TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 15, No. 3, September 2015, pp. 445 ~ 451
DOI: 10.11591/telkomnika.v15i3.8276

Using Particle Swarm Optimization, Genetic Algorithm, Honey Bee Mating Optimization and Shuffle Frog Leaping Algorithm for Solving OPF Problem with their Comparison

Sajjad Ahmadnia*, Ehsan Tafehi,
University of Birjand, Birjand/ Iran addres
*Corresponding author, e-mail: sajjadahmadnia@gmail.com

Abstract

Today using evolutionary programing for solving complex, nonlinear mathematical problems like optimum power flow is commonly in use. These types of problems are naturally nonlinear and the conventional mathematical methods aren't powerful enough for achieving the desirable results. In this study an Optimum Power Flow problem solved by means of minimization of fuel costs for IEEE 30 buses test system by Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Honey Bee Mating Optimization (HBMO) and Shuffle Frog Leaping Algorithm (SFLA), these algorithms has been used in MATLAB medium with help of MATHPOWER to achieving more precise results and comparing these results with the other proposed results in other published papers.

Keywords: *PSO, GA, HBMO, SFLA, OPF, MATHPOWER*

Copyright © 2015 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

In power system operation, the economic dispatch (ED) problems is and important optimization problem. Moreover it has complex and nonlinear characteristics with heavy equality and inequality constraint [2]. The OPF problem can be describe as the optimal allocation of power system controls to satisfy the specific objective function such as fuel cost, power loss, and bus voltage deviation. The control variables include the generation real power, the generation bus voltage, the tap ration of transformer and the reactive power generation of VAR sources. To solve this large-scale, highly constrained, nonlinear non-convex optimization problem needs unconventional method because conventional methods based on mathematical technique cannot give a guarantee to find the global optimum [1]. In additional performance of these traditional approaches also depends on the starting points and is likely to converge to local minimum or even diverge [1]. Recently, many attempts to overcome these limitation has been proposed such as honey bee mating optimization (HBMO), Improved Evolutionary Programing (IEP), Modified Differential Evolution OPF (MDE-OPF) and the goal for all of these methods is to find the global optimization faster and more precise.

In this paper we used HBMO as intelligence heuristic algorithm and solved OPF problem by means of minimization of fuel costs and consequently cost decreases of the generated power while satisfying specified physical and operational limitations as constraint. By comparison between HBMO with the above mentioned algorithms the difference is more obvious. This paper is organized as follows: Section II explain the optimal power flow an in section III describes the problem formulation and constrain. Section IV describes the HBMO strategy. Section V expresses the strategy of Particle Swarm Optimization. Section VI describes the Genetic algorithm and its formulation. In section VII present the shuffle frog leaping algorithm and its formulation. And in section VIII is the numerical results and the comparison between achieved results and results in the other papers.

Received June 5, 2015; Revised July 30, 2015; Accepted August 13, 2015

2. Optimal Power Flow Problem

One of the objective for planning and operating is to minimum the generation cost. This objective to determine the optimized combination of real power generation, voltage magnitude, compensator capacitor and transformer tap position. These conditions make the OPF a large scale non-linear constrained optimized problem. Other conventional methods are not suitable for solving this type of great optimization problem. OPF problem has been solved by non-linear program (NLP) [13], linear program (LP) [14], Newton methods which can minimize a quadratic of the lagrangian function by increasing the number of problem variables for each iteration [15, 16], quadratic program [17], decomposition method [18] and fast function linear programing algorithm [19]. By increasing the cost of the fossil fuels which lead to increasing the cost of the generated electricity, it is important to minimum the cost, while satisfying certain constraints. Also rising concern about environmental problems, the utilities need to modify their operation strategies for power generation.

3. Problem Formulation

In this paper the goal of OPF problem is to minimize the fuel costs while satisfying operating and loading constraints. Generally, an optimization problem may formulate as below:

$$\text{Min } f(x) \tag{1}$$

Subject to:

$$g(x) = 0 \tag{2}$$

$$h(x) \leq 0 \tag{3}$$

$$x_{min} \leq x \leq x_{max} \tag{4}$$

Where x is the vector of system variables, f(x) is the objective function: g(x) and h(x) are equality and inequality constraint respectively. Equation (4) represents the control variable limits (which are, in fact, a set of inequality constrain). In this problem:

$$x = [\, P_G^l \; V_L^T \; Q_G^T \; S_l^T \,] \tag{5}$$

Where (P) is the vector of active power in all buses, (Q) represents the vector of reactive power in all buses while (V) is the vector of voltage magnitude in all buses and (θ) is the vector of voltage angle in all buses.

3.1. The Load Flow Equations

$$P_{Gi} - P_{Di} - V_i \sum_{j \neq i} V_j (G_{ij} \sin \theta_{ij} + B_{ij} \sin \theta_{ij}) = 0 \tag{6}$$

$i \in n$, where set of numbers of buses except the swing bus.

$$Q_{Gi} - Q_{Di} - V_i \sum_{j \neq i} V_j (G_{ij} \sin \theta_{ij} - B_{ij} \sin \theta_{ij}) = 0 \tag{7}$$

$i \in n$, where set of numbers of buses except the swing bus. The fuel cost function is as given:

$$F = \sum_{i=1}^{NG} f_i \; (\$/h) \tag{8}$$

And the generation curves are represented by quadratic function as:

$$f_i = (a_i + b_i P_{Gi} + c_i P_{Gi}^2) \; (\$/h) \tag{9}$$

3.2. Generation Constraints

Generator voltages, real power outputs and reactive power outputs are restricted by their lower and upper limits as follows:

$$VGi\ min \le VGi \le VGi\ max\ ,\ i = 1,2,\ldots NG,$$
$$PGi\ min \le PGi \le PGi\ max\ ,\ i = 1,2,\ldots..NG,$$
$$QGi\ min \le QGi \le QGi\ max\ ,\ i = 1,2,\ldots NG \quad (10)$$

3.3 Transformer Constraints

Transformer tap settings are bounded as follows:

$$Ti\ min \le Ti \le Ti\ max\ ,\ i = 1,2,\ldots NT. \quad (11)$$

3.4. Security Constraint

The constraints of the voltage at load buses and transmission line loading are considered as follows:

$$VLi\ min \le VLi \le VLi\ max\ ,\ i = 1,2,\ldots NL, \quad (12)$$

$$Sli \le Sli\ max\ ,\ i = 1,2,\ldots Nl, \quad (13)$$

Where F is objective function, g equality constraints, h operating constraint, PGl slack bus power, PGi real power output of generator, PDi real power load of bus i, QGi reactive power output of generator i, QDi reactive power load of bus i, VL load bus voltages, Vi Voltage magnitude of bus i, θi voltage phase angle of bus i and j, Bij mutual susceptance between buses i and j, NG number of generator buses, NL number of load buses, NT number of Transformers, nl number of lines, Si transmission line loadings, VGi min, VGi max bus voltage limit, PGi min, PGi max generator real power limit, QGi min, QGi max generator reactive power limit, Ti min , T i max transformer tap position limit [3].

4. Honey Bee Mating Optimization

The frame of HBMO algorithm grasps from the mating process of the queen in the hive. Each bee is randomly generated as a candidate solution. Afterwards, the fitness of each individual is calculated and the nest member of the hive whereas all the other members of the population are the drones. The mating process of the queen is started when the queen flights away from the hive performing the mating with her in the air. In the original HBMO algorithm, the procedure of mating of the queen with the drones has been describe.

In [10], a drone mates with a queen probabilistically using an annealing function as follows [9]:

$$Prob(D) = \exp\ (\_\Delta(f)/S(k)) \quad (14)$$

Where Prob(D) is the probability of adding the sperm of drones D to the sperm theca of the queen, $\Delta(f)$ is the absolute difference between the fitness of D and the iteration k. the probability of mating is higher when the queen is with the high speed level, or when the fitness of drone is as good as that of the queen. After each transition in space, the queen's speed decreases according to the following equation:

$$S(k+1) = \alpha \times S(k) \quad (15)$$

Where α is a factor ϵ [0,1] and is the amount of speed reduction after each transition and each step. The speed of the queen is initialized at random. At the start of a mating flight drones are the realized. At the start of a mating flight drones are generated randomly and the queen selects a drone using the probabilistic rule in (14). If the mating is successful, the drone's sperm is stored in the queen's sperm theca. By suing the crossover of the drone's and the queen genotypes, a new brood (trial solution) is generated, which can be improved later by employing workers to conduct local search. This crossover operator is a kind of adaptive memory procedure. Initially, the adaptive memory has been proposed by Rochat and Taillard [8].

5. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a relatively new evolutionary algorithm that may be used to find optimal or (near optimal) solution to numerical and qualitative problems. PSO was originally developed by James Kennedy and Russell Eberhart in 1995, and emerged from earlier experiments with algorithm that modeled the flocking behavior seen in many species of birds.in simulations, birds would begin by flying around with no particular destination and spontaneously formed flocks until one of the birds used to set their directions and velocities, a bird pulling away from the flock in orders to land at the roost would result in nearby bird moving toward the roost. Once these birds discover the roost, they would land there, pulling more birds towards it, and so on until the entire flock had landed. Finding a roost is analogous to finding a solution in a field of possible solution in a solution space. The manner in which a bird who has found the roost, leads it neighbors to move towards it, increase the chances that they will also find it. This is known as the socio-cognitive view of mind. It means a particle learns primarily from the success of its neighbor's.

The concept of the PSO consist of, at each time step, changing the velocity of (accelerating) each particle toward its pbest and gbest location (local version of OPS). Acceleration is weight by a random terms, with separate random numbers being generated for acceleration toward pbest and gbest locations. In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods [6].
This modification can be represented by the concept of velocity. Velocity of each agent can be modified by the following equation:

$$Vk+1 = W.Vk+C1rabd\ (Pbest\ \_\ Xk) + C2\ rand\ (gbest\ \_\ Xk) \quad (16)$$

Using the above equation, a certain velocity, which gradually gets close to pbest and gbest can be calculated. The current position (searching points in the space) can be modified by the following equation:

$$X\ k+1 = X\ k + V\ k+1,\ k = 1,2,\ldots,\ n \quad (17)$$

Where X k is current searching point, X k+1 is modified searching point, V k is current velocity, V k+1 is modified velocity of agent. Vpbest is velocity based on pbest, Vgbest is velocity based on gbest, n is number of particle in a group, m is number of members in a particle, pbest of agent k, gbesti is gbest of the group, w is weight function for velocity of agent k, Ci is weight coefficient for each term. C1 and C2 are two positive constant. r1 and r2 are two randomly generated numbers with a range of [0,1]. W is the inertia weight and it is defined as a function of iteration indes k as follows:

$$W(k) = W\ max\ \_\ [\ (\ Wmax\ \_\ W\ min\ )\ /\ (\ Max.\ Iter.)]\ {*}k \quad (18)$$

Where Max.Iter, k is maximum number of iteration and the current number if iterations, respectively. To insure uniform velocity through all dimensions, the maximum velocity is as:

$$V\ max = (X\ max\ \_\ X\ min)\ /\ N \quad (19)$$

Where N is a chosen number of iterations [2].

For implementation of the PSO algorithm each unit's active power generation is assumed particle. Therefore, there is ng particles in solution space. Here the objective function should be modified with a penalty factor and become a fitness function to give feasible solution as follow:

$$F(P) = \Sigma\ \ i=1\ Costi\ (Pi)\ x\ (1+0.5x(\sim success)) \quad (20)$$

For implementation the following procedure is done:

1) Set j=1;

2) Set all particle's positios to initial values, which are random numbers in the acceptable range [0; PGmax];
3) Set initial velocity to zero;
4) Evaluate objective function for all particles;
5) Find the best position of each particle and the best position of all;
6) Update velocities and positions;
7) Set j = j+1;
8) Go to set 2 until factor and learning rates are set according to Trelea's second set of parameters [7] (w=0.7968, C1 = C2 = 1.4962).

6. Genetic Algorithm

GA is a general purpose optimization algorithm based on mechanics of natural selection and genetics. It operates on string structure (chromosomes), typically a concatenated list of binary digits representing a coding of the control parameters (phenotype) of a given problem. Chromosomes themselves are composed of genes. The real value of a control parameter, encoded in a gene, is called an allele [7]. The GA and most AI-based heuristics methods are attractive alternative to other optimization methods because of their robustness. There are three major differences between these two new methods (GA and PSO and conventional optimization algorithms. First these new ones operate on the encoded string of the problem parameters rather than the actual parameters of the problems. Each string can be thought of as a chromosome or a swarm that completely describe one candidate solution to the problem. Second, they use a population of points rather than a single point in their search. This allows them to explore several areas of the search space simultaneously, reducing the probability if finding local optimum point. Third they don't require any prior knowledge, space limitation, or special properties of the function to be optimized, such as smoothness, convexity, uni-modality or existence of derivatives [7]. For implementation of this method each units assumed a gene, and all these genes in a row, make up a chromosome. So there are ng genes in each chromosome. Since all possible sets of solutions are not feasible, we need a routine to take out infeasible ones, for each set of solution feasibility will be checked. The penalty factor is as follow [7]:

$$F(P) = \Sigma_{i=1} Cost_i (P_i) \times (1+0.5x(\sim success)) \quad (21)$$

The penalty factor is not directly added, but is multiplied to the cost function in order to make a softer fitness function and to avoid convergence into sub-optimal solutions. So:

Set j=1;
Creat first generation randomly in an acceptable range [0,P_{gmax}]; (P_G is the vector of active power of generators).
Evaluate each chromosome in the generation.
Produce new chromosomes using crossover operator (0.8 of the new population is created this way).
Make a new population from the new chromosomes with mutation operator.
Set j = j+1;
Go to step 3 until some stopping criteria is met.

7. Shuffle Frog Leaping Algorithm

SFLA mimics the metaphor of natural biological evolution that is based on population of frogs in nature searching for food. The SFLA is a decreased based stochastic search algorithm which is started with an initial frog population whose characteristics represent the decision variables of the optimization problem. An initial population of F frog is created randomly. For k-dimensional problems (k variables), a frog i is represented as $X_i = (x_{i1}, x_{i2}, \dots, x_{ik})$. Initially, the objective function is calculated for each frog, and afterwards frogs are sorted in a descending manner according to their fitness. In SFLA, the total population is divided into groups that search independently. In this process, the first frog goes to fist group, the second frog goes to second group, frog m goes to the qth group, and frog m+1 goes to the first group, and so on. In each group, the frogs with the best and the worst fitness are recognized as X_b and X_w , respectively. Also the frog with best fitness in all groups as recognized as X_g. Then, the following process is applied to improve only frog with the worth fitness (not all frogs) in each iterate. Correspondingly, the location of the frog with the worst fitness is regulated as follows:

$$\text{Change in the location } V_i = \text{rand()} \times (X_b - X_w) + \text{rand ()} \times (X_g - X_w) \quad (22)$$

$$X_{w(new)} = X_w + V_i \quad -V_{max} \leq V_i \leq V_{max} \quad (23)$$

Where rand() is a random number between 0 and 1, and V_{max} is the maximum permitted change in a frog's location. If thus process generates a better solution, it replaces the worst frog. Otherwise, the calculation in (22) and (23) are repeated for specific iterations (Iter. $_{max}$). in addition, to provide the opportunity for random generation of improved information, random virtual frogs are generated and substituted in the population if the local search cannot find the better solution respectively in each iteration. After a number of iterations (Iter $_{max}$), al groups are combined and share their ideas with themselves through a shuffling process. The local search and shuffling processes continue until the defined convergence criteria are satisfied. The aim of the entire process is to determine global optimal solutions.

8. Case Study

Table 2 is the results of algorithm that are mentioned in above (PSO, GA, HBMO, FSLA) which has been collected from other papers on IEEE 30-buses test system and in comparison is Table 3. The results which is presented in this paper in also on IEEE 30-buses test system in MATLAB 2011 a medium and programing in MATHPOWER in Core i5-2430M. the specification for this test system can be seen on Table 1 and Figure 1. And it has been obtained from [4]. The 30-bus test system consist of six generators at buses #1, #2, #5, #8, #11 and #13, four transformers (T6-9, T6-9, T4-12, T27-28) and two shunt capacitor buses (bus#10, bus #24). Also bus#1 is considered as the slack bus and voltage magnitude limit of all buses is considered 0.95≤ V_g ≤ 1.05 for buses #14 and #30.

Table 1. Cost Coefficint, Power Limitation for IEEE 30-Bus Test System

Unit	a	b	c	d	P_{min}	P_{max}
G_1	0.00375	2	0	18	0	250
G_2	0.0175	1.75	0	16	0	80
G_3	0.0625	1	0	14	0	50
G_4	0.0083	3.25	0	12	0	55
G_5	0.025	3	0	13	0	30
G_6	0.025	3	0	13.25	0	40

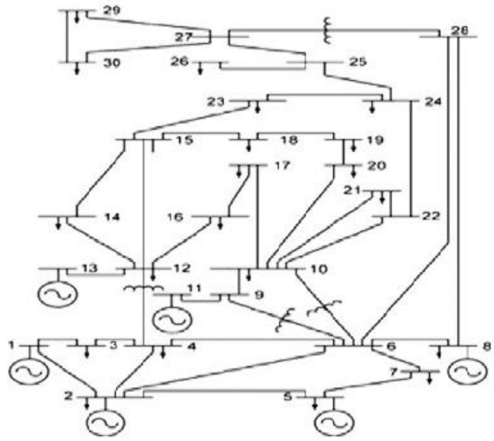

Figure 1. IEEE 30-buses test system [11]

Table 2. Results of the minimum fuel cost compared with different algorithm for IEEE 30-bus (From other papers)

Method	P_{G1} (MW)	P_{G2} (MW)	P_{G5} (MW)	P_{G8} (MW)	P_{G11} (MW)	P_{G13} (MW)	Cost
PSO [3]	179.1929	48.9804	19.8833	20.7275	12.7280	12.0000	802.8351
GA [3]	183.1419	46.2438	21.9576	17.2385	11.7363	13.0954	803.2332
SFLA [3]	179.0337	49.2580	20.3183	21.3269	11.5420	11.6655	802.5092
HBMO [4]	178.4646	46.274	21.4596	21.446	13.207	12.0134	802.211

Table 3. Results Results of the minimum fuel cost compared with different algorithm for IEEE 30-bus (presented in this paper)

Method	P_{G1} (MW)	P_{G2} (MW)	P_{G5} (MW)	P_{G8} (MW)	P_{G11} (MW)	P_{G13} (MW)	Cost
PSO	177.0934	48.9512	21.5214	21.8346	12.2031	11.3272	802.3224
GA	177.0691	48.96	21.53	21.85	12.18	11.33	802.2863
SFLA	173.9909	51.58	22.48	18.12	16.28	10.32	803.1098
HBMO	176.1328	48.96	21.05	23.14	12.5	11.08	802.3601

References

[1] Jong-Yul Kim, Hwa-seok Lee, June-Ho Park. A Modified Particle Swarm Optimization for Optimal Power Flow. *Journal of Electrical Engineering & Technology*. 2007; 2(4): 413-419.

[2] Dijilani Ben Attous, Yancine Labbi. Particle Swarm Optimization based Optimal Power Flow for Units with Non-Smooth fuel Cost Function. *Electrical and Electronic Engineering ELECO.* 2009; 1: 377-381.

[3] Prabha Umapathy, C Venkataseshaiah, M Senthil Arumugam. Particle Swarm Optimization with various Inertia Weight Variants for Optimal Power Flow Solution. Hindwal Publishing Corporation. 2010.

[4] T Niknam, MR Narimani, J Ghei, S Tabatabaei, M Nayeripour. Modified Honey Bee Mating Optimisation to solve dynamic optimal power flow considering generator constraints. *IET Generation, Transmission & Distribution.* 2011; 5(10): 989-1002.

[5] Taher Niknam, Mohammad Rasoul Narimani, Masoud Jabbari, Ahmad Reza Malekpour. Shuffle Frog Leaping Algorithm for multi-objective optimal power flow. *Journal energy on Elsevier*. 2011; 36(11): 6420-6432.

[6] Pathak Smita, Prof BN Vaidya. *Particle Swarm Optimization based Optimal Power Flow for Reactive loss Minimization*. IEEE students' Conference on Electrical, Electronic and Computer Science (SCEECS). 2012: 1-4.

[7] Mostafa Majidpour, Ashkan Rahimi-kian. *Solving OPF Problem with the Hybrid GA and the Hybrid PSO Algorithms and Comparing with the Gradient-based Methods*. 21[Th] International Power System Conference. 1385.

[8] Y Rochat, R Taillard. Probilistic diversification and intensification in local search for vehicle routing. 1995: 147-167.

[9] T Niknam, Olamanie, R Khorshidi. A Hybrid fuzzy algorithm for multi-objective distribution feeder reconfiguration. 2008; 4(2): 308-315.

[10] T Niknam. Application of Honey bee mating optimization on distribution state estimation including distributed generation. *J. Zhejian Univ. Sci*. 2008: 1753-1764.

[11] Avaiable: http://www.ee.washington.edu/research/pstca/indes.html.

[12] O Alasce, B Stott. Optimal load flow with steady-state security. *IEEE Trans. Power Appar*. 1994: 745-751.

[13] Momoh Ja, El-Hawary ME, Adapa R. A review of selected optimal powr literature to 1993. Part 1: non-linear and quadratic programing approaches. *IEEE Transaction on Power System.* 1999: 96-104.

[14] Momoh Ja, El-Hawary ME, Adapa R. A review of selected optimal powr literature to 1993. Part II: Newton, linear programing and interior point methods. *IEEE Transaction on Power System.* 1999: 105-111.

[15] Ambriz-Perez H, Acha E, Fuerte-Esquivel CR, De La Torre. Incorporation of a UPFS model in an optimal power flow using Newton's method. *IEEE Proceeding on Generation Transmission Distribution.* 1998: 336-366.

[16] Ambriz-Perez H, Acha E, Fuerte-Esquivel CR. Advanced SVC models for Newton-Raphson load flow and newton optimal power flow studies. *IEEE Transaction on Power system.* 2000: 129-136.

[17] Momoh JA, Guo SX, Ogbubiri EC, Adapa R. The quadratic interior point method solving power system optimization problem. *IEEE Transaction on Power system.* 1994: 1327-1336.

[18] Alasc O, Stoot B. Optimal Power Flow with steady state security. *Power Apparatus and System IEEE Transaction.* 1974: 745-751.