2} (pu)	1.0300	1.05267	1.02717
V_{G3} (pu)	1.0190	0.98723	1.05625
V_{G6} (pu)	1.0100	0.99352	0.96450
V_{G8} (pu)	1.0080	0.99358	1.03835
V_{G9} (pu)	0.9840	0.98515	0.99435
V_{G12} (pu)	0.9840	0.99514	0.97703
Cost (\$/hr)	3176.39	3175.579	3173.982
P_L (MW)	19.06	19.06	19.06
Time (S)	6.85	103.75	56.375

Table 10. Results of GAOPF compared with classical methods and GAMatpowerOPF for the IEEE 57-bus system.

Variable	MatpowerOPF	GAOPF	GAMatpowerOPF
P_{G1} (MW)	265.33	266.850	268.047
P_{G2} (MW)	100.00	100.000	100.000
P_{G3} (MW)	140.00	140.000	140.000
P_{G6} (MW)	100.00	100.000	100.000
P_{G8} (MW)	276.97	280.438	275.268
P_{G9} (MW)	100.00	100.000	100.000
P_{G12} (MW)	287.56	281.875	282.275
Q_{G1} (Mvar)	72.730	195.321	136.833
Q_{G2} (Mvar)	50.000	- 11.768	- 10.341
Q_{G3} (Mvar)	36.740	18.6129	6.70500
Q_{G6} (Mvar)	6.3700	4.45390	- 5.3714
Q_{G8} (Mvar)	55.780	16.1860	54.2315
Q_{G9} (Mvar)	9.0000	- 1.6061	- 0.2815
Q_{G12} (Mvar)	48.250	55.7004	95.1240
V_{G1} (pu)	1.0320	0.96900	0.97.22
V_{G2} (pu)	1.0300	1.03900	0.96.17
V_{G3} (pu)	1.0190	1.04540	0.95.14
V_{G6} (pu)	1.0100	1.00800	1.00196
V_{G8} (pu)	1.0080	1.00850	0. 9962
V_{G9} (pu)	0.9840	1.03940	1.01581
V_{G12} (pu)	0.9840	0.94680	1.00446
Cost (\$/hr)	3176.39	3171.785	3158.09
P_L (MW)	19.06	18.40	14.80
Time (S)	6.85	97.75	46.875

Therefore, the combined method is presented in this paper. This method consists of two parts. The first part employs the GAs to obtain a near global solution, while the other part employs Matpower to determine the solution closer to the global solution. Result of the study demonstrates that the proposed method can give the lowest production cost within reasonable time.

5. CONCLUSIONS

This paper presents a methodology for solving OPF including active power dispatches using GA and Matpower. The method employs advantage of the GAs which can provide a near global solution at the beginning. Then Matpower which can tune the control variables to obtain the global solution is applied. Results of the study demonstrate that the proposed method give a better solution than GA or Matpower alone.

REFERENCES

1. A. J. Wood and B. F. Wollenberg, *Power Generation Operation and Control*, Wiley and Sons, 1996.
2. H. W. Dommel, "Optimal power dispatch," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS93, 1974, pp. 820-830.
3. O. Alsac, J. Bright, M. Prais, and B. Stott, "Further developments in LP-based optimal power flow," *IEEE Transactions on Power Systems*, Vol. 5, 1990, pp. 697-711.
4. J. Nanda, D. P. Kothari, and S. C. Srivatava, "New optimal power-dispatch algorithm using Fletcher's quadratic programming method," in *Proceedings of the IEE*, Vol. 136, 1989, pp. 153-161.
5. J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, USA, 1975.
6. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Publishing Company, 1989.
7. W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-86, 1967, pp. 1449-1460.
8. B. Stott and O. Alsac, "Fast decoupled load flow," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-93, 1974, pp. 859-869.
9. R. D. Zimmerman and D. Gan, *MATPOWER – A MATLAB Power System Simulation Package*, User's Manual, School of Electrical Engineering, Cornell University, 1997, <http://www.pserc.cornell.edu/matpower/manual.pdf>.
10. R. van Amerongen, "A general-purpose version of the fast decoupled loadflow," *IEEE Transactions on Power Systems*, Vol. 4, 1989, pp. 760-770.
11. C. Mészáros, "The efficient implementation of interior point methods for linear programming and their applications," Ph.D. Thesis, Eötvös Loránd University of Sciences, 1996.
12. A. Grace, *Optimization Toolbox*, the MathWorks, Inc., Natick, MA, 1995.
13. C. Li, R. B. Johnson, and A. J. Svoboda, "A new unit commitment method," *IEEE Transactions on Power Systems*, Vol. 12, 1997, pp. 113-119.
14. M. Rahli, "Applied linear and nonlinear programming to economic dispatch," Ph.D. Thesis, Electrical Institute, University of Science and Technology of Oran, Oran, Algeria, 1996.
15. M. Younes, "Economic dispatch using genetic algorithm," M.S. Thesis, Electrical Engineering Institute of the University of Sidi-bel-Abbes, Algeria, 2003.
16. M. Younes and M. Rahli, "On the choice genetic parameters with Taguchi method applied in economic power dispatch," *Leonardo Journal of Sciences*, Issue 9, 2006, pp. 9-24.
17. A. H. Mantawy, Y. L. Abdel-Magid, and S. Z. Selim, "Unit commitment by Tabu search," *IEE Proceedings of Generation Transmission Distribution*, Vol. 145, 1998, pp. 56-64.
18. G. W. Stagg and A. H. el Abiad, *Computer Methods in Power Systems Analysis*, Mc Graw Hill international Book Company, 1968.
19. Z. Michalewicz, "A survey of constraint handling techniques in evolutionary computation methods," in *Proceedings of the 4th Annual Conference on Evolutionary Programming*, 1995, pp. 135-155.



Mimoun Younes was born on March 24th, 1965 in Sidi Belabbes, Algeria. He received his B.S. degree in 1990, the M.S. degree in 2003 from the Electrical Engineering Institute of The University of Sidi Belabbes (Algeria). He is currently Professor of Electrical Engineering at The University of Sidi Belabbes (Algeria). His research interests include operations, planning and economics of electric energy systems, as well as optimization theory and its applications.



Mostefa Rahli was born on October 24th, 1949 in Mosta-Douz, Algeria. He received his B.S. degree in 1979, the M.S. degree in 1985, and the Ph.D. degree in January 1996 from the Electrical Engineering Institute of The University of Sciences and Technology of Oran (USTO). From 1987 to 1991, he was a visiting professor at the University of Liege (Montefiore's Electrical Institute) Liege (Belgium) where he worked on Power Systems Analysis analysis under Professors Pol Pirotte and Jean Louis Lilien. He is currently Professor of Electrical Engineering at The University of Sciences and Technology of Oran (USTO), Oran, Algeria. His research interests include operations, planning and economics of electric energy systems, as well as optimization theory and its applications. He is IEEE member.



Lahouari Abdelhakem-Koridak was born on March 12nd, 1966 in Oran, Algeria. He received his B.S. degree in 1993 and the M.S. degree in 2003 from the Electrical Engineering Institute of The University of Sciences and Technology of Oran (USTO). He is currently Professor of electrical engineering at The University of Sciences and Technology of Oran (USTO). His research interests include operations, planning and economics of electric energy systems, as well as optimisation theory and its applications.