Abstract: Optimization techniques play vital role in many problems. Many optimization algorithms have been proposed in past decades. In this paper various optimization techniques like Particle Swarm Optimization, Modified Particle Swarm Optimization, Artificial Bee Colony Algorithm, Fish School Search Algorithm and Gravitational Search Algorithm are compared and tested in an IEEE-14 Bus power system. By using these five optimization techniques, the power losses like active power and reactive power are reduced drastically. By reducing the losses, the voltage profile improvement and voltage stability enhancement can be maintained. Also this paper shows the better optimization technique which suits the power system to maintain stability by reducing power losses. The results are tabulated to show how much losses have been reduced from the determined actual losses in the power system.

Keywords: Particle Swarm Optimization (PSO), Modified Particle Swarm Optimization (MPSO), Artificial Bee Colony (ABC) Algorithm, Fish School Search (FSS) Algorithm and Gravitational Search Algorithm (GSA)

1. Introduction

The required reactive power optimization at various locations determines the objective function. The problem on reactive power influences on operation of power systems significantly. Due to this power generated in power system, it results in transmission losses. This is one of the most important problems which are present in this field [1-2]. The problem that has to be solved in a reactive power optimization is to determine the required reactive generation at various locations so as to determine the required reactive generation at various locations so as to optimize the objective function. Decrease in power factor can also be due to the loss of reactive power [3].

PSO is a stochastic, population based technique and it is well adapted to multidimensional space [4-5]. MPSO has a faster convergence compared to PSO due to its acceleration weight and inertia factor introduced in MPSO [6]. In ABC, the best solution is observed by a scout due to randomly produced solution. When the target nectar is found the bees will interact and the intensity of the bee interactions gives the solution of the problem [7-8].

Fish School Search was inspired by some fishes in gregarious behavior. FSS is to improve the survivability of the entire group. The success of FSS search space is mostly due to the members of the population [9]. In GSA, the particles move towards the best particles. Due to the heavy mass inertia, the particles search the space locally. Higher attraction of agents is due to the more gravitational mass. The performance of GSA is better when compared to all other algorithms [10].

In this paper, GSA algorithm shows better performance compared to other algorithms. Since GSA is inspired by physical phenomenon. Due to this algorithm the losses are reduced drastically compared to other algorithms. Hence by GSA the voltage stability enhancement is improved due to the reduced losses. GSA algorithm is tested in IEEE 14-Bus power system. The results are tabu-
lated and shown clearly to prove the advantages of this algorithm.

2. Proposed Methodology

2.1 PSO Algorithm

2.1.1 Overview

Dr. Russell Eberhart and Dr. James Kennedy developed particle swarm optimization algorithm in 1995. This algorithm is inspired by behaviour of bird flocking or fish schooling. This technique uses the number of particles (agents) to search the solution present in the population.

The basic definitions used in PSO algorithm are:
- **Particle**: The elements which are interacting in problem space.
- **Swarm**: Movement of group of ‘n’ number of particles in problem space.
- **Optimization**: The opportunity of availing best ones perfectly.
- **PSO**: In total, it is the opportunity of finding the optimal solutions.
- **Pbest**: The best value achieved so far by the individual particle.
- **Lbest**: The best value achieved so far by a particle in competitions of its neighbourhood particles.
- **Gbest**: The best fitness value which is achieved from the individuals.

2.1.2 Algorithm

**Step 1**: Generate ‘n’ number of population randomly.

**Step 1.1**: Initialize particles position and velocity randomly.

**Step 2**: For each particles position, evaluate fitness.

**Step 2.1**: Initialize Pbest (Particle best), Lbest (Local best) and Gbest (Global best) for each particle and iteration count.

**Step 3**: Update iteration count.

**Step 4**: Update particles position, Velocity, Pbest, Lbest and Gbest.

**Step 4.1**: Update velocity of the component using:

\[
v(k, j, i + 1) = v(k, j, i) + C_1 \cdot \text{rand} \cdot (p\text{best}(j, k) - x(k, j, i)) + C_2 \cdot \text{rand} \cdot (g\text{best}(k) - x(k, j, i))
\]

where,
- \(v\) = velocity of the particle at the \(i^{th}\) iteration.
- \(\text{rand}\) = random number should be between 0 and 1.
- \(C_1, C_2\) is taken as 1.

**Step 4.2**: Update the Position Component by using:

\[
x(k, j, i + 1) = x(k, j, i) + v(k, j, i)
\]

**Step 5**: Check for stopping condition;

**Step 5.1**: If 100 iterations are reached, stop it.

**Step 5.2**: Else update iteration count.

2.2 MPSO Algorithm:

2.2.1 Overview

MPSO is a technique which is used for fast convergence compared to PSO algorithm by updating inertia weight and acceleration factor. The ability of breaking away the local optimum is greatly improved. It also avoids the premature convergence problem effectively.

2.2.2 Algorithm

**Step 1**: Generate ‘n’ number of population randomly.

**Step 1.1**: Initialize particles position and velocity randomly.

**Step 2**: For each particles position, evaluate fitness.

**Step 2.1**: Initialize Pbest (Particle best), Lbest (Local best) and Gbest (Global best) for each particle and iteration count.

**Step 3**: Update iteration count.

**Step 4**: The population can be updated by using MPSO technique:

\(i)\) **Inertia Weight Factor**:

The inertia weight factor is calculated by using:

\[
W^k_i = W_{min} + \frac{p^k_{p\text{best}} \times [p^k_{p\text{best}} - p^k_{p\text{best}}]}{p^k_{p\text{best}} \times [p^k_{p\text{best}} - p^k_{p\text{best}}]} (3)
\]

\(ii)\) **Modified Acceleration Factor**

The acceleration factor can be calculated to speed up the convergence.

\[
C^k_{1,i} = \sqrt{\frac{p^k_{p\text{best}}}{p^k_{g\text{best}}}} (4)
\]

\[
C^k_{2,i} = \sqrt{\frac{p^k_{p\text{best}}}{p^k_{g\text{best}}}} (5)
\]
2.3 ABC Algorithm:

2.3.1 Overview

ABC is a stochastic, population based method developed by Dervis Karaboga in 2005. This algorithm is motivated by the intelligent behaviour of honey bees. Here the bee colony comprises of three groups of bees: employed bees, onlookers and scouts. Each employed bee is assumed for each source of food. After coming back to hive, the bees dance on their area. Onlookers choose the food sources depending on the dances of the employed bees. If the food sources of the employed bees are abandoned, it becomes stouts.

2.3.2 Algorithm

Step 1: Generate ‘n’ number of population randomly.

Step 2: Each solution vector is generated by using:

\[ X_{ij} = X_{\text{min}j} + \text{rand}(0,1) \times (X_{\text{max}j} - X_{\text{min}j}) \]  

where,

- \( X_{\text{max}j} \) and \( X_{\text{min}j} \) are the upper and lower bounds of j dimension.

Step 3: Fitness of each food source is evaluated.

Step 4: To determine new food source, each employed bee searches its current food in the neighbourhood;

\[ V_{ij} = X_{ij} + \Phi_{ij}(X_{ij} - X_{kj}) \]  

where,

- \( k_j \) : randomly chosen number, taken as 1.
- \( \Phi_{ij} \) : random number chosen between -1 and +1.

Step 5: If the fitness of the new food source is equal to or better than that of \( X_i \); the new food source takes the place of \( X_i \) in the population and becomes a new member Else it (employed bee) leaves the position and moves to a new food source.

Step 6: After evaluating the information received from the employed bees, the onlooker bee selects a food source. To select the food source \( (i) \), the probability \( (P_i) \) is determined by:

\[ P_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \]  

Step 7: The onlooker bee generates a new food source and fitness evaluation is determined.

Step 8: If the food source cannot be further improved by upcoming iterations, it will be abandoned and the employed bee of the concerned food source becomes a stout. The stout generates a new food source by using:

\[ V_{ij} = X_{\text{min}j} + \text{rand}(0,1) \times (X_{\text{max}j} - X_{\text{min}j}) \]  

Step 9: Checking for stopping criteria.

Step 9.1: If termination condition is met, best food source is reported.

Step 9.2: Else the iteration is repeated from step 4.

2.4 FSS Algorithm

2.4.1 Overview

FSS is a computational intelligent technique developed by Bastos-Filho and Lima-Neto in 2007. It is a stochastic, bio-inspired, population based technique. This search process is carried out by fish having limited memory individuals. The FSS algorithm can be grouped into two classes: feeding and swimming. The operator eating determines the quality of the solution. There are three swimming operators which drive the fish movements.

2.4.2 Algorithm

Step 1: Generate ‘n’ number of population (fishes) randomly.

Step 2: Initialize iteration count.

Step 3: Initialize individual movement of each fish.

Step 4: Evaluate the fitness of each fish’s new position.

Step 5: Each fish weight increases depending on the feeding process. To calculate new weight;
Feeding Process:

\[ W_i(t+1) = W_i(t) + \frac{\Delta f_i}{\max[|f[X_i(t+1)]-f[X_i(t)]|]} \]  
(10)

where,
\[ W_i(t+1) \]: Weight of the fish.
\[ X_i \]: Position of ith fish.
\[ f[X_i(t)] \]: Fitness function.

**Step 6** : The position of each fish can be determined by;

Individual Movement:

\[ \text{new}_i(t) = X_i(t) + \text{rand}(-1,1)\times(\text{step}_{\text{index}}) \]  
(11)

where,
\[ \text{step}_{\text{index}} \]: Percentage of search space amplitude bounded by parameters (\text{step}_{\text{index min}} and \text{step}_{\text{index max}}).
\[ X_i \]: Current position of fish.
\[ \text{rand}(-1,1) \]: random number between -1 and +1.

**Step 7** : If new position is better than the previous, the movement will occur.

**Step 8** : After the movement of fishes, the positions are updated.

Collective-Instinctive Movement:

\[ X_i(t+1) = X_i(t) + \sum_{j=1}^{N} \Delta X_{\text{index}}[f[X_j(t+1)]-f[X_j(t)]] \sum_{j=1}^{N} [f[X_j(t+1)]-f[X_j(t)]] \]  
(12)

where,
\[ \Delta X_{\text{index}} \]: Position alteration of the ith fish during its individual movement.
Whole school move towards set of the most successful fishes.

**Step 9** : The school will be successful, if the collective-volitive movement contracts. If the weight increases then the fish school is successful and new positions are:

\[ X_i(t+1) = X_i(t) - \text{step}_{\text{pole}} \times \text{rand} \times [X_i(t) - \text{Bari}(t)] \]  
(13)

**Step 10** : If the weight decreases, then the fish school is not successful and new positions are:

\[ X_i(t+1) = X_i(t) + \text{step}_{\text{pole}} \times \text{rand} \times [X_i(t) - \text{Bari}(t)] \]  
(14)

where,
\[ \text{Bari}(t) = \frac{\sum_{j=1}^{N} X_j(t) \times W_j(t)}{\sum_{j=1}^{N} W_j(t)} \]  
(15)

**Step 11** : Stopping condition criteria.

**Step 11.1** : If definite amount of cycles are reached, stop it.

**Step 11.2** : Else update the iteration count.

2.5 GSA Algorithm

2.5.1 Overview

GSA is a meta-heuristic algorithm. It is based on law of gravity and law of motion. It has a good convergence rate compared to PSO and MPSO. The particles in search space are the collection of masses. Here objects are the particles and each of the particles performance is measured by masses. Each object attracts the others by a gravity force and this force causes a movement of all objects towards the heavier mass objects.

2.5.2 Algorithm

**Step 1** : Generate ‘n’ number of particles randomly.

**Step 2** : Evaluate the fitness for each particle.

**Step 3** : Initialize gravitational constant G(t) and iteration count.

**Step 4** : Update:

i. **Gravitational Constant**:

\[ G(t) = G(t_0) \times \left( \frac{t}{t_0} \right)^\beta \]  
\[ \beta < 1 \]  
(16)

where,
\[ G(t_0) \]: Gravitational Constant at time interval (t_0).

ii. **best(t) and worst(t)**:

\[ \text{best}(t) = \min_{j \in \{1, \ldots, N\}} \text{fit}_j(t) \]  
(17)

\[ \text{worst}(t) = \max_{j \in \{1, \ldots, N\}} \text{fit}_j(t) \]  
(18)

where,
\[ \text{fit}_i(t) \]: Fitness value of the particle i at time t.

iii. **Inertia Mass**:

\[ M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \]  
(19)

where,
\[ m_i(t) = \frac{\text{fit}_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)} \]  
(20)
Step 5 : Calculate Total Force.

\[ F_i^d(t) = \sum_{j=1}^{N} rand_j F_{ij}^d(t) \] (21)

\[ F_{ij}^d(t) = G(t) \times \frac{M_{ai}(t) \times M_{aj}(t)}{R_{ij}(t) + e} \times (x_i^d(t) - x_j^d(t)) \] (22)

\[ R_{ij}(t) = \|x_i(t), x_j(t)\|_2 \] (23)

where,
- \( rand_j \) : Random number between 0 and 1
- \( M_{ai}(t) \) : Active gravitational mass
- \( M_{aj}(t) \) : Passive gravitational mass
- \( R_{ij}(t) \) : Euclidean distance between two particles \( i \) and \( j \).

Step 6 : Calculate Acceleration.

\[ a_i^d(t) = \frac{x_i^d(t)}{M_{ii}(t)} \] (24)

where,
- \( M_{ii}(t) \) : Inertial mass of the \( i^{th} \) particle.

Step 7 : Update velocity and position.

\[ v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \] (25)

\[ x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \] (26)

Step 8 : Check for stopping condition.
Step 8.1 : If 100 iterations are reached, stop it.
Step 8.2 : Else repeat the steps from 3-7.

3. Problem Definition

The IEEE 14-bus power system is used to test the power losses using various optimization techniques. The one line diagram of an IEEE-14 bus power system is shown in the figure 1. The line data, reactive power limit, shunt capacitor data, bus data and transformer tap setting data are given in table 1-5. The data used for this power system are:

- Bus 1 and 2 are considered as generator bus.
- Bus 3-14 is considered as load bus.
- Bus 3, 6 and 8 are considered as synchronous compensators.

By using conventional method like Newton Raphson method, the losses are calculated manually. The various optimization techniques are used to reduce the power losses in the system.

![IEEE 14-Bus Power System](image)

**Figure 1. IEEE 14-Bus Power System**

The one line diagram of IEEE 14-Bus system is shown in figure 1.

**Table 1. Line Data of IEEE 14-Bus Power System**

<table>
<thead>
<tr>
<th>Line No.</th>
<th>From Bus</th>
<th>To Bus</th>
<th>Line Impedance</th>
<th>Half Line charging B (p.u.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>R (p.u.)</td>
<td>X (p.u.)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0.0193</td>
<td>0.0591</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>0.0469</td>
<td>0.1979</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>0.0581</td>
<td>0.1763</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>0.0540</td>
<td>0.2230</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>5</td>
<td>0.0569</td>
<td>0.1738</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>4</td>
<td>0.0670</td>
<td>0.1710</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>5</td>
<td>0.0133</td>
<td>0.0421</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>6</td>
<td>0.0000</td>
<td>0.2520</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>7</td>
<td>0.0000</td>
<td>0.2091</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>8</td>
<td>0.0000</td>
<td>0.1761</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>9</td>
<td>0.0000</td>
<td>0.5561</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>9</td>
<td>0.0000</td>
<td>0.1100</td>
</tr>
<tr>
<td>13</td>
<td>9</td>
<td>10</td>
<td>0.0318</td>
<td>0.0845</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>11</td>
<td>0.0949</td>
<td>0.1989</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>12</td>
<td>0.1229</td>
<td>0.2558</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>13</td>
<td>0.0661</td>
<td>0.1302</td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>14</td>
<td>0.1271</td>
<td>0.2703</td>
</tr>
<tr>
<td>18</td>
<td>10</td>
<td>11</td>
<td>0.0820</td>
<td>0.1920</td>
</tr>
<tr>
<td>19</td>
<td>12</td>
<td>13</td>
<td>0.2209</td>
<td>0.1998</td>
</tr>
<tr>
<td>20</td>
<td>13</td>
<td>14</td>
<td>0.1709</td>
<td>0.3480</td>
</tr>
</tbody>
</table>
### Table 2. Reactive Power Limits

<table>
<thead>
<tr>
<th>Bus No.</th>
<th>$Q_{\text{min}}$ (p.u.)</th>
<th>$Q_{\text{max}}$ (p.u.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-0.40</td>
<td>0.50</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.40</td>
</tr>
<tr>
<td>6</td>
<td>-0.06</td>
<td>0.24</td>
</tr>
<tr>
<td>8</td>
<td>-0.06</td>
<td>0.24</td>
</tr>
</tbody>
</table>

### Table 3. Shunt Capacitor Data

<table>
<thead>
<tr>
<th>Bus No.</th>
<th>Susceptance (p.u.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0.19</td>
</tr>
</tbody>
</table>

### Table 4. Bus Data of IEEE-14 Bus Power System

<table>
<thead>
<tr>
<th>Bus No.</th>
<th>Bus Voltage V (p.u.)</th>
<th>Generation P (p.u.)</th>
<th>Generation Q (p.u.)</th>
<th>Load P (p.u.)</th>
<th>Load Q (p.u.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.060</td>
<td>2.32</td>
<td>0.16</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>1.045</td>
<td>0.40</td>
<td>0.00</td>
<td>0.217</td>
<td>0.127</td>
</tr>
<tr>
<td>3</td>
<td>1.010</td>
<td>0.00</td>
<td>0.00</td>
<td>0.942</td>
<td>0.191</td>
</tr>
<tr>
<td>4</td>
<td>1.000</td>
<td>0.00</td>
<td>0.00</td>
<td>0.478</td>
<td>0.039</td>
</tr>
<tr>
<td>5</td>
<td>1.000</td>
<td>0.00</td>
<td>0.00</td>
<td>0.076</td>
<td>0.016</td>
</tr>
<tr>
<td>6</td>
<td>1.070</td>
<td>0.00</td>
<td>0.00</td>
<td>0.112</td>
<td>0.075</td>
</tr>
<tr>
<td>7</td>
<td>1.000</td>
<td>0.00</td>
<td>0.00</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>8</td>
<td>1.090</td>
<td>0.00</td>
<td>0.00</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>9</td>
<td>1.000</td>
<td>0.00</td>
<td>0.00</td>
<td>0.295</td>
<td>0.166</td>
</tr>
<tr>
<td>10</td>
<td>1.000</td>
<td>0.00</td>
<td>0.00</td>
<td>0.090</td>
<td>0.058</td>
</tr>
<tr>
<td>11</td>
<td>1.000</td>
<td>0.00</td>
<td>0.00</td>
<td>0.035</td>
<td>0.018</td>
</tr>
<tr>
<td>12</td>
<td>1.000</td>
<td>0.00</td>
<td>0.00</td>
<td>0.061</td>
<td>0.016</td>
</tr>
<tr>
<td>13</td>
<td>1.000</td>
<td>0.00</td>
<td>0.00</td>
<td>0.035</td>
<td>0.058</td>
</tr>
<tr>
<td>14</td>
<td>1.000</td>
<td>0.00</td>
<td>0.00</td>
<td>0.149</td>
<td>0.050</td>
</tr>
</tbody>
</table>

### Table 5. Transformer Tap Setting Data of an IEEE 14-Bus Power System

<table>
<thead>
<tr>
<th>From Bus</th>
<th>To Bus</th>
<th>Tap Setting Value (p.u.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>7</td>
<td>0.978</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>0.969</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>0.932</td>
</tr>
</tbody>
</table>

### Table 6. System Status after Optimization

<table>
<thead>
<tr>
<th>Variables</th>
<th>PSO Voltage (p.u.)</th>
<th>MPSO Voltage (p.u.)</th>
<th>ABC Voltage (p.u.)</th>
<th>FSS Voltage (p.u.)</th>
<th>GSA Voltage (p.u.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Variables</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V1</td>
<td>1.0600</td>
<td>1.0600</td>
<td>1.0600</td>
<td>1.0600</td>
<td>1.0600</td>
</tr>
<tr>
<td>V2</td>
<td>1.0450</td>
<td>1.0450</td>
<td>1.0450</td>
<td>1.0450</td>
<td>1.0450</td>
</tr>
<tr>
<td>V3</td>
<td>1.0100</td>
<td>1.0100</td>
<td>1.0100</td>
<td>1.0100</td>
<td>1.0100</td>
</tr>
<tr>
<td>V6</td>
<td>1.0700</td>
<td>1.0700</td>
<td>1.0700</td>
<td>1.0700</td>
<td>1.0700</td>
</tr>
<tr>
<td>V8</td>
<td>1.0900</td>
<td>1.0900</td>
<td>1.0900</td>
<td>1.0900</td>
<td>1.0900</td>
</tr>
<tr>
<td>Dependent Variables</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V4</td>
<td>0.9796</td>
<td>0.9561</td>
<td>0.9964</td>
<td>0.9777</td>
<td>0.9265</td>
</tr>
<tr>
<td>V5</td>
<td>0.9653</td>
<td>1.0041</td>
<td>0.9714</td>
<td>0.9324</td>
<td>0.9801</td>
</tr>
<tr>
<td>V7</td>
<td>0.9692</td>
<td>0.9821</td>
<td>0.9905</td>
<td>0.9507</td>
<td>0.9749</td>
</tr>
<tr>
<td>V9</td>
<td>0.9504</td>
<td>0.9121</td>
<td>0.9414</td>
<td>0.9767</td>
<td>1.0073</td>
</tr>
<tr>
<td>V10</td>
<td>0.9534</td>
<td>0.9410</td>
<td>0.9254</td>
<td>0.9697</td>
<td>1.0017</td>
</tr>
<tr>
<td>V11</td>
<td>0.9526</td>
<td>0.9541</td>
<td>1.0033</td>
<td>0.9301</td>
<td>1.0046</td>
</tr>
<tr>
<td>V12</td>
<td>0.9819</td>
<td>0.9405</td>
<td>0.9175</td>
<td>0.9389</td>
<td>0.9735</td>
</tr>
<tr>
<td>V13</td>
<td>0.9194</td>
<td>0.9338</td>
<td>0.9121</td>
<td>0.9373</td>
<td>0.9646</td>
</tr>
<tr>
<td>V14</td>
<td>0.9606</td>
<td>0.9458</td>
<td>0.9853</td>
<td>0.9952</td>
<td>1.0061</td>
</tr>
</tbody>
</table>

### 4. Results and Discussions

The results are shown in table 6-7. The GSA algorithm is compared with the various algorithms. The reactive power is minimized much by using GSA algorithm. The active and reactive power losses are minimized much which is tabulated in table 7. By reducing the losses, the voltage profile can be improved. This makes the stable power system. The voltage profile improvement is shown in figure 2. Here the control variables are taken as generator voltages and the rest are taken as dependent variables.
Table 7. Result Comparisons for an IEEE 14-Bus Power System

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>14-Bus Power System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional Method</td>
<td>13.593 MW + j56.910 MVAR</td>
</tr>
<tr>
<td>PSO Algorithm</td>
<td>9.9159 MW + j37.1336 MVAR</td>
</tr>
<tr>
<td>MPSO Algorithm</td>
<td>8.5053 MW + j39.8343 MVAR</td>
</tr>
<tr>
<td>ABC Algorithm</td>
<td>6.4611 MW + j35.6122 MVAR</td>
</tr>
<tr>
<td>FSS Algorithm</td>
<td>7.8458 MW + j48.1797 MVAR</td>
</tr>
<tr>
<td>GSA Algorithm</td>
<td>3.2764 MW + j31.8845 MVAR</td>
</tr>
</tbody>
</table>

Figure 3 and figure 4 depicts the real and reactive power loss of the tested system of table 7. Here the dependent variables depend on control variables. The voltage values are improved by using optimization techniques. By improving this voltage profile, the losses like real and reactive power losses are minimized. As losses are minimized, the power system will remain stable. Hence from all the simulation results shown, GSA optimization technique proves well with reduced losses.

5. Conclusion

In this paper, PSO, MPSO, ABC, FSS and GSA algorithm are proposed. The IEEE 14-Bus power system is tested using these algorithms. The GSA algorithm minimizes the reactive power drastically compared to the other algorithms. It has a good convergence compared to other algorithms. Hence by minimizing the losses like active power and reactive power, the voltage profile is improved and makes the power system much stable. Simulation results prove the effectiveness of GSA algorithm.

6. References


