*International Journal of Innovative Computing, Information and Control*
Volume **8**, Number **3(A)**, March **2012**
ICIC International ©2012 ISSN 1349-4198
pp. 1705–1716

# APPLICATION OF PARTICLE SWARM OPTIMIZATION TO OPTIMAL POWER SYSTEMS

Ahmed A. A. Esmin[1] and Germano Lambert-Torres[2]

[1]Department of Computer Science
Federal University of Lavras
Lavras, MG 37200-000, Brazil
ahmed@dcc.ufla.br

[2]Department of Electrical Engineering
Federal University of Itajuba
Itajuba 37500-503, Brazil
germano@unifei.edu.br

Received December 2010; revised May 2011

Abstract. *This paper presents the particle swarm optimization (PSO) algorithm for solving the optimal distribution system reconfiguration problem for power loss minimization. The proposed methodology determines control variable settings, such as the number of shunts to be switched, for real power loss minimization in the transmission system. The problem is formulated as a nonlinear optimization problem. The PSO is a relatively new and powerful intelligent evolution algorithm for solving optimization problems. It is a population-based approach. The proposed approach employs the PSO algorithm for the optimal setting of optimal power flow (OPF) based on loss minimization (LM) function. The proposed approach has been examined and tested on standard IEEE 14, IEEE 30 and IEEE 118 bus test systems. The obtained results are compared with those using other techniques in a previous work to evaluate the performance.*
**Keywords:** PSO, Evolutionary algorithm, Optimal power flow, Loss power minimization

## 1. Introduction.

The subject of minimizing distribution systems losses and the problem of optimal power flow (OPF) have gained a lot of attention due to the high cost of electric energy; therefore, much of current research on distribution automation is focused on the minimum loss configuration problem. There are many alternatives available for reducing losses at the distribution level: reconfiguration, capacitor installation, load balancing, and introduction of higher voltage levels.

The OPF problem solution aims at optimizing specific objective functions such as loss of power by adjusting the power control variables and at the same time satisfying the equality and the inequality constraints. The inequality constraints are the upper and the lower limits at the control and some state variables, while the equality constraints are the power flow equations.

A number of mathematical optimization techniques have been proposed in literature to solve the OPF problem. For decades, conventional optimization techniques such as linear programming (LP), quadratic programming (QP), gradient method, Newton method and interior point methods have been used for solving the optimal reactive power dispatch problem [1-4]. Even though these methods present some drawbacks, they provide, in general, satisfactory performance. This paper analyzes, once more, the problem of loss reduction.

The OPF problem is a nonlinear optimization problem. Nonlinear optimization problems, however, may require a more complex formulation, as the set of equations involved may not be linearized. In this case, nonlinear techniques must be employed. One of the nonlinear problems analyzed in power systems is in regard to loss reduction [5], which considers, as control variables, the active power generations at the machines, generators, voltage level, tap settings and AVR sources. This nonlinear problem may be important for optimal power flow and for voltage stability analysis or merely to improve the system operating conditions [5].

In recent years, metaheuristic methods have been studied for solving combinatorial optimization problems to obtain an optimal solution of global minimum. Typical metaheuristic methods include simulated annealing (SA), genetic algorithm (GA) and tabu search (TS) [6-9].

In this paper, the tool that was used to analyze the loss reduction problem belongs to the family of the evolutionary algorithms. The stochastic technique, known as particle swarm optimization (PSO), is employed.

The application of this tool in power systems is wide ranging. For example, [9] focuses on the problems of fuel cost minimization, voltage profile improvement and voltage stability enhancement. In [8], the hybrid model was employed for loss power minimization.

The PSO is a relatively new and powerful intelligence evolution algorithm for solving optimization problems. It is a population-based approach [10-12].

The PSO was originally inspired by the social behavior of bird flocks and fish schools. It was observed that they take into consideration the global level of information to determine their direction. Hence, the global and the local best positions are computed at each instant of time (iteration), and the output is the new direction of search. Once this direction is detected, it is followed by the cluster of birds.

In this paper, a PSO-based approach is proposed to solve the loss reduction problem by the shunt capacitor installation. This is done in two phases: first, the critical area of the power system is identified using the tangent vector technique [5]; second, the PSO techniques are used to optimize the amount of shunt reactive power compensation in each bus. This model is developed and integrated into a previously written load flow program.

The remainder of the paper is organized as follows. In Section 2, a brief introduction of the PSO is given. The problem formulation is described in Section 3. Section 4 discusses the experimental results. Finally, the paper is concluded in Section 5.

## 2. The Particle Swarm Optimization.

**2.1. Overview.** The particle swarm optimization algorithm (PSO) is a population-based optimization method that was first proposed by Kennedy and Eberhart [10]. The PSO technique finds the optimal solution using a population of particles. Each particle represents a candidate solution to the problem. PSO is basically developed through the simulation of bird flocking in two-dimensional space. Some of the attractive features of the PSO include ease of implementation and the fact that no gradient information is required. It can be used to solve a wide array of different optimization problems; some example of applications include neural network training and function minimization.

**2.2. PSO algorithm definition.** The PSO definition is presented as follows:

1) Each individual particle $i$ has the following properties: A current position in search space, $x_i$, a current velocity, $v_i$, and a personal best position in search space, $y_i$.

2) The personal best position, $y_i$, corresponds to the position in search space where particle $i$ presents the smallest error as determined by the objective function $f$, assuming a minimization task.

3) The global best position represents the position yielding the lowest error among all the $y_i$.

Equations (1) and (2) define how the personal and the global best values are updated at time $t$, respectively. In the following, it is assumed that the swarm consists of s particles, Thus $i \in 1 \ldots s$.

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(y_i(t) \leq f(x_i(t+1))) \\ x_i(t+1) & \text{if } f(y_i(t) > f(x_i(t+1))) \end{cases} \tag{1}$$

$$\begin{aligned} &\hat{y}(t) = \min\{f(y), f(\hat{y}(t))\} \\ &y \in \{y_0(t), y_1(t), \ldots, y_s(t)\} \end{aligned} \tag{2}$$

During each iteration, every particle in the swarm is updated using Equations (3) and (4).

Two pseudorandom sequences, $r_1 \sim U(0,1)$ and $r_2 \sim U(0,1)$ give the stochastic nature of the algorithm. For all dimensions $j \in 1 \ldots n$, let $x_i;\ j$, $y_i;\ j$ and $v_i;\ j$ be the current position, the current personal best position, and the velocity of the $j$th dimension of the $i$th particle. The velocity update step is:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,j}(t)[y_{i,j}(t) - x_{i,j}(t)] + c_2 r_{2,j}(t)[\breve{y}_j(t) - x_{i,j}(t)] \tag{3}$$

The new velocity is then added to the current position of the particle to obtain its next position:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{4}$$

The value of each dimension of every velocity vector $v_i$ is clamped to the range $[-v_{\max}, v_{\max}]$ to reduce the likelihood of the particle leaving the search space. The value of vmax is usually chosen to be:

$$v_{\max} = k \times x_{\max}, \text{ where } 0.1 \leq k \geq 1.0,$$

where $x_{\max}$ denotes the domain of the search space. Note that this does not restrict the values of $x_i$ to the range $[-v_{\max}, v_{\max}]$. Rather than that, it merely limits the maximum distance that a particle will move.

The acceleration coefficients, $c1$ and $c2$, control how far a particle will move in a single iteration. Typically, both these are set to a value of 2.0, although it has been shown that setting c1 $\neq c2$ can lead to a good performance [10]. The inertia weight, $w$, in (3), is used to control the convergence behavior of the PSO. Small values of $w$ usually result in a more rapid convergence on a suboptimal position, while too large a value may prevent divergence. Typical implementations of the PSO adapt the value of $w$ during the training stage, for example, linearly decreasing it from 1.0 to near 0 during the execution. Convergence can be obtained with fixed values as shown by Kennedy in [10]. In general, the inertia weight $w$ is set according to the following equation:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \cdot iter \tag{5}$$

where $iter_{\max}$ is the maximum number of iterations and $iter$ is the current iteration number.

The PSO system combines two models: a social-only model and a cognition-only model [10]. These models are represented by the velocity update, shown in Equation (3). The second term in the velocity update equation, $c_1 r_{1,j}(t)[y_{i,j}(t) - x_{i,j}(t)]$, is associated with cognition as it only takes into account the particle's own experiences. The third term in the velocity update equation, $c_2 r_{2,j}(t)[\breve{y}_j(t) - x_{i,j}(t)]$, represents the social interaction between the particles. It suggests that individuals ignore their own experience and adjust their behavior according to their perception of successful individuals in the neighborhood.

Figure 1 lists the pseudo-code for the basic PSO algorithm.

```
Create and initialize:
        i – current particle;
        s – PSO of n-dimensional :

Start
repeat:
        for ecah particle    i = [1..s]
                if f(s.x_i) < f(s. y_i)
                        then s. y_i = s.x_i
                if   f(s.y_i) < f(s. ŷ)
                        then s.  ŷ  = s. y_i
        end-for
        Update    s using equations    (3e 4)
until   the    stop condition is true
end
```

FIGURE 1. The PSO algorithm

The initialization step mentioned in the algorithm consists of the following:

1. Initialize each coordinate $x_{i,j}$ to a value from the uniform random distribution on the interval $[-x_{\max}, x_{\max}]$, for all $i \in 1 \ldots s$ and $j \in 1 \ldots n$. It is important to note that the choice of good random algorithms has a direct impact on the good initial distribution of the particles along the search space.
2. Initialize each $v_{i,j}$ to a value drawn from the uniform random distribution on the interval $[-v_{\max}, v_{\max}]$, for all $i \in 1 \ldots s$ and $j \in 1 \ldots n$. Alternatively, the velocities of the particles could be initialized to 0, as the starting positions are already randomized.

The stopping criterion mentioned in the aforementioned algorithm depends on the type of problem being solved. Usually, the algorithm is run for a fixed number of iterations (objective function evaluations) or until a specified error bound is reached.

It is important to note that the velocity term models the rate of change in the position of the particle. The changes induced by the velocity update Equation (2) therefore represent acceleration.

The description of how the algorithm works is as follows: Initially, based on particle fitness information, some particle is identified as the best particle. Then, all the particles are accelerated in the direction of this particle, but at the same time in the direction of their own best previously encountered solutions. Occasionally, the particles will overshoot their target, exploring the search space beyond the current best particles. All particles also have the chance to discover better particles en route, in which case the other particles will change direction and head toward the new best particle. Because most functions have some continuity, chances are that a good solution will be surrounded by equally good, or better, solutions. By approaching the current best solution from different directions in search space, the chances that these neighboring solutions will be discovered by some of the particles are good [13].

The next section illustrates how this method can be applied in a power system.

3. **The Problem Formulation.** The LM mode of the OPF problem can be formulated as follows:

$$\begin{aligned} \text{Minimize} \quad & f(x,\sigma) \\ \text{subject to} \quad & g(x,\sigma) = 0, \\ \text{and} \quad & h(x) < 0,\ x_l < x < x_u \end{aligned} \tag{6}$$

where $f(x)$ are scalar functions, the sum of branch losses in an area of interest; $g(x)$ are functional equality constraints; the power flow equations; $h(x)$ are functional inequality constraints and the limits on the control variables; and $x$ are the state variable vectors that consist of both the control and the dependent variables (voltage magnitudes and angles, shunt susceptances, active power generations, etc.); $\sigma$ is the system loading parameter; whereas the lower and the upper limits are $x_l$ and $x_u$.

The solution process of the LM consists of optimizing the objective function and satisfying the following constraints:

1) Power flow equations;
2) Branch flow limits;
3) Bus voltage limits;
4) Control variable limits.

3.1. **Stopping criteria and power flow program limits.** Several stopping criteria could be used. One option was a combination of number of iterations and errors in the loss value. In addition, penalty is used, because in power flows with all the system limits, and if some violation is observed, an error is computed. This approach is usually employed in genetic algorithms (GAs). Several limits have been considered in the power flow program [15], for instance: (a) for transmission lines loading, all overloads flagged during the power flow computation are incorporated in the Jacobian matrix and solved by generator redispatch; (b) for LTC tap blocking, the problem has been handled by blocking the LTC tap changes when this action prevents several problems for the system and allows recovery of a larger amount of constant impedance load; (c) for loss reduction—it is encountered by computation of generator redispatch or shunt compensation, with a larger load margin; (d) for load shedding, solved by an augmented Jacobian, it is made in the system critical bus with the purpose of controlling the undervoltages.

3.2. **Critical bus identification.** The technique used in this paper is to reduce the active power loss in a critical area under the point of view of voltage instability. This critical area is identified with tangent vector help. Such a vector is given by Equation (7), and the reader is referred to [8].

$$\begin{bmatrix} \Delta\theta_g \\ \Delta\theta_l \\ \Delta V_l \end{bmatrix} \frac{1}{\Delta\sigma} = [J]^{-1} \begin{bmatrix} P_{go} \\ P_{lo} \\ Q_{lo} \end{bmatrix} \tag{7}$$

This vector shows how the state variables change as a function of a system parameter variation, and its largest entries indicate the buses most likely to drive the system to voltage collapse. The application of this technique to loss sensitivity studies is proposed in [5].

4. **Methodology and Simulation Results.** In this section, the practical results associated with PSO applied to LM are obtained. The proposed PSO approach for LM is tested on standard IEEE 14 [16], IEEE 30 [17] and IEEE 118 [18] bus test systems. The IEEE test systems have been designed to incorporate all characteristics of real power systems in a concentrated system and provide a unique system for comparisons among

different strategies. The control strategy used is based on shunt compensation. Table 1 gives the details of the test systems.

Table 1. Description of test systems

| Description | IEEE 14 | IEEE 30 | IEEE 118 |
|---|---|---|---|
| No. of buses $N_B$ | 14 | 30 | 118 |
| No. of generators $N_G$ | 5 | 6 | 54 |
| No. of transformers $N_T$ | 3 | 4 | 9 |
| No. of shunts $N_{sh}$ | 2 | 2 | 12 |
| No. of branches $N_l$ | 20 | 41 | 186 |
| No. of equality constraints | 28 | 60 | 236 |
| No. of inequality constraints | 65 | 125 | 566 |
| No. of control variables | 10 | 12 | 75 |
| No. of discrete variables | 5 | 6 | 21 |

A comparative study with the IP primal-dual interior-point algorithm (IP) [9] was done to verify the performance of the proposed algorithm. The PSO and the IP algorithms were implemented using MATLAB 6.12 [19] running on PC Pentium (Dual Core) with 2GB.

In the IEEE 14-bus system shown in Figure 2, there are 14 buses, out of which 5 are generator buses. Bus 1 is the slack bus; 2, 3, 6 and 8 are taken as PV generator buses; and the rest are PQ load buses. The network has 20 branches, 17 of which are transmission lines and 3 are tap-changing transformers. It is assumed that capacitor compensation is available at buses 9 and 14. Totally, there are nine control variables, which consist of four PV generator voltages, three tap-changing transformers with 20 discrete steps of 0.01 p.u. each, and two shunt compensation capacitor banks with three discrete steps of 0.06 p.u. each.

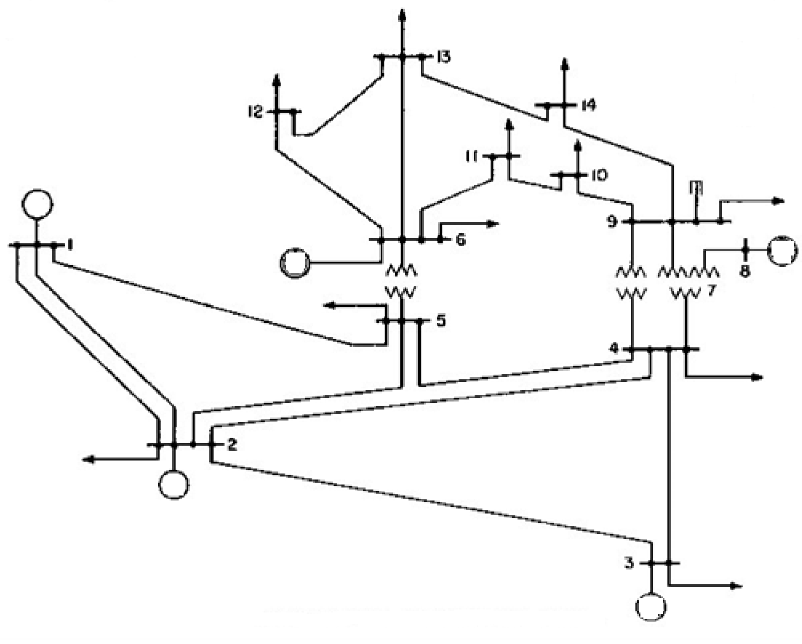


Figure 2. Network diagram of IEEE 14-bus system

To calculate the amount of shunt capacitor to be installed, the following methodology is used: Initially, using the tangent vector and the loss sensitivity technique, the critical area of the power system is identified. Once this area is identified, the PSO technique is used to optimize the amount of shunt reactive power compensation that should be available in each bus.

The results obtained by PSO are compared with those achieved using primal-dual interior-point algorithm (IP) presented in a previous work [15], where a similar methodology was used. Many simulations have been done using PSO and the results are discussed in more detail later.

The PSO parameters used are given in Table 2:

Table 2. The PSO simulation parameters

| | |
|---|---|
| Population numbers | 5, 10 e 15 |
| Iteration | 100 |
| Wmax | 0.9 |
| Wmin | 0.2 |
| C1; C2 | 1 |

Table 3 shows the critical buses for the IEEE 14, IEEE 30 and the IEEE 118-bus systems. These results are obtained by executing the tangent vector and the loss sensitivity program.

Table 3. Critical buses

| System | The Critical Buses | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| IEEE 14 | 14 | 13 | 12 | 10 | | | | |
| IEEE 30 | 30 | 29 | 26 | 19 | 24 | 18 | | |
| IEEE 118 | 41 | 39 | 33 | 117 | 35 | 43 | 2 | 3 |

Once the bus candidates for shunt compensation have been identified, the optimization process takes place. The optimal amount of shunt compensation and the associated power loss are the results of interest at this stage. For each bus set, the amount of shunt reactive compensation suggested by the optimization program is implemented, and the load flow is executed. This process in continued until the maximum number of iterations is reached or a stop criterion is satisfied.

4.1. **IEEE 14-bus:** Tables 4 and 5 show the test results for the IEEE 14-bus system. The former shows the results obtained with the LM using IP and the latter the results using the PSO algorithm. The graphics in Figure 3 illustrates the behavior of the PSO with pop = 5, 10 and 15 with 100 iterations.

Table 4. Results obtained by IP–IEEE 14

| | |
|---|---|
| Iteration | 6 |
| Initial Loss (p.u) | 0.091 |
| Shunt Compensat. (p.u) | 0.10215 |
| Final Loss (p.u) | 0.0903 |
| Total Reduction (p.u) | 0.0005 |
| Simulation Time (s) | 0.89 |

Table 5. Results obtained by PSO IEEE 14

| Population Number = 15 - Initial Loss (p.u) = 0.09099 | | | | |
|---|---|---|---|---|
| Iteration | Final Loss (p.u) | Total Redu. (p.u) | Simulation Time (s) | Shunt Comp. (p.u) |
| 5 | 0.09046721 | 0.00062296 | 4.80 | 0.09582646 |
| 25 | 0.09034439 | 0.00064578 | 25.03 | 0.10215723 |
| 50 | 0.09033803 | 0.00065213 | 51.11 | 0.10346126 |
| 75 | 0.09033797 | 0.00065219 | 76.89 | 0.10342244 |
| 100 | 0.09033795 | 0.00065222 | 101.41 | 0.10341871 |

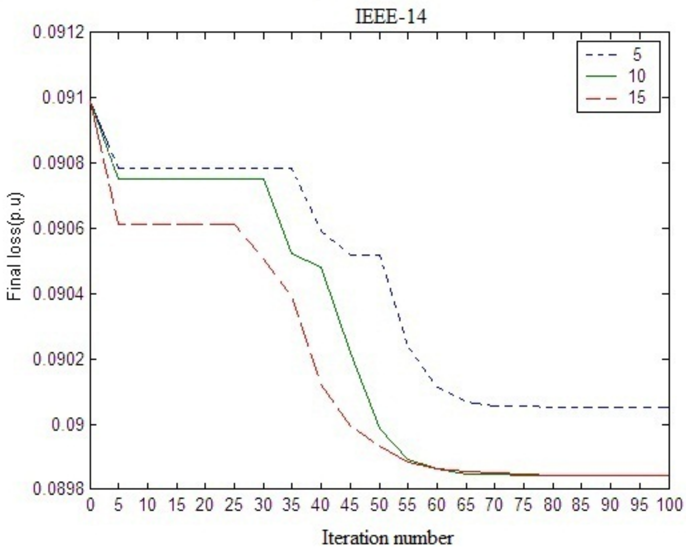


Figure 3. Graphic of PSO behavior for IEEE-14 system

4.2. **IEEE 30-bus.** Tables 6 and 7 show the test results for the IEEE 30-bus system. The former shows the results obtained with the LM using IP and the latter the results using the PSO algorithm. Table 8 shows a summary result using PSO with a different population numbers (5, 10 and 15) and with 100 iterations.

The graphics in Figures 4 and 5 illustrate the behavior of the PSO using pop = 5, 10 and 15 with 100 iterations. The Figure 5 shows the graphics of the PSO behavior for the IEEE-30 system with zoom-starting from 5a iteration.

Table 6. Results obtained by IP–IEEE 30

| | |
|---|---|
| Iteration | 6 |
| Initial Loss (p.u) | 0.1896 |
| Shunt Compensat. (p.u) | 0.6196 |
| Final Loss (p.u) | 0.1840 |
| Total Reduction (p.u) | 0.0056 |
| Simulation Time (s) | 4.96 |

4.3. **IEEE 118-bus.** Tables 9 and 10 show the test results for the IEEE 118-bus system. The former shows the results obtained with the LM using IP, and the latter the results using the PSO algorithm. The graphics in Figure 6 illustrate the behavior of the PSO with pop = 5, 10 and 15 with 100 iterations.

Table 7. Results obtained by PSO/IEEE-30

| Population Number = 15 - Initial Loss (p.u) = 0.18962255 | | | | |
|---|---|---|---|---|
| Iteration | Final Loss (p.u) | Total Redu. (p.u) | Simul. Time (s) | Shunt Comp. (p.u) |
| 5 | 0.18164544 | 0.00797711 | 12.17 | 0.26565748 |
| 10 | 0.18158854 | 0.00803401 | 23.45 | 0.28925055 |
| 15 | 0.18158854 | 0.00803401 | 35.89 | 0.28925055 |
| 20 | 0.18158854 | 0.00803401 | 47.25 | 0.28925055 |
| 25 | 0.18158854 | 0.00803401 | 59.02 | 0.28925055 |
| 30 | 0.18158854 | 0.00803401 | 70.97 | 0.28925055 |
| 35 | 0.18158854 | 0.00803401 | 81.70 | 0.28925055 |
| 40 | 0.18151395 | 0.00810860 | 92.14 | 0.28608188 |
| 45 | 0.18136576 | 0.00825679 | 102.20 | 0.28574470 |
| 50 | 0.18136273 | 0.00825982 | 112.34 | 0.28806400 |
| 55 | 0.18133842 | 0.00828413 | 122.47 | 0.28565299 |
| 60 | 0.18131220 | 0.00831035 | 132.53 | 0.28630616 |
| 65 | 0.18128750 | 0.00833505 | 142.63 | 0.28419335 |
| 70 | 0.18127340 | 0.00834915 | 152.73 | 0.28593340 |
| 75 | 0.18127150 | 0.00835105 | 162.77 | 0.28611939 |
| 80 | 0.18126746 | 0.00835509 | 172.86 | 0.28566032 |
| 85 | 0.18126433 | 0.00835822 | 182.91 | 0.28621453 |
| 90 | 0.18126267 | 0.00835988 | 192.94 | 0.28629764 |
| 95 | 0.18126213 | 0.00836042 | 203.02 | 0.28632045 |
| 100 | 0.18126211 | 0.00836044 | 213.05 | 0.28632191 |

Table 8. Summary of the results obtained by PSO/IEEE 30 with population (5,10 and 15) and 100 iterations

| POP/Iter | Final Loss (p.u) | Total Redu. (p.u) | Shunt Comp. (p.u) |
|---|---|---|---|
| 5 /100 | 0.18126211 | 0.00836044 | 0.28632191 |
| 10/100 | 0.18139671 | 0.00822584 | 0.28629105 |
| 15/100 | 0.18126211 | 0.00836044 | 0.28632191 |

Table 9. Results obtained by IP–IEEE 118

| | |
|---|---|
| Initial Loss (p.u) | 1.9728 |
| Shunt Compensat. (p.u) | 0.6196 |
| Final Loss (p.u) | 1.9710 |
| Total Reduction (p.u) | 0.0018 |
| Simulation Time (s) | 285.3 |

Table 10. Results obtained by PSO IEEE 118

| Population Number = 15 - Initial Loss (p.u) = 0.18962255 | | | | |
|---|---|---|---|---|
| Iteration | Final Loss (p.u) | Total Redu. (p.u) | Time (s) | Shunt Comp. (p.u) |
| 5 | 1.97026274 | 0.00257382 | 75.06 | 0.81632292 |
| 25 | 1.96856163 | 0.00427493 | 391.95 | 1.49754764 |
| 50 | 1.96855999 | 0.00427657 | 756.08 | 1.49829563 |
| 75 | 1.96855999 | 0.00427657 | 1119.19 | 1.49829565 |
| 100 | 1.96855695 | 0.00427960 | 1480.64 | 1.49811538 |

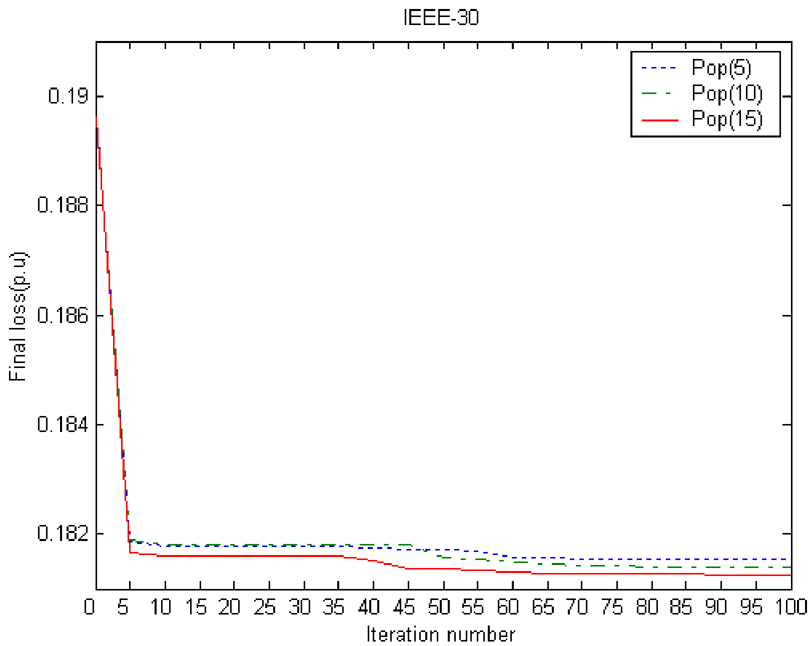


Figure 4. Graphic of PSO behavior for IEEE-30 system

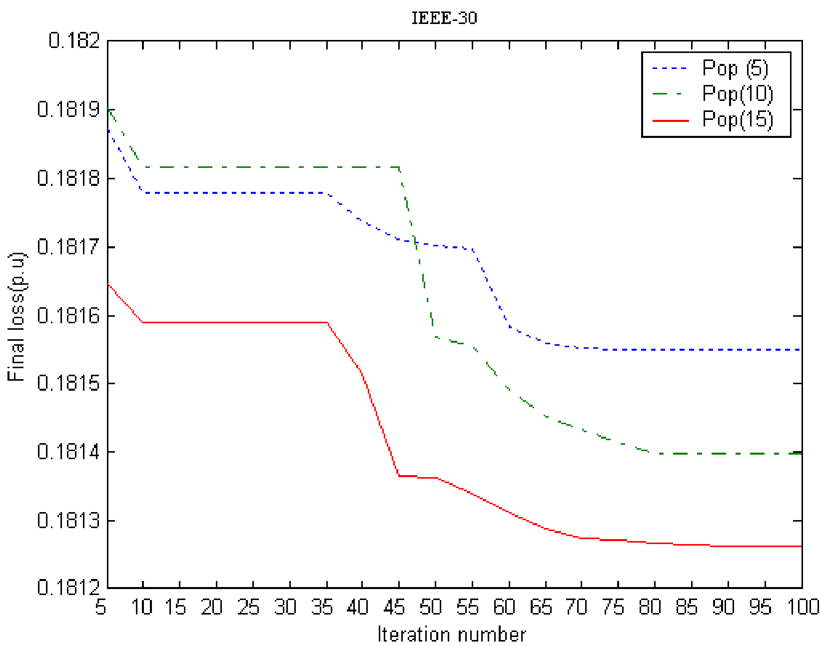


Figure 5. Graphic of PSO behavior for IEEE-30 system (Zoom-starting from 5a. iteration)

The results obtained by the PSO algorithm are better than the ones observed by IP as shown in Table 10. From the beginning, the PSO with 5 iterations obtained best results and became better with the increase of the iteration number. The graphics in Figure 6 illustrate the behavior of the PSO using population (5, 10 and 15). The best results are obtained with population = 15.

As shown in Tables 4-8, the computational time obtained by our approach using the PSO is higher than the IP. It occurs for small systems, because the PSO usually needs a

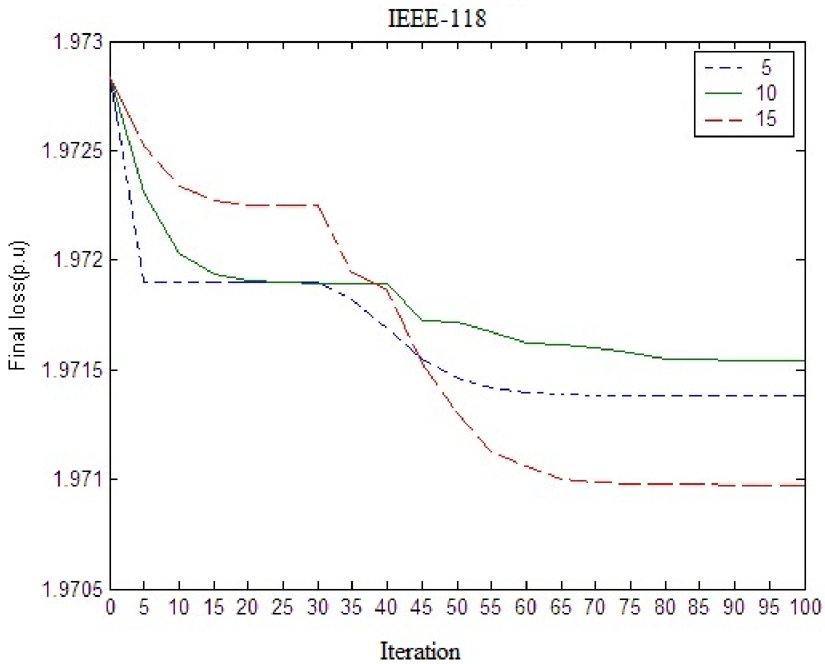


Figure 6. Graphic of PSO behavior for IEEE-118 system

large number of fitness evaluations before a satisfactory result can be obtained, and thus increases the computational time [20]. However, in these cases, the computation times are within the time expected in real-life situations. The PSO is better than the IP for large systems (in time and results – final loss and total reduction) and the real power systems are large, usually around more than 300 buses.

5. **Conclusions.** This paper has proposed the PSO algorithm as a new evolutionary technique to optimize the power loss.

The proposed approach utilizes the local and the global capabilities to search for optimal loss reduction by installing the shunt compensator.

The approach can be applied for a wide range of Power System optimization problems. It presents good results for the loss reduction in particular. These results were compared to those reported in the literature, confirming the potential and the effectiveness of the proposed approach.

**Acknowledgment.** This work is partially supported by CNPq and FAPEMIG, in Brazil.

## REFERENCES


[1] N. Deeb and S. M. Shahidepour, Linear reactive power optimization in a large power network using the decomposition approach, *IEEE Trans. Power Syst.*, vol.5, no.2, pp.428-435, 1990.

[2] S. Granville, Optimal reactive dispatch through interior point methods, *IEEE Trans. Power Syst.*, vol.9, no.1, pp.136-146. 1994.

[3] J. A. Momoh, S. X. Guo, E. C. Ogbuobiri and R. Adapa, The quadratic interior point method solving power system optimization problems, *IEEE Trans. Power Syst.*, vol.9, no.3, pp.1327-1336, 1994.

[4] A. M. Chebbo and M. R. Irving, Combined active and reactive despatch – Part 1: Problem formulation and solution algorithm, *IEE Proc. Gener. Transm. Distrib.*, vol.142, no.4, pp.393-400, 1995.

[5] A. C. Zambroni de Souza, Tangent vector applied to voltage collapse and loss sensitivity studies, *Electric Power Systems Research*, vol.47, no.1, pp.65-70, 1998.

[6] S. Kaitwanidvilai, P. Olarnthichachart and I. Ngamroo, PSO based automatic weight selection and fixed-structure robust loop shaping control for power system control applications, *International Journal of Innovative Computing, Information and Control*, vol.7, no.4, pp.1549-1563, 2011.

[7] K. Y. Lee, X. Bai and Y. M. Park, Optimization method for reactive power planning by using a modified simple genetic algorithm, *IEEE Trans. Power Syst.*, vol.10, no.4, pp.1843-1850, 1995.

[8] A. A. A. Esmin, G. Lambert-Torres and A. C. Zambroni de Souza, A hybrid particle swarm optimization applied to loss power minimization, *IEEE Trans. Power Syst.*, vol.20, no.2, pp.859-866, 2005.

[9] M. A. Abido, Optimal power flow using particle swarm optimization, *Electrical Power & Energy Systems*, no.24, pp.563-571, 2002.

[10] J. Kennedy, The particle swarm: Social adaptation of knowledge, *Proc. of 1997 IEEE International Evolutionary Computation ICEC'97*, Indianapolis, USA, pp.303-308, 1997.

[11] P. Angeline, Evolutionary optimization versus particle swarm optimization philosophy and performance differences, *Proc. of the 7th Annual Conference on Evolutionary Programming*, pp.601-610, 1998.

[12] Y. Shi and R. Eberhart, Parameter selection in particle swarm optimization, *Proc. of the 7th Annual Conference on Evolutionary Programming*, pp.591-600, 1998.

[13] J. Kennedy, Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance, *Proc. of the 1999 Congress of Evolutionary Computation*, vol.3, pp.1931-1938, 1999.

[14] A. C. Zambroni de Souza, C. A. Cañizares and V. H. Quintana, New techniques to speed up voltage collapse computations using tangent vectors, *IEEE Trans. Power Syst.*, vol.12, pp.1380-1387, 1997.

[15] A. C. Zambroni de Souza, L. M. Honório, G. L. Torres and G. Lambert-Torres, Increasing the loadability of power systems through optimal-local-control actions, *IEEE Trans. Power Syst.*, vol.19, no.1, pp.188-194, 2004.

[16] H. Yoshida, Y. Fukuyama, K. Kawata, S. Takayama and Y. Nakanishi, A particle swarm optimization for reactive power and voltage control considering voltage security assessment, *IEEE Trans. Power Syst.*, vol.15, no.4, pp.1232-1239, 2001.

[17] Q. H. Wu, Y. J. Cao and J. Y. Wen, Optimal reactive power dispatch using an adaptive genetic algorithm, *Electr. Power Energy Syst.*, vol.20, no.8, pp.563-569, 1998.

[18] *Power System Test Case Archive*, http://www.ee.washington.edu/research/pstca/pf118/pg_tca118 bus.htm, 2006.

[19] *Matlab Optimization Toolbox 2*, User's Guide, 1999.

[20] Z. Cui, J. Zeng and G. Sun, A fast particle swarm optimization, *International Journal of Innovative Computing, Information and Control*, vol.2, no.6, pp.1365-1380, 2006.