Turk J Elec Engin, VOL.16, NO.3 2008, © TÜBİTAK

Role of Artificial Intelligence in the Reliability Evaluation of Electric Power Systems

Chanan SINGH and Lingfeng WANG
Department of Electrical and Computer Engineering
Texas A&M University, College Station, TX, 77843 USA
e-mail: singh@ece.tamu.edu • l.f.wang@ieee.org

Abstract

Every reliability analysis effort, in some way, involves searching the state space of the system for those states that represent the events of interest, typically failure of the system or a given node to meet the demand. This essentially translates into a search procedure to efficiently identify states to be examined and then using a mechanism to evaluate these states. Traditionally, reliability analysis methods are based either on an implicit or explicit enumeration process or Monte Carlo sampling. More recently, methods based on artificial intelligence have been investigated both as an alternative to Monte Carlo for the search process as well as state evaluation techniques in conjunction with the Monte Carlo methods. This paper will examine the conceptual basis of overall reliability evaluation process and explore the role of artificial intelligence methods in this context. It will also provide some examples of application to the reliability analysis of hybrid systems involving conventional and alternative energy sources.

1. Introduction

Many probabilistic methods [1, 2] have been developed over the past several decades, and are now being used more widely in power system operations and planning to deal with a variety of uncertainties involved. Examples of these uncertainties are equipment outages, load forecast uncertainties, weather conditions, uncertainties in the availability of basic energy and operating considerations. In the new restructured environment, the probabilistic methods have even a stronger potential for application because of their inherent ability to incorporate economic analysis market uncertainties. By including the cost of reliability to customers, strategies for optimum planning and operation can be designed more systematically than is possible with deterministic methods.

The basic steps of reliability assessment are shown in Figure 1. The first step is to define the system being analyzed and its operating policies. The system consists of components and therefore the models of the components and system need to be defined and specified. The combination of component states and system operating strategies describes system states. A possible approach would be a complete enumeration, i.e., to select each possible state in turn and evaluate it for its status as success or failure defined for a given node or the system. In power systems, the failure of the system often means that the load can not be satisfied and thus some part of it needs to be curtailed. Then based on the probability of the failed states and the magnitude and location of load loss, the relevant reliability indices can be computed. It can be seen from this process that the following are needed for the reliability evaluation:

1. Component and system models, data and operating strategies;
2. A state evaluation procedure;
3. Specification of reliability indices to be computed.

In all but very small systems, complete enumeration scheme is not feasible as the number of states increases exponentially with the increase of components. This is sometimes called the curse of dimensionality. The two categories of methods that have been developed in the past can be classified into analytical and Monte Carlo simulation (MCS). The analytical methods deal with this problem by several basic approaches such as state space reduction, state space truncation, implicit enumeration and contingency ranking [3]. The Monte Carlo method deals with this problem of dimensionality by sampling states using the basic concept that they occur proportional to their probabilities of occurrence [1, 4]. More recently, computational methods based on metaheuristic techniques have been developed for this purpose and show the promise of more intelligent search of state space [5, 6] than Monte Carlo. In a sense these metaheuristic techniques provide a more systematic and intelligence based truncation or pruning of state space. The intelligent methods have also been used for faster evaluation of the selected system states [13].

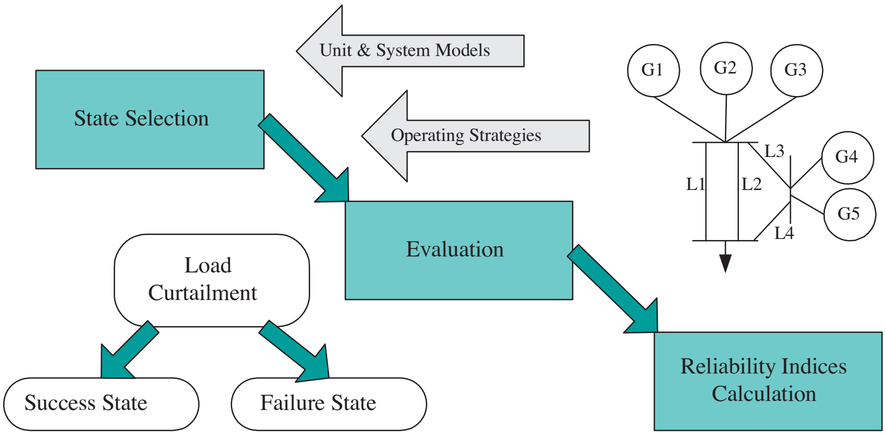

Figure 1. Reliability evaluation steps.

This paper examines the conceptual basis of the overall reliability evaluation process and describes the role artificial intelligence methods can play in this context. It also provides examples of such applications to the reliability analysis of conventional and alternative energy sources.

2. Conceptual Considerations

Methods of power system reliability analysis can be considered to fall into two broad categories: analytical and computational methods, where computational methods include Monte Carlo simulation and intelligent search techniques. Basically, there are three stages inherent in any reliability method: state selection, state evaluation and index calculation. The analytical techniques and computational techniques differ mostly in the process of state selection as the number of possible states is extremely large for most practical applications. The analytical techniques use some device to circumvent the problem of straightforward enumeration, such as state merging, truncation, implicit enumeration and sequential model building [1, 3]. The computational methods select system states based on their respective sampling or searching mechanisms.

For instance, Monte Carlo techniques accomplish this by sampling states proportional to the probabilities of their occurrence while Intelligent Search (IS) techniques choose system states based on their fitness values in relation to the target problem. Analytical techniques represent the system by mathematical models and compute reliability indices using mathematical solutions.

2.1. State space

The whole state space is graphically illustrated in Figure 2 by classifying all system states into different sets.

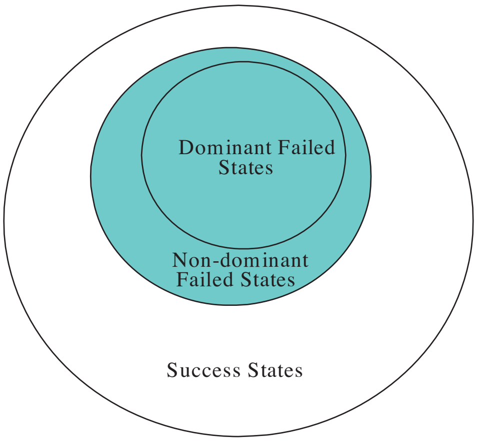

Figure 2. Classification of system states in the whole state space.

The total state space can be broadly divided into two sets: success, and failed system states. The failed states can be further classified into dominant and non-dominant failed states. Dominant failed states here mean states that have more dominant effect on the computation of reliability indices. Any power system reliability model using computational methods comprises at least the following steps:

1) Sampling of states: The states may be selected using an analytical approach, random and sequential sampling in MCS, or intelligent (fitness-guided) sampling in IS. The sampled state is defined by the status of all components comprising the system.

2) Evaluation of states: This step is to determine whether the load can be satisfied given the status of generators and other components depending on the scope of investigation.

3) Estimation of indices: Reliability indices are estimated from the repeated use of the two previous steps. The stopping criterion is based on the coefficient of variation being less than a stipulated value or other suitable consideration.

It is important to note that any selected state first needs to be evaluated before it can be classified as a failed or success state. The state evaluation may be simple in some situations as in single area generation reliability studies where the sum of capacities is compared with the load to determine loss of load. In multi-area or composite reliability studies a flow calculation method is used to determine magnitude and location of loss of load. The flow algorithm could be transportation flow method, as in multi-area studies or DC/AC power flow for composite system reliability studies. The state evaluation in such applications can be computationally intensive and may constitute the most significant part of computational burden. Since every state selected needs to be evaluated, the number of states selected for the computation of the indices has a significant effect on the computational efficiency. There are two important observations from this discussion:

1. The number of states sampled or selected for evaluation should have as higher percentage of failed states as possible within the computational framework of the method.
2. The technique for state evaluation should be efficient.

In the remainder of this paper, we discuss the role of IS in the state selection and this will be the main focus of our paper. For comparison, we use Monte Carlo simulation (MCS). Later, we also describe the role of artificial intelligence in the state evaluation process and refer to the reader to relevant literature.

3. Comparison Of Monte Carlo To Intelligent Search

Let us first examine the MCS by considering its application to the estimation of the loss of load probability (LOLP) index. The various steps are outlined below.

Step 1: Select the seed for the random number generator. Set the maximum iteration number and let the initial iteration number $k = 1$;

Step 2: Sample the system state randomly (load level, generation status and line status) and perform a flow calculation to classify it as loss-of-load or otherwise:

$$X_i = \begin{cases} 0 & \text{if sampled state is loss-of-load} \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

Step 3: Calculate LOLP, variance of the estimated LOLP and the coefficient of variation:

$$\overline{LOLP} = \frac{1}{k}\sum_{i=1}^{k} X_i \tag{2}$$

$$V(L\hat{O}LP) = \frac{1}{k}(L\hat{O}LP - L\hat{O}LP^2) \tag{3}$$

$$\sigma = \frac{\sqrt{V(\overline{LOLP})}}{\overline{LOLP}} \tag{4}$$

Step 4: Check whether the coefficient of variation σ is less than a specified threshold δ. If $\sigma < \delta$ or $k > K_{\max}$, stop; otherwise $k = k + 1$, and go to step 2.

It can be seen from equation (2) that, in MCS, both the success and failed states enter the index calculation. Therefore one can not focus on the identification of the failed states alone, but a proportional number of success states need also be generated to calculate the reliability index. One should keep in mind though that both success and failure states will need to be evaluated before they can be classified as such.

Distinguished from the random sampling in MCS, in IS sampling can be interpreted as the “optimization process.” The process of applying IS optimization operators in deriving the next generation of individuals is the sampling mechanism of IS algorithms. Here the individuals with higher fitness values have higher chances to be sampled in each iteration. The general computational flow of any population-based intelligent search (PIS) algorithms can be illustrated in the following:

Step 1: A population of individuals is randomly created.

Step 2: Each individual is evaluated based on the specified objective function, which is used to measure the “fitness” of each individual. Here the term “fitness” is slightly abused to generally indicate the "goodness" of each individual with respect to the specific problem, though it is usually used in genetic algorithms.

Step 3: Determine if any stopping criterion is satisfied. If yes, halt the PIS algorithm; otherwise, go to next step.

Step 4: Different PIS operations are applied to each individual in order to create the next generation of individuals.

Return to Step 2 until any stopping criterion is satisfied.

It can be appreciated here that in IS, only the failure states contribute to the index calculation. Thus the focus here is to generate the dominant failure states. Success states also will be created during this process but the efficiency depends on the design of the fitness function to minimize the generation of success states. In this fashion, much fewer states need to be evaluated than the MCS. Therefore, unlike MCS, PIS is rather problem dependent, where system states with higher failure probabilities have higher chances to be selected and evaluated. Here, in PIS the failure probability of system state is used to guide the search. In some sense, this characteristic enables PIS to have promise to outperform MCS for some type of problems due to its potentially higher algorithmic efficiency. The driving force behind each PIS renders the search more purposeful by avoiding problem-independent random sampling. Due to the difference of estimation philosophies between MCS and PIS, the deviations of estimated results in relation to the "real" values may be different between them. For instance, in MCS, the estimated values of indices may be larger or smaller than the actual values; however, in PIS, the estimated values are always somewhat smaller than the actual ones. Especially, in highly reliable systems, since failure states are scattered in the state space in an extremely sparse fashion, it is possible that, in a given sampling window, the MCS method can not sample the failure states in their "real" ratio with respect to the total number of system states. This will inevitably lead to larger estimation errors of the intended reliability indices or even cause convergence problem. It should be noted that PIS-based algorithms can have a special advantage in cases where flow calculations using DC/AC load flow are needed to evaluate a sampled state. When a state is sampled, it can be identified to be loss of load only after the evaluation process. Since in MCS, majority of the states sampled are success states, this flow calculation will need be carried out more often. On the other hand, in PIS the states are sampled in a more directed fashion and thus the evaluation process will be used more efficiently.

In PIS algorithms, each individual is regarded as a potential solution and many individuals comprise a population. For a specific PIS algorithm, each individual has different names. For instance, in GA, each chromosome is an individual, which is a composite of genes. In ACS, the tour traveled by each ant (referred to as "ant" for brevity) is deemed a potential solution. In PSO, each particle flying in the search space is thought of as a candidate solution. In AIS, each antibody is seen as a potential solution. A binary coding scheme may be used to represent each individual, where each bit takes one or zero to indicate the component state. "One" and "zero" represent the working and failed status of each component. The target problem is concerned with combinatorial optimization, and its objective is to find the failure state array which can be used to calculate different adequacy indices. There are two major stages in the evaluation procedure: First the failure-state array with respect to the maximum load demand is derived using PIS, and then the reliability indices are calculated by convoluting the effective total capacity with the hourly load based on the state array achieved previously. The computational flow of the proposed evaluation procedure is laid out in the following.

Step 1: Generate a population of individuals randomly. The states of components are initialized by binary numbers.

Step 2: Evaluate each individual i based on the defined objective function, for example LOLP with respect to the defined load. If its value is less than the specified LOLP threshold (a small LOLP value below which the corresponding states are filtered out), it is assigned a very small fitness value in order to reduce its chances of participating in subsequent PIS operations. Also, if the state is a success state, the fitness of corresponding individual is assigned a very small value so as to reduce its chances to contribute to next generation.

Step 3: Increase the iteration number by one.

Step 4: Check if any stopping criterion is met. If yes, halt the algorithm and output the state array derived. If no, go to the next step.

Step 5: Different PIS operators are applied for producing the next generation, and then repeat the procedure from Step 2 to Step 4 until any stopping criterion is satisfied.

Step 6: Calculate the adequacy indices based on the achieved state array.

4. State Evaluation: Neural Net Based Methods

From the steps of the straight Monte Carlo simulation, we can make two observations: 1) for each sampled state, a flow calculation has to be performed to determine its load-loss status; 2) because of the random sampling, many similar states are sampled in the simulation and their characteristics determined repeatedly. Therefore, a straight Monte Carlo simulation is very time-consuming.

Two neural net-based methods have been proposed for state evaluation [13]. The first method (designated Method A in this paper) is to more efficiently determine the load loss characteristic of the sampled state. In this method, the Self-Organizing Map (SOM) is trained to recognize the loss-of-load states. Once this training is complete, the SOM is used along with the Monte Carlo simulation to estimate the reliability of the system. This method overcomes the first disadvantage of the straight Monte Carlo simulation. Incidentally, another method that has been used to identify system states in flexible manufacturing systems [14] is based on Group Method of Data Handling (GMDH). This method can also be easily used for power system applications.

The second method (designated method B) proposes to cluster the sampled states before determining their load loss characteristics. In this method, Monte Carlo simulation is used first to accumulate states, then SOM is used to cluster these states and flow calculation is used for analysis of clustered states. This method overcomes the second disadvantage of the straight Monte Carlo simulation.

It has been shown that Monte Carlo simulation can be made more efficient using SOM [13]. We will first discuss the input training features for SOM. Then approaches to marry SOM and MCS will be discussed.

4.1. Input Training Features

For the problem of loss-of-load state identification, a power system state can be characterized by load conditions, network topology and availability status of generators. Because the probability of outage of a transmission line is very small, it may be assumed that the lines are fully available all the time. The input vector corresponding to a system state then is

$$X_i = [P_{i1}, \mathrm{Q}_{\mathrm{i1}}, ..., P_{in}, Q_{in}, PG_{i1}, ..., PG_{im}]. \tag{5}$$

Here, P_{ik} denotes real power load of bus k for state i; Q_{ik} is the reactive power load of bus k for state i; PG_{im} denotes available real power generation of bus m for state i; n is the number of load buses; and m is the number of generation buses.

4.2. Marriage between MCS and SOM

It may be recalled from the earlier discussion that the two drawbacks of MCS are the excessive time taken by state characterization and the sampling and characterization of similar states repeatedly. This subsection describes two methods to overcome these drawbacks.

Method A

Method A can determine the load loss characteristics of the sampled state more efficiently. In method A, the SOM is trained to recognize the loss-of-load states. Once this training is complete, the SOM is used along with the Monte Carlo simulation to estimate the reliability of the system. In this version of MCS-SOM the state evaluation is done by the trained SOM rather than OPF calculation. Thus this method overcomes the first disadvantage of the straight Monte Carlo simulation. The overall procedure of method A consists of the following steps.

Step 1: Prepare the training patterns for SOM. Training patterns are obtained by OPF calculations which characterize each training pattern as loss of load or otherwise.

Step 2: Carry out SOM training with the prepared training patterns.

Step 3: Label a neuron in the map as loss-of-load or no-loss-of-load according to the majority label voting of the training patterns mapped to that neuron.

Step 4: After the SOM network is trained, Monte Carlo simulation follows the same procedure as in section III except that state classification is performed by SOM instead of OPF. The class (loss-of-load or no-loss-of-load) of each sampled state is determined by the label of the nearest neuron in the map.

Method B

In Method B, states sampled by MCS are clustered before determining their load loss characteristics. Monte Carlo simulation is used first to accumulate states, then SOM is used to cluster these states and OPF is used for analysis of clusters. Thus Method B overcomes the second disadvantage of the straight Monte Carlo simulation. The overall procedure of method B consists of the following steps.

Step 1: Perform Monte Carlo sampling of the system state space to get N samples. These samples are taken as training vectors to SOM. N is decided by experience, for example, $N = 10{,}000$.

Step 2: Put the training vectors generated in step 1 into Self-Organizing Map and train it. After training, the weight vector of each neuron represents a kind of equivalent power system state and maintains the original data's topological relationships. Also some neurons do not map any of the training vectors, and some neurons map one or more training vectors.

Step 3: Perform OPF calculation for each neuron that has mapping of the training vectors. Determine the load-loss status of the neuron by using its weight vector as inputs to the OPF program. Label the neuron as "1" if it is loss-of-load and "0" if it is not loss-of-load.

Step 4: Count the number of sampled states n_i which are mapped to each loss-of-load neuron i in the Self-Organizing Map.

Step 5: Calculate the estimated LOLP value:

$$LOLP = \frac{\sum\limits_{i} n_i}{N} \tag{6}$$

5. Case Studies

In this section, two case studies using PIS and MCS-SOM methods for reliability evaluation are discussed.

5.1. Pis Based State Selection

Some studies are reported in Refrence [6] on a WTGs-augmented IEEE RTS-79. The original RTS has 24 buses (10 generation buses and 17 load buses), 38 lines and 32 conventional generating-units [7]. The system annual peak load is 2850 MW. The total installed generating capacity is 3405 MW. In this study, one unconventional subsystem comprising of multiple identical WTGs is added to the RTS. Each WTG has an installed capacity of 1 MW, a mean up time of 190 hours and a mean down time of 10 hours. The hourly derating factors for WTG output can be found in [8]. Reliability indices are calculated for a time span of one week and the load cycle for week 51 with peak load 2850 MW, low load 1368 MW and weekly energy demand 359.3 GWh. The impact of wind power penetration is examined by incorporating installed wind power capacity of 400 MW.

For peak load of 2850 MW with wind power penetration of 400 MW, the system adequacy indices obtained using the exact method [8], MCS, and proposed PIS methods are listed in Table 1. The PIS techniques include ant colony system (ACS) [9], artificial immune system (AIS) [10], binary particle swarm optimization (BPSO) [11], and genetic algorithm (GA) [12]. The units for LOLE, EENS, and LOLF are h/week, MWh, and occ./week, respectively. Time is in seconds. Here, all four discrete PIS optimizers are used to derive meaningful system states [15]. The population sizes for all PIS algorithms are set 300. We can see that the performance of MCS is the worst among all methods in this scenario of our problem in terms of solution quality and computational time. The solutions derived by all PIS algorithms are comparable to the exact ones. Among them, the solutions from ACS are slightly more accurate than those of others. GA is the most computationally expensive one primarily due to its time-consuming genetic operations. BPSO has the shortest convergence time because of its simpler operations.

Table 1. Reliability indices for unconventional capacity 400 MW.

Method	LOLE	EENS	LOLF	Time
ACS	0.789780	98.921	0.193233	21.6
AIS	0.789768	98.912	0.193229	22.7
BPSO	0.789760	98.909	0.193221	15.4
GA	0.789740	98.900	0.193213	29.3
MCS	0.771991	96.211	0.190632	59.4
Exact method	0.789840	99.085	0.193275	29.9

To measure the efficiency of the various methods, we define a ratio to measure the convergence performance (i.e. sampling efficiency) of different PIS algorithms for the scan and classification task.

$$\lambda = \frac{\text{Number of meaningful states sampled}}{\text{Number of total samples}} \tag{7}$$

This ratio can be used in each generation or across the whole optimization process. It varies depending on the algorithm efficiency and solution density in the search space. It should be noted that although this ratio is defined for measuring the convergence performance of PIS, it also has significance in the context of MCS which is virtually the estimate of LOLP as defined in (2), if the “meaningful states” are also interpreted as the “dominant failed states”. As compared with PIS, in MCS a smaller proportion of sampled system states are expected to be dominant failed states.

5.2. MCS-SOM Based State Evaluation

Here studies were performed on the original IEEE RTS-79. The reliability analysis was performed at the peak load level. Studies using hourly load level can be found in [13].

Method A

A. Input selection The total load is fixed at the peak load of 2850 MW. The input features for the SOM network consist of the generating unit statuses only with the lines assumed available at all times. There are 32 units distributed at 10 buses leading to the input vector:

$$X = [PG_1, PG_2, \ldots, PG_9, PG_{10}] \tag{8}$$

B. SOM training A total of 500 different training patterns were used to train the SOM network. These training patterns were non-repetitive and from the high probability region of the state space. This was achieved by varying the availability status of units through a preliminary Monte Carlo experiment and evaluation by OPF. Table 2 [13] shows the characteristics of the SOM training.

Table 2. Characteristics of SOM (method A, peak load level).

Input dimension	10
Number of training patterns	500
Kohonen layer (x*y)	20*20
Topology	rectangular
Neighborhood type	bubble
Learning rate type	linear function
Iteration number for phase I	2000
Initial neighborhood radius for phase I	15
Initial learning rate for phase I	0.8
Iteration number for phase II	20000
Initial neighborhood radius for phase I	3
Initial learning rate for phase II	0.03

After training, the map was calibrated and labeled according to the samples in the input data file. The best matching neuron in the map corresponding to each data vector was found. The neurons were then labeled as loss-of-load or no-loss-of-load according to the majority of labels “hitting” a particular map neuron. The neurons that got no “hits” were left unlabeled. Using this procedure, 143 neurons were labeled as loss-of-load or no-loss-of-load.

C. Monte Carlo simulation Monte Carlo simulation was performed to estimate the loss of load probability (LOLP) but for each sampled system state, SOM, instead of OPF, was used to characterize it. The label of the nearest neuron to each sampled system state was the estimate of load-loss status. Ten thousand states were sampled in the simulation; there were 8759 no-loss-of-load states and 1241 loss-of-load states characterized by SOM. Thus the estimated LOLP is 0.1241.

Monte Carlo simulation using OPF was performed to obtain the benchmark value of LOLP at peak load level. For the 10000 system states sampled above, there were 8774 no-loss-of-load states and 1226 loss-of-load states characterized by OPF. The computed benchmark value of LOLP is thus 0.1226. Among the 8774 no-loss-of-load state classified by OPF, 8720 states were classified as no-loss-of-load correctly by

SOM in method A, resulting in a classification accuracy of 99.38%. Among the 1226 loss-of-load states, 1187 states were classified as loss-of-load correctly by SOM, giving a classification accuracy of 96.82%. It should be noted that calculations for the classification accuracy were made for the benchmark case and not for the case where only SOM was used for calculating the LOLP.

D. Computing time Five (5) seconds were required for phase I of SOM training (global ordering) and 31 seconds for phase II of SOM training (fine-tuning). For characterization of all the 10000 sampled states, the computing time was 3 seconds. Compared to the straight Monte Carlo simulation, which needs to perform 10000 OPFs, the computing time is greatly reduced. The program was implemented in C language and run on a Sun Solaris 2.5.

Method B

Reliability analysis discussed below was also performed at the peak load level.

E. SOM training The total load is fixed at the peak load of 2850 MW. The input features for the SOM network are the same as (8). A total of 10,000 training vectors generated from Monte Carlo sampling were used to train the SOM network. The training parameters of SOM are listed in Table 3.

Table 3. Characteristics of SOM (Method B, peak load level).

Input dimension	10
Number of training patterns	10000
Kohonen layer (x×y)	30×30
Topology	rectangular
Neighborhood type	bubble
Learning rate type	linear function
Iteration number for phase I	3000
Initial neighborhood radius for phase I	20
Initial learning rate for phase I	0.9
Iteration number for phase II	30000
Initial neighborhood radius for phase I	3
Initial learning rate for phase II	0.03

F. LOLP calculation After SOM training, there were 368 neurons in the map that had mapping of the training vectors. These neurons were labeled as loss-of-load or no-loss-of-load by using their weight vectors as inputs to OPF. After the map was labeled, the total number of samples mapped to the loss-of-load neurons was counted. Among the 10000 samples, there were 1187 samples mapped into loss-of-load neurons. Thus the estimated LOLP value is 0.1187.

As shown before in straight Monte Carlo simulation, there were 8774 no-loss-of-load states and 1226 loss-of-load states of the total 10000 samples and the benchmark value of LOLP at peak load level is 0.1226. Among the 8774 no-loss-of-load states classified by OPF, 8763 states were classified correctly by SOM in method B, resulting in a classification accuracy of 99.87%. Among the 1226 loss-of-load states classified by OPF, 1176 states were classified correctly by SOM, giving a classification accuracy of 95.92%.

G. Computing time The computation time required for method B consists of two major components: the time required for training the SOM, and that for using OPF to label the neurons as loss-of-load or not. For the peak load condition, it required 6 seconds for the phase I of SOM training, and 55 seconds for phase II. Also because there were 368 neurons that had mapping of the training vectors, 368 OPFs were performed to label these neurons after the SOM training. Compared to the straight Monte Carlo simulation that needs 10000 OPFs, the computing time is greatly reduced. Method B was also implemented in the C language and run on a Sun Solaris 2.5.

6. Concluding Remarks

Artificial intelligence techniques have drawn much attention in dealing with complex and challenging problems in power systems. Among them, reliability evaluation is a type of representative applications. In this paper, some concepts on reliability evaluation based on population-based intelligent search as well as neural network enhanced MCS are presented. Also some case studies are presented to demonstrate the effectiveness of the proposed methods. It appears that the intelligence based methods hold promise for reliability studies and merit further investigation.

References

[1] C. Singh, R. Billinton, System Reliability Modelling and Evaluation, Hutchinson, UK, 1977.

[2] G. J. Anders, Probability Concepts in Electric Power Systems, Wiley, New York, 1990.

[3] C. Singh, "Reliability Calculations of Large Systems", Proc. of 1975 Annual Reliability and Maintainability Symposium, Washington, D.C., 1975.

[4] C. Singh, P. Jirutitijaroen, "Monte Carlo Simulation Techniques for Transmission System Reliability Analysis", A Tutorial Paper presented at IEEE Power Engineering Society General meeting, Tampa, Florida, May, 2007.

[5] N. Samaan, C. Singh, "Adequacy Assessment of Power System Generation Using A Modified Simple Genetic Algorithm", IEEE Transactions on Power Systems, Vol. 17, No 4, November 2002, pp. 974–981

[6] L. F. Wang, C. Singh, "Adequacy assessment of power-generating systems including wind power integration based on ant colony system algorithm", IEEE Proceedings of Power Tech Conference (PowerTech), Lausanne, Switzerland, July 2007.

[7] IEEE Committee Report, "IEEE reliability test system", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-98, No. 6, pp. 2047–2054, 1979.

[8] S. Fockens, A. J. M. van Wijk, W. C. Turkenburg, C. Singh, "Reliability analysis of generating systems including intermittent sources", International Journal of Electric Power & Energy Systems, Vol. 14, No. 1, pp. 2–8, 1992.

[9] M. Dorigo, T. Stutzle, Ant Colony Optimization, Cambridge: The MIT Press, 2004.

[10] L. N. de Castro, F. J. Von Zuben, "Learning and optimization using the clonal selection principle", IEEE Transactions on Evolutionary Computation, Vol. 6, No. 3, June, pp. 239–251, 2002.

[11] J. Kennedy, R. Eberhart, "Particle swarm optimization", IEEE Proceedings of the International Conference on Neural Networks, Perth, Australia, pp. 1942–1948, 1995.

[12] D. E. Goldberg, Genetic Algorithm in Search, Optimization, and Machine Learning, Reading, MA: Addison-Wesley, 1989.

[13] C. Singh, X. Luo, H. Kim, “Power system adequacy and security calculations using Monte Carlo Simulation incorporating intelligent system methodology”, IEEE Proceedings of the 9th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), Stockholm, Sweden, 2006.

[14] P. Yuanidis, M. A. Styblinski, D. R. Smith, C. Singh, “Reliability Modeling of Flexible Manufacturing Systems”, Microelectronics and Reliability, Vol. 34, No. 7, pp. 1203–1220, 1994.

[15] C. Singh and L. F. Wang, “An alternative to Monte Carlo simulation for system reliability evaluation: search based on artificial intelligence”, Proceedings of the 3rd International Conference on Reliability and Safety Engineering (INCRESE07), Udaipur, India, December, 2007.