Load Flow Computation and Power Loss Reduction Using New Particle Swarm Optimization Technique

[1]Pruthviraja L, [2]Pradeepkumar F H, [3]Ashok M, [4]Mahesh Kumar B T

[1]Assistant Professor, [2,3,4]UG Students
[1]E&E Department, [2,3,4]EEE Department
Jain Institute of Technology-Davanagere, Karnataka, India

Abstract: **The paper presents a new algorithm, Particle Swarm Optimization (PSO) for optimization of power loss taking in to account the voltage limits. The proposed algorithm has been applied for standard 6-bus system. The algorithm applied shows better result compared to existing Newton Raphson method. The power flow study provides the system status in the steady-state and it is fundamental to the power system operation, planning and control. PSO is applied in a new computational model for the system power flow obtainment. This model searches for a better convergence, as well as a wider application in comparison with traditional methods as the Newton-Raphson method.**

Keywords **- Particle Swarm Optimization (PSO), NR method, Loss minimization, Evolutionary computation, Global best, Local best.**

I. INTRODUCTION

One of the important operating tasks of the power utilities is to keep the voltage level within the acceptable limits and to maintain the load requirements of the customer and also power quality has to be maintained. Electric power loads vary from hour to hour and voltage can be varied by change of the power load. Power utility operators in control centres handle various equipment such as generators, transformers, static condenser (SC), and shunt reactor (ShR), so that they can inject reactive power and control voltage directly in target power systems in order to follow the load change.

Particle Swarm Optimization (PSO) is a relatively new evolutionary algorithm that may be used to find optimal (or near optimal) solutions to numerical and qualitative problems. Particle Swarm Optimization was originally developed by a social psychologist James **Kennedy** and an electrical engineer Russell **Eberhart** in 1995, and emerged from earlier experiments with algorithms that modelled the flocking behaviour seen in many species of birds.

The main objective function is to compute the required load flow for the power system and to find out the loss by using this new technique and comparing this with the existing techniques. The control variables are generators bus voltages, transformer tap positions and switch-able shunt capacitor banks. The equality constraints are power/reactive power equalities, the inequality constraints include bus voltage constraints, generator reactive power constraints, reactive source reactive power capacity constraints and the transformer tap position constraints, etc. The equality constraints can be automatically satisfied by load flow calculation, while the lower/upper limit of control variables corresponds to the coding on the Particle Swarm Optimization (PSO) Algorithm, so the inequality constraints of the control variables are satisfied. The algorithm saves the time and memory space required to store the computational results in each iteration.

II. LOAD FLOW REVIEW

The primary function of an Electric Power System is to supply the power demand in an efficient, economic, high quality and reliable way . The power system can operate in an infinite number of states – voltage and power sets in the buses – in order to comply with standard requisites. The solution of a Power Flow problem consists in the determination of these possible operational states through the knowledge *a priori* of certain variables of the system buses. The objective of this kind of problem is to obtain the system buses voltages – module and angle – in order to determine later the power adjustments in the generation buses and the power flow in the system lines. The power flow study provides the system status in the steady-state that is, its parameters do not vary with the time variation.

Once the steady-state of the system known, it is possible to estimate the amount of power generation necessary to supply the power demand plus the power losses in the system lines, moreover the voltage levels must be kept within the boundaries and overloaded operations, besides the operations in the stability limit must be avoided . The general form of the Static Load Flow Equations (SLFE) is given by equation (1),

$$P_i - jQ_i - y_{i1} V_1 (V_i^*) - y_{i2} V_2 (V_i^*) - \dots - y_{in} V_n (V_i^*) = 0 \qquad (1)$$

Where: i = 1,..., n, bus number; Pi = active power generated or injected in the bus i; Q_i = reactive power generated or injected in the bus i; $|V_i|$ = voltage module of the bus i; δi = voltage angle of the bus i; $V_i = |V_i| ej\delta i$, i. e., the voltage in the polar form; y_{ik} = element of the nodal admittance matrix *Ybus.*

The nodal admittance matrix is obtained through the following explanation: if i = k, y_{ik} is the sum of the admittances that come out of the bus i; and if $i \neq k$, y_{ik} is the admittance between the buses i and k, multiplied by -1.

The power system buses are classified in types, according to the variables known *a priori* and to the variables that will be obtained through the SLFE.

- Type 1 Bus OR PQ Bus: Pi & Qi are specified and $|Vi|$ and δi are obtained through the SLFE;
- Type 2 Bus OR PV Bus: Pi & $|Vi|$ are specified and Qi and δi are obtained through the SLFE;
- Type 3 Bus OR Vδ Bus ("Slack Bus"): $|Vi|$ and δi are specified and Pi and Qi are obtained through the SLFE.

Equation (1) performs a complex and non-linear equations system, and its solution is obtained through approximations using numeric computational methods. In the existing Gauss

Seidel and Newton Raphson methods consist in the adoption of initial estimated values to the bus voltages, module 1,0 [pu] and angle 0 [rad], for instance, and in the application of the SLFE in successive iterations, searching for better approximations for the voltages. The stop criterion varies according to the required accuracy.

III. PARTICLE SWARM OPTIMIZATION

Kennedy and Eberhart developed a particle swarm optimization algorithm based on the behaviour of individuals (i.e., particles or agents) of a swarm. It has been noticed that members of the group seem to share information among them to lead to increased efficiency of the group. The particle swarm optimization algorithm searches in parallel using a group of individuals similar to other AI-based heuristic optimization techniques. Each individual corresponds to a candidate solution to the problem. Individuals in a swarm approach to the optimum through its present velocity, previous experience, and the experience of its neighbours. In a physical *n*-dimensional search space, the position and velocity of individual *i* are represented as the velocity vectors. Using these information individual *i* and its updated velocity can be modified under the following equations in the particle swarm optimization algorithm.

Velocity and position update at discrete intervals is given by

$$v_i(t+1)=v_i(t)+c1R1\left(p_i^{best}(t)-x_i(t)\right)+c2R2\left(p_g^{best}(t)-x_i(t)\right) \quad (2)$$

And

$$x_i(t+1)=x_i(t)+v_i(t+1) \quad (3)$$

Where R1 and R2 are random numbers uniformly distributed within [0, 1], 'i' indicates the local best and 'g' indicates the global best locations. The procedure for the PSO is shown in second flowchart. From the above two equations, we can identify that the PSO algorithm can find the best global set values for the optimal solution of the problem.

Because of the above advantage we can apply the procedure for loss minimization of power system, so that active power loss can be minimized and we can set all the inequality constraints to their best values which meet the required customer load without any loss in power and frequency and maintaining the efficiency of the system to a great. PSO uses a set of particles in which each one of them is a candidate to the solution of the treated problem. Such particles are distributed in an *n*-dimensional space, and each particle has a position and a velocity in each time instant. The best individual position of a particle is defined as *local best*, and the best position of all the particles is defined as *global best*. The PSO particles have knowledge about their performances and about their neighbour's performances. The interaction between the particles and the environment they are inserted is made by the *rule function*, which is related to the problem modeling. The search of best position requires a strong attraction of particles towards their bounds, so that the new parameter called *inertia weight* is used in (2), the velocity now became as shown below:

$$\mathbf{v_i(t+1) = w(t)v_i(t) + c1R1\,(p_j(t)-x_i(t)) + c2R2\,(p_g(t)-x_i(t))} \quad (4)$$

In general, a linearly decreasing scheme for *w* can be mathematically described as follows:

$$w(t)=w_{up}-(w_{up}-w_{low})*\{t/T_{max}\}, \quad (5)$$

where *t* stands for the iteration counter; w_{low} and w_{up} are the desirable lower and upper bounds of *w*; and T_{max} is the total allowed number of iterations. Equation (4) produces a linearly decreasing time-dependent inertia weight with starting value, w_{up}, at iteration, $t = 0$, and final value, w_{low}, at the last iteration, $t = T$max.

The basic PSO pseudocode for random uniform initialization.

Input: Number of particles *N*, dimension *n*, velocity bounds [-*v*max, *v*max], and search space, $A = [a1, b1] \times [a2, b2] \times ... \times [an, bn]$

Step 1. **Do** ($i = 1…N$)
Step 2. **Do** ($j = 1…n$)
Step 3. **Set** particle component $xij = a_j + \text{rand}(\,)*(b_j-a_j)$.
Step 4. **Set** best position component $p_{ij} = x_{ij}$.
Step 5. **Set** velocity component $v_{ij} = -v\text{max} + 2\,\text{rand}(\,)*v\text{max}$.
Step 6. **End Do**
Step 7. **End Do**

Where a_i=lower limits of swarms, b_i= upper limits of swarms.

If there are items of information available regarding the location of the global minimize in the search space, it makes more sense to initialize the majority of the swarm around it.

IV. PSO METHODOLOGY APPLIED TO LOAD FLOW

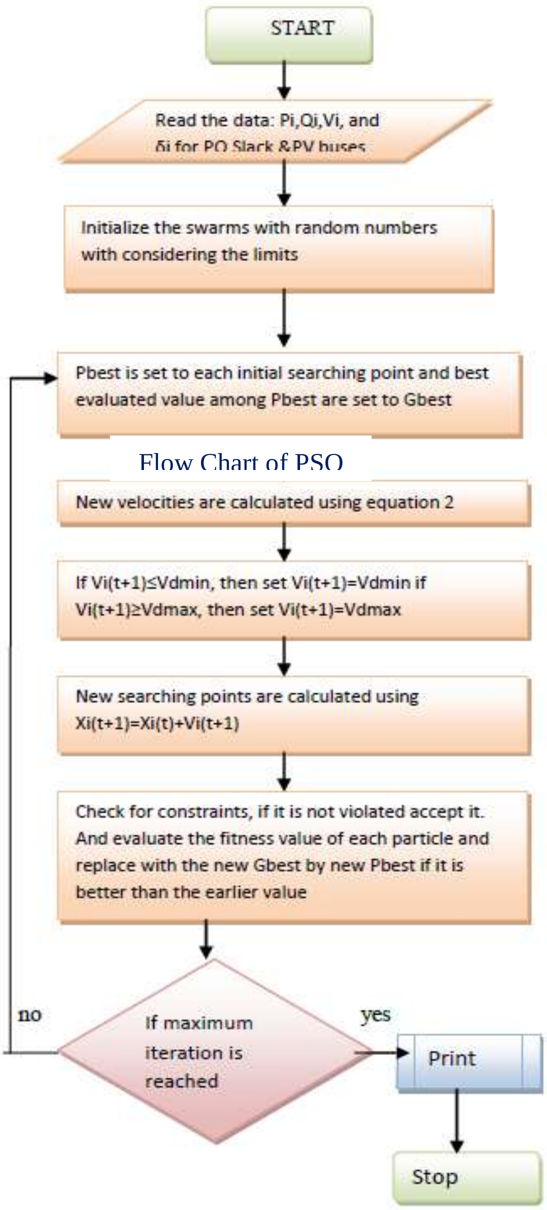
Before the initialization of the module value of each particle, the bus type needs to be verified and related in the equation. In the case of a PQ bus, the voltage module receives a random value within the specified boundary; for a PV bus, the voltage module receives the related value specified in the input data. The initial velocities are null. The local best parameters receive the particles positions values and the global best parameter receives the first particle value, arbitrarily. The grades are initialized with high values in order to be minimized later. Having that accomplished, the iterations are initialized. The following process is accomplished to each particle of the swarm. Firstly the buses voltages receive the particles positions. The reactive power of the PV buses is calculated using equation (1), then the active and reactive power of the slack bus are also calculated using this equation. Finally the power flow in the system lines is calculated in accordance to the equation .

$$S_{ij} = P_{ij} - jQ_{ij} = V_i(V_i * -V_j^*)Y_{ij} * -V_iV_i * Y_{sh},i \quad (6)$$

Where: S_{ij} = complex apparent power between the buses *i* and *j*; *Pij* = active power between the buses *i* and *j*; Q_{ij} = reactive power between the buses *i* and *j*; V_i = bus *i* voltage; V_j = bus *j* voltage; Y_{ij} = admittance between the buses *i* and *j*; *Ysh,i =shunt* admittance of the bus *i*.

Thus once all the power of the buses and of the lines is known, the active and reactive power mismatches of each bus are calculated. They are calculated as the sum of the injected power in the approached bus. The apparent power mismatches arithmetic mean is obtained, and this is the value that is desired to be minimized. The local best is replaced by the current particle position in case of the particle current grade is considered better than the local grade. Thus, after all the particles pass through the described process, a similar criterion is used to the global best updating. Next each particle is verified in the following criteria: whether the local grade or global grade is best, the best global is replaced by the approached best local. The velocities as well as the particles positions are updated. The pseudo code for the PSO algorithm is reported as follows:

Input: N, c1, c2, x_{min}, x_{max} (lower and upper bounds), f(x) (objective function), N=population size.
Step 1. **Set** $t \leftarrow 0$.
Step 2. **Initialize** $xi(t)$, $vi(t)$, $pi(t)$, $i = 1, 2,…, N$.
Step 3. **Evaluate** $f(xi(t))$, $i = 1, 2,…, N$.
Step 4. **Update** indices, g_i, of best particles.
Step 5. **While** (stopping condition not met)
Step 6. **Update** velocities, $v_i(t+1)$, and particles, $xi(t+1)$, $i = 1, 2,…, N$.
Step 7. **Constrain** particles within bounds [x_{min}, x_{max}].
Step 8. **Evaluate** $f(xi(t+1))$, $i = 1, 2,…, N$.
Step 9. **Update** best positions, $pi(t+1)$, and indices, g_i.



Flow Chart of PSO

Step 10. **If** (local search is applied) **Then**
Step 11. **Choose** a position $p_q(t+1)$, $q \in \{1, 2,\ldots, N\}$,
Step 12. **Apply** local search on $p_q(t+1)$ and obtain a solution, y.
Step 13. **If** $(f(y) < f(p_q(t+1))$ **Then** $p_q(t+1) \leftarrow y$.
Step 14. **End If**
Step 15. **Set** $t \leftarrow t+1$.
Step 16. **End While**

Depending on the problem formulation the changes will take place in the above algorithm. The parameter values also depend on the velocity limits and the inertia weight. Normally c1 and c2 are kept constant.

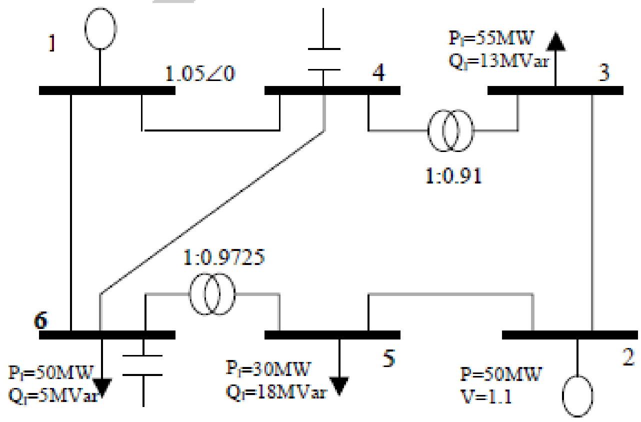
PSO Algorithm is as follows:

The basic elements of the PSO techniques are briefly stated and defined as follows:

a) **Particle X_i(t):** It is a candidate solution represented by a k-dimensional real-valued vector, where k is the number of optimized parameters. At time t, the i^{th} particle Xi(t) can be described as $Xi(t)=[x_{i,1}(t); x_{i,2}(t); \ldots\ldots; x_{i,k}(t)]$.
b) **Population:** it is a set of n particles at time t.
c) **Swarm and its direction:** it is an apparently disorganized population of moving particles that tend to cluster together while each particle seems to be moving in a random direction.
d) **Particle velocity V (t):** It is the velocity of the moving particles represented by a k-dimensional real-valued vector. At time t, the i^{th} particle Vi (t) can be described as $Vi(t)=[v_{i,1}(t); v_{i,2}(t); \ldots\ldots; v_{i,k}(t)]$.
e) **Inertia weight w(t):** it is a control parameter that is used to control the impact of the previous velocity on the current velocity. All the control variables transformer tap positions and switch-able shunt capacitor banks are integer variables and not continuous variables. Therefore, the value of the inertia weight is considered to be 1 in this study.
f) **Individual best X^* (t):** As the particle moves through the search space, it compares its fitness value at the current position to the best fitness value it has ever attained at any time up to the current time. The best position that is associated with the best fitness encountered so far is called the individual best X^* (t). For each particle in the swarm, X^* (t) can be determined and updated during the search.
g) **Global best X^{**} (t):** It is the best position among all of the individual best positions achieved so far.
h) 8. **Stopping criteria:** These are the conditions under which the search process will terminate. In this study, the search will terminate if one of following criteria is satisfied:
   1. The number of the iterations since the last change of the best solution is greater than a pre-specified number.
   2. The number of iterations reaches the maximum allowable number.

V. *EXAMPLE ANALYSIS*

The proposed algorithm has been run using Matlab-2011b for IEEE 6-bus system[1] as shown below. Bus 1 is the swing bus, bus 2 is a PV bus, while Bus 3 and 6 are reactive power installation buses. The two branches with tap-setting transformers are branches 1-4 and 6-5. The line data, the control variables constraints, and the state variable constrains for the IEEE 6-Bus system are shown in Tables 1, 2 and 3.



IEEE-6 bus system

Table 1: IEEE 6-BUS SYSTEM DATA (p.u.)

Start Bus	End Bus	Branch Impedance	Transformer Tap
6	3	0.123+0.518j	
6	4	0.080+0.370j	
4	3	0.097+0.407j	
5	2	0.282+0.640j	
2	1	0.723+1.050j	
6	5	0.000+0.300j	0.9725
4	3	0.000+0.133j	0.9100

Table 2: CONTROL VARIABLE CONSTRAINTS

	Transformer Tap	Generator Bus Voltage		VAR Installation (MVAR)	
	T_{65}, T_{43}	V_1	V_2	Q_6	Q_4
Lower Limit	0.910	1.0	1.1	0.0	0.0
Upper Limit	1.110	1.1	1.15	5.0	5.0
Discrete Value	0.91+16*1.25%			10*0.5	10*0.5

In table 2 the incremental steps are multiplied with the total discrete size of the control variables. The method has been applied to standard IEEE-6 bus system. And also compared with NR method, the reference paper [1], and with the present applied method. The results are shown in below Table 4. By observing this table, we can identify that the applied method has given the best result while others are with little variation.

Table 3: STATE VARIABLE CONSTRAINTS

	PQ Bus Voltage	PV Bus Reactive Power (MVAR)
Lower Limit	0.9	-20
Upper Limit	1.1	100

The obtained result and the comparison is tabulated in Table 4. The applied method has given the good values of voltages and it minimized the losses.

<table>
<tr><th rowspan="2">Method</th><th colspan="6">Total Power transmission loss</th></tr>
<tr><th colspan="3">Real Power (MW)</th><th colspan="3">Reactive Power (Mvar)</th></tr>
<tr><td>NR method</td><td colspan="3">9.655</td><td colspan="3">36.707</td></tr>
<tr><td>PSO method [1]</td><td colspan="3">8.720</td><td colspan="3">—</td></tr>
<tr><td rowspan="2">PSO,present paper</td><td>Average</td><td>Best</td><td>Worst</td><td>Average</td><td>Best</td><td>Worst</td></tr>
<tr><td>8.70</td><td>8.716</td><td>9.5</td><td>28.55</td><td>26.247</td><td>31.655</td></tr>
</table>

TABLE 4: COMPARISON OF RESULT FOR DIFFERENT METHODS

VI. CONCLUSION

This paper presented a new PSO algorithm; the program has been modified by new idea in order to get the converged results using a new PSO algorithm. The developed PSO algorithm provides a flexibility to add or delete any system constraints and objective functions. Having this flexibility will help electrical Engineers analyzing other system scenarios and contingency plans. The method had superior computational efficiency and better convergence in less iterations. It was suitable for reactive power and voltage integrated control of power system greatly. In this algorithm all cases, i.e. PQ, PV and slack/reference buses are considered and the results shows the acceptable values. The major advantage of the proposed method is that, it is simple in programming and takes less time to get converge.

REFERENCES

[1] H. Mantawy, M. S. Al-Ghamdi Electrical Engineering Department King Fahd University of Petroleum & Minerals on "A New Reactive Power Optimization Algorithm" , Dhahran 31261 Saudi Arabia, 2003 IEEE Bologna Power Tech Conference, June 23th-26th, Bologna, Italy.

[2] Numphetch Sinsuphun, Uthen Leeton, Umaporn Kwannetr, Dusit Uthitsunthorn, and Thanatchai Kulworawanichpong, on "Loss Minimization Using Optimal Power Flow Based on Swarm Intelligences", ECTI TRANSACTIONS On Electrical Eng., Electronics, And Communications Vol.9, No.1 February 2011.

[3] Pathak Smita, Prof. B.N.Vaidya, on "Optimal Power Flow by Particle Swarm Optimization for Reactive Loss Minimization" International Journal of Scientific & Technology Research Volume 1,Issue 1,Feb 2012 **ISSN 2277.8616.**

[4] Hirotaka Yoshida Yoshikazu Fukuyama on "particle swarm optimization For reactive power and voltage control Considering voltage security assessment", IEEE Trans. on Power Systems, Vol.15, No.4, pp.1232-1239, November 2001, presented at IEEE PES Winter Meeting, Columbus, Jan.28-Feb.1, 2001.

[5] Ahmed A. A. Esmin, Germano Lambert-Torres, and Antônio C. Zambroni de Souza, *on* "A Hybrid Particle Swarm Optimization Applied to Loss Power Minimization", IEEE TRANSACTION On Power System,Vol. 20.N0 2 , MAY, 2005.