

Particle Swarm Optimization Applied to Optimal Power flow solution

Ibrahim Oumarou, Prof. Daozhuo Jiang, Prof. Cao Yijia
College of Electrical Engineering, Zhejiang University, Hangzhou, China
ibraoumar@yahoo.fr dzjiang@zju.edu.cn

Abstract--This paper presents solution of optimal power flow (OPF) problem of a power system via a simple particle swarm optimization (PSO) algorithm. This method is dynamic in nature and it overcomes the shortcomings of other evolutionary computation techniques such as premature convergence and provides high quality solutions. The objective is to minimize the fuel cost and keep the power outputs of generators, bus voltages, shunt capacitors/reactors and transformers tap-setting in their secure limits. The effectiveness of PSO was compared to that of OPF by MATPOWER. The potential and superiority of PSO have been demonstrated through the results of IEEE 30-bus system.

Key Words: Particle Swarm Optimization, Optimal Power Flow, load Flow.

I . INTRODUCTION

The main target of electric power utilities is to provide high quality reliable supply to the consumers at the lowest possible cost while operating to meet the limits and constraints imposed on the generating units. Optimal Power Flow (OPF) is a large-scale, nonlinear, non convex, static optimization problem which computes optimal settings of electric variables under the condition of constraints imposed by operational and physical particulars of the electric network. As the name suggests, optimal power flows attempt to optimize the power system according to a specific function. This function is called the objective function and is generally minimized by the OPF program. The most common objective function is the sum of all production costs of the system; however other functions such as system losses may be used. The optimal power flow has been frequently solved using classical optimization methods such as nonlinear programming, quadratic programming, linear programming, interior-point method, The constraints involved are the physical laws governing the power generation-transmission systems and the operating limitations of the equipment. Conventional optimization methods are based on successive linearization using the first and the second derivatives of objective functions and their constraint as the search directions [1][4]. The conventional optimization methods usually converge to a local minimum [5]. Recently, intelligence heuristic algorithms, such as genetic algorithm [6], evolutionary programming [7], and metaheuristic algorithms [8] have been proposed for solving the OPF

problem. Like other metaheuristic algorithms, particle swarm optimization (PSO) algorithm was developed through simulation of a simplified social system such as bird flocking and fishing school. PSO is an optimization method based on population [9], and it can be used to solve many complex optimization problems, which are nonlinear, non differentiable and multi-modal. The most prominent merit of PSO is its fast convergence speed. In addition, PSO algorithm can be realized simply for less parameters need adjusting. PSO has been applied to various power system optimization problems with impressive success [10]. The results for a 30-bus system shows that PSO is an effective method to solve OPF problem.

II . PROBLEM FORMULATION

The most commonly used objective in the OPF problem formulation is the minimization of the total cost of real power generation. The individual costs of each generating unit are assumed to be function, only, of real power generation and are represented by quadratic curves of second order. Generally, this objective is given by

$$F = \sum_{i=1}^{ng} (\alpha_i + \beta_i P_{g_i} + \gamma_i P_{g_i}^2) \quad (1)$$

where ng is the number of generation including the slack bus, P_{g_i} is the generated real power at bus i , α_i , β_i , γ_i are the unit costs curve for i^{th} generator.

A . Types of equality constraints.

While minimizing the cost function, it's necessary to make sure that the generation still supplies the load demands plus losses in transmission lines. Usually the power flow equations are used as equality constraints.

$$P_i(V, \theta) = P_{g_i} - P_{d_i} = \sum_{j=1}^{nb} V_i V_j (g_{ij} \cos \theta_{ij} + b_{ij} \sin \theta_{ij}) \quad (2)$$

$$Q_i(V, \theta) = Q_{g_i} - Q_{d_i} = \sum_{j=1}^{nb} V_i V_j (g_{ij} \sin \theta_{ij} + b_{ij} \cos \theta_{ij})$$

Where,

P_{g_i}, Q_{g_i} : the total real and reactive power generation at bus i .

P_{d_i}, Q_{d_i} : the total real and reactive power load at bus i .

V_i : the voltage magnitude at bus i .

g_{ij} : the real part of admittance matrix.

θ_{ij} : the voltage angle of the ij^{th} element of admittance matrix.

b_{ij} : the imaginary part of admittance matrix.

nb : number of bus.

B. Types of inequality constraints

The most usual types of inequality constraints are upper bus voltage limits at generations and load buses, lower bus voltage limits at load buses, VAR limits at generation buses, maximum active power limits corresponding to lower limits at some generators, maximum line loading limits and limits on tap setting of TCULs and phase shifter. The inequality constraints on the problem variables considered include:

Upper and lower bounds on the active generations at generator buses

$$Pg_i^{\min} \leq Pg_i \leq Pg_i^{\max}, i = 1, ng$$

Upper and lower bounds on the reactive power generations at generator buses and reactive power injection at buses with VAR compensation

$$Qg_i^{\min} \leq Qg_i \leq Qg_i^{\max}, i = 1, npv$$

Upper and lower bounds on the voltage magnitude at the all buses

$$V_i^{\min} \leq V_i \leq V_i^{\max}, i = 1, nbus$$

Upper and lower bounds on the bus voltage phase angles

$$\theta_i^{\min} \leq \theta_i \leq \theta_i^{\max}, i = 1, nbus$$

It can be seen that the generalized objective function f is a non-linear, the number of the equality and inequality constraints increase with the size of the power distribution systems. Applications of a conventional optimization technique such as the gradient-based algorithms, to large power distribution systems with a very non-linear objective functions and great number of constraints, are not good enough to solve this problem. Because they depend on the existence of the first and the second derivatives of the objective function and on the well computing of these derivative in large search space.

III. PARTICLE SWARM ALGORITHM IN OPTIMAL POWER FLOW

A. Description of particle swarm optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberharth and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary

computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied. As stated before, PSO simulates the behaviors of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what's the best strategy to find the food? The effective one is to follow the bird which is nearest to the food. PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored). This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest. Compared with genetic algorithms (GAs), the information sharing mechanism in PSO is significantly different. In GAs, chromosomes share information with each other. So the whole population moves like a one group towards an optimal area. In PSO, only gbest (or lbest) gives out the information to others. It is a one-way information sharing mechanism. The evolution only looks for the best solution. Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases. The basic principles in "classical" PSO are very simple. A set of moving particles (the swarm) is initially "thrown" inside the search space. Each particle has the following features:

- It has a position and a velocity
- It knows its position, and the objective function value for this position
- It knows its neighbors, best previous position and objective function value (variant: current position and objective function value)

- It remembers its best previous position
- At each time step, the behavior of a given particle is a compromise between three possible choices:
- To follow its own way
 - To go towards its best previous position
 - To go towards the best neighbor's best previous position, or towards the best neighbor (variant)

This compromise is formalized by the following equations:

$$v_{k+1} = \bar{a} \otimes \bar{v}_k + \bar{b}_1 \otimes \bar{r}_1 \otimes (\bar{p}_1 - \bar{x}_k) + \bar{b}_2 \otimes \bar{r}_2 \otimes (\bar{p}_2 - \bar{x}_k) \quad (3)$$

$$\bar{x}_{k+1} = \bar{x}_k + \bar{v}_{k+1} \quad (4)$$

where :

\bar{v}_{k+1} : the current velocity.

\bar{a} : The inertia weighting function.

\bar{v}_k : the previous velocity.

\bar{b}_1, \bar{b}_2 : the cognitive and the social parameters, respectively.

\bar{r}_1, \bar{r}_2 : random numbers uniformly distributed within [0,1] .

\bar{p}_1 : the best previous position of the k^{th} particle.

\bar{p}_2 : the global best in the k^{th} swarm.

\bar{x}_{k+1} : the current position.

\bar{x}_k : the previous position.

The first part of equation (3) is the inertia velocity of particle, which reflects the memory behavior of particle; the second part is cognition part, which represents the private thinking of the particle itself; the third part is the social part, which shows the particle's behavior system from the experience of other particles in the population. The particles find the optimal solution by cooperation and competition among the particles [10]. Using the above equation, a certain velocity, that gradually gets close to \bar{p}_1 and \bar{p}_2 , can be calculated. The position of each particle (searching point in the solution space) can be modified according to equation (4).

B. Application to Optimal Power Flow.

The cost function is defined as:

$$F = \sum_{i=1}^{ng} (\alpha_i + \beta_i P g_i + \gamma_i P g_i^2) \quad (5)$$

$$P g_i^{\min} \leq P g_i \leq P g_i^{\max}$$

Minimize F is equivalent to getting a maximum fitness value in the searching process. The particle that has lower cost function should be assigned a larger fitness value. The

objective of OPF has to be changed to the maximization of fitness to be used as follows:

$$fitness = \begin{cases} f_{\max} / F; & \text{if } f_{\max} \neq F \\ 0; & \text{otherwise} \end{cases} \quad (6)$$

The PSO algorithm applied to OPF can be described in the following steps:

Step 1: Input parameters of system, and specify the lower and upper boundaries of each variable.

Step 2: Initialize randomly the particles of the population.

Step 3: Calculate the evaluation value of each particle using the objective function.

Step 4: Calculate the fitness value of objective function of each particle using (6), \bar{p}_1 is set as the k^{th} particle's initial position; \bar{p}_2 is set as the best one of \bar{p}_1 , and the current evolution is $t=1$.

Step 6: Initialize learning factor \bar{b}_1, \bar{b}_2 , inertia weight \bar{a} and the initial velocity \bar{v}_1

Step 7: Modify the velocity \bar{v} of each particle according to (3).

Step 8: Modify the position of each particle according to (4). If a particle violates its position limits in any dimension, set its position at the proper limits. Calculate each particle's new fitness, if it is better than the previous \bar{p}_2 , the current value is set to be \bar{P}_2 .

Step 9: To each particles of the population, employ the Newton-Raphson method to calculate power flow and the transmission loss.

Step 10: Update the time counter $t=t+1$.

Step 11: If one of the stopping criteria is satisfied then go to step 12. Otherwise go to step 7.

Step 12: The particle that generates the latest \bar{p}_2 is the Pareto optimal value.

C. Load flow calculation

After the search goal is achieved, or an allowable generation is attained by the PSO algorithm, it's required to perform a load flow solution in order to make better adjustments on the optimum values obtained from the PSO-OPF procedure. This will provide updated voltages, angles and transformer taps and points out generators having exceeded reactive limits. to determine all reactive power of all generators and to determine active power that should be given by the slack generator using into account the deferent 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. 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 [12][13].

IV. APPLICATION STUDY

The PSO-OPF is tested using the modified IEEE 30-bus system [Terra91]. The system consists of 41 lines, 6 generators, 4 Tap-changing transformers, and shunt capacitor banks located at 9 buses. The system is optimized using the PSOOPF algorithm developed. The parameter settings to execute PSO-OPF are $a = 0.9$, $b_1 = 0.5$, $b_2 = 0.05$, $V_{inc} = 1.98$, $numb_particles = 20$, $max_generation = 20$, the power mismatch tolerance is 0.0001 p.u, the maximum voltage magnitude of all bus is 1.1 p.u while the minimum voltage magnitude is 0.95. the maximum voltage angle of all bus is 0° while the minimum voltage angle is -14° . Other parameters are presented in Table I. To compare these results with conventional methods using the same cost objective function. The optimal power flow problem was solved by Matpower [15]. Matpower was created by Ray Zimmerman and Deqiang Gan of PSERC at Cornell University under the direction of Robert Thomas. Its main aim was to provide a simulation tool within Matlab that was easy to use and modify. The comparison results (active and reactive powers, voltages, cost and power losses) are presented in Table II.

Table II: Comparison of Matpower and PSOPF results on the IEEE 30-Bus system

Variables	Matpower	PSOOPF
P ₁ (MW)	176.28	176.28
P ₂ (MW)	48.79	48.301
P ₅ (MW)	21.48	20.924
P ₈ (MW)	22.07	20.561
P ₁₁ (MW)	12.19	11.576
P ₁₃ (MW)	12.00	12.484
Q ₁ (Mvar)	-12.02	-2.983
Q ₂ (Mvar)	30.63	41.837
Q ₅ (Mvar)	29.48	27.676
Q ₈ (Mvar)	46.89	22.467
Q ₁₁ (Mvar)	5.41	29.534
Q ₁₃ (Mvar)	2.80	33.118
V ₁ (p.u.)	1.06	1.06
V ₂ (p.u.)	1.047	1.045
V ₅ (p.u.)	1.02	1.010
V ₈ (p.u.)	1.029	1.025
V ₁₁ (p.u.)	1.06	1.082
V ₁₃ (p.u.)	1.06	1.071
θ_1 ($^\circ$)	0.000	0.000
θ_2 ($^\circ$)	-3.520	-3.608
θ_5 ($^\circ$)	-10.157	-10.301
θ_8 ($^\circ$)	-7.965	-8.116
θ_{11} ($^\circ$)	-8.320	-8.783
θ_{13} ($^\circ$)	-9.679	-10.409
Generation Cost (\$/hr)	802.1	801.995
Real Power Loss (MW)	9.41	9.384

Table 1: Power generation limits and generator cost parameters of IEEE 30-bus system in p.u (SB=100MVA)

Bus	P_{min}	P_{max}	Q_{min}	Q_{max}	V_{min}	V_{max}	b(\$/MWhr)	$c \cdot 10^{-4}$ (\$/MW ² hr)
1	0.50	2.00	-0.20	2.00	0.95	1.10	200	037.5
2	0.20	0.80	-0.20	1.00	0.95	1.10	175	175.0
5	0.15	0.50	-0.15	0.80	0.95	1.10	100	625.0
8	0.10	0.35	-0.15	0.60	0.95	1.10	325	083.0
11	0.10	0.30	-0.10	0.50	0.95	1.10	300	250.0
13	0.10	0.40	-0.15	0.60	0.95	1.10	300	250.0

The results show that PSO algorithm gives much better results than the classical method solved by MATPOWER. The difference in generation cost between these two methods (801.995 \$/hr compared to 802,1 \$/hr) and in Real power loss (9.384 MW compared to 9.41 MW) clearly shows the advantage of this method. In addition, it is important to point out that this simple PSO algorithm OPF converge in an acceptable time. For this system was approximately 15 seconds, and it converged to highly optimal solutions set after 20 generations tested with PV 1.5 GHz 128MO. The security constraints are also checked for

voltage magnitudes and angles. The voltage magnitudes are from the minimum of 0.9592 p.u. to maximum of 1.06 p.u (Fig.I), and the angles are from the minimum of -14.531° to the maximum of 0.0° (Fig.2). No load bus was at the lower limit of 0.95 p.u. The minimum of the magnitude and angle is in the 30th bus with (0.9525p.u., -14.531°).

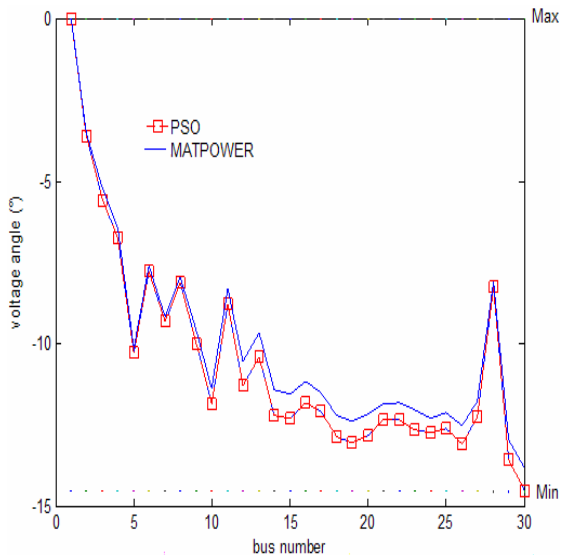


Fig 1: Comparison between MATPOWER and PSO for the voltage angle after optimization

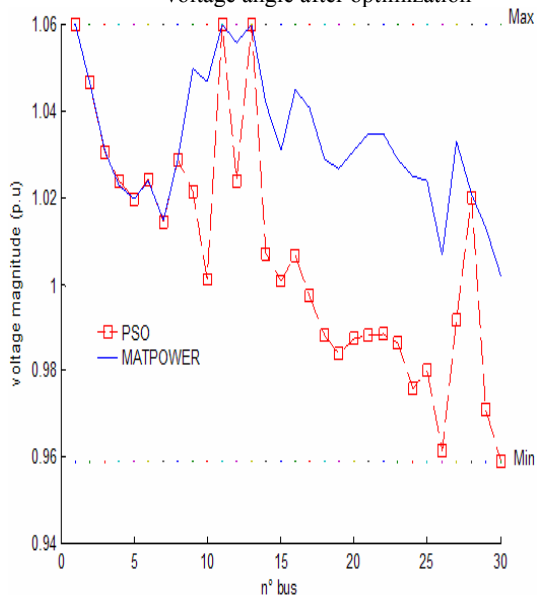


Fig. 2: Comparison between MATPOWER and PSO for the voltage magnitude after optimization

V. CONCLUSION

For the solution of constrained optimization problems such as OPF problem in power systems, penalty function based methods is the most popular approach. However, since the penalty function approach is generic and applicable to any type of constraints, their performance is not always satisfactory and consistent. In order to overcome this drawback, in this paper a PSO method is used for the solution of OPF problem. In view of the memory characteristics of the PSO, a new constraints handling strategy is developed. The proposed PSO approach is

efficiently and effectively minimizing the total generation production cost in an Optimal Power Flow problem. As a study case, the IEEE 30 Bus system with 6-generating units has been selected. The simulation results show that a simple PSO can give best result than MATPOWER. The effectiveness of the PSO for solving OPF problem with environmental pollution caused by fossil based thermal generating units will be investigated in the future research work. The simulation results show that the proposed PSO method is superior to the MATPOWER and computationally faster. Besides, the obtained computational results for the solution of OPF problem confirm its good performance in terms of the solution quality, computational cost as well as the convergence stability.

REFERENCES

- [1] H. W. Dommel, W. F. Tinney, Optimal Power Flow Solutions, IEEE Transactions on power apparatus and systems, Vol. PAS.87, N°10, pp.1866-1876, October 1968.
- [2] K. Y. Lee, Y.M. Park, and J. L. Ortiz, A United Approach to Optimal Real and Reactive Power Dispatch, IEEE Transactions on Power Systems, Vol. PAS-104, pp.1147-1153, May 1985.
- [3] M. Sasson, Non linear Programming Solutions for load flow, minimum loss, and economic dispatching problems, IEEE trans. power apparatus and systems, Vol. Pas-88, N°4, April 1969.
- [4] T. Bouktir, M. Belkacemi, K. Zehar, Optimal power flow using modified gradient method», Proceeding ICCEL2000,13-15 November 2000, U.S.T.Oran, Algeria, Vol.02, pp.436-442
- [5] R. Fletcher, Practical methods of optimisation, John Wiley & Sons, 1986.
- [6] L. L. Lai and J. T. Ma, Improved genetic algorithms for optimal power flow under both normal and contingent operation states, Int. J. Electrical power and Energy Syst., vol. 19, no 5, pp. 287- 292, 1997.
- [7] J. Yurjevich and K. P. Wong, Evolutionary programming based optimal power flow algorithm, IEEE Trans. Power systems, vol. 14, n° 4, pp. 1245-1250, 1997.
- [8] Q. Guo, R. M. Hierons, M. Harman, and K. Derderian, Construct multiple unique input/output sequences using metaheuristic optimisation techniques, IEEE Proceedings - Software. Vol. 152, No. 3, June 2005.
- [9] J. Kennedy and R. Eberhart, Particle swarm optimization . In : IEEE Int. Conf on Neural Networks, Perth, Australia, 1942-1948, 1995.
- [10] C-R. Wang, H-J. Yuan, Z-Q. Huang, J-W. Zhang, C-J. Sun, A modified particle swarm optimization algorithm and its application in optimal power flow problem. Proceedings of the fourth International Conference on machine learning and Cybernetics, Guangzhou, 18-21 Augusts 2005.
- [11] R. Poli, W. B. Langdon, and O. Holland, Extending particle swarm optimisation via genetic programming, Proceedings of EuroGP, 2005.
- [12] G. W. Stagg, A. H. El Abiad, Computer methods in power systems analysis, Mc Graw Hill international Book Company, 1968.
- [13] S. Kumar, R. Billinton, Low bus voltage and ill-conditioned network situation in a composite system adequacy evaluation, IEEE trans. on Power Syst., vol. PWRS-2, No. 3, August 1987.
- [14] L. Terra, M. Short, Security constrained reactive power dispatch, IEEE transaction on power systems, Vol. 6, no.1 February 1991.
- [15] R. D. Zimmerman, C. E. Murillo-Sánchez and D. Gan, Power Systems Engineering Research Center (PSERC).
- [16] School of Electrical Engineering, Cornell University, Ithaca N1997-2004, NY 14853. <http://www.pserc.cornell.edu/matpower>.
- [17] Eberhart, R. C. and Shi, Y. Particle swarm optimization: developments, applications and resources. Proc. congress on evolutionary computation 2001 IEEE service center, Piscataway, NJ., Seoul, Korea., 200