Optimal coordination of overcurrent relays using a modified particle swarm optimization

H.H. Zeineldin*, E.F. El-Saadany, M.M.A. Salama

Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ont., Canada N2L 3G1

Received 13 September 2005; received in revised form 29 November 2005; accepted 1 December 2005

Available online 24 January 2006

Abstract

In this paper, a new problem formulation is proposed to calculate the optimal relay settings of directional overcurrent relays in power systems. The proposed coordination problem is formulated as a mixed integer nonlinear problem to take into account the discrete values for the pickup current settings. A modified particle swarm optimization (PSO) algorithm is proposed to calculate the optimal relay settings. A comparison between the original particle swarm optimization based method, the proposed PSO algorithm and the GAMS solver is presented.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Directional overcurrent relays; Optimization techniques; Particle swarm optimization; Power system protection

1. Introduction

The problem of coordinating protective relays in electric power systems consists of selecting their suitable settings such that their fundamental protective function is met under the requirements of sensitivity, selectivity, reliability and speed. Directional overcurrent relays are commonly used as an economical means for protecting power systems. The selection of the settings of these types of relays plays an important role in reducing the impact of the fault on the power system.

The calculation of the time dial setting (TDS) and the pickup current (Ip) setting of the relays is the core of the coordination study. Several optimization techniques have been proposed for coordinating directional overcurrent relays. In [1], both the TDS and Ip were assumed to be continuous and the generalized reduced gradient nonlinear optimization technique was proposed to calculate the optimal relays’ settings. The discrete Ip solutions were obtained by rounding off the continuous Ip solutions to their nearest discrete values. Unfortunately, rounding the Ip values could lead to a solution that is outside the feasible region. In [2], it is assumed that the pickup currents are predetermined by choosing one of the available pickup current settings as the predetermined value, thus, the problem becomes a linear programming problem. The simplex two-phase method was proposed to determine the optimal TDS of the relays. However, there could be better pickup current setting for the relays, other than the predetermined one, that would provide a better optimal solution for the coordination problem. The coordination problem was reformulated to take into consideration the dynamic changes in the network in [3]. Similarly, the problem was formulated as a linear programming problem by assuming a fixed value for Ip. In [4], a proposed method based on only constraints was developed. Minimization was inherently included by setting the TDS to minimum and increasing their values gradually and the problem was formulated as a nonlinear programming problem. In [5], genetic algorithm (GA) are applied to the coordination problem to reach the global optimum value with less computational time compared to conventional single point searching methods.

Recently, a new evolutionary computation technique, particle swarm optimization (PSO), was proposed [6]. Like GA, PSO is initialized with a population of random solutions. Its development was based on observations of the social behavior of animals such as bird flocking, fish schooling and swarm theory. Each random solution in PSO is assigned with a randomized velocity according to its own and its neighbors’ flying experiences, and the random solutions, called particles, are then flown through search space. Compared with GA, PSO has some attractive characteristics. It has memory, so knowledge of good solutions is stored by all particles; whereas in GA, previous knowledge of the problem is destroyed once the population changes. It has con-
The original PSO is capable of finding optimal solutions for unconstrained problems. Since the coordination problem is a constrained optimization problem, the original PSO has to be modified.

In this paper, knowing that directional overcurrent relays allow for continuous time dial settings and discrete pickup current settings, the problem of protective relay coordination formulated in [7], as a mixed integer nonlinear programming (MINLP) was used. To verify the importance of the new problem formulation, a comparison between the results obtained for the conventional and reformulated problem is presented using the General Algebraic Modeling System (GAMS) solvers. To enhance the performance of the PSO, a modified PSO is proposed to solve the reformulated coordination problem to calculate the optimal relay settings and the results are compared with the original PSO algorithm.

2. Problem formulation

For the coordination problem, the main objective is to calculate the TDS and \( I_p \), which would minimize the time of operation of the relays. This section presents the conventional and proposed problem formulation, respectively.

2.1. Conventional problem formulation

The coordination problem of directional overcurrent relays in a power system can be stated as follows:

\[
\min \sum W_k T_k \quad (1)
\]

where \( T_k \) indicates the operation time of relay \( R_k \) for a fault in zone \( k \) and \( W_k \) is a coefficient which indicates the probability of the occurrence of the fault on a line and is usually set to 1, thus assuming equal probability of fault occurrence on each line [4].

The constraints can be stated as follows:

**Coordination criteria**

\[
T_{ak} - T_{ak} \geq \Delta T 
\]

where \( T_{ak} \) is the operation time of the first backup relay \( R_{ak} \) for relay \( R_k \) for a given fault in protection zone \( k \). \( \Delta T \) is the coordination time interval and it can take a value between 0.2 and 0.5 s. In this work, a coordination time interval of 0.2 s was adopted.

**Bounds on relay settings and operation times**

\[
\begin{align*}
TDS_{min} & \leq TDS \leq TDS_{max} \\
I_{p_{min}} & \leq I_p \leq I_{p_{max}} \\
T_{min} & \leq T_i \leq T_{max}
\end{align*}
\]

where \( TDS, T_i, I_p \) are the time dial, the relay operation time and the pickup current settings of relay \( R_i \).

**Relay characteristics**

All relays were assumed identical and with characteristic functions approximated by [1]:

\[
T_k = \frac{0.14 \times TDS_k}{\left( I_{ik}/I_{ik}^{P_{max}} \right)^{0.41} - 1} \quad (6)
\]

where \( I_{ik} \) is the short circuit current passing through the relay.

2.2. Proposed problem formulation

From the above section, it can be seen that the coordination problem is a nonlinear programming problem whether the pickup current settings are discrete or assumed to be continuous. In order to take into account the discrete pickup current values, an additional binary variable \( y_{m} \) is added to the problem formulation. The pickup current of each relay is written as a sum of each of its available pickup current settings multiplied by the binary variable. Thus, if a relay has six available pickup settings, it can be formulated as follows:

\[
I_p = \sum_{m} y_{m} I_{pm} \quad \forall i \quad (7)
\]

where \( I_{pm} \) are the available relay pickup current settings. In general,

\[
I_p = \sum_{m} y_{m} I_{pm} \quad \forall i \quad (8)
\]

where \( n \) is the number of relays and \( m \) is the number of available pickup current settings.

In order to assure that not more than one pickup current setting is chosen for each relay, the constraint shown below is included in the problem formulation.

\[
\sum_{m} y_{m} = 1 \quad \forall i \quad (9)
\]

By adding (8) and (10) to the conventional problem formulation, the coordination problem becomes a mixed integer nonlinear programming problem.

3. Particle swarm optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995 [6]. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The problem is initialized with a population of feasible random solutions; however, PSO has no evolution operators such as crossover and mutation. In PSO, the feasible solutions, called particles, fly through the problem space by following the current optimum particles. The particle adjusts its position according to its own experience and the experience of the neighboring particles. Let \( s \) and \( v \) denote the particle’s position and its velocity.
in the search space. Thus, the position of the nth particle in a D-dimensional space is represented as \( x_n = [x_{n1}, x_{n2}, \ldots, x_{nD}] \). The best previous position explored by the nth particle is recorded and denoted as \( p_{n} \). Another value that is recorded 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 is known as gbest. Each particle tries to modify its position using the current velocity and its distance from pbest and gbest. The modification can be represented by the concept of velocity and can be calculated as shown in the following formulas:

\[
\begin{align*}
\mathbf{v}_{n+1} &= w \cdot \mathbf{v}_{n} + c_1 \cdot \mathbf{r}_1 \cdot (p_{n} - \mathbf{x}_{n}) \\
&\quad + c_2 \cdot \mathbf{r}_2 \cdot (\mathbf{g}_{\text{best}} - \mathbf{x}_{n}) \\
\mathbf{x}_{n+1} &= \mathbf{x}_{n} + \mathbf{v}_{n+1}
\end{align*}
\]  
(11)

The first term in (11) represents the inertia of the particle, which is the contribution of the previous velocity. The second and third terms represent the memory and cooperation between particles, respectively. The parameter \( V_{\text{max}} \) represents the resolution with which regions within the feasible search space are to be searched. Choosing a high number for \( V_{\text{max}} \), the particles would be driven to explore sufficiently and the particle becomes trapped in a local optimal solution. The constants \( c_1 \) and \( c_2 \) represent the learning rate or the importance of the particle's position in the first dimension, and then check if after this change, the particle is still within the feasible search space. If the particle is still feasible, then accept the updated position; otherwise, keep the particle at its old position. The following weighting function is usually used in (11):

\[
w = \frac{w_{\text{max}} - w_{\text{min}}}{\text{iter}_{\text{max}}} \times \text{iter}
\]  
(14)

where \( w_{\text{max}} \) and \( w_{\text{min}} \) are the maximum and minimum weight values that are constant and iter is the iteration number.

The role of the inertia weight \( w \) is considered important for the PSO’s convergence. It regulates the trade off between the global and local exploration. A large inertia weight facilitates exploration (searching new area), while a small one tends to facilitate exploitation (fine tuning of the current solution). A proper value of the inertia weight provides a balance between the global and local exploration [8]. It can be seen from (14) that the PSO starts with a high inertia weight \( w_{\text{max}} \) and decrease as the number of iterations increases.

For the proposed MINLP coordination problem, some of the variables are continuous (TDS), while the remaining variables \( \gamma \) are binary variables, which take a value of 0 or 1. For the discrete binary variables, the formula in (11) and (12) still holds except that the particle position takes a value either 0 or 1. To accomplish this modification, a sigmoid function can be used to transform the particle position values to a value between 0 and 1. This is done through the following formula:

\[
sig(v_{n}^{\gamma}) = \frac{1}{1 + \exp(-v_{n}^{\gamma})}
\]  
(15)

The overall change in the particle’s position is governed by the following rule:

If \( \text{rand} < \sig(v_{n}^{\gamma}) \) then \( x_n^{\gamma} = 1 \)
Else \( x_n^{\gamma} = 0 \)

The function of \( V_{\text{max}} \) for the binary variables is to limit exploration after the population has converged. It is important to note that, while high values for \( V_{\text{max}} \) for continuous variables, increases the range explored, the opposite occurs for the binary variables [9].

4. Proposed modified PSO algorithm

Significant research has been done in the area of constrained nonlinear optimization problems (CNOPs). The key issue in the constrained optimization problem is to deal with the constraints. During the initialization process of the PSO, only the solutions that are within the feasible search space are used to initialize the PSO algorithm. The particles search the whole search space and only store in their memory feasible solutions [10]. Despite the fact that, this method of searching for feasible solutions is simple, it is problem dependent. Applying this method of initialization to the coordination problem, would take a large amount of time to find feasible solutions. Another concern regarding the performance of the PSO, is that during the updating process, where each particle modifies its position, the resultant particle position could be outside the feasible search space. This reduces the possibility of finding an optimal or close to optimal solution.

The original PSO is modified to overcome the aforementioned problems. In the proposed algorithm, the interior point method is used to obtain the initial feasible solutions. This is done by initializing the pickup currents randomly, thus the problem becomes linear and the TDS values are calculated using the interior point method. The initial feasible solutions are then applied to the PSO algorithm. Another major modification is applied to the original PSO to overcome the occurrence of any infeasible particles. Instead of updating the entire particle’s position in all D-dimensions at the same time, the positions are updated one after the other. In other words, update the particle’s position in the first dimension, and then check if after this change, the particle is still within the feasible search space. If the particle is still feasible, then accept the updated position for the first dimension; otherwise, keep the particle at its old position.

It has been also examined that, regarding the binary variables, updating the binary variables depending on their feasibility status is not enough. In some cases, where the binary variables are updated with another feasible solution that has less fitness value, the convergence of the PSO to a good solution is not guaranteed. The algorithm has been modified such that, for binary variables, the update is done under two conditions: feasibility...
Fig. 1. General flowchart of the modified PSO used for the coordination problem.

and better fitness value. As for the continuous variables, the only condition that must be satisfied is feasibility.

Fig. 1 illustrates the general flowchart for the modified particle swarm algorithm used to calculate the optimal settings of the relays.

The detailed steps of the algorithm for coordination of directional overcurrent relays using PSO are given below:

Step 1: Initialize the particles with random discrete values of $I_p$ and calculate the TDS using the interior point method. The particles are initialized with the $I_p$ and TDS values.

Step 2: For each particle calculate its fitness value using the fitness function given in (16): \[
\text{fitness function} = 1 - \frac{\sum r_i}{b \times n} \]

where $b$ is the number of constraints and $r$ represents the penalty value if a solution does not satisfy the constraints. It can take a value of either 0, if all constraints are satisfied or 1 if the solution is not feasible.

Step 3: Compare each particle’s fitness value with its pbest. If the fitness value is greater than pbest then update pbest with this new value. Determine the current gbest among all particles’ pbest positions. Compare the current gbest position with the previous gbest position and update gbest.

Step 4: For a particle $P$, update the position of the $i$th dimension for continuous variables. Check if the solution is feasible with the new change. As for binary variables, check feasibility and fitness value before updating. Repeat until all dimensions of particle $P$ are updated.

Step 5: Go to the next particle and repeat step 4 until all the particles are updated.

Step 6: Repeat until the stopping criterion is satisfied. The stopping criterion could be either the number of iteration or the computational time.

5. Simulation results

The proposed method for solving the new problem formulated for optimal coordination of protective relays will be illustrated using two different systems. The first is the 8-bus system shown in Fig. 2, which has a link to another network, modeled by a short circuit power of 400 MVA. The second is a more realistic system represented by the IEEE 14-bus system shown in Fig. 7. Tables 1–3 presents the 8-bus system data. At bus 4, there is also a link to another network modeled by a short circuit power of 400 MVA. The results can be divided under three main categories:

Fig. 2. Single line diagram of the 8-bus network.
Table 1
Line characteristic

<table>
<thead>
<tr>
<th>Nodes</th>
<th>R (Ω/km)</th>
<th>X (Ω/km)</th>
<th>Y (S/km)</th>
<th>Length (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–2</td>
<td>0.004</td>
<td>0.05</td>
<td>0.0</td>
<td>100</td>
</tr>
<tr>
<td>1–3</td>
<td>0.0057</td>
<td>0.0714</td>
<td>0.0</td>
<td>70</td>
</tr>
<tr>
<td>3–4</td>
<td>0.005</td>
<td>0.0563</td>
<td>0.0</td>
<td>80</td>
</tr>
<tr>
<td>4–5</td>
<td>0.005</td>
<td>0.045</td>
<td>0.0</td>
<td>100</td>
</tr>
<tr>
<td>5–6</td>
<td>0.0045</td>
<td>0.0409</td>
<td>0.0</td>
<td>110</td>
</tr>
<tr>
<td>2–6</td>
<td>0.0044</td>
<td>0.05</td>
<td>0.0</td>
<td>90</td>
</tr>
<tr>
<td>1–6</td>
<td>0.005</td>
<td>0.05</td>
<td>0.0</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 2
Generator data

<table>
<thead>
<tr>
<th>Node</th>
<th>Sg (MVA)</th>
<th>Vp (kV)</th>
<th>x (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>150</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>150</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>

5.1. Comparison between problem formulations

In this section, the protective relay coordination problem is first solved using the conventional problem formulation and the pickup currents are assumed to be continuous. The optimal pickup current values are then rounded to the nearest discrete value. Then, the new proposed MINLP formulation is applied. The conventional problem formulation is solved using the CONOPT solver in GAMS and the new formulation is solved using the DICOPT solver in GAMS for the MINLP. The results are presented in Table 4.

The technique used (CONOPT) to solve the nonlinear programming problem (NLP) is to first search for a feasible solution. Once a feasible solution is found, the Jacobian of the constraints is calculated. Then the reduced gradient approach is used to determine the optimal solution.

The technique used (DICOPT) to solve such MINLP problems, first, relaxes the problem by assuming that the binary variables are continuous. This will give a lower bound to the solution of the problem since it is a minimization problem. If the solution of the relaxed problem gives integer solutions for the relaxed integer values, then the search stops and this will be the solution to the problem. Otherwise the problem is divided in to subproblems and a master problem. The subproblems are nonlinear programming problems (NLP) with fixed values for the binary variables. The master problem is a mixed integer-programming problem (MIP) where the variables calculated from the NLP subproblems are used to obtain the optimal solution.

It can be seen from Table 4, that by rounding off the discrete pickup values highlighted in bold, some of the coordination constraints which were given in (2) in the problem are not satisfied, thus leading to an infeasible solution. Besides that, if the pickup currents are assumed fixed at any discrete value for the pickup current other than the solutions obtained from the MINLP problem, the solution would not be the optimal solution. It can be seen that formulating the problem as an MINLP problem prevents the possibility of obtaining infeasible settings and guarantees a better optimal setting for the relays other than the predetermined pickup settings.

Table 3
Transformer data

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Sn (MVA)</th>
<th>Vp (kV)</th>
<th>Vs (kV)</th>
<th>x (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7–1</td>
<td>150</td>
<td>10</td>
<td>150</td>
<td>4</td>
</tr>
<tr>
<td>8–6</td>
<td>150</td>
<td>10</td>
<td>150</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 4
Optimal settings of the relays

<table>
<thead>
<tr>
<th>Relay</th>
<th>NLP (GAMS)</th>
<th>MINLP (GAMS)</th>
<th>NLP (rounding)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDS1</td>
<td>0.1</td>
<td>0.100</td>
<td>0.1</td>
</tr>
<tr>
<td>TDS2</td>
<td>0.324</td>
<td>0.247</td>
<td>0.324</td>
</tr>
<tr>
<td>TDS3</td>
<td>0.197</td>
<td>0.207</td>
<td>0.207</td>
</tr>
<tr>
<td>TDS4</td>
<td>0.143</td>
<td>0.143</td>
<td>0.143</td>
</tr>
<tr>
<td>TDS5</td>
<td>0.1</td>
<td>0.100</td>
<td>0.1</td>
</tr>
<tr>
<td>TDS6</td>
<td>0.217</td>
<td>0.221</td>
<td>0.217</td>
</tr>
<tr>
<td>TDS7</td>
<td>0.204</td>
<td>0.195</td>
<td>0.204</td>
</tr>
<tr>
<td>TDS8</td>
<td>0.215</td>
<td>0.218</td>
<td>0.215</td>
</tr>
<tr>
<td>TDS9</td>
<td>0.1</td>
<td>0.100</td>
<td>0.1</td>
</tr>
<tr>
<td>TDS10</td>
<td>0.1</td>
<td>0.100</td>
<td>0.1</td>
</tr>
<tr>
<td>TDS11</td>
<td>0.206</td>
<td>0.212</td>
<td>0.206</td>
</tr>
<tr>
<td>TDS12</td>
<td>0.346</td>
<td>0.353</td>
<td>0.346</td>
</tr>
<tr>
<td>TDS13</td>
<td>0.1</td>
<td>0.100</td>
<td>0.1</td>
</tr>
<tr>
<td>TDS14</td>
<td>0.149</td>
<td>0.151</td>
<td>0.149</td>
</tr>
</tbody>
</table>

Objective 16.67 s 17.25 s 17.5771 s
Feasibility Feasible Feasible Infeasible

5.2. Applying the modified PSO on the 8-bus network

The original and modified PSO are applied to the MINLP coordination problem and were coded in MATLAB. In this case, each relay had only three available pickup settings. This will
give a total number of 56 variables in the problem (14 continuous and 42 binary). Several values for the population size and the maximum number of iterations were simulated in order to find the suitable values, which provide the convergence of the algorithm. A population size of 30 particles is used. The number of iterations used is 100. The constants $c_1$ and $c_2$ are both set to 1.5. The value of $w_{max}$ and $w_{min}$ are taken to be equal to 0.9 and 0.4. The above values were determined by trial and error such that the solution converges to a close optimal solution. For demonstration, one of the particles was chosen randomly and the plot of its TDS and fitness value is presented to examine the performance of the PSO algorithm. The fitness value of one of the particles and its TDS using the original PSO algorithm are shown in Figs. 3 and 4. Figs. 5 and 6 present the fitness value of one of the particles and its TDS using the modified PSO algorithm.

From Fig. 3, the fitness value of the particle is oscillating with negative values when the original PSO is used. A fitness of negative value indicates that the particle is outside the feasible region. As shown in Fig. 3, at the first iteration the particle’s fitness is positive (starting with a feasible solution) and then suddenly the particle jumps outside the feasible search space leading to negative fitness values. As for the TDS values calculated, using the original PSO, it can be seen that it changes during each iteration. The reason that the method does not converge is because all the dimensions of the particle change at the same time leading to a solution that is not feasible.

The modified PSO takes care of this problem as it modifies each dimension at a time. The fitness of the particle is always positive and the particle is converging towards a better fitness value (minimizing the objective) as shown in Fig. 5. As for the TDS value, the changes are limited since only the moves that will satisfy the constraints are considered. Table 5 presents the numerical value for the TDS and pickup current settings obtained for both the original and modified PSO.
Table 5
Optimal relay settings

<table>
<thead>
<tr>
<th>Relay</th>
<th>Original PSO</th>
<th>Modified PSO</th>
<th>GAMS solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDS1</td>
<td>0.1044</td>
<td>0.13</td>
<td>0.100</td>
</tr>
<tr>
<td>TDS2</td>
<td>0.3452</td>
<td>0.334</td>
<td>0.247</td>
</tr>
<tr>
<td>TDS3</td>
<td>0.3326</td>
<td>0.2096</td>
<td>0.209</td>
</tr>
<tr>
<td>TDS4</td>
<td>0.192</td>
<td>0.1434</td>
<td>0.143</td>
</tr>
<tr>
<td>TDS5</td>
<td>0.108</td>
<td>0.1005</td>
<td>0.100</td>
</tr>
<tr>
<td>TDS6</td>
<td>0.2733</td>
<td>0.2669</td>
<td>0.221</td>
</tr>
<tr>
<td>TDS7</td>
<td>0.2387</td>
<td>0.2059</td>
<td>0.195</td>
</tr>
<tr>
<td>TDS8</td>
<td>0.2849</td>
<td>0.2179</td>
<td>0.218</td>
</tr>
<tr>
<td>TDS9</td>
<td>0.1</td>
<td>0.1244</td>
<td>0.100</td>
</tr>
<tr>
<td>TDS10</td>
<td>0.2881</td>
<td>0.2162</td>
<td>0.214</td>
</tr>
<tr>
<td>TDS11</td>
<td>0.3514</td>
<td>0.2133</td>
<td>0.212</td>
</tr>
<tr>
<td>TDS12</td>
<td>0.4877</td>
<td>0.4497</td>
<td>0.353</td>
</tr>
<tr>
<td>TDS13</td>
<td>0.1165</td>
<td>0.1</td>
<td>0.100</td>
</tr>
<tr>
<td>TDS14</td>
<td>0.186</td>
<td>0.157</td>
<td>0.151</td>
</tr>
</tbody>
</table>

The original PSO is not capable of finding a close to optimal solution since most of its particles jump to the infeasible region. As for the modified PSO, the results obtained are close to optimal as compared with the results of the GAMS solver presented before in Table 5. The results in this section prove that the modified PSO is working properly and is capable of finding a close to optimal solution.

5.3. Applying the modified PSO to the IEEE 14-bus system

The modified PSO is applied to the IEEE 14-bus network shown in Fig. 7. In this case, six pickup settings are made available for each relay. The IEEE 14-bus system consists of 40 relays, thus making the number of variables in the problem equal to 280 (40 continuous and 240 binary). The problem was solved using the OSL solver in GAMS for 10 million iterations. Then, the PSO was applied to the same problem with 20 particles for 200 numbers of iterations. The results of both simulations are presented in Table 6.

By examining the results presented in Table 6, the modified PSO is capable of finding a better solution than the GAMS solver with less number of iterations (less computational time). The main drawback when using deterministic techniques is that they depend to a great extent on the initial starting point. On the other hand, heuristic techniques overcome this problem by using a population of random starting points. This could be seen from the results in Table 6 where the modified PSO was capable of finding a better feasible solution than the GAMS solver with less number of iterations. This makes the modified PSO suitable for online adaptive protection where the relay settings could be changed adaptively every time the configuration of the system changes.

Besides that, as the problem gets harder (more binary variables), the probability that a heuristic technique finds a global optimal solution decreases. The modified PSO guarantees a close to optimal solution for the MINLP coordination problem with less computational time. For the 8-bus system, the GAMS solver was capable of finding a solution, which was better and close to the results obtained by the modified PSO. On the other hand, for the IEEE 14-bus network, the modified PSO proves to be much more efficient than the GAMS solver.
A new problem formulation was presented in this paper for the optimal coordination of directional relays. The new problem formulation takes into account the discrete values for the pickup current and the time dial settings. Formulating the protective relay coordination problem as an MINLP problem determines both the optimal pickup current settings and the time dial settings.

Applying deterministic optimization techniques to the proposed problem formulation required a huge number of iterations and computational time. Besides that, for large systems with increased number of variables, deterministic techniques were trapped in a local optimal solution. This is due to the dependency of deterministic techniques on the initial starting point.

Particle swarm optimization was successfully applied to the new optimal coordination formulation and was capable of overcoming the drawbacks of deterministic techniques by starting with a population of feasible solutions. The original PSO algorithm was modified to deal with constrained optimization problems such as the coordination problem presented in this paper. The modified PSO succeeded in finding a close to optimal solution for the coordination problem as compared with the original PSO algorithm. For a larger problem, the modified PSO was capable of finding a much better solution than deterministic techniques.

References