Paper accepted for presentation at 2009 IEEE Bucharest Power Tech Conference, June 28th - July 2nd, Bucharest, Romania

Multi-deme Parallel Genetic Algorithm in Reliability Analysis of Composite Power Systems

Lingfeng Wang and Chanan Singh, *Fellow, IEEE*

***Abstract*—Intelligent search based techniques such as genetic algorithm (GA) have been proposed to deal with reliability evaluation of complex power systems recently. In this type of methods, the guided search is carried out on a population scale trying to find all the dominant failure states, based on which different reliability indices can be calculated accordingly. However, the process may be time-consuming when power flow analysis is involved in deciding the status of a system state in complex power systems such as composite systems. To speed up the computing process, parallel implementation of GA is proposed in this study by using multi-deme based search, where multiple subpopulations are distributed in different processors. In this way, simultaneous search is achieved through parallel implementation. An IEEE reliability test system is used for simulation studies. It turns out that the proposed parallel method is effective in increasing the computing efficiency of GA when it is used for reliability evaluation of composite power systems.**

***Index Terms*—Genetic algorithm, parallel computation, reliability evaluation, multi-deme parallelization, computational efficiency, intelligent search, composite power system, hash table.**

I. Introduction

Due to uncertainties in electric power systems, probabilistic reliability evaluation is becoming more commonly used [1]. The methods for reliability analysis of electric power systems can be broadly classified into analytical and computational methods. The former demands strict mathematical analysis, which usually circumvents exhaustive enumeration using devices such as state truncation, state merging, and implicit enumeration in order to improve the evaluation efficiency. In some way, the latter can be divided into Monte Carlo simulation (MCS) and Artificial Intelligence (AI) based iterative computations. MCS has turned out to be very effective in dealing with highly complex power systems through its random sampling mechanism. It treats the problem as a series of experiments, and estimates reliability indices by simulating the actual process using probability distributions of state residence times. However, MCS may become less efficient when its convergence criteria are different to fulfill in certain evaluation scenarios. For instance, when MCS is used to deal with highly reliable systems, its efficiency may become low since a large number of system states need to be sampled and evaluated. Especially, when a time-consuming flow analysis is needed to determine the status of each sampled system state, MCS will be quite time-consuming. To avoid this difficulty, artificial intelligence based methods such as intelligent search has been proposed for reliability evaluation [2-8], and it has shown to be able to outperform MCS in some scenarios in terms of solution quality and computing cost. In this work, we will use a common intelligent search algorithm termed Genetic Algorithm (GA) [9] for reliability analysis of composite power systems, which include both generation and transmission systems.

Parallel implementation of iterative algorithms is an effective way to increase the computational efficiency [10]. From its mechanism, GA is a typical iterative computation procedure, thus it is quite suitable for parallel computing. Various parallel topologies can be used to partition the GA tasks. For instance, because the evaluation of each individual (system state) is independent of one another, GA can be parallelized based on genetic operations. This parallelization scheme has turned out to be quite effective in reliability evaluation of composite power systems [7]. In this work, we will use a different parallelization scheme to deal with the problem. Instead of using a single population, here multiple sub-populations (multi-deme) will be distributed into multiple processors for searching for meaningful system states simultaneously. And the genetic exchange between different sub-populations will be introduced so as to enhance the search efficiency in each processor.

The remainder of the paper is organized as follows. Loss of load state identification using DC flow analysis is introduced in Section II. In Section III, the multi-deme parallel GA is discussed. Section IV presents the inner working of GA-based reliability evaluation for composite power systems including state representation, computing flow, and hash table based state storage. Simulation results are presented in Section V. Finally, the paper ends up with the conclusion and future research directions.

II. Loss of Load Identification

Loss of Load Probability (LOLP) may be the most commonly used reliability index in current power industry practice. Therefore, this study will focus on the LOLP estimation using GA-based method. For this purpose, the loss of load status of system states of interest should be identified. In this investigation, DC flow analysis is used to identify loss of load status, which intends to minimize load shedding at each bus subject to a set of constraints. The result of the objective function minimization can then be used to determine if a system state is loss-of-load or not. The problem can be stated as follows [8]:

L. F. Wang is with the Electrical and Computer Engineering Department, Texas A&M University, College Station, TX, 77843, USA (e-mail: l.f.wang@ieee.org).

C. Singh is with the Electrical and Computer Engineering Department, Texas A&M University, College Station, TX, 77843, USA (e-mail: singh@ece.tamu.edu).

978-1-4244-2235-7/09/$25.00 ©2009 IEEE

Minimize:

$$\sum_{i=1}^{N_b} C_i \tag{II.1}$$

Subject to:

$$\hat{B}\theta + G + C = D \tag{II.2}$$

$$G \leq G^{max} \tag{II.3}$$

$$C \leq D \tag{II.4}$$

$$b\hat{A}\theta \leq F_f^{max} \tag{II.5}$$

$$-b\hat{A}\theta \leq F_r^{max} \tag{II.6}$$

$$G, C \geq 0 \tag{II.7}$$

$$\theta \quad \text{unrestricted} \tag{II.8}$$

where
N_b: Number of buses;
N_t: Number of transmission lines;
C: N_b-vector of bus load curtailments;
C_i: i-th element of C, i.e., unsatisfied demand at bus i;
D: N_b-vector of bus demands;
G^{max}: N_b-vector of available generation at buses;
F_f^{max}: N_t-vector of forward flow capacities of transmission lines;
F_r^{max}: N_t-vector of reverse flow capacities of transmission lines;
G: N_b-vector of dispatched generation at buses;
θ: N_b-vector of bus voltage angles;
b: $N_t \times N_t$ primitive matrix of transmission line susceptances;
$\hat{A}$: $N_t \times N_b$ element-node incidence matrix;
$\hat{B}$: $N_b \times N_b$ augmented node susceptance matrix $= \hat{A}^T b\hat{A}$.

If the objective function in the above formulation is zero, the power system state does not result in loss-of-load. If the objective function is greater than zero, then the system state does constitute loss-of-load.

III. Multi-deme Parallel GA

Parallel processing is a natural way to handle computationally-expensive problems. There are several parallelization topologies which may be used to partition all the tasks for computation speedup [11]. For example, master-slave parallelization is based on the partition of genetic operations. It uses only a single population, where fitness function evaluation and/or genetic operations are carried out in a parallel manner. Here, the generation process is decoupled from the evaluation process in GA. Depending on different operation strategies, this parallelization scheme can be implemented in synchronous or asynchronous mode.

On the contrary, the multiple-deme parallelization scheme used in this study is based on partition of the entire population, where several subpopulations (demes) are distributed in multiple processors. There are several migration topologies which can be used to move (copy) individuals from one deme to another, such as ring, star, hypercube, 2D/3D mesh, and torus. Figures 1 shows the parallelization scheme of ringe topology used in this study. In this parallelization topology, migration is carried out between two neighboring demes in a certain direction. It is also possible that there are several variants for each parallelization topology. For instance, the migration between demes may be bi-directional.

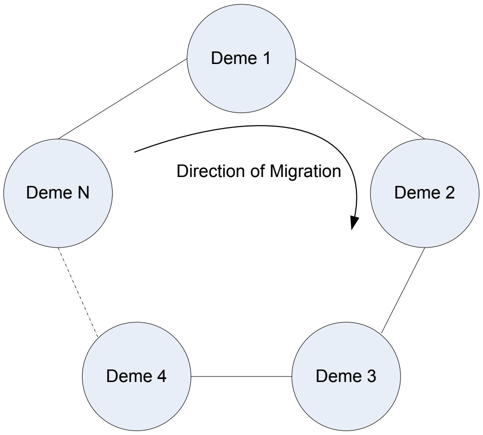

Fig. 1. Ring topology of multi-deme parallel GA.

The parallel GA based on multi-deme evolution is not only the parallelization implementation of traditional sequential GA, it can also be regarded as a new class of GA. This is because in the parallel GA, speciation is allowed in multiple populations, which is the process where different populations evolve towards different optimal solutions. This characteristic is very promising to increase the evaluation efficiency in our problem. Here GA can be more appropriately seen as a scanner instead of a single-solution searcher, since its task is to find out a set of meaningful states instead of a single optimal or sub-optimal solution. When it is used for optimization, the intermediate solutions found in the search process lead to the final solution. Somehow different from this function, when it is used as a scanning tool in our problem, all the eligible system states found during the search contribute to the final solution. Thus, when GA is parallelized by multiple subpopulations for the scanning purpose, its search efficiency is expected to be improved more significantly than that when it is used for optimization tasks.

Here is a bit concise conceptual comparison between different parallel methods of reliability analysis from the perspective of their information communication mechanisms in the sampling process. In the parallel Monte Carlo Simulation (MCS), the computing task in each slave processor is independently executed and there is no information communication or exchange between them. This is also true for master-slave parallel GA. However, in the population partition scheme of parallel GA (i.e., multi-deme parallel GA), migration operations are conducted between some computing tasks in multiple processors in order to improve the evaluation efficiency. This feature is especially useful for our problem, where GA is used for scanning the state space. In the following simulation work, we will be focusing on this parallelization method by comparing its performance with other topologies.

IV. Multi-deme GA based Reliability Evaluation for Composite Power Systems

Intelligent search has shown its promise in dealing with reliability evaluation of power generation systems [2-8]. In this study, a commonly used intelligent search procedure termed genetic algorithm will be used for reliability evaluation of composite power systems, which include both generation and transmission systems. The composite power systems are much more complicated since the transmission topologies and capability should also be considered to determine if loss of load is caused. To determine the loss of load status of each system state, a flow calculation is usually needed, which may demand a significant computing load when the number of system states under evaluation is large. In this section, we will show how reliability evaluation can be accomplished using parallel GA for achieving higher computing efficiency.

A. State Representation

In GA, a population is made up of a set of chromosomes (i.e., individuals). In this problem, each individual can be regarded as a system state. Note that the identical generators connecting to the same bus are put in the same group for the convenience of computation. For loss-of-load state identification, a composite power system state can be characterized by availability statuses of generators and transmission lines. Thus, the system state is encoded by a set of binary numbers as follows:

$$X_i = [P_{i1}, \ldots, P_{ik}, \ldots, P_{in}, TL_{i1}, \ldots, TL_{ik}, \ldots, TL_{it}] \tag{IV.9}$$

where P_{ik} is the status of generation k for state i, TL_{ik} is the status of transmission line k for state i, n is the number of generators, and t is the number of transmission lines. Each bit takes the value of “1” or “0”: the former indicates an up status of a generator/line, and the latter represents a down status.

B. Computational Flow

There are three stages inherent in any reliability evaluation methods: state selection, state evaluation, and index calculation. There is no exception in intelligent search based reliability evaluation method. However, in this method, the first two stages are somehow interwoven. GA is used to select system states with high failure probability (i.e., dominant failure states) based on its optimization mechanism. In this process, optimal power flow is carried out to determine the loss of load status of each selected state, which is needed to determine the fitness of each state. Those failure states of high probability will have high chances to be selected in the GA search process. The state selection is guided by the results from state evaluation. In this way, both system state selection and evaluation are needed to find dominant failure states in the intelligent search process. Thus, the first two stages cannot be explicitly separated since they interact with one another throughout the search. All the derived eligible failure states are saved for the subsequent index calculation, which is the third stage in reliability evaluation. The data-flow diagram of this method is shown in Figure 2. The major steps of using this method for reliability evaluation with regards to the maximum load level is described in the following.

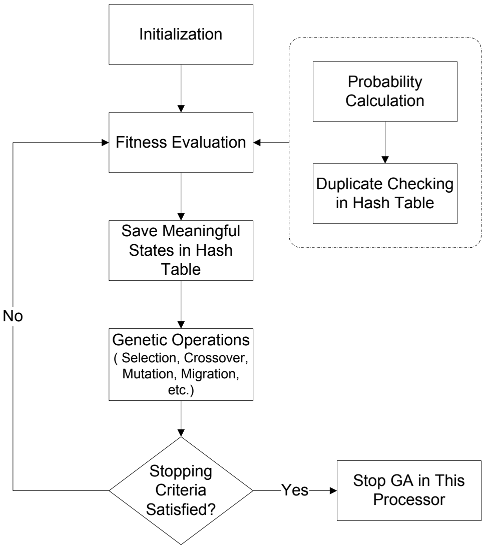
Initialization
Probability Calculation
Fitness Evaluation
Duplicate Checking in Hash Table
Save Meaningful States in Hash Table
No
Genetic Operations (Selection, Crossover, Mutation, Migration, etc.)
Stopping Criteria Satisfied?
Yes
Stop GA in This Processor

Fig. 2. Computational procedure of GA-based reliability evaluation

- Step 1: Create a population of individuals in a random fashion. The states of generators and lines are initialized by binary numbers.
- Step 2: Evaluate each individual i based on the defined objective function (probability of not satisfying load with respect to the maximum load demand L_{max}).

  The objective function value of state i can be calculated as follows:

  - To reduce the number of load flow calculations, here the probability of system state is calculated first:

    $$P_i = \prod_{j=1}^{gt} p_j \tag{IV.10}$$

    where gt is the total number of generators and transmission lines. For the generators, p_j can take one of the following two values: if j-th generator is up, then $p_j = 1 - FOR_j$; Otherwise, then $p_j = FOR_j$. FOR_j is the forced outage rate (FOR) of generator j. Likewise, the state of each transmission line can also be included in the probability calculation based on its availability. If the state probability is less than the specified threshold (a small value below which the corresponding states are neglected), it is assigned a very small fitness value for reducing its chance of participating in subsequent GA operations. This is because it is a small-probability system state which contributes very trivially to the reliability index no matter it is a success or failure state.
  - Query the hash table to determine if it is an existing system state. If it is, a small value will be assigned as its fitness.
  - Determine the loss of load status for state i with respect to the maximum load level L_{max} using the load flow procedure discussed in Section II.

  - If there is no load curtailment for the system state, the fitness of its corresponding individual is assigned a very small value in order to reduce its chance to contribute to the next generation, since it represents a success state. The probability of the failure state i can be calculated using (IV.10).
  - Calculate the number of equivalent states of the evaluated failure state i, which can be obtained through the permutations in each group:

$$Copy_i = \binom{G_1}{O_1} \dots \binom{G_j}{O_j} \dots \binom{G_n}{O_n} \quad \text{(IV.11)}$$

    where O_j is the number of "ones" in group j of length G_j.
  - The fitness of the failure state is defined as

$$Fit_i = Copy_i * P_i \quad \text{(IV.12)}$$

    This is the objective function to be maximized by GA, which is the total loss of load probability of the system state and its equivalents. We can see that its value is determined by the state of each generator.
  - Save information on the failure states as a record in the hash table with high lookup efficiency, whose inner working will be detailed in the following subsection. Besides each generator status, the stored information includes P_i, Fit_i, and $Copy_i$, which will be used in calculating reliability indices.
  - Repeat the above procedure for the remaining individuals until all of them are evaluated. Before each evaluation, the individual under consideration will be checked to ensure it is neither identical nor equivalent to any previously evaluated ones. If it is a previously evaluated state, its fitness will be assigned a very small number so as to eliminate it as quickly as possible in the following optimization operations.
- Step 3: Increase the iteration number by one.
- Step 4: Check if any stopping criterion is met. If it is, halt the algorithm; otherwise, proceed to the next step.
- Step 5: Different GA operators are applied for producing the next generation, and then repeat the procedure from Step 2 to Step 4 until the stopping criterion is met.
- Step 6: Wait until GA execution in each processor has ended.
- Step 7: Calculate the reliability index LOLP based on the achieved state array, which is stored in the hash table.

$$LOLP = \sum_{i=1}^{FS} Copy_i * P_i \quad \text{(IV.13)}$$

  where FS is the number of failure states found out by the GA.

C. Hash Table Based State Storage and Retrieval

In the proposed method, the derived dominant failure states are stored in a table, whose size keeps becoming larger as the search goes on. Each time when a new system state is evaluated, first it is compared with the existing system states to see if it has been selected and evaluated. In doing so, usually a linear search is needed to scan the table from the first record until any possible identical record is located, which results in the computational complexity of $O(n)$, where n is the number of records in the table. Furthermore, to compare two system states, some extra calculations are needed. This kind of linear search is usually not time-efficient when the table size becomes large. For instance, for a system state which has never been selected and evaluated, the entire table needs to be scanned to determine its uniqueness. In this study, since extensive interactions are involved, a storage table with high lookup efficiency will greatly improve the computing efficiency. Thus, to speed up the lookup process, hash table [12, 13] is used to store the dominant failure states found out by the parallel GA. Hash table boosts the search efficiency to $O(1)$, which means that the search effort for a specific record is constant. Namely, the lookup is not or only affected very trivially by the table size. Hash function is used to map the keys (system states) to a unique index (usually an integer) in most situations. Although there are still small chances for the occurrence of collisions when two different keys are mapped to the same index, the hash table provides various mechanisms to avoid and resolve this kind of undesirable collisions in order to ensure each key only corresponds to a single record in the table. The adoption of hash table is a useful way to render the record search highly immune to the increasing table size during the system-state search process.

Extensive table operations are needed in the proposed parallel implementation since the table should contain all distinct system states. They are not a one-time query, instead, they are continuous interactions. Each time before the processor starts evaluating a system state, it needs to look up the shared table to ensure that it is not a previously evaluated one. This is so done that unnecessary load flow analysis can be avoided for ineligible and duplicate system states.

V. Simulation Studies

The IEEE Reliability Test System (RTS) [14] was chosen to test the proposed method. It has 24 buses (10 generation buses and 17 load buses), 38 lines and 32 units. The system annual peak load is 2850 MW and the total installed generating capacity is 3405 MW. Its one-line diagram is shown in Figure 3. The generation data including generator sizes, numbers, types, and forced outage rate (FOR) are given in Table I.

TABLE I
Generating Unit Data.

Unit Size (MW) & Number	Unit Type	Unit FOR
12 (5)	Oil	0.02
20 (4)	Oil	0.10
50 (6)	Hydro	0.01
76 (4)	Coal	0.02
100 (3)	Oil	0.04
155 (4)	Coal	0.04
197 (3)	Oil	0.05
350 (1)	Coal	0.08
400 (2)	Nuclear	0.12

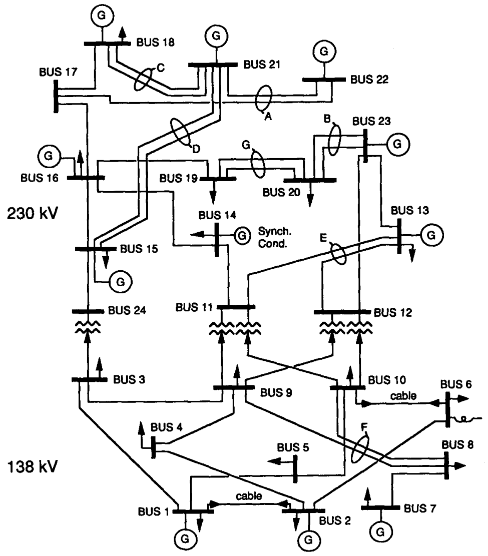

Fig. 3. One-line diagram of the IEEE Reliability Test System (RTS).

A. Simulation Results

In this section, we will report some simulation results obtained from a couple of topologies in the multiple-deme parallelization method. The total execution time for the synchronous parallel program T_p includes four segments: parallel time, serial time, synchronization time, and communication time [15]. The parallel time is the time period when all the processors are executing tasks. Serial time refers to the time used to execute the tasks which cannot be parallelized. The synchronization time is that used to wait for other processors to finish the tasks. The communication time is used to exchange information between processors. Usually the performance of a parallel program can be measured by speedup S_p and efficiency η_p, which are defined as follows:

$$S_p = T_s/T_p \tag{V.14}$$

$$\eta_p = S_p/P \tag{V.15}$$

where T_s is the total time needed to execute its corresponding serial program, and P is the total number of processors.

A modified GA [3] is used for the simulation studies. The crossover rate is 0.85, mutation rate is 0.05, migration rate is 0.1, and deme size is 60. The program stops in each processor when no new failure states can be searched out for ten consecutive iterations. The parallel program is implemented in a 64-processor system with distributed-shared-memory. All memory is physically distributed, which can be accessed by all processors as a single shared address space. In the parallel implementation, message passing interface (MPI) [16] is used as the communication protocol. Here some simulation results based on multi-deme parallel GA of ring topology, master/slave parallel GA, and parallel MCS utilizing different numbers of processors are listed from Table II to IV.

From the simulation results, we can see that the performance of multi-deme parallel GA is comparable to or better than other two parallelization topologies in terms of speedup T_p and efficiency η_p. As the major parallelization involved in the latter two parallel algorithms is the distribution of state-evaluation tasks among multiple processors [7], their results are quite comparable. Furthermore, we find that the multi-deme GA consumes the least time, and the solution quality is also somewhat higher since more failure states are found. It should be noted there is a master processor in the master/slave GA and MCS methods which does not conduct sampling, but there is no such a coordinating processor in the multi-deme GA. This is also one of the reasons why the multi-deme GA exhibits higher sampling efficiency than the master/slave topologies.

TABLE II
SIMULATION RESULTS USING 3 PROCESSORS

	Multi-deme GA	Master/slave GA	Parallel MCS
LOLP	0.1471	0.1469	0.1476
$T_p(sec)$	632.74	723.7	1011.8
S_p	2.23	1.95	1.93
$\eta_p(\%)$	74.33	65.00	64.33

TABLE III
SIMULATION RESULTS USING 7 PROCESSORS

	Multi-deme GA	Master/slave GA	Parallel MCS
LOLP	0.1473	0.1472	0.1467
$T_p(sec)$	235.95	244.15	337.27
S_p	5.98	5.78	5.79
$\eta_p(\%)$	85.43	82.57	82.71

B. Some Discussions

In this study, the role of GA is somehow different from that when it is used for function optimization. As a result, the GA versions of high convergence performance are not suitable anymore since it may avoid some non-optimal meaningful states. There is a tradeoff between the number of system states sought and search time in this state scanning problem. For this purpose, diversity preservation should be ensured during the search process in order to find out as many dominant failure states as possible in an efficient manner. Besides the hash table that is used to punish the duplicate states, there are at least a couple of other possible ways for enhancing the state diversity in the search process.

- Niching and fitness sharing: The selection pressure in GA may lead to a population made almost entirely of replicas of the best individuals found so far. As long as the optimization continues, more mutants of the best individuals will be created. From the inner working of niching and fitness sharing [17], the impact of genetic shift can be reduced or even eliminated since the shared fitness of the replicas is reduced when their number increases. In doing so, other individuals with lower raw fitness or belonging to less populated niches will have higher chances to compete for selection and reproduction.
- Different objective functions: From the mechanism of GA, the search is guided by the specified objective functions. Thus, the change of objective function will have a direct impact on the search trajectory. Here we defined the loss of load probability as the objective

TABLE IV
SIMULATION RESULTS USING 15 PROCESSORS

	Multi-deme GA	Master/slave GA	Parallel MCS
LOLP	0.1475	0.1475	0.1484
$T_p(sec)$	117.19	118.09	164.10
S_p	12.04	11.95	11.90
$\eta_p(\%)$	80.27	79.67	79.33

function attempting to find out the dominant failure states which contribute most significantly to the system LOLE. This has turned out to be an effective definition of objective function in this problem. However, it is possible that there are alternative ways to define the objective function. Especially when other reliability indices need to be calculated, this may not be the only effective objective function that can be stipulated. For instance, when the expected unserved energy needs to be computed, in some processors the objective function may be defined as the unserved energy of each system state. In doing so, the system states leading to high unserved energy of relatively low probability will have higher chances to be found out by the search algorithm. Furthermore, bi-objective optimization procedure can also be used to enhance the diversity of the solutions found out. For instance, in [8] two objective functions are defined in terms of loss of load probability and unserved energy of each system state.

VI. CONCLUSIONS AND FUTURE WORK

Reliability evaluation for composite power systems may be time-consuming when the number of system states is large. since power flow analysis is involved in deciding the status of each sampled system state. In this work, parallelization implementation of genetic algorithm based on multiple sub-populations is used for reliability evaluation to expedite the computing process. For each sub-population, the dominant failure states are sought out by the search algorithm, which can later be used for estimating the system's loss of load probability. Some measures such as hash tagging are taken for diversity preservation during the search of these states. As compared with its sequential counterpart, the parallel method has turned out to be effective in improving the computing efficiency and solution quality. One future research direction is to figure out other schemes in order to find out the meaningful system states in a more efficient fashion, which may involve using other objective functions. Meanwhile, the exploration of other parallelization topologies is also worthwhile since it may lead to even higher computational efficiency.

REFERENCES

[1] Anders, G. J. (1990). *Probability Concepts in Electric Power Systems*, Wiley, New York.

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

[3] Samaan, N. and Singh, C. (2002). Adequacy assessment of power system generation using a modified simple genetic algorithm, *IEEE Transactions on Power Systems*, Vol. 17, No. 4, pp. 974–981.

[4] Samaan, N. and Singh, C. (2003). Genetic algorithms approach for the evaluation of composite generation-transmission systems reliability worth, *Proceedings of IEEE PES Transmission and Distribution Conference and Exposition*, Vol. 1, pp. 113–119, Sept. 2003.

[5] Samaan, N. and Singh, C. (2004). Genetic algorithms approach for the assessment of composite power system reliability considering multi-state components, *Proceedings of the 9th International Conference on Probabilistic Methods Applied to Power Systems*, Ames, Iowa.

[6] Wang, L. and Singh, C. (2007). Genetic algorithm based adequacy evaluation of hybrid power generation system including wind turbine generators, *IEEE Proceedings of International Conference on Intelligent Systems Applications to Power Systems*, November, 2007.

[7] Wang, L. and Singh, C. (2009). Reliability evaluation of composite power systems using parallel genetic algorithms: Some conceptual and simulation studies, *Proceedings of IEEE Power System Conference and Exposition (PSCE09)*, Seattle, WA, March, 2009.

[8] Patra, S. B., Mitra, J., Earla R. (2006). A new intelligent search method for composite system reliability analysis, *IEEE PES Transmission and Distribution Conference and Exhibition*, pp. 803–807, 2006.

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

[10] Bahi, J. M., Contassot-Vivier, S., and Couturier, R. (2008). *Parallel Iterative Algorithms: From Sequential to Grid Computing*, Boca Raton: Chapman & Hall/CRC Press.

[11] Cantu-Paz, E. (2000). *Efficient and Accurate Parallel Genetic Algorithms*, Boston: Kluwer Academic Publishers.

[12] Hash table, Available online: http://en.wikipedia.org/wiki/Hash_table.

[13] Booker, G., Sprintson, A., Singh, C., and Guikema, S. (2008). Efficient availability evaluation for transport backbone networks, *Proceedings of 12th Conference on Optical Network Design and Modeling*, Vilanovaila Geltru, Spain, March 2008.

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

[15] Gubbala, N. and C. Singh (1995). Models and considerations for parallel implementation of Monte Carlo simulation methods for power system reliability evaluation, *IEEE Trans. on Power Systems*, Vol. 10, No. 2, pp. 779–787, May 1995.

[16] Snir, M., Otto, S., Huss-Lederman, S., Walker, D., and Dongarra, J. (1998). *MPI-The Complete Reference*, Vol. 1, The MPI Core, Cambridge, MA: The MIT Press.

[17] Della Cioppa, A., De Stefano, C., and Marcelli, A. (2004). On the role of population size and niche radius in fitness sharing, *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 6, pp. 580–592.