Real time coordination of directional overcurrent relays by ACO

Meng Yen Shih, Arturo Conde Enríquez
UANL-FIME
con_de@yahoo.com

RESUMEN

La coordinación de relevadores direccionales de sobrecorriente se estudia comúnmente en base a una topología fija en un sistema eléctrico de potencia interconectado dada su complejidad y no linealidad, la coordinación se formula como un problema de optimización. Los sistemas de distribución suelen sufrir consecuencias debido a los cambios dinámicos de topología de red y la operación de elementos. Dichos cambios son entradas y salidas de generadores, líneas y cargas. Las consecuencias son la reducción de sensibilidad y selectividad de relevadores. El objetivo principal de este trabajo es el coordinar los relevadores de sobrecorriente en tiempo real. El objetivo secundario es presentar la formulación de un algoritmo de colonia de hormiga y una comparación de la misma con el algoritmo genético. Los objetivos fueron cumplidos a través del desarrollo de un algoritmo en tiempo real que ha funcionado en conjunto con los algoritmos de optimización.

PALABRAS CLAVE

Relé de coordinación de sobrecorriente, dinàmica de sistemas de potencia, dinàmica de topologías de red, algoritmo genético, algoritmo de colonia de hormigas.

ABSTRACT

The coordination of directional overcurrent relays is most commonly studied based on a fixed network topology within a mesh power system. Due to its complexity and nonlinearity, the coordination is formulated as an optimization problem. Distribution systems often suffer consequences due to the dynamic changes of network topology and operation of elements. Such changes are inputs and outputs of generators, lines and loads. The consequences are reduction of sensitivity and selectivity of relays. The principal objective of this paper is to coordinate the directional overcurrent relays on a real time basis. The secondary objective is to present the formulation of ant colony algorithm and a comparison of it with the genetic algorithm. The objectives were accomplished through the development of a real time algorithm which worked alongside with the optimization algorithms.

KEYWORDS

Overcurrent relay coordination, power system dynamics, dynamic network topologies, genetic algorithm, ant colony algorithm.

Artículo basado el trabajo "Real time coordination of overcurrent relays by means of optimization algorithm", Premio a la Mejor Tesis de Maestría UANL 2014 en el área de Ingeniería y Tecnología y Arquitectura.

INTRODUCCIÓN

The fundamental objective of a protective relay is to detect and isolate the faulted element as soon as possible, so that the impact to the rest of the system is minimized, leaving intact as many non-faulted elements as possible. As different protections are used in different voltage levels of the power system, the directional overcurrent relays (DOCRs) are widely implemented in the sub-transmission and distribution systems due to their competing costs. The purpose of coordinating the DOCRs is to encounter settings that minimize the operation time for faults within the protective zone, while at the same time offering pre-specified timed backup for relays that are in the adjacent zones. Two settings (degrees of freedom) were considered: "dial" which is also known as time-dial setting and "k" which is the security factor that multiplies with the load current in order to obtain the pickup current setting.

Although DOCRs have nonlinear characteristic curves (nonlinear function), the coordination is carried out as a linear problem. This is because the coordination of a pair of relays is performed based on one point, which is the maximum coordination current of the pair of relays. Therefore, the relays guarantee coordination at this point; however, there may be a loss of coordination for faults that are located far from this point.

Over the past decades, manual coordination of DOCRs has been the most common practice performed by protection experts. However, due to its complexity and nonlinearity, manual coordination has been formulated as an optimization problem. Several optimization methods have been proposed to attack this problem. Coordination of DOCRs in the frame of deterministic optimization theory using linear programming (LP) was an approach first reported in 1988. The problem was presented as a linear function in which dials were computed for given values of pickup currents.[1] LP was then studied more for this problem due to its simplicity.[2, 3] Heuristic methods, such as the genetic algorithm (GA) and particle swarm optimization (PSO) [4] of the artificial intelligence (AI) family have quickly gained popularity in solving coordination problems.[5-8] GA has been frequently reported in different literatures due to its simplicity, robustness and easy implementation. This algorithm is based on the evolutionary ideas of natural selection of genes which consist of selection, reproduction and mutation. In this case, the problem was presented as a nonlinear function in which both the dial and the pickup current parameters were computed.

Recently, hybrid methods have also arisen in solving coordination problem. Their main attractions are the reduction of search space, execution time and the number of iterations required in encountering the solution. The hybrid GA and mixed PSO are newly developed methods that are combined with LP, in which their search space are drastically reduced by encoding only the pickup currents as input variable strings, leaving the dials as task for LP to solve.[9, 10] In other words, these hybrid methods solve coordination problems by the linearization of the relay function.

Although DOCRs are the simplest and cheapest, they are the most difficult to apply and the fastest to need re-setting as system changes. Coincidentally, they are mostly implemented in the distribution network, which is the most dynamic part of the whole power system. Consequently, these dynamic changes affect their sensitivity and selectivity, which cause inappropriate operations. However, due to the fact that DOCRs need to meet the fundamental requirements of sensitivity, selectivity, reliability and speed,[11] coordination on a real time basis is proposed in this paper.

Coordination of DOCRs considering different network topologies has been reported in different occasions.[9, 12, 13] A set of relay settings are encountered which will satisfy the coordination constraints of different cases of the network topology. On the other hand, the real time coordination proposed in this paper is not to find a set of relay settings that will satisfy the coordination constraints of different cases of the network topology, but to re-coordinate all DOCRs for every change of network topology. The advantages by doing so are minimum relay operation time, increment of sensitivity, and the ability to withstand another unknown contingency. Moreover, the idea is to coordinate DOCRs online, which as a result enhances in meeting the fundamental requirements mentioned above.

The developed real time algorithm first updates data from the latest changes of the system, and then computes load flow and fault analysis in order to obtain input data for the optimization algorithms. In this paper, ant colony optimization (ACO) and GA were selected to work hand in hand with the real time algorithm. ACO has lately been used for the study of reactive power flow planning,[14] power flow economic dispatch,[15] and power generation scheduling.[16] ACO has reported to be a powerful tool in solving complex problems in different areas.[17] The advantage of this algorithm compared to GA is the role of global memory played by pheromone matrix, which leads to better and faster solution convergence. Hence, the idea to formulate the coordination problem using ACO is original. GA, which is widely known in the coordination area, is used as the comparison reference. In addition, GA is improved and selected due to its simplicity, robustness and easy implementation.

FORMULATION OF THE REAL TIME COORDINATION

As presented in the previous section, the coordination of DOCRs on a real time basis enhances meeting the fundamental requirements of sensitivity, selectivity, reliability and speed.[11] Therefore, the real time coordination algorithm and the formulation of coordination problem are presented sequentially.

Real time coordination

The flow diagram of real time coordination of DOCRs is presented in figure 1.

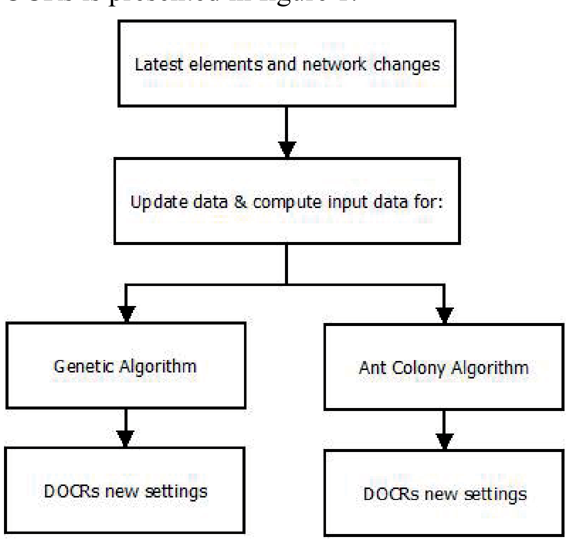

Fig. 1. Real time coordination flow diagram.

The real time algorithm is a very important segment of this paper, as presented in figure 1. It consists of collecting data of the latest elements and network changes, from which input data for posterior relay coordination are computed. The online update hardware system is assumed to have already been manufactured; the hardware requires only the installation of an appropriate real time algorithm. The GA and the ACO presented in figure 1 are for comparison purpose only.

The detailed description of the algorithm is as follows: first, the system's data is updated according to elements and network changes. Then, the Ybus is constructed or modified from the obtained data using the Incident method and the inverse of Inspection method. Next, both lists of "Relay Names" and "Coordination Pairs" are generated automatically. After that, the load flow analysis is run using the Newton Raphson or another method. Then, the Zbus is constructed or modified by the Block construction method and Partial Inversion Motto. Finally, fault analysis is run using Thevenin's method or Symmetrical Components.[18] When all of the above are done, the algorithm will have defined the coordination pairs and computed the maximum load currents and fault currents (3-ph principal, 3-ph backup, 2-ph backup, 1-ph) of each relay for the optimization algorithms of the original network topology.

However, to ensure that relay settings obtained from the posterior coordination algorithm are suitable for at least one element output without coordination loss, the maximum load and fault currents must be computed according to the different n-1 contingency topologies. All elements are taken out one at a time and the simulation is carried out over and over again for the different n-1 contingency topologies. Only the maximum load and fault currents are stored as data for coordination use.

Finally, this algorithm performs a sensitivity filtration before passing the data to the optimization algorithms which coordinate the overcurrent relays. This step ensures that all coordination pairs can be coordinated. The coordination pairs that do not satisfy the requirement of sensitivity analysis will be omitted from the coordination process. In this way, the optimization algorithms will not spend extra time on trying to find settings for these insensitive pairs of relays, which have no settings that will suit them.

Formulation of Coordination Problem

Objective Function:

The purpose of formulating the coordination of DOCRs as an optimization problem is to minimize the principal and backup operation time of relays while maintaining selectivity. It is of great importance to establish a good objective function that evaluates the fitness of the settings because this is the key to encounter optimum solutions using optimization algorithms. The fitness is given in (1):

$$\text{fitness} = \left(\frac{\sum_{a=1}^{NCP} t_{primary_a}}{NCP} \right) * h_1 + \left(\frac{\sum_{b=1}^{NCP} t_{backup_b}}{NCP} \right) * h_2 + \left(\sum_{L=1}^{NCP} E_{CTI_L} \right) * h_3 \qquad (1)$$

where h_1, h_2 and h_3 are factors that increase or decrease the influence of each sub-objective function and will do for any other system, NCP is the number of coordination pairs, $t_{primarya}$ is the primary operation time of relay a, $t_{backupb}$ is the backup operation time of relay b, and E_{CTIL} is the CTI error of $_L$-th coordination pair.

Primary and Backup Relay Constraints:

To coordinate the relays, there must be a time difference between the primary and backup relay. This time difference is called coordination time interval (CTI). In this way, whenever the primary relay fails to extinct a fault, the backup relay enters and tries to extinct the fault after a pre-specified delay. It is normally between 0.2 and 0.5 seconds, but 0.3 seconds is used in this paper. The coordination constraint for every coordination pair is given in (2):

$$CTI_L = t_{backup} - t_{primary} \qquad (2)$$

where CTI_L is the CTI of the L-th coordination pair, t_{primay} is the primary operation time, t_{backup} is the backup operation time.

There is also a range for each relay setting, from which feasible solutions are encountered. This is given in (3) and (4):

$$dial_{min} \le dial \le dial_{max} \qquad (3)$$

$$I_{pickup_{min}} \le I_{pickup} \le I_{pickup_{max}} \qquad (4)$$

where dial is the relay dial setting found within its maximum $dial_{max}$ and minimum $dial_{min}$ range. And $Ipick_{up}$ is the relay pickup current found within its maximum $I_{pickupmax}$ and minimum $I_{pickupmin}$ range.

Relay Characteristic Curve:

The relays function according to the relay characteristic curve (inverse time curve). This inverse time curve operates with less time as fault magnitude raises and more time as fault magnitude drops. IEEE standard norm C37.112-1996 is used in this paper and is given in (5):

$$t = \left[\frac{A}{\left(\frac{I_{sc^{3\varnothing}_{max}}}{I_{pickup}} \right)^n - 1} + B \right] * dial \qquad (5)$$

Where t is the relay operation time, $I_{sc}{}^{3\varphi}{}_{max}$ is the maximum 3-ph short circuit current, I_{pickup} is the pickup current, dial is the relay dial setting, and A,B,n are constants of the IEEE standard.

The IEEE constants of DOCRs are shown in table I. These are the conventional curves: moderate inverse (MI), very inverse (VI) and extremely inverse (EI). The IEEE VI curve is used in this paper, but other curves such as the IEC standard can be used as well.

Table I. IEEE Relay Parameters.

Norm	Curve type	A	B	n
IEEE	Moderate inverse	0.0515	0.114	0.02
	Very inverse	19.61	0.491	2
	Extremely inverse	28.2	0.1267	2

IMPROVED GENETIC ALGORITHM

The GA is a simple and robust algorithm that has gained popularity over the past decades. It performs a heuristic search based on evolutionary ideas of natural selection of genes. A population of search space containing sets of feasible solutions (chromosomes) is created. Decision variables (*dial, k*) are encoded as genes into the chromosome strings. Then, genes are evaluated, penalized, ranked and selected according to their fitness value of the objective function. After that, principles of genetic evolution (crossover, mutation) are applied, and the new population is formed. The whole process is repeated until the stopping criterion is met.

The population size indicates how many chromosomes are in the population (in one generation). If there are too few chromosomes, the algorithm will have few possibilities to perform crossover and only a small part of the search space is explored. On the other hand, if there are too many chromosomes, the algorithm will explore more variety of feasible solutions, but the execution time is excessively increased.

The population is as shown in (6):

$$P=\begin{bmatrix} dial_{(1,1)} & \cdots & dial_{(1,NR)} & k_{(1,NR+1)} & \cdots & k_{(1,NR*2)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ dial_{(NC,1)} & \cdots & dial_{(NC,NR)} & k_{(NC,NR+1)} & \cdots & k_{(NC,NR*2)} \end{bmatrix} \quad (6)$$

where NC is the number of chromosomes and NR number of relays. The population size is the number of chromosomes multiplied by the number of relays times two ($NC,NR*2$). Although chromosomes can be encoded as binary strings, both dial and k parameters are encoded as continuous integer strings in this paper.

Because GA is a well known algorithm in the coordination area, penalization, ranking and genetic operators such as selection, reproduction, mutation are not presented.

Steps of Protection Coordination using Genetic Algorithm

Detailed steps of the GA performed in this paper are presented as followed:

Randomly generate the initial population of n chromosomes, in which each gene is a possible solution to the problem. Genes must be found within the specified ranges introduced in (3) and (4).

Compute the primary and backup time of each relay according to each chromosome.

Evaluate the fitness f(x) of each chromosome x in the population.

Creating a new population in each iteration:

Selection: select parent chromosomes from the population according to their fitness by performing roulette wheel, rank and elitism.

Reproduction: use a crossover probability to crossover the parents to form a new child or children. Perform non-uniform crossover.

Mutation: use a mutation percentage to mutate the genes of the chromosomes. Perform non-uniform mutation at the first stages of the algorithm, then perform intelligent mutation at the posterior stages.

New Population: place the results of reproduction, mutation (new children) and elitism in the new population.

Execute the algorithm again using the new population.

Terminate the algorithm if stopping criteria is met, otherwise, repeat steps 2 to 5.

The improvement of GA here is the addition of intelligent mutation. This is performed at posterior stages of the algorithm. It consists of detecting coordination pairs that are not coordinated, and mutating only the settings of those relays leaving the coordinated pairs untouched. As intelligent mutation was incorporated into the algorithm, numerous chromosomes and iterations are no longer necessary. This was a very successful improvement of this particular coordination problem because the objective function results in all coordination pairs being coordinated most of the time. The solution reported here may not be the global optimum, but it is excusable because coordination is performed on a real time basis.

ANT COLONY OPTIMIZATION

The ACO algorithm is part of the swarm intelligence computing, it is a meta-heuristic optimization aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their home colony and food source. Real ants are capable of finding the shortest path between their home colony and the food source using only information of chemical deposits called pheromone trails. Ants walk on pheromone trails deposited previously be other ants, while at the same time depositing pheromone trails on the ground for future ants to follow. The pheromone trails are volatile over time. When an ant comes across more than one pheromone trail, it selects the most intense trail to follow according to the transition rule. The intensity of a pheromone trail deposited on a route depends on the number of ants traveled on it and the amount of food found through it. This biological behavior inspired the ACO algorithm, in which a set of artificial ant agents adopt the behavior of real ants in solving a problem by exchanging information via pheromones deposited on a graph.

Nomenclature

Ant agents are a number of artificial ants that build solutions to an optimization problem and exchange information on their quality through a communication scheme that is reminiscent of the one adopted by real ants.[17]

The *AS-graph (search space)* is a matrix that contains discrete settings (states) of the control variables (stages).[14] In other words, this graph or matrix contains the set of feasible solutions to the optimization problem, which will be explored by the ant agents. Another matrix, called the pheromone matrix, is created to represent the attractiveness of each discrete setting.

The *Pheromone matrix* is a matrix that contains information about the chemical pheromone deposited by ants. The matrix shows the pheromone intensity of each discrete setting, therefore describing the attractiveness of every possible route to the solution. The more intense a setting is, the more probability it has to be chosen by an ant agent as part of the solution.

The *Transition rule* is the probabilistic and stochastic mechanism that ant agents use to evaluate the pheromone intensity in order to decide which point is the most attractive to visit next.

The *Pheromone update* is the process in which pheromone intensities are increased or decreased according to the evaluated results, regardless of whether the settings lead to good or bad solutions. This is achieved by decreasing the pheromone values through *pheromone evaporation* and increasing the pheromone levels by depositing more pheromone if it is a set of good solution.

The algorithm is started with many sets of solutions (states); together, all the states form the AS-graph search space. This AS-graph is held constant throughout the whole searching process; therefore it does not change from iteration to iteration. On the other hand, pheromone matrix is a representation of attractiveness of each discrete setting (edge) that does change not only in all iterations but in every ant tour. The settings that fit well will consequently lead the ant agents to deposit more and more pheromone until all ants converge on this route (set of settings). In this way, the optimal solution is found. The whole process is repeated until the stopping criterion is met.

The AS-graph size indicates how many states are in the AS-graph. If there are too few states, the algorithm will have fewer possibilities to obtain the optimal solution to the problem, and only a small part of the search space is explored. On the other hand, if there are too many states, the algorithm increases the possibility to encounter the optimal solution, but it drastically slow down the whole process.

Formulation of Ant Colony Optimization

1) AS-graph:

The size of the AS-graph is a (*m,n*NR*) matrix where m represents number of states, *n* number of stages and *NR* number of relays. For example if a system has 5 relays with 2 degrees of freedom (*dial and k*), then the size of the AS-graph for 20 states will be (20, 10). There will be a total of 200 discrete settings of relays. As more degrees of freedom are added to the AS-graph, the size of AS-graph increases.

The AS-graph is as shown in (7):

$$AS = \begin{bmatrix} dial_{(1,1)} \cdots dial_{(1,NR)} & k_{(1,NR+1)} \cdots k_{(1,NR*2)} \\ \vdots \quad \ddots \quad \vdots & \vdots \quad \ddots \quad \vdots \\ dial_{(m,1)} \cdots dial_{(m,NR)} & k_{(m,NR+1)} \cdots k_{(m,NR*2)} \end{bmatrix} \quad (7)$$

In order to create the AS-graph matrix, necessary data such as upper and lower limits and steps of the control variables are needed. For example if a system has 2 relays with 2 degrees of freedom (dial,k) which the range of dial is [0.5, 1.4] in steps of 0.1 and the range of k is [1.4, 1.6] in steps of 0.05, then the AS-graph is a (10, 4) matrix. Note that there are 10 values in the range of dial setting including upper and lower limits, but there are only 5 values in the range of k setting including upper and lower limits. Under this circumstance, complete the rest of the matrix of the *k* settings by repeating the upper limit of *k*. This is to have a sequential order.

Pheromone Matrix:

The pheromone matrix γ(m,n) is constructed according to the size of AS-graph, where m is the number of states and n is the number of stages. This matrix is initialized as shown in (8):

$$\gamma(m,n)=\gamma_0(m,n)=\tau_{max} \tag{8}$$

where τ_max is the maximum pheromone trail and is given in (9):

$$\tau_{max}=\frac{1}{\alpha * f_{gbest}} \tag{9}$$

where fgbest is the global best solution (best over all the past iterations) and α is an empirical value that best suits in the range [0.88, 0.99] [11]. In the case of initializing the pheromone matrix, f_gbest is an initial estimation of the best solution.

In this paper, pheromone matrix was first constructed with all equal edges as presented in (8). However, as in the construction of the AS-graph, the smallest settings of relays occupy the first rows of the AS-graph. These settings are the ideal settings for coordination due to the reason that they will result minimum operation time. Hence, after the pheromone matrix is constructed, the pheromones of the first rows of this matrix are increased. This was done to help the algorithm find the best time operation settings in less time, so one decide how many rows to change and in what amount. One shall not change too many rows and the changes should not be large because this will significantly affect the performance of the exploration of ants.

Transition Rule:

When ant-j is at the r-state of the (i-1)-stage, it will choose the s-state of the (i)-stage as the next visit according to the transition rule shown in (10):

$$p(r,s)=\frac{\gamma(r,s)}{\sum_l \gamma(r,l)} \quad s,l\in N_r^j \tag{10}$$

where N_r^j is a memory tabu list of ant- j that defines the set of points still to be visited when it is at point *r*. The pheromone of the next possible visit of (*i*)-stage currently under evaluation is γ(*r*,*s*) and $\sum_l\gamma(r,l)$ is the pheromone sum of the entire column of the (*i*)-stage under evaluation.

Pheromone Update:

Local pheromone update:

The pheromone trail on each edge of the ant-j tour is updated immediately as the ant-j agent finishes its tour. This is given in the (11):

$$\gamma(r,s)=\alpha*\gamma(r,s)+\Delta\gamma^k(r,s) \tag{11}$$

$$\Delta\gamma^k(r,s)=\frac{1}{Q*f} \tag{12}$$

where α is the persistence of the pheromone trail within the range 0<α<1, (1-α) represent the pheromone trail evaporation and $\Delta\gamma^k(r,s)$ is the amount of pheromone that ant-j puts on edge (*r*,*s*). The desirability of the edge (*r*,*s*) is represented by $\Delta\gamma^j$, such as a shorter distance, a better performance, and in this case a smaller operation time. The objective function evaluation of the settings of ant-*j* tour is represented by f and *Q* is a positive constant. It is observed from (12) that as constant *Q* increases, the amount of pheromone deposited by an ant decreases. Here *Q* was chosen to be 100.

Global pheromone update:

After all ant agents have completed their tours in the iteration, the primary and backup operation times are computed. The objective function is evaluated for each ant tour and all pheromone edges (*r*,*s*) of the best ant tour (the ant tour with best fitness value) are updated according to (13):

$$\gamma(r,s)=\alpha*\gamma(r,s)+\frac{R}{f_{best}} \quad r,s\in J_{best}^j \tag{13}$$

where f_{best} is the best solution of this iteration, *R* is a positive constant and J_{best}^j is the location list of the best ant tour that records the state of each stage when ant-j moves from one stage to another. It is observed from (13) that as constant *R* increases, the amount of pheromone deposited by an ant increases. Here *R* was chosen to be 5.

In this work, the global pheromone update was not performed as described in the previous paragraph. This was because updating only the best ant tour leads to premature convergence. Therefore, the global pheromone update was performed by updating a percentage of the best ant tours. For example, 30% of the ant tours that ranked the best were updated.

Empirical tests have shown that the ACO converges faster when both Q and R are arbitrarily large numbers and almost equal to one another. Studies in the literatures might use $Q=R=1{,}000{,}000$ for other problems, but such large constant is not very suitable to use here because the algorithm converges around 30 iterations, leaving many coordination pairs uncoordinated. This is the reason for choosing $Q=100$ and $R=5$. They were chosen empirically but can work for any network.

Intelligent Exploration:

Intelligent exploration of the AS-graph consists of exploring the setting (*dial*) of specific relays after a pre-specified amount of iterations. This was programmed to help coordinating all coordination pairs. The dial setting was chosen because it has more influence on the relay operation time.

First, the coordination pairs that are not coordinated are detected. Then, a pair is selected to start with. Next, the setting of this specific relay (primary) is obtained from the best ant tour and is used as the upper limit. Afterwards, (13) is applied again but with the modification as given in (14).

$$\gamma(r,s)=\alpha*\gamma(ran1,s)+\frac{R}{f_{best}} \quad r,s\in J_{best}^{j} \tag{14}$$

where ran1 is a random number selected from the interval [1:*r*]. Pheromone is then deposited on this edge. Note that *r* represent the upper limit (state) and s represent the specific relay (stage). Depositing pheromone on this specific edge will lead the ant agents to explore the corresponding setting from the AS-graph. And because it has an upper limit, the newly explored edge will correspond to a smaller setting in the AS-graph, leading to a reduction of primary operation time.

The setting of the specific relay (backup) is obtained from the best ant tour of the same coordination pair that was selected previously, and it is used as the lower limit. Then (13) is applied again but with the modification as given in (15).

$$\gamma(r,s)=\alpha*\gamma(ran2,s)+\frac{R}{f_{best}} \quad r,s\in J_{best}^{k} \tag{15}$$

where ran2 is a random number selected from the interval [*r:end of state*]. The Pheromone is then deposited on this edge. Note that r represent the lower limit (state) and s represent the specific relay (stage). Depositing pheromone on this specific edge will lead the ant agents to explore the corresponding setting from the AS-graph. Because it has a lower limit, the newly explored edge will correspond to a bigger setting in the AS-graph, leading to an increment of backup operation time.

If all coordination pairs were coordinated then coordination pairs that have a greater CTI than the pre-specified are detected and reduced. A coordination pair is chosen to start with and the settings of both relays (primary and backup) are obtained from the best ant tour and used as the upper limit. Then, (13) is applied again, this time, for both the primary and backup relay. By doing so, pheromone trails are deposited on these specific edges (both primary and backup). This leads ant agents to explore the corresponding settings from the AS-graph. And because they have upper limit, the newly explored edges will correspond to a smaller setting in the AS-graph, leading to a reduction of both primary and backup operation time.

Test system

The IEEE 14-bus test system was chosen as shown in figure 2. The system consists of 32 phase relays and they are located as shown in figure 2. The voltages were selected to be 34.5 kV for buses at high voltage side of transformers and 22 kV for buses at low voltage side of transformers. All relays are considered to have very inverse time characteristic curve as was presented in table I.

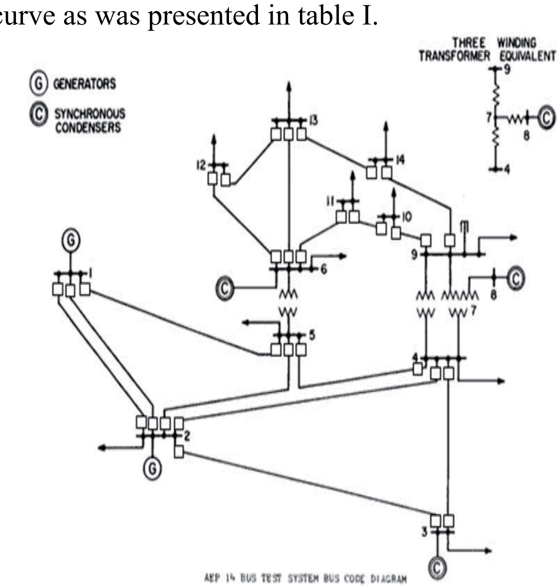

Fig. 2. IEEE 14 bus test system.

In this present work, coordination of phase relays will be studied and ground relays will be omitted. As ground protection is an easier problem, the study here will concentrate on the complicated coordination of phase relays.

The relay names are not assigned as a number, as was done conventionally, but are instead generated automatically by the real time algorithm as a string of numbers. These relay names (string of numbers) consist of 3 digits. The first digit is the name of the nearby bus, the second digit is the name of the remote bus and the third digit represents the number of parallel lines between two buses. For example, the relays between buses 1 and 2 that are near bus 1 are assigned as [1 2 1] and [1 2 2], while the relays that are near bus 2 are assigned as [2 1 1] and [2 1 2] respectively.

The two lines between buses 1 and 2 have the same impedance value. Therefore the relays [1 2 1], [1 2 2], [2 1 1] and [2 1 2] sense the same amount of maximum load currents of 815 A, but due to the n-1 contingency analysis described in section II-A, the maximum load currents of these relays are 1,849 A. The current values are based on minimum load operation.

The CTI is proposed to be 0.3 seconds. Both dial and k are considered continuous in GA, their range are [0.5:2.0] and [1.4:1.6] respectively. However, dial and k are considered as discrete values in ACO, their range and step are [0.5:2.0], [1.4:1.6] and 0.05, 0.01 respectively. Other parameters of the ACO are Q = 100, R = 5. The only stopping criterion is to stop when the algorithms have reached the maximum iteration of 1,000. Although each algorithm has its own stopping criteria, they were disabled in order to be comparable between them. The GA is simulated with 500 chromosomes and the ACO is simulated with 500 ant agents.

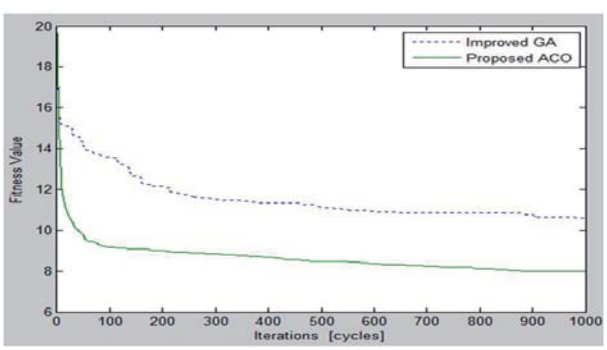

Fig. 3. Averaged fitness convergence of the main network topology of both GA and ACO in ten simulations operating at maximum load.

The fault currents are calculated with the remote end opened. This was done due to two considerations: to obtain the maximum fault current that the relay senses and the very small probability for the remote end relay to malfunction. Note that as the operation of the elements or the network topology changes, load flow and fault analysis must be computed again through real time algorithm.

RESULTS AND DISCUSSIONS

Fixed Network Simulation

The main network is simulated using both the GA and the ACO with the corresponding parameters presented in previous section. There are a total of 39 relay coordination pairs after the sensitivity filter. Both algorithms were simulated 10 times. The convergence of each algorithm was averaged using the best fitness of each iteration of the 10 simulations. This is presented in figure 3 for comparison.

The averaged number of violations of coordination constraints, averaged fitness convergence and averaged time of both the GA and the ACO are 0.1, 10.611, 5507 seconds and 0, 7.994, 441 seconds respectively. From these results, it is concluded that ACO has fewer violations of coordination constraint, better convergence and faster performance than GA. Note that in ACO, all coordination pairs are coordinated for the 10 simulations. On the contrary, some of the 10 GA simulations do not have all coordination pairs coordinated.

The averaged relay settings and operation time, as well as CTI and sensitivity of the 10 averaged GA and ACO simulations are presented in table II.

Table II. Averaged relay settings, operation time, cti and sensitivity of both Ga and aco for main network topology at maximum load.

Algorithm	Dial	k	Blackp Time	Primary Time	CTI	Sensitivity
GA	1.1324	1.4952	3.1425	1.1817	1.9650	3.12005
ACO	0.9875	1.5004	2.3968	1.0383	1.3581	3.12624

Although it is observed from table II that the averaged dial is near 1, there are many relays that use the minimum dial value, which leads to faster operation time. Some relays use a dial near minimum, and the rest use bigger dials due to the necessity for coordination. All relays use small k which leads to greater sensitivity. This observation goes for both the GA and the ACO.

Real Time Coordination

The results presented in section VI-A are computed based on the maximum load currents. Now suppose that there is a decrement of load flow to minimum due to a seasonal change. The relays of the system can have faster operation time and better sensitivity if they are computed again using real time algorithm (to obtain the latest fault and load currents) and the coordination algorithms are run again. There are a total of 48 relay coordination pairs after the sensitivity filter as the load decreased. Both algorithms were simulated 10 times. The convergence of each algorithm was averaged using the best fitness of each iteration of the 10 simulations. This is presented in figure 4 for comparison.

The averaged number of violations of coordination constraints, averaged fitness convergence and averaged time of both the GA and the ACO are 1.4, 11.748, 5825 seconds and 0, 6.968, 438 seconds respectively. From these results, it is concluded that the ACO has fewer violations of coordination constraints, better convergence and faster performance than GA. Note that in the ACO all coordination pairs are coordinated for the 10 simulations. On the contrary, some of the 10 GA simulations do not have all coordination pairs coordinated.

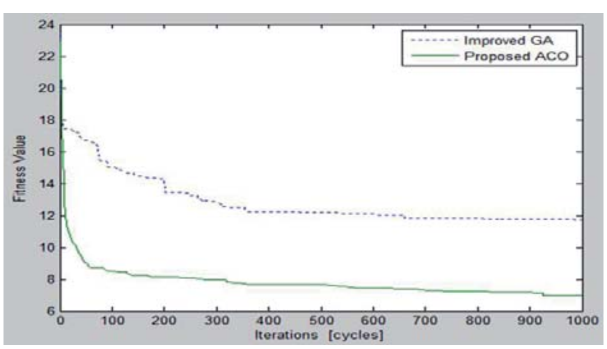

Fig. 4. Averaged fitness convergence of the main network topology of both GA and ACO in ten simulations operating at minimum load.

The relay operation time of the newly coordinated original system due to season change is shown in table III.

Table III. Averaged relay settings, operation time, cti and sensitivity of both ga and aco for main network topology at minimum load.

Algorithm	Dial	k	Blackp Time	Primary Time	CTI	Sensitivity
GA	1.1336	1.4909	2.7546	0.8511	1.9266	4.03250
ACO	1.1165	1.4434	2.1331	0.7973	1.3357	4.17205

A better comparison between the GA and the ACO algorithms and the averaged relay operation results before and after seasonal change is presented in a chart form as presented in figure 5.

The "NV" represents the amount of not coordinated pairs, or in other words the amount of violations of coordination constraints, "tp" and "tb" represent primary and backup time, "CTI" is the coordination time interval, "sen" represents the sensitivity, and "dial" is the time dial and "k" is the security factor that multiplies the load current.

From figure 5 it is observed that the NV is zero for the maximum and minimum loads using ACO, while it is 0.1 for the GA at maximum load and 1.4 at minimum load. The increase in the averaged NV for the GA at the minimum load is because the number of coordination pairs increased as the load current decreased, leading the GA being incapable to satisfy all the coordination constraints using only 500 individuals. However, due to the need of fast execution time for the purpose of real time coordination, the amount of individuals must not be increased if a serious consideration is not taken into account.

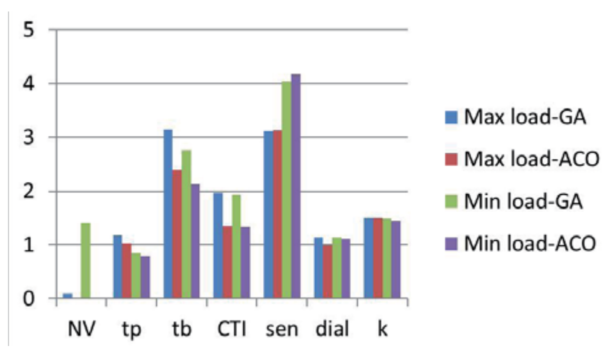

Fig. 5. Comparison chart of the averaged values of both GA and ACO operating the main network topology at maximum and minimum load.

The tp and tb of both the GA and the ACO algorithms operating at the minimum load is smaller than operating at the maximum load. This proves the idea of obtaining better relay operation time performance using real time coordination. The tp and tb of the ACO are smaller than those of the GA. The CTIs of the ACO are smaller than those of GA. It is a very remarkable improvement of sensitivity when the system is executed by the real time algorithm at the minimum load operation. There was a small increase of dial using the ACO when the system is operating at the minimum load. This is reasonable because the dial is the major factor that affects the operation time. When load current decreases the relay operation time decreases (faster operation) as well, so the dial increases to maintain coordination. On the contrary, if the load current increases, the dial decreases. There was a small decrease of k using the ACO when the system is operating at the minimum load. This decrement of k leads to increment of sensitivity.

Modified Network Simulation

Suppose now that the main network topology suffered the following change when it was operating at the minimum load: an outage of distribution line [1 2 2] due to maintenance.

As contingency analysis was taken into account in the load flow and fault simulation in the real time algorithm, the outage of line [1 2 2] did not cause inappropriate operation of the nearby relays which would have lead to severe loss of selectivity. All relays maintained coordinated for this change of network topology. If the n-1 contingency analysis was not taken into account, same relays would have loss selectivity for this change of topology. Now if another element is out of service, the actual relay settings will no longer maintain selectivity. Hence, in order to let all relays maintain coordinated for another outage (expected or unexpected), the new topology must be simulated again from real time algorithm to coordination.

There are 41 coordination pairs for the new system due to network change. The averaged number of violation of coordination constraints, averaged fitness convergence and averaged time of both GA and ACO are 0.8, 11.637, 4028 seconds and 0, 7.694, 377 seconds respectively. The relay operation time of the modified network is shown in table IV.

Table IV. Averaged relay settings, operation time, cti and sensitivity of both Ga and aco for modified network topology at minimum load.

Algorithm	Dial	k	Blackp Time	Primary Time	CTI	Sensitivity
GA	1.1468	1.4909	3.2725	1.1006	2.2031	3.77672
ACO	1.0340	1.4982	2.4744	0.9577	1.5167	3.75726

A comparison between table III and IV and the averaged relay operation results before and after network change at minimum load is presented in chart form as shown in figure 6.

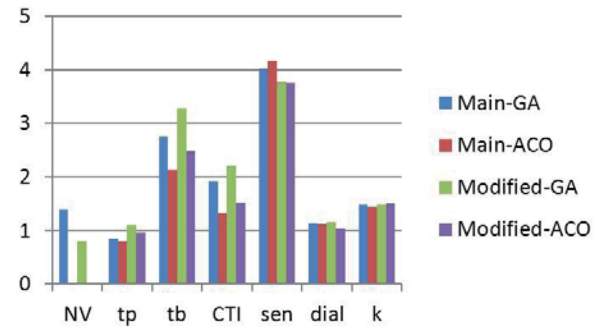

Fig. 6. Comparison chart of the averaged values of both GA and ACO operating the modified network topology at minimum load.

There are zero NV for ACO algorithm. The NV reduced for GA because the number of coordination pairs reduced, leading to less coordination constraints. The increments of both GA and ACO such as tp, tb, and the decrements such as sensitivity, dial are due to the outage of line [1 2 2] that caused the increment of load current for the remaining DOCRs.

CONCLUSION

In this paper, the idea of real time coordination is presented by formulating the coordination problem using the ACO. The main contribution of this research work is introducing the brilliant idea of real time (online) coordination. In other words, the aim is to find as quickly as possible a set of solution that will guarantee the sensitivity and selectivity of relays not only for the current topology, but tolerating an n-1 contingency on a real time basis.

The implementation of the proposed algorithm can be performed in various ways, such as considering discrete or continuous relay settings. Also the update can be performed every change of season (less frequently) or every change of demand (more frequently).

REFERENCES

1. A. J. Urdaneta, R. Nadira, and L. G. Perez, Optimal coordination of directional overcurrent

relays in interconnected power systems, IEEE Transactions on Power Delivery, vol. 3, No. 3, pp. 903-911, July.1988.

2. A. J. Urdaneta, H. Restrepo, S. Marquez, and J. Sanchez Coordination of directional overcurrent relay timimg using linear programming, IEEE Transactions on Power Delivery, vol. 11, No.1, pp. 122-129, January 1996.
3. B. Chattopadhyay, M. S. Sachdev, T. S. Sidhu An on-line relay coordination algorithm for adaptive protection using linear programming technique, IEEE Transactions on Power Delivery, vol. 11, No.1, pp. 165-173, January 1996.
4. Mohamed M. Mansour, Said F. Mekhamer, and Nehad El-Sherif El-Kharbawe, A modified particle swarm optimizer for the coordination of directional overcurrent relays, IEEE Transactions on Power Delivery, vol. 22, No.3, pp. 1400-1410, July 2007.
5. Reza Mohammadi, Hossein Askarian Abyaneh, Farzad Razavi, Majid Al-Dabbagh, Seyed H. H. Sadeghi, Optimal relays coordination efficient method in interconnected power systems, journal of Electrical Engineering, vol. 61, No. 2, pp. 75-83, 2010.
6. Manohar Singh, B. K. Panigrahi, Optimal overcurrent relay coordination in distribution system, IEEE, 2011.
7. Dharmendra Kumar Singh and S. Gupta, Optimal coordination of directional overcurrent relays: A genetic algorithm approach, IEEE Students' Conference on Electrical, Electronics and Computer Science, 2012.
8. Dusit Uthitsunthorn and Thanatchai Kulworawanichpong, Optimal overcurrent relay coordination using genetic algorithms, International Conference on Advances in Energy Engineering, 2010.
9. Abbas Saberi Noghabi, Javad Sadeh, and Habib Rajabi Mashhadi, Considering different network topologies in optimal overcurrent relay coordination using a hybrid GA, IEEE Transactions on Power Delivery, vol. 24, No.4, pp. 1857-1863, Octuber 2009.
10. A. Motie Girjandi, M. Pourfallah, Optimal coordination of overcurrent relays by mixed genetic and particle swarm optimization algorithm and comparison of both, International Conference on Signal, Image Processing and Applications, vol. 21, 2011 IACSIT Press, Singapore.
11. J. Lewis Blackburn, and Thomas J. Domin, Protective relaying, principles and applications, 3rd edition, 2006.
12. A. J. Urdaneta, L. G. Perez, and H. Restrepo, Optimal coordination of directional overcurrent relays considering dynamic changes in the network topology, IEEE Transactions on Power Delivery, vol. 12, No. 4, pp. 1458-1464, October.1997.
13. Abbas Saberi Noghabi, Javad Sadeh, and Habib Rajabi Mashhadi, Optimal coordination of directional overcurrent relays considering different network topologies using interval linear programming, IEEE Transactions on Power Delivery, vol. 25, No.3, pp. 1348-1345, July 2010.
14. Kwang Y. Lee, and John G. Vlachogiannis, Optimization of power systems based on ant colony system algorithms: an overview, ISAP, 2005.
15. Yun-He Hou, Yao-Wu Wu, Li-Juan Lu, Xin-Yin Xiong Generalized ant colony optimization for economic dispatch of power systems, IEEE, 2002.
16. Shyh-Jier Huang, Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches, IEEE Transactions on Energy Conversion, vol. 16, No. 3, September 2001.
17. Marco Dorigo, Mauro Birattari, and Thomas Stutzle, Ant colony optimization: artificial ants as a computational intelligence technique, IEEE Computational Intelligence Magazine, November 2006.
18. Hadi Saadat, Power system analysis, McGraw-Hill, ISBN 0-07-561634-3, 1999.