ADEQUACY ASSESSMENT IN POWER SYSTEMS USING GENETIC ALGORITHM AND DYNAMIC PROGRAMMING

A Thesis

by

DONGBO ZHAO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2010

Major Subject: Electrical Engineering

Adequacy Assessment in Power Systems Using Genetic Algorithm and Dynamic Programming

ADEQUACY ASSESSMENT IN POWER SYSTEMS USING GENETIC ALGORITHM AND DYNAMIC PROGRAMMING

A Thesis

by

DONGBO ZHAO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Chanan Singh
Committee Members,	Aniruddha Datta
	Georgia-Ann Klutke
	Mehrdad Ehsani
Head of Department,	Costas Georghiades

December 2010

Major Subject: Electrical Engineering

ABSTRACT

Adequacy Assessment in Power Systems Using Genetic Algorithm and Dynamic Programming.

(December 2010)

Dongbo Zhao, B.S., Tsinghua University

Chair of Advisory Committee: Dr. Chanan Singh

In power system reliability analysis, state space pruning has been investigated to improve the efficiency of the conventional Monte Carlo Simulation (MCS). New algorithms have been proposed to prune the state space so as to make the Monte Carlo Simulation sample a residual state space with a higher density of failure states.

This thesis presents a modified Genetic Algorithm (GA) as the state space pruning tool, with higher efficiency and a controllable stopping criterion as well as better parameter selection. This method is tested using the IEEE Reliability Test System (RTS 79 and MRTS), and is compared with the original GA-MCS method. The modified GA shows better efficiency than the previous methods, and it is easier to have its parameters selected.

This thesis also presents a Dynamic Programming (DP) algorithm as an alternative state space pruning tool. This method is also tested with the IEEE Reliability Test System and it shows much better efficiency than using Monte Carlo Simulation alone.

DEDICATION

To My Girl Friend Bin Liu

ACKNOWLEDGEMENTS

I would like to firstly thank my committee chair, Dr. Singh, and my committee members, Dr. Ehsani, Dr. Datta, and Dr. Klutke, for their guidance and support throughout the course of this research. Special thanks to Dr. Singh for his kind help, support, and supervision, which made my study and research at Texas A&M University a wonderful experience with joy.

Thanks also go to my friends, colleagues, and the department faculty and staff at Texas A&M University for accompanying me and helping me all the way till now. And I would like to thank my parents and my girlfriend for their love and encouragement.

Finally, thanks to Jesus Christ for bringing me here, and for giving me a cheerful time at Texas A&M University.

NOMENCLATURE

MCS	Monte Carlo Simulation
GA	Genetic Algorithm
DP	Dynamic Programming

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

This thesis is about the improved methods for state space pruning in the reliability calculation of power systems. These methods are expected to help prune the state space of a power system more efficiently and effectively. In order to discuss these methods, introduction to power system reliability analysis is made in this chapter.

A power system can consist of several functional parts: the generating part, the transmission and distribution part, and the load. Generally speaking, the generating part is collection of the generators, no matter what energy sources they are using (thermal, nuclear, hydro, etc.); the transmission and distribution part refers to the transmission and distribution lines and their accessories, like the circuit breakers; the load often refers to the utilization of the electric energy, with specific energy consumption expressed in MWh.

Power systems are connected to ensure energy flow, and hence the planning and the analysis of the connected power systems become essential for the power system operation and energy consumption. The reliability indices help to ensure reliable power system operation, although target setting of indices might be different among different power systems. The entire electric power system is aimed to be running in the most economical fashion while meeting the required reliability limits, i.e., maintaining a

---

This thesis follows the style of *IEEE Transactions on Power Systems.*

reasonable level of reliability.

The field of power system reliability analysis was firstly brought into practice based on the computational methods developed 1970s, that have been widely used in the electric power systems in USA. Improvements of the computational algorithms have been continuously proposed to deal with higher complexity and computing the reliability indices more efficiently.

The power systems have some characteristics that have made their reliability analysis different from other fields. The most typical one is the transmission and distribution constraints, which have made the analysis complicated. These constraints need to be met in order to maintain the normal operation of the power system, while the change in generation and in load would certainly bring change to the flow over the transmission lines. Therefore the transmission and distribution line constraints are to be met at all times. In order to have these constraints met, it is not the simply having the generation equal to or greater than the load.

The generation parts of the power system consist of different groups of generators, which have certain capacities and probabilities of failing. The data of the probability of a generator's failing is normally based on the statistics collected from the field experience. A generator can fail completely, or to a degraded level which is normally called the "derated" status. Generators with the same capacity and failure rate are often categorized into one group, which makes the generation part consist of several groups of generators. We can consider the generators within one group identical to one another, which can simplify adequacy assessment problems in the reliability analysis.

When taking the transmission line constraints into consideration, the power system reliability analysis is called the composite power system reliability analysis. The composite system analysis involves the power flow calculation, and comparison with the transmission line constraints, which becomes quite complicated. When considering the improvement over existing reliability analysis methods, we are more tending to start from the single-area evaluation, which is concerned with the adequacy of generation to supply load, while not considering the transmission constraints.

The load of a power system is changing all the time, which can be understood intuitively by the switching of any household appliance from up to down. There are curves showing the fluctuation of the load during a day or a week, which reveals that within a certain area of the power system, there is always maximum load which comes during certain period of time. In reliability analysis, the load can be modeled as different time variant constants. When considering the generation adequacy, it is important to consider that the peak load is satisfied.

The calculation of reliability indices will be discussed in detail in the following chapters of this thesis. This thesis focuses mostly on the state space pruning technique in adequacy assessment problems, introducing new algorithms or improvements over existing ones, to help truncate the state space of the power system more efficiently and effectively. Case studies are performed to show the improvement of the state space pruning techniques.

This thesis describes two methods of adequacy assessment using state space pruning techniques: modified genetic algorithm and the dynamic programming. The two techniques help the state space pruning process converge faster than conventional methods. These pruning methods are then combined with Monte Carlo simulation to get the final reliability indices. Detailed definitions are included in the following chapters.

In this thesis, Chapter I is the introduction to power system adequacy assessment and the brief introduction of this thesis;

Chapter II is the explanation of the calculation of reliability indices and state space pruning of the power system;

Chapter III is about the modified genetic algorithm used in the state space pruning process in adequacy assessment problems, with case study illustrating the improvement;

Chapter IV is about the dynamic programming algorithm used in the state space pruning process in adequacy assessment problems, with case study;

Chapter V is the conclusion of this thesis.

CHAPTER II

STATE SPACE PRUNING IN ADEQUACY ASSESSMENT

The basic function of a modern electric power system is to provide an adequate supply of electrical energy to its customers as economically as possible and with a reasonable level of continuity and quality. Quality in this sense implies constancy of system voltages and frequencies. The ability of the power system to meet its load requirements at any time is referred to as the 'reliability' of the system.

In power system reliability analysis, the objective is commonly the calculation of the reliability indices of a given power system, and their comparison with the pre-set reliability requirement limit. The reliability requirement limit is set beforehand for the specific system, with detailed range of indices required for a certain level of operation.

System reliability can be classified into two distinct aspects of system security and system adequacy. System security involves the ability of the system to respond to disturbances arising internally, whereas system adequacy relates to the existence of sufficient facilities within the system to satisfy the customer load demand.

Although the calculation of the reliability indices of the power system is not in real time, it will need to be done as fast as possible. This is because reliability calculations may need to be performed many times for a given planning study.

In this chapter, the calculation of reliability indices of single area power systems will be shown, followed by the description of state space pruning in adequacy assessment process.

A. Single Area Power System Reliability Analysis

In single area power system reliability analysis, the adequacy of generation to supply load is considered, without concerning the transmission constraints, i.e., assuming that transmission system can transport generation to load points. So the reliability analysis of single area power system is mostly dealing with the generating capacity evaluation, which is the adequacy assessment. Generation adequacy is usually predicted using one or more indices which quantify expected system reliability performance, and implemented using criteria based on acceptable values of these indices. A complete reliability evaluation of a power system involves a comprehensive analysis of its three principal functional zones, namely generation, transmission and distribution. These functional zones can be combined to give the hierarchical levels (HL) under which the various techniques used in adequacy assessment are grouped. Adequacy assessment at HLI is concerned only with the generation facilities. The transmission and distribution facilities are assumed to be fully reliable and capable of moving the generated electrical energy from the generation points to the customer load points. At HLI therefore, only the total system generation is examined to determine its adequacy to meet the total system load requirements. HLII assessment includes a composite appraisal of both generation and transmission facilities and HLIII involves all three functional zones in an assessment of a customer load point adequacy. This thesis is restricted to generation adequacy assessment at HLI which deals with generation adequacy assessment.

Therefore, the consideration of single area reliability becomes the adequacy assessment which is:

- Usually measured through the use of some reliability index, which is also the adequacy index that quantifies system reliability performance and it is enforced through a criterion based on an acceptable value of this index;
- The load used for the determination of whether the generation is adequate or not can be either the peak load of or the load cycle over the period of investigation.

Reliability indices can be generally divided into two categories:

- Deterministic indices: the indices that reflect postulated conditions. They are not directly indicative of the factors that affect power system reliability and are therefore of little use. Their calculation is simple and requires little data. These indices include the percent reserve margin, which reflects the excess of installed generation capacity over annual peak load, and the reserve margin in terms of largest unit, which recognizes the importance of unit capacities in relationship to reserve margin.
- Probabilistic indices: the indices that directly reflect the uncertainty in the power system reliability and have the capability of reflecting the various parameters which can impact the system reliability. These indices permit the quantitative evaluation of system alternatives through direct consideration of parameters that affect reliability. This is the most

commonly used criterion of the power system reliability analysis.

The most commonly used probabilistic index is Loss of Load Expectation (LOLE). This index is usually obtained from the calculation of Loss of Load Probability (LOLP). The other indices are Loss of Load Frequency (LOLF), Loss of Load Duration (LOLD) and Expected Energy Not Supplied (EENS). The relationship between LOLE and other indices is:

$$\mathrm{LOLE} = \mathrm{LOLD} * \mathrm{LOLF}$$

B. State Space Pruning

In this study, the state space of a power system is defined to consist of all possible states of generators, with each generator having its success (up) and failure (down) states only. In order to make the adequacy assessment, we only consider the generating capacity in comparison with the total load of the system, so as to show the improvement over existing state space pruning methods. Power flow and transmission congestion issues are not included. A success state means that the generating capacity is no less than the total load, while the failure state means that the generating capacity is less than the total load, where the total load is set at a constant value in both cases. When the system is having "n" generating units, with each unit having 2 states: up and down, the total number of states of the system is:

$$K = 2^n$$

Each of the K states represents a system state, and each system state is defined by the status of n binary components.

State space pruning as approached in this thesis is based on the findings presented in reference [1]. This paper proposed a computationally efficient linear program for calculating the DC load flow model while describing the pruning based on the decomposition-simulation methodology. This reference also developed the theory and mathematics behind this methodology in a solid and concise fashion. State space pruning itself is a methodology applied in simulations, particularly Monte Carlo Simulation (MCS), in order to reduce the number of states sampled while awaiting convergence of the algorithm. This is discussed in detail in reference [1] where success states are pruned away from the main state space in order to create a higher density of failure states. The figures from that paper best describe the application and usage of state space pruning:

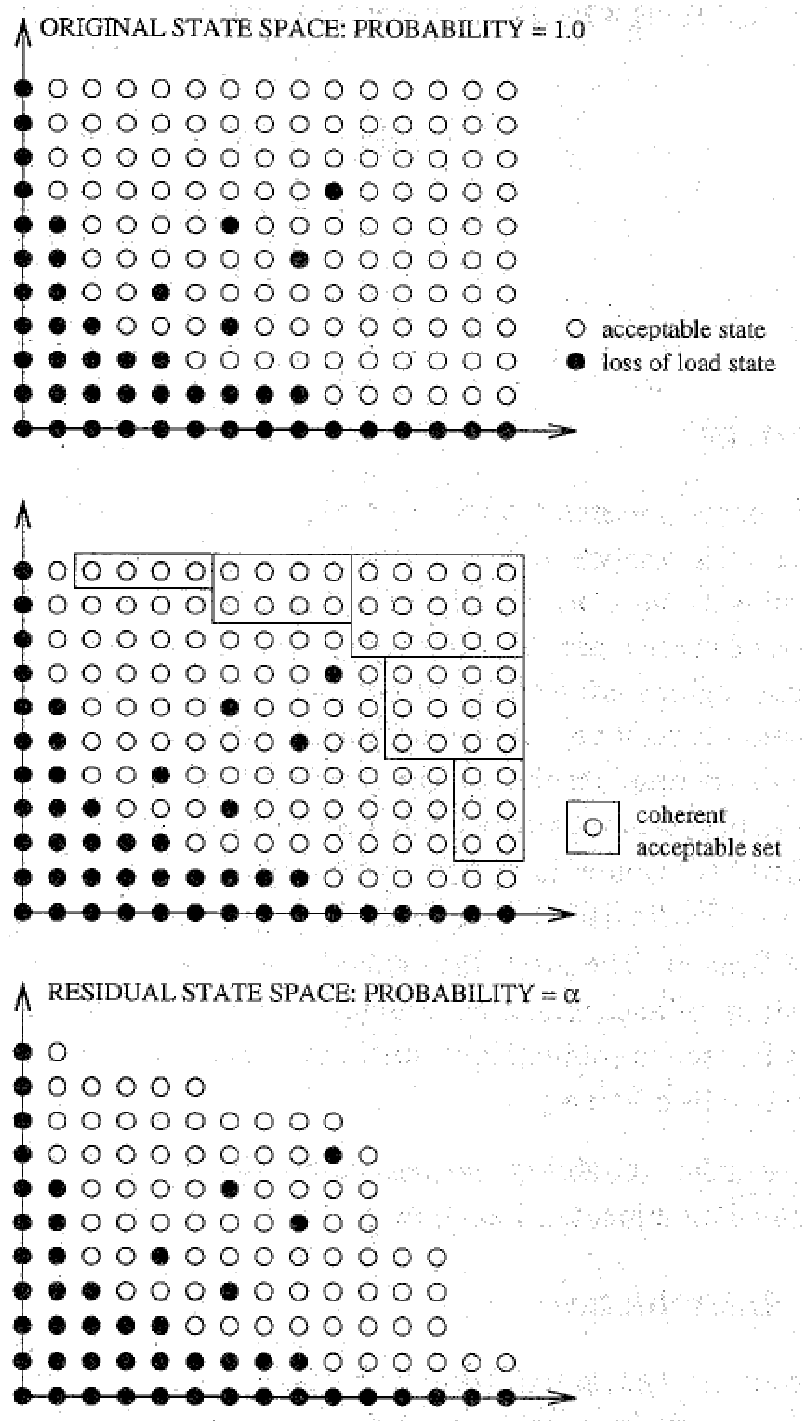

Fig. 1 State Space Pruning Description [1]

After truncating some of the coherent acceptable sets (success states), the state space is truncated into a partial state space, i.e., the pruned state space with high density of unacceptable sets (failure states). This higher density of failure states allows the MCS to sample more failure states which, in turn, forces the MCS to converge faster. Once the MCS has converged, the reliability indices calculated must be re-calculated through the reintroduction of the pruned states. This process of pruning, simulation, and re-calculation allows the estimation of the original state space more efficiently.

As an example, consider a very small state space that is comprised of 50 success states and 50 failure states for a total of 100 states. Without pruning, the density of failure states is 50%. When this state space is subjected to pruning, a portion of the success states will be removed. Consider the removal of 40 success states through pruning. Not only has the entire state space shrinks by 40%, but the density of failure states has also increased from 50% to 50/60= 83.3%. When MCS is run against this pruned state space it should converge much more quickly as a larger portion of sampled states will be failure states. Of course, the indices over the pruned state space will need to convert into the corresponding indices over the original entire state space, with the conditional probability of the pruned state space, which is the α shown in Fig. 1. The value of α or (1- α) is obtained with the calculation of the probability of each success state, and have the ratio of their sum over the probability of the whole state space, which is 1 in the figure. The conversion of the pruned state space to the original state space will be described in detail with equations in the next two chapters, along with specific state space pruning techniques.

C. Monte Carlo Simulation and Pruning Techniques

Adequacy assessment analysis of power systems can be treated as a large and complex combinatorial problem. Because of this, the difficultly in computation increases dramatically as the dimensionality of the problem increases. Probabilistic methods are increasingly becoming popular in power system reliability evaluation due to their capability of accounting for increasing system uncertainties. Recent work has been done in both Monte Carlo Simulation (MCS) and Population-based Intelligent Search (PIS) in order to develop new, improved, and computationally more efficient methods for power system adequacy evaluation.

Reliability indices of an actual physical system could be estimated by collecting data on the occurrence of failures and durations of repair. The Monte Carlo method mimics the failure and repair history of components and the system by using the probability distributions of component states. Statistics are collected and indices estimated by statistical inference.

Monte Carlo simulation is based on stochastic simulation which can be used for evaluating reliability indices of power systems at various levels. MCS itself comes in two forms: sequential and non sequential. In sequential MCS, system states are typically sampled in time order over different periods. This usually requires greater computational effort. In non-sequential MCS, the system states are sampled randomly. This enhances computational efficiency and it is the preferred method for this work unless sequential correlations need to be considered.

Generally speaking, the use of MCS comes with both advantages and disadvantages. The major advantage of MCS is that it is able to deal with large and complex power systems. The disadvantage of MCS is that the time for convergence for highly reliable systems can become very long. Although the convergence time of the MCS does not depend on size but the computational time for evaluating the states for large systems can be quite long. The disadvantage in the use of MCS is that as the dimensionality of the modeled system increases so does the computational time. Large or complex systems with high reliability may require such a large number of states to be sampled that MCS will run for an unacceptable amount of time before converging. Because of this, efforts have been made to reduce the computational time of MCS by improving convergence. This is the main objective of the state space pruning process.

PIS is another methodology that has been applied to the reliability evaluation of power systems. As applied to system adequacy, PIS has turned out to be an effective alternative to existing methods since it is able to achieve higher convergence performance in some scenarios. This is mainly for two reasons: PIS does not sample states in a totally random fashion and PIS evaluates multiple states simultaneously. It is a directed search for improving the objective placed before it. Thus objective functions must be chosen to fit the given situation. Typically when measuring adequacy this means designing objective functions that encourage the generation of dominant failure states which leads to the sampling of less states for convergence. With regards to the simultaneous evaluation of multiple states, PIS algorithms do not work with one possible state at a time but with a population of states at a time. Each iteration or generation of a

PIS generates and evaluates multiple states. PIS has been applied to multiple reliability problems in the field of power systems as detailed in reference papers. This thesis will examine the use of PIS algorithms in order to prune the given state space so that MCS may accomplish better convergence.

In this thesis, MCS is used as the basis for comparison, i.e., newly proposed approaches for state space pruning are to be compared for efficiency with the method of using straight MCS without state space pruning. The state space pruning process is the essential part to be considered in the two newly proposed methods in this thesis: the modified genetic algorithm and the dynamic programming algorithm. However, as complete adequacy assessment methods to calculate the final reliability indices of a power system, the combination of the state space pruning algorithms with MCS will be needed to complete the calculation.

CHAPTER III

ADEQUACY ASSESSMENT USING MODIFIED GENETIC ALGORITHM

Several methods have been previously developed to assess the power system reliability and calculate the reliability indices. Adequacy assessment of power systems is an important issue in the reliability analysis. The methods in the adequacy assessment can be primarily divided into two categories: analytical algorithms and numerical simulation algorithms. Recent studies on this topic focused mostly on the numerical simulation algorithms, and have concentrated on Monte Carlo Simulation [1] and intelligent methods [2] such as Genetic Algorithm. Most intelligent methods are population based, and aim at generating certain kinds of states so as to analyze the state space. Intelligent methods have been shown to introduce more computationally efficiency, and clearer steps in the analysis of state space.

Genetic Algorithm has been previously used in the calculation of reliability indices of power systems. In power system reliability calculations, it has been used primarily as a search tool to identify states with specific characteristics.

Previous studies on the application of Genetic Algorithm in reliability evaluation of power system was originally developed by Singh's group at Texas A&M and later by many other researchers. Reference [3] discussed the usage of GA as a state sampling tool for the composite generation and transmission system, and [4] provided the GA sampling techniques regarding multi-state components. More detailed work has been done to by Wang and Singh, with focus of the application of GA together with Monte

Carlo Simulation to analyze the state space of composite power system [5], [6]. Various ways of developing the Genetic Algorithm in order to have faster convergence of the calculation or to have better computational efficiency have been described. These include parallel GA in combination with Monte Carlo simulation.

The mechanism of Genetic Algorithm is described in detail in [7] and [8]. In [7], a unique simple Genetic Algorithm is proposed in the adequacy assessment of power system generation, without any additional technique like Monte Carlo Simulation. It uses only the GA as the sampling tool and does the calculation of reliability indices only from the result of GA, which provides an easier coding and faster convergence than many conventional methods. In [8], the combination of GA and Monte Carlo simulation (GA-MCS) as a new approach is proposed, with better control of the state space pruning status and the overall efficiency. This thesis presents a modified GA-MCS method that has better efficiency and parameter control, based on the GA method in [7] and the combination of GA and Monte Carlo simulation in [8].

A. Basis of the Proposed Method

- The State Space to Be Pruned

As described in the last chapter, the state space of a power system is defined to consist of all possible states of generators, with each generator having its success (up) and failure (down) states only. In order to make the adequacy assessment, we only consider the generating capacity in comparison with the total load of the system, so as to show the improvement over existing state space pruning methods.

When the system has "n" generating units, with each unit having 2 states: up and down, the total number of states of the system is:

$$K = 2^{n}$$

Each of the K states represents a system state, and each system state is defined by the status of n binary components.

- Permutations

Normally, there are many generators in a power system that have identical generating capacity and identical failure rate. As discussed above, when considering each generator as a two-state component, generators with the same capacity can be represented in an identical manner.

For example, if 5 generators all have capacity of 400MW each, then the total output of the 5 generators will be 0, 400, 800…, and 2000MW. When only the output generating capacity of the 5 generator group is taken into account, the cases of having a certain output, like 800 MW, are following the permutation of having 2 of the 5 up and 3 of the 5 down. Then the total number of states of having 800 MW is:

$$N_{800MW} = \begin{pmatrix} 2 \\ 5 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix} = 10$$

The probability of having 800MW output is:

$$P_{800MW} = \begin{pmatrix} 2 \\ 5 \end{pmatrix} * (1 - FOR)^{2} * \begin{pmatrix} 3 \\ 5 \end{pmatrix} * FOR^{3}$$

Where FOR is the forced outage rate of these 5 generators, i.e., the probability of its being "down".

- Genetic Algorithm

The primary use of the GA is to find the optimal value of a certain function under some constraints. A GA is a simulation of evolution where the rule of survival of the fittest is applied to a population of individuals. In the basic GA, an initial population is randomly created. Population individuals, called chromosomes, are then evaluated by applying some function or formula. A new population is selected from the old one based on the fitness value of the individuals. Some genetic operators are then applied to some of the newly selected population to create the final new generation. The most commonly used genetic operators are crossover and mutation. The process is repeated from one generation to another until reaching a stopping criterion.

Using binary number to represent up and down as 1 and 0, each state in the state space becomes a chromosome in GA. The length of the chromosome, i.e., the number of generators, is equals to n.

As discussed above, generators can be divided into groups, based on the generation capacity. Generators with identical capacity and failure rate are defined to be within the same group. Therefore, the chromosome can be divided into q groups, with the i th group having Li generators,

$$\sum_{i=1}^{q} L_i = n$$

Normally, the Genetic Algorithm has three major steps in generating new states:

- Selection: selecting from former generation, based on certain criteria;
- Crossover: having certain pairs of chromosomes exchange in certain bits;
- Mutation: having certain bits of certain chromosomes change from 0 to 1 or from 1 to 0.

GA commonly follows the procedure from selection, crossover, and then to mutation. The mechanisms of deciding which bit or which pair of chromosomes should be operated, the crossover or mutation is based on the parameters: crossover rate and mutation rate, which means to have a designated proportion of the bits or pairs to be operated. The first generation is usually generated randomly, following similar random number generation process as Monte Carlo. The selection process normally follows probability gain, i.e., the one with larger probability in the "father" generation is more likely to be selected.

- Genetic Algorithm in State Space Pruning

In this paper, GA is used primarily in the state space pruning process. After pruning the state space, the residual state space can be analyzed using various kinds of tools, in which Monte Carlo is the most typical one. GA has shown its efficiency in generating success states that are needed to be pruned. When the success states generated are truncated, the pruned space will have a higher density of failure states, so it will converge faster in the reliability calculation process. For example, if a state space is composed of 80 success states and 20 failure states, after pruning 60 success states, the percentage of failure states will be raised significantly from 20% to 50%. Usually, the probability of the pruned states is the key point of the pruning. If a state space is having success states occupying probability 0.9 and failure states 0.1, then after pruning 0.8 success probability, the state space is having 0.1/0.2=50% rather than 0.1/1=10% at the beginning. Taking both the pruning and calculation steps into account, the total time and accuracy will be the criteria to judge whether a method increases efficiency.

In [7], GA is used as the final calculation tool of reliability indices. In [8], GA is used only in the pruning process to generate success states, which are later truncated to form the pruned state space so as to increase the density of failure states in the residual space. Monte Carlo simulation is used to obtain the final calculation results, so that this method is named GA-MCS. Both the methods in [7] and [8] are generally faster in the total reliability analysis time than conventional methods.

- Modified GA-MCS and Its Parameter Selection

The method proposed in this paper is based on the GA-MCS but has significant differences. With the introduction of permutation within generation groups, the GA part will be faster in generating new states.

In the GA process, the three operations are in the sequence as: selection, mutation, and crossover. This is because from the past experience, the mutation rate is hard to decide, but has important influence on the pruning speed. Intuitively speaking, when the selection process selects states from the first generation, the states and their permutations that have large probabilities are most likely to be selected. If followed by the crossover operation, it will not introduce many new states because they are mostly "success" states, i.e., having very few 0s but mostly 1s. Then the essential step in generating new states is the mutation operation. But by putting mutation before crossover, not only this mutation but also the crossover process will notably produce new states.

GA and the modified GA are showing to have fast pruning speed at the beginning but becomes flat soon. Hence the criterion of the GA in the pruning process can be determined using the slope of increasing success states pruned within certain unit of time.

In the previous study, it has been shown that a suitable set of GA parameters is essential to have the best pruning process. It is discovered in [7] that GA is not strongly dependent on the population size, or the crossover rate, but is significantly affected by the mutation rate. The change of mutation rate of GA shows difference in the number of states and the probability of these states generated, but is not indicating a clear trend. Therefore, mutation rate is often set as GA recommended number, like 0.06 in [7].

In the modified GA-MCS method proposed in this thesis, since the mutation operation is moved before crossover, experiments have shown that the change of mutation rate will have obvious effect and will be easy to find the tendency of the change on the states and their probability generated. The other parameters, the crossover rate and the population size, are shown to be still not having significant influence. Therefore, the selection of parameters will have clearer rules.

B. Algorithm Structure

- First Generation

The first generation is generated randomly. Meanwhile, the following tasks need to be done:

- Select parameters: population size, mutation rate, crossover rate, stopping parameter
- Input system information: generator capacities, generator groups, reliability parameters of generators (FOR, λ, μ), peak load of the system
- Calculate the generating capacity of each state generated, and compare with the peak load

$$Gen_i = \sum_{j=1}^{n} b_j * cap_j$$

$$state\ i = \begin{cases} success & Gen_i > peak\ load \\ failure & Gen_i < peak\ load \end{cases}$$

where bj is the binary number of the ith generator representing its up or down status, and capj is the capacity of the ith generator.

- Calculate the probability of each chromosome

$$P_i = \prod_{j=1}^{n} gp_j$$

$$gp_j = \begin{cases} 1 - FOR_j & if\ b_j = 1 \\ FOR_j & if\ b_j = 0 \end{cases}$$

- For the state generated, examine whether it has been generated or its permutation states have been generated. If not, calculate its permutations:

$$Perm_i = \prod_{j=1}^{q} \binom{L_j}{O_j}$$

where L represents the success states in its group and O is the number of generators in that group.

- Evolution of New Generation

  - Selection: the selection process is described in detail in [7]. In the selection operation, a state is selected based on its permutation probability, which is:

$$ps_i = \frac{perm_i * P_i}{\sum perm * P}$$

and random number is generated to fall within one of the intervals of ps.

  - Mutation: for the states selected from the "father" generation, mutation operation is applied by having each bit of the chromosome examined with a random number generated and compared with the designated mutation rate. If it is less than the mutation rate, then the bit is changed.
  - Crossover: for each pair of the states after mutation, generate random number to compare with the crossover rate. If it is less than the crossover rate, then apply the crossover by generating an integer from [0, 1… n-1] to decide from which bit the two chromosomes exchange.
  - Keep generating new generations and store all success states and their probability, until the program meets its stopping criterion.

- Stopping Criterion

During the generation of success states, keep calculating the slope, i.e., the incremental speed of the success probability generated. Set a constant parameter Sc to compare with the slope. When the slope is less than Sc, the pruning program will be stopped.

Another way of setting the stopping criterion is to set a constant parameter Ps, which means the desired pruned success probability. When the total probability of success states becomes or exceeds Ps, the program is stopped. This stopping criterion is easier to set but may cause some problems. When Ps is too large, the pruning process will have lack of efficiency because it will converge much slower. When Ps is too small, it will not fully use the effect of the pruning operation.

- Monte Carlo Simulation

In the pruned state space, Monte Carlo simulation is then applied, until its stopping criterion is met. Monte Carlo simulation will sample states within the residual state space, and store all the failure states it has generated, in order to calculate the reliability indices.

- Calculation of Reliability Indices

The calculation of reliability indices involves the calculation of LOLP and other indices. The calculation needs to consider both the states generated by Monte Carlo simulation, and the state generated by the pruning process. It will then obtain the final LOLP based on the conditional probability of the pruned space. The pruned state space has a probability of:

$$P_{MCS} = 1 - \sum perm_i * P_i$$

The LOLP based on the LOLP of the pruned space is:

$$LOLP = LOLP_{MCS} * P_{MCS}$$

Other reliability indices can also be obtained from the results of the Monte Carlo simulation, and can be converted to the original state space as shown in the equation above.

C. Case Studies and Results

We tested the modified GA method and the previous method in the pruning of the state space on the same computer and with same coding person and coding technique. The software in this study was written in Matlab.

The methods were tested using two different test systems: IEEE-RTS79 (RTS79) [9] and the Modified Reliability Test System (MRTS) [10], [11]. IEEE RTS79 system is a test system consisting of 24 buses, 38 transmission lines, and 32 generators. The annual peak load for the system is 2850 MW and the total generating capacity is 3405 MW. MRTS is the same as RTS79 except that all generation levels are doubled and all load levels are raised to 1.8 times. Hence the peak load of the system becomes 5130 MW and the generating capacity is 6810 MW. These systems were chosen based on the fact that most GA methods proposed so far have used these systems to test.

In this study, only the peak loads in these test systems are considered to give the adequacy assessment. When testing in the MRTS system, it gives similar results as in RTS 79 system. The following figures are mostly results from RTS79 system.

- State Space Pruning Process

The following parameters are used: input population size =40, crossover rate =0.5, mutation rate =0.5. The mutation rate is selected based on the discussion below in B part. The stopping criterion of the pruning process is Sc=40,000. The time in the figure is the CPU time. Figure 2 is the number of the success states pruned by the proposed method.

With the same parameters, the previous GA will prune as shown in Figure 3. In these figures, the number of the success states pruned has “jumps” because of the permutation number of each success state is taken into account. This is an advantage of the application of the permutation numbers because the success states will be truncated in groups when one of the permutation states is generated.

It can be seen that the modified method is much faster than the previous GA in the pruning process when counting the number of success states. For the following procedure, the pruned space will be used with Monte Carlo simulation.

Figure 4 and Figure 5 are the probabilities pruned by the proposed method and by previous method. As a searching tool, GA cannot always get the same result in every run. But the tendency will be clearly shown after running 20 times. Figures in this paper are typical results from the 20 runs.

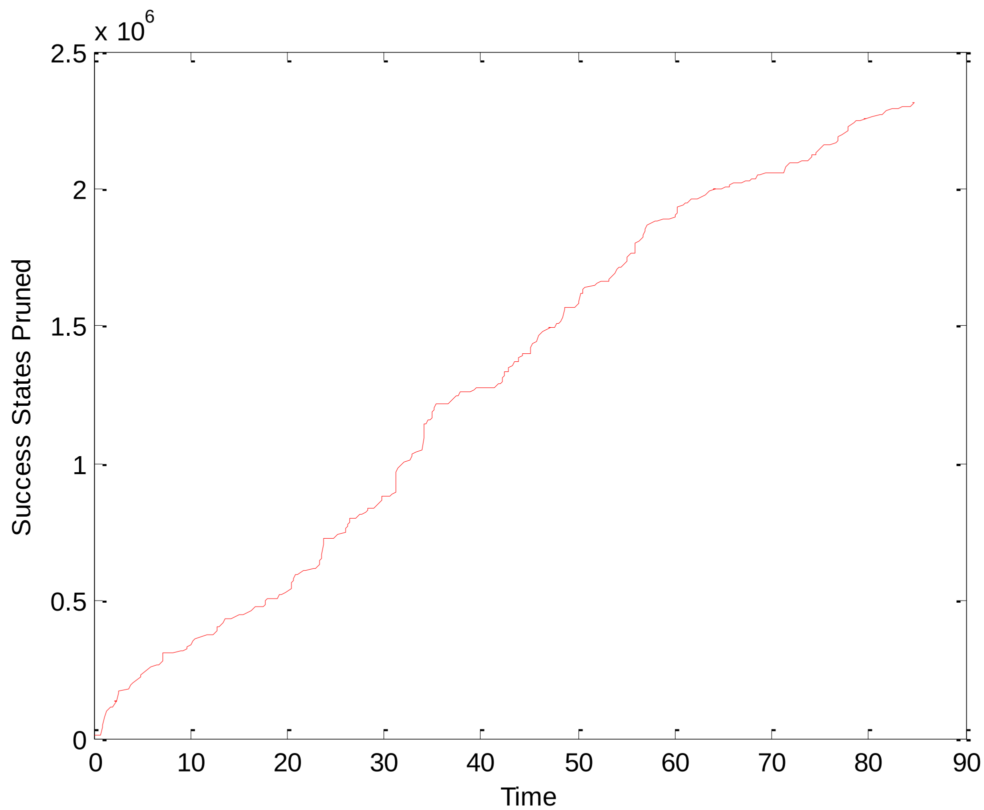

Fig. 2 Success States Pruned by Modified Method vs. Time

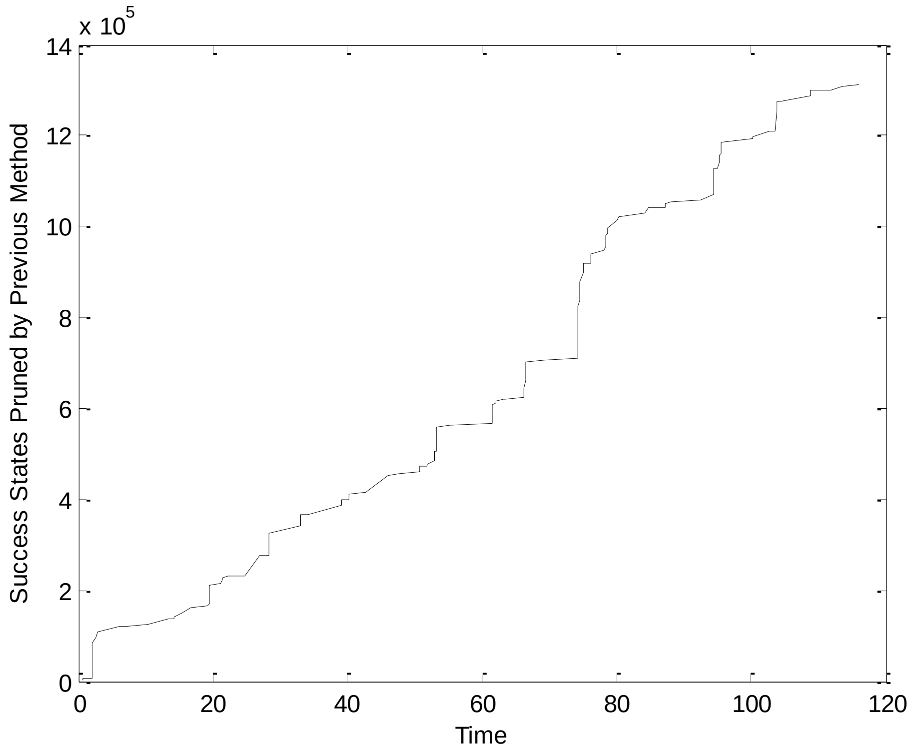

Fig. 3 Success States Pruned by Previous Method vs. Time

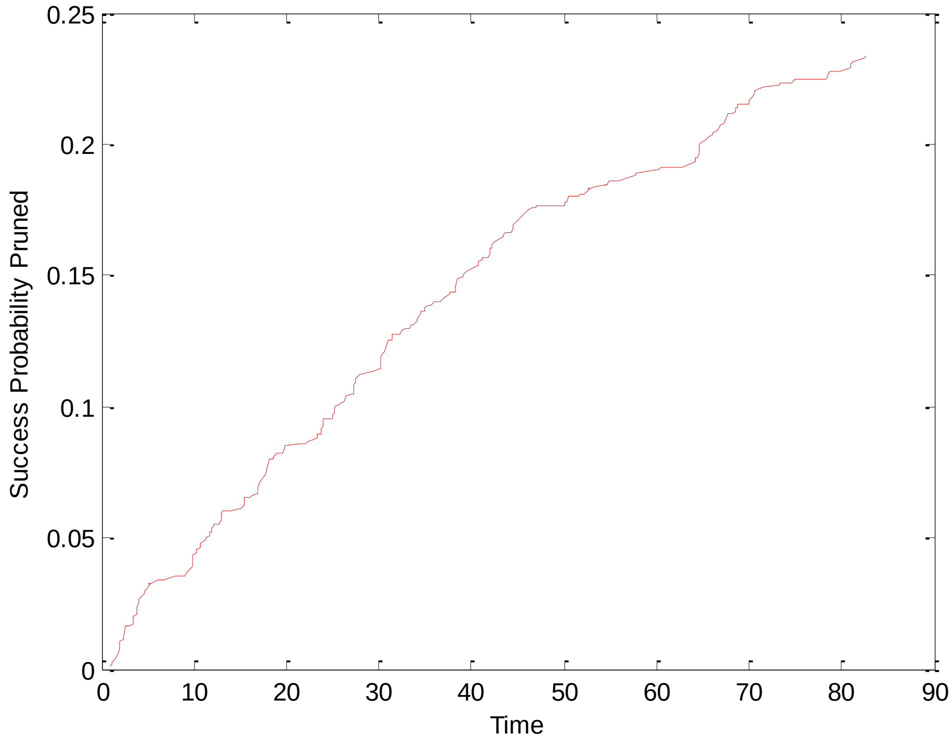

Fig. 4 Success Probability Pruned by Modified Method vs. Time

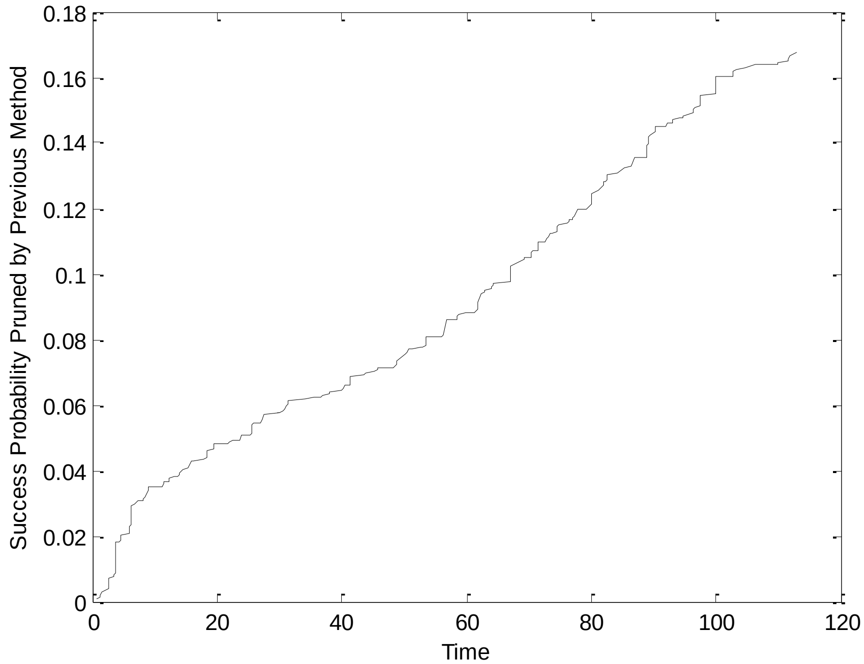

Fig. 5 Success Probability Pruned by Previous Method vs. Time

- Parameter Selection

As discussed above, the selection of parameters is focused primarily on the decision of mutation rate. If we consider the mutation rate as a variable, and iterate it from 0.02 to 1, we can get the success state pruning curve: number of success states pruned versus different mutation rate as in Figure 6. The numbers in boxes indicate the mutation rate of the corresponding curve. If we consider the success probability pruned, it will gives out similar results. We can see that the mutation rate affects the pruning speed significantly, and generally speaking, the number of success states pruned will be larger when the mutation rate is increased. Normally the mutation rate cannot be too large, or it will bring about errors, as it will convert the intelligent search into random search. It can be seen from the figure that selecting a mutation rate for the method proposed to be between [0.51, 0.68] would be the best.

Figure 7 shows a curve similar to that of Figure 6 under the previous method but it is hard to tell the tendency of the change of mutation rate and its impact on the states pruned. Also as discussed in [7], it is hard to decide the selection of the mutation rate. Figure 7 has longer running time because under some mutation rate, the converging time is longer. Figure 7 indicates that the mutation rate does not have controllable impact on the pruning process, and in addition, when the system changes, there will be no clear criterion to select the mutation rate. But in the modified method, as indicated in Figure 6, the mutation rate always has a clear tendency of its impact on the pruning.

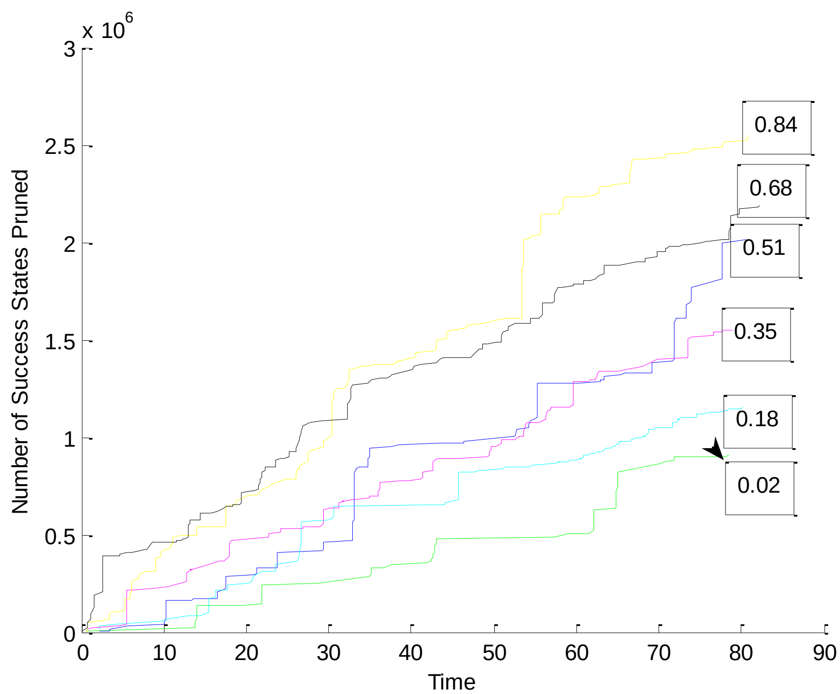

Fig. 6 Success States Pruned by Modified Method vs. Time with Various Mutation Rates

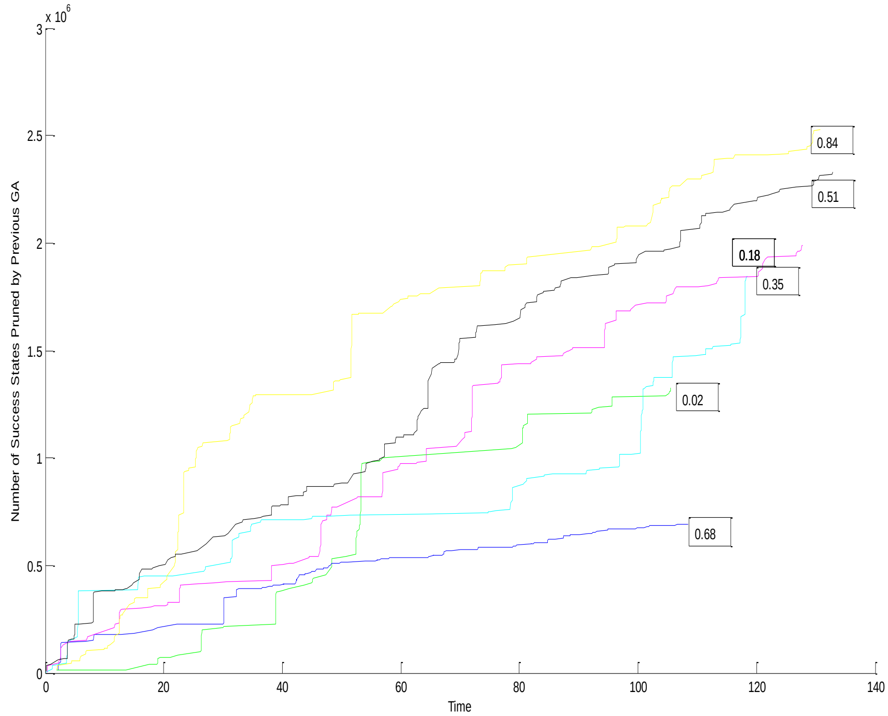

Fig. 7 Success States Pruned by Previous Method vs. Time with Various Mutation Rates

D. Some Remarks

This work is based on the observations of the Genetic Algorithm applied in the reliability analysis, especially in the state space pruning process [12]. It is only an initial step that introduces the usage of number of permutations of states generated, and the change of the sequence of the three operations within GA. It appears that the method is more efficient and the parameter setting is more controllable. However, more detailed exploration is necessary to help this method become generally applied. One constraint is the coding software and the limitation of programming, which will certainly introduce difference when the programmer and the platform differ [13]. In software like GAlib, numbers of packages may be easier to use to help evaluate the method proposed.

About the parameter selection, it is almost impossible to form mathematical model for every parameter [14]. Experimentation is essential, but will be different for different cases. A common conclusion is that the mutation rate can be within a certain range so as to ensure the pruning process, but this advantage over the conventional ones may also vary when the system changes.

The criterion in this thesis is the time of running the program, but sometimes the selection of application of a method in calculating reliability indices depends on many other aspects. This requires further development.

E. Conclusion

This chapter has presented a modified Genetic Algorithm used in the state space pruning process to help faster pruning of success states. The major differences of this method with the previous ones are:

- This method uses permutation numbers when pruning success states
- This method changes the order of mutation and crossover within traditional GA in order to get controllable mutation rate
- This method uses stopping criterion by selecting the increasing slope of the success probability pruned
- By looking into the trends of the effect of mutation rate over the pruning, the selection of GA parameters becomes reasonable

The proposed method not only has better computational efficiency but also provides a feasible selection of parameters.

CHAPTER IV

ADEQUACY ASSESSMENT USING DYNAMIC PROGRAMMING ALGORITHM

Dynamic Programming has been previously used in the optimization problems of power systems, but has been barely applied in the analysis of adequacy assessment of power systems. This chapter investigates the use of dynamic programming as a tool for state space pruning to improve the efficiency of the Monte Carlo Simulation.

A. Dynamic Programming

The concept of basic Dynamic Programming will be described with the help of examples. The following two examples have characteristics similar to the problem of adequacy assessment, and are included here to describe the model of adequacy assessment more clearly.

Example 1: This is a simple capital budgeting problem taken from a tutorial of Dynamic Programming from Carnegie Mellon University [15]:

Problem: A corporation has set aside $5 million to allocate to its three plants for possible expansion. For each plant, there are a few proposals on how it intends to have the money spent, each proposal including the cost of the expansion (c) and the total revenue expected (r).

The proposals generated are tabulated below and numbered from 1 to 4, as shown in Table 1.

Table 1 Investment Possibilities [15] (Example 1)

Proposal	Plant 1.		Plant 2.		Plant3	
	c_1	r_1	c_2	r_2	c_3	r_3
1	0	0	0	0	0	0
2	1	5	2	8	1	4
3	2	6	3	9	—	—
4	—	—	4	12	—	—

Each plant will be allowed to execute only one of its proposals and the goal of analysis to maximize the firm's revenue. It is assumed that the unspent money will be lost.

Solution: Intuitively speaking, the easiest way to analysis this problem is to calculate all allocation possibilities and judge for the best, i.e., the one with the largest revenue. In this case, there are 3 proposals of plant 1, 4 of plant 2, and 2 of plant 3. There are altogether 3 * 4 * 2 = 24 ways of allocating the money.

It should be noted that that not all of these protocols are feasible. For example, if proposal 3 is chosen for plant 1, proposal 3 is chosen for plant 2, and proposal 2 is for plant 3, then the total cost of the three plants will be $6 million which is over the budget. Also some proposals are not suitable although feasible, like the ones with less than $5 million allocation.

Therefore, we can see that solving this problem just by enumerating could have many disadvantages:

- When the size of the problem is grows, the number of combinations will increase significantly, which will make the enumeration much less efficient if not impractical;
- The enumeration cannot distinguish the protocols that are infeasible, which will then add unnecessary classification process;
- There is no storage of any precedent combination which may provide guidance over subsequent selections of combinations

Since the return function is not showing to have any obvious relationship with the allocation, it cannot be modeled as a linear optimization problem.

Dynamic Programming can in this case be used as follows:

- Decompose the problem into stages

Each stage is a step in the allocation process. For this example, the steps are easily identified. The 3 stages represent the money allocated to the 3 plants separately. Stage 1 represents the money allocated to plant 1, stage 2 the money to plant 2, and stage 3 the money to plant 3. Although no order of allocating the funds has been specified, we can arbitrarily place a sequence on the stages, saying that we will first allocate to plant 1, then plant 2, then plant 3.

- Analyze the states in each stage

Each stage is divided into states. A state represents an option in this stage. In this case the states for stages 1, 2, and 3 are

States in stage 1: the amount of money spent on plant 1, labeled as x1:

{0, 1, 2, 3, 4, 5}

States in stage 2: the amount of money spent on plants 1 and 2, labeled as x2: {0, 1, 2, 3, 4, 5}

States in stage 3: the amount of money spent on plants 1, 2, and 3, represented by x3: {5}

A revenue is associated with each state. As long as we know the money spent in the first two stages, we are able to decide how much to spend on stage 3, without the requirement to know in detail how the money was spent in the previous two stages.

Let's make some illustrative description of how to figure out the revenues associated with each state.

In stage 1, there are 6 possible allocation protocols from allocating 0 to 5. Table 2 gives the revenue associated with the allocation x1.

Table 2 Stage 1 Computations [15] (Example 1)

If the available capital x_1 is	Then the optimal proposal is	And the revenue for stage 1 is
0	1	0
1	2	5
2	3	6
3	3	6
4	3	6
5	3	6

In stage 2, we want to find the best solution for both plants 1 and 2. When x2 is designated, we can search from all the plant 2 proposals and allocate the given amount of funds to plant 2, and calculate the remainder which will be the allocation of x1.

For instance, suppose we want to determine the best allocation for state x2 = 4. In stage 2 we can do one of the following proposals:

Proposal 1: stage 2 with revenue 0, and proposal 3 for stage 1. Total revenue: 6.

Proposal 2: stage 2 with revenue 8, and proposal 3 for stage 1. Total revenue: 14.

Proposal 3: stage 2 with revenue 9, and proposal 2 for stage 1. Total revenue: 14.

Proposal 4: stage 2 with revenue 0, and proposal 1 for stage 1. Total revenue: 12.

With the consideration of optimization of revenue, proposals 2 and 3 of plant 2 give the maximum. In either case, the revenue for being in state x2 = 4 is 14. The rest of Table 3 can be filled out similarly.

Table 3 Stage 2 Computations [15] (Example 1)

If the available capital x_2 is	Then the optimal proposal is	And the revenue for stages 1 and 2 is
0	1	0
1	1	5
2	2	8
3	2	13
4	2 or 3	14
5	4	17

In stage 3, the only value we are interested in is x3 = 5. We will again go through all the proposals for this stage, determine the amount of money remaining and use Table 3 to decide the value for the previous stages.

Proposal 1: state 3 has revenue 0, x2=5. Total revenue: 17.

Proposal 2: state 3 has revenue 4, x2=4. Total revenue: 18.

Therefore, the optimal solution is:

Select proposal 2 at plant 3, proposal 2 or 3 at plant 2, and proposal 3 or 2 at plant 1. This gives a total revenue of 18.

We can see that stage 2 calculations are based on stage 1, and stage 3 only on stage 2. Indeed, given you are at a state, all future decisions are made independent of how you got to the state. This is the basic idea of Dynamic Programming. We are able to make the recursion either from forward or from backward.

Figure 8 describes the forward and backward recursion of this problem.

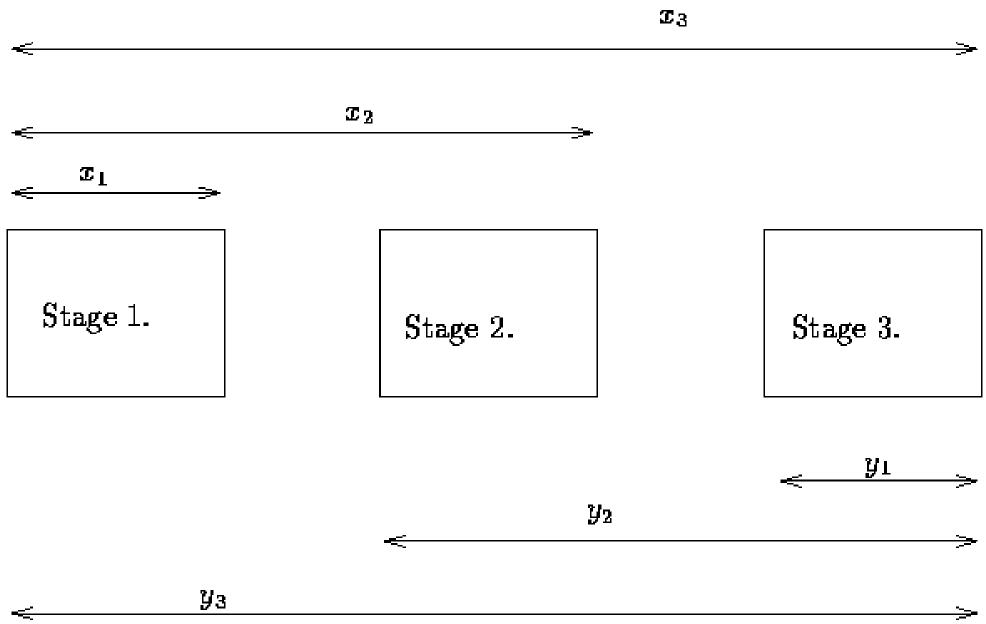

Fig. 8 Forward vs. Backward Recursion [15]

When making the calculations, two kinds of recursion will come up with the same answer.

In this particular case, the ordering of the stages made no difference. In other cases, like in the power generating case, there may be computational advantages of choosing backward over forward recursion, due to the ordering of generators. In general, the backward recursion has been found to be more effective in most applications.

Example 2: This is the typical example used for illustrating Dynamic Programming. The model in this example is very similar to the model we will be describing later in this chapter for the adequacy assessment.

Problem: This is the shortest path problem where we wish to get from A to J in the road network shown in Figure 9 using the shortest path. The numbers on the arcs represent distances.

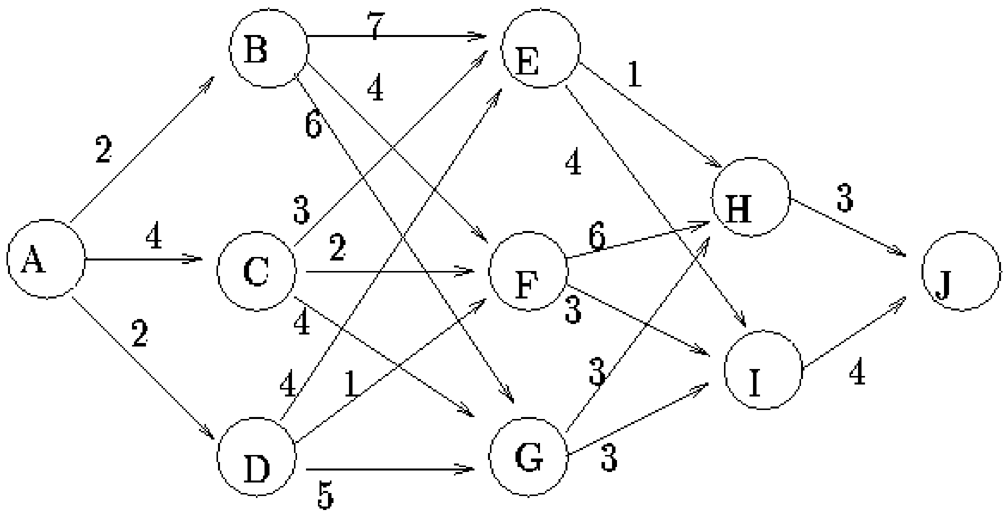

Fig. 9 Road Network [15] (Example 2)

Solution: After examining the steps, we can tell that the paths can be divided into 5 stages. This is the essential part of solving this problem, which means that in the calculation of possible paths, there are certain stages that we can follow.

Then we will clarify the states in each stage.

Stage 1: node A;

Stage 2: nodes B, C and D;

Stage 3: node E, F and G;

Stage 4: node H and I;

Stage 5: node J.

Here, the states in each stage correspond just to the node names. For example, stage 2 has the states B, C and D.

Formulas can be obtained by modeling the increased length of coming to each stage while aiming at the minimum of total path length. Here since we are making illustrative descriptions, we will just have the analysis.

Here are the backward recursions:

- Stage 4: In stage 4, there is only one choice.

The only path is to simply go to destination J.

So we get:

Going from H to J, the distance h=3

Going from I to J, the distance i=4.

- Stage 3: In stage 3, there are more choices.

From F we can either go to H or I. The cost of going to H is 6. The following cost from H to J is 3, which is the formulation in stage 4. This comes with the total of 9.

The cost of going to I is 3. The following cost from I to J is 4, which is the formulation in stage 4. This comes with the total of 7. Therefore, if we are at F, the best thing to do is to go to I. The total minimum cost is 7.

Similarly, we can analysis the states E and G in stage 3. The overall results are:

From state E, the best way is to go to H, which has the total cost of stage 3 and 4 to be 4;

From state G, the best way is to go to H, which has the total cost to be 6;

From state F, the best way is to go to I, which has the total cost to be 7.

With these data, the shortest paths from each of states of stage 3 to stage 5 are definite. We are able to store all these shortest paths as the function of the states, which means, when you come to a certain state in stage 3, it will bring along the unique shortest path to stage 5 afterwards.

Similarly, the analysis of the first two steps is done by backward recursion. The results are:

- Stage 2:

At state B, decision is to go to E or F with minimum cost from B to stage 5 to be 11;

At state C, decision is to go to E with minimum cost from C to stage 5 to be 7;

At state D, decision is to go to E or F with minimum cost from D to stage 5 to be 8;

- Stage 1:

At state A, decision is to go to C or D with minimum cost from A to stage 5 to be 11;

Therefore, the overall shortest path is found.

Here we are only making illustrative descriptions of how the Dynamic Prgramming will work with the idea of backward recursion, without detailed optimization formulas.

We can conclude from the examples that the most typical characteristic of Dynamic Programming is the division of problem into stages. As shown in the second example, the storage of the shortest path from each state to the last stage is the essential part for making the decision in the stage before it. Simply speaking, the algorithm has the storage that makes the decision of going from one state to the final stage only related to the choices of one step, which is from this state to the states in the next stage, without considering the following procedure. This will introduce the idea of sequence and calculating from the largest capacity of total generation output as below.

B. The Simplification of Adequacy Assessment Model

As described in previous chapters, the generation side consists of different generators with individual capacities. But generally speaking, generators can be categorized into groups by identical capacities and failure rates, i.e., the generators with the same capacity and failure rate will be in the same group. In single area power systems, there can be a few generation groups.

Based upon the permutation method described in last chapter, one group of generators can have many generation outputs, where "outputs" refer to the total generation capacity levels of one group, with each output representing several generating states with permutation calculation.

Here is an example similar to the one in the last chapter. A generation group with 5 generators each with capacity of 200MW will have 6 outputs, which are just the 6 generation capacity levels that this group can have: 0MW, 200MW, 400MW, 600MW, 800MW, 1000MW.

Corresponding to each generation level, there are several states of the five generators. For instance, the generation level of 400MW means two of the generators being "up" while three of the generators being "down", which has

$N_{400MW} = \binom{2}{5} = \binom{3}{5} = 10$ states. These 10 states have the same effect over the power system, which means that we can treat them equally without distinguishing which specific generator is up or not in one specific state. Therefore, when one of the permutation states is proved to be success state, all the permutation states are success states.

Therefore, each generation group can be abstracted as a stage, with (n+1) states when there are n generators in this group. The i th state represents the generation capacity level of (i-1) generator being up, i.e., the generation output being (i-1) * Cap. (Cap means the capacity of each generator)

The load in the adequacy assessment here is the peak load of the system, which is a constant in the evaluation.

Based upon this modeling, the adequacy assessment model becomes the comparison of outputs of all "stages" with the load.

C. Dynamic Programming for Adequacy Assessment

From the simplified model, we can formulate the Dynamic Programming problem as the consideration of multiple stages and the sum of their outputs, followed by the comparison with the load.

In state space pruning, as described in the previous chapters, the target of using Dynamic Programming is to truncate the state space by pruning success states to help

Monte Carlo Simulation sample a pruned state space with higher density of failure states. Therefore, like using the Modified Genetic Algorithm in last chapter to sample success states from the state space, the Dynamic Programming method is also aiming at generating success states.

Compared with the example of the shortest path problem, the objective here with Dynamic Programming is to find the longest (largest in capacity) path from each node, or in other word, from each state at each stage. The Dynamic Programming can be intuitively understood like finding the longest path, which will definitely meet the requirement, i.e., larger than the load; and then, DP goes to search for the second longest path, and compare it with the load, and so on. Steps in this method can be summarized as:

1. Sequence the stages by individual generator capacity in each stage, from the largest to the smallest. For example, if generators are grouped with the capacity of 200MW, 300MW, 400MW separately, than the sequence will be from the group with every 400MW, with every 300MW, to the group with every 200MW;
2. Identify the states in each stage based on the number of generators in that group. For example, if stage 1 is the group with 5 generators that all have the capacity of 400MW, then the states in stage 1 will be 0,1,2,3,4,5, which have 6 states representing the output capacity level of 0 MW, 400 MW, 800 MW,1200 MW, 1600 MW, 2000 MW;

3. Identify the peak load, which is a constant number in this case. For example, the peak load = 3000MW;
4. Delete some states by observing the difference between the total load of all groups of generators, and the peak load. This is a simplification process, and will be discussed in detail in the next part. This step is not always necessary, but will reduce calculations.
5. Follow the Dynamic Programming; look for the longest path, which is the largest in the sum of the states in each stage. Intuitively, the longest path will certainly be the path with all full output states in each stage. And this path will certainly be longer than the peak load, or otherwise, there are no success states in this system. Store the success state.
6. Search from the last stage, which is the group with the least individual generation capacity. Try the second largest state in this stage and see if it comes up with a system success state, i.e., the path is longer than the peak load. Store the success state.
7. Try all the states in the last stage, with all states in other stages unchanged. Judge all the paths with comparison with the peak load to see if they are success states. Store the success states.
8. After judging all the states above, start over from step 5 again to step 7 with these changes: find the path with the state in the second stage counted from the last to be the second largest state in that stage; re-do all the calculations by changing the state in the last stage. Store the success states.

9. Start over step 8 again with the change that it starts with the second largest state of the third stage counted from the last. Search for success states by getting the path with first changing the state in the last stage while the state of the second last stage stays at the largest, and then changing the state in the last stage while the stage of the second last stays at the second largest, etc. Store the success states.
10. Do similar change in the states of the next stage counted from the last, and re-do steps above. Store the success states.

There needs to be some illustration of step 4. This step is a simplification process. The method is simple:

- First calculate the difference between the total generation capacity and the peak load. For example, if there are 3 groups, each with 4 generators. The individual capacities of the generators in these 3 groups are 500MW, 400MW, 300MW. The peak load is 4200 MW. The total generation capacity is 4 * 500 + 4 * 400 + 4 * 300 = 4800 MW. Therefore the difference is 4800 – 4200 = 600.
- Then find the states in each stage that clearly have a difference more than 600MW than the largest state in that stage. For example, in stage 1, the states are 2000, 1500, 1000, 500, 0. Among these states, the states 1000, 500, and 0 will be the states that can be deleted. The reason is simple. The calculation of the difference of total generation capacity and the

peak load represents the largest capacity that the generating units can be short from full capacity. In the example, the 600MW difference means that there can be at the most 600 MW of generation that can not be put into use. Therefore, if a state shows that it has already had more than 600 MW in its stage, then this state can be deleted. This is because when searching for the success states of the system, this state in the stage will never give success states.

In the example above, originally the first stage contains states: 2000, 1500, 1000, 500, 0; after this step, the first stage only contains states: 2000, 1500. The second stage originally contains states: 1600, 1200, 800, 400, 0; after this step, the second stage will only contain: 1600, 1200. The third stage originally has: 1200, 900, 600, 300, 0; after this step, the third stage contains: 1200, 900, 600.

Process of simplification will be described more in detail in the example and case study in next part of this thesis.

D. An Example Illustrating the Process of the Algorithm

Let’s take an example to show how the steps described above work.

The system has 20 generators, with different generating capacities. They are grouped with capacities shown in Table 4 below.

Table 4 Grouped Generator Capacities

Number of Generators in the Group	Individual Generator Capacity (MW)
5	400
5	250
4	300
3	100
3	200

The peak load is 4500MW.

So the problem can be defined as follows. There are 5 groups of generators, each with an individual generation capacity. The target of the Dynamic Programming process is to find the success states.

We will illustrate the algorithm step by step.

Step 1: Sequence the groups. The result is shown in Table 5.

Table 5 Sequenced Grouped Generator Capacities

Number of Generators in the Group	Individual Generator Capacity (MW)
5	400
4	300
5	250
3	200
3	100

The capacity level and the state below are all in MW.

Step 2: Find the states. The results are shown in Table 6.

Table 6 States in Each Stage

Stage	1	2	3	4	5
States	2000	1200	1250	600	300
	1600	900	1000	400	200
	1200	600	750	200	100
	800	300	500	0	0
	400	0	250		
	0		0		

Step 3: The peak load = 4500

Step 4: The difference between full generation output and the peak load is: 2000 + 1200 + 1250 + 600 + 300 – 4500 = 850. This means that the entire system can have at the most 850 MW that is not generated, counting from the full capacity of all generators. Therefore we delete the 4th, 5th, 6th states in stage 1, the 4th, 5th states in stage 2, and the 5th, 6th states in stage 3. The results are shown in Table 7.

Table 7 States after Simplification

Stage	1	2	3	4	5
States	2000	1200	1250	600	300
	1600	900	1000	400	200
	1200	600	750	200	100
			500	0	0

Step 5: Longest path: the result is shown in Table 8.

Table 8 Longest Path

Stage	1	2	3	4	5	Total
States	2000	1200	1250	600	300	5350

Step 6: Search system states by changing the state in last stage. The results are shown in Table 9.

Table 9 Path after Changing the States in Last Stage (1st)

Stage	1	2	3	4	5	Total
States	2000	1200	1250	600	200	5250

It is a success system state.

Step 7: Search system states by changing the states in last stage. The results are shown in Table 10.

Table 10 Path after Changing the States in Last Stage (2nd)

Stage	1	2	3	4	5	Total
State	2000	1200	1250	600	100	5150
State	2000	1200	1250	600	0	5050

These are two success states.

Step 8: Change the state in stage 4. The results are shown in Table 11.

Table 11 Path after Changing the States in 2^{nd} Last Stage (1st)

Stage	1	2	3	4	5	Total
State	2000	1200	1250	400	300	5150
State	2000	1200	1250	400	200	5050
State	2000	1200	1250	400	100	4950
State	2000	1200	1250	400	0	4850

These are 4 success states.

Changing the state in stage 4 to 200 in the similar way, the states are shown in Table 12.

Table 12 Path after Changing the States in 2^{nd} Last Stage (2nd)

Stage	1	2	3	4	5	Total
State	2000	1200	1250	200	300	4950
State	2000	1200	1250	200	200	4850
State	2000	1200	1250	200	100	4750
State	2000	1200	1250	200	0	4650

Changing the state in stage 4 to 0 in the similar way, the states are shown in Table 13.

Table 13 Path after Changing the States in 2nd Last Stage (3rd)

Stage	1	2	3	4	5	Total
State	2000	1200	1250	0	300	4750
State	2000	1200	1250	0	200	4650
State	2000	1200	1250	0	100	4550
State	2000	1200	1250	0	0	4450

The last state is a failure state, which has the total capacity of 4450 with stage 1 to 5 to be: 2000, 1200, 1250, 0, 0.All the other states generated are success states.

Step 9: Change the state in stage 3:

Change the state in stage 3 to 1000, and iterate the search by changing the states in last 2 stages, the results are shown in Table 14.

Table 14 Path after Changing the States in 3rd Last Stage

Stage	1	2	3	4	5	Total
State	2000	1200	1000	600	300	5100
State	2000	1200	1000	600	200	5000
State	2000	1200	1000	600	100	4900
State	2000	1200	1000	600	0	4800
State	2000	1200	1000	400	300	4900
State	2000	1200	1000	400	200	4800
State	2000	1200	1000	400	100	4700
State	2000	1200	1000	400	0	4600
State	2000	1200	1000	200	300	4700
State	2000	1200	1000	200	200	4600
State	2000	1200	1000	200	100	4500
State	2000	1200	1000	200	0	4400
State	2000	1200	1000	0	300	4500
State	2000	1200	1000	0	200	4400
State	2000	1200	1000	0	100	4300
State	2000	1200	1000	0	0	4200

The states that have total path length (total generation capacity) less than 4500 are failure states. In the result above, there are 4 failure states and 12 success states.

Similar searches can be conducted with the criterion described above to find success states. After finding the success states, we will need to identify the permutation states and their probabilities, with exactly the same method as in the last chapter.

The calculation of reliability indices is exactly the same as in last chapter.

E. Case Studies and Results

We tested the method of the pruning of the state space with comparison of using Monte Carlo Simulation on the same computer and with the same coding person and coding technique. The software in this study was written in Matlab.

The methods were tested using two different test systems: IEEE-RTS79 (RTS79) [9] and the Modified Reliability Test System (MRTS) [10], [11]. Figure 10 and Figure 11 show the probability of success states pruned from the state space by the proposed Dynamic Programming algorithm, and the Monte Carlo Simulation. We can tell that the proposed Dynamic Programming method is more efficient than the Monte Carlo Simulation.

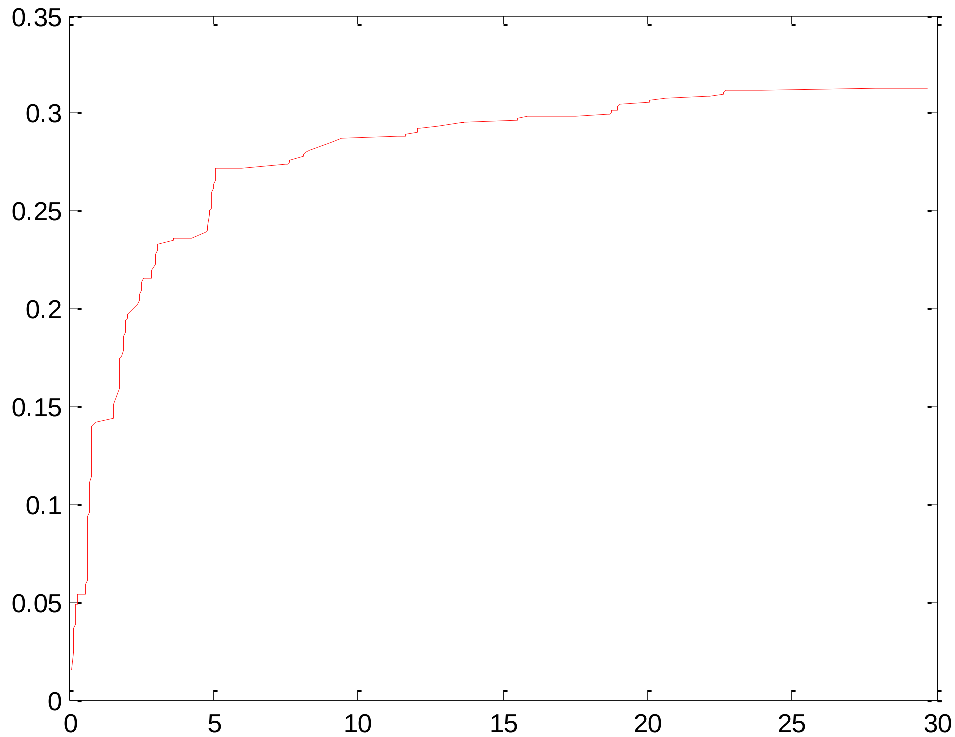

Fig. 10 Success Probability Pruned by Dynamic Programming vs. Time

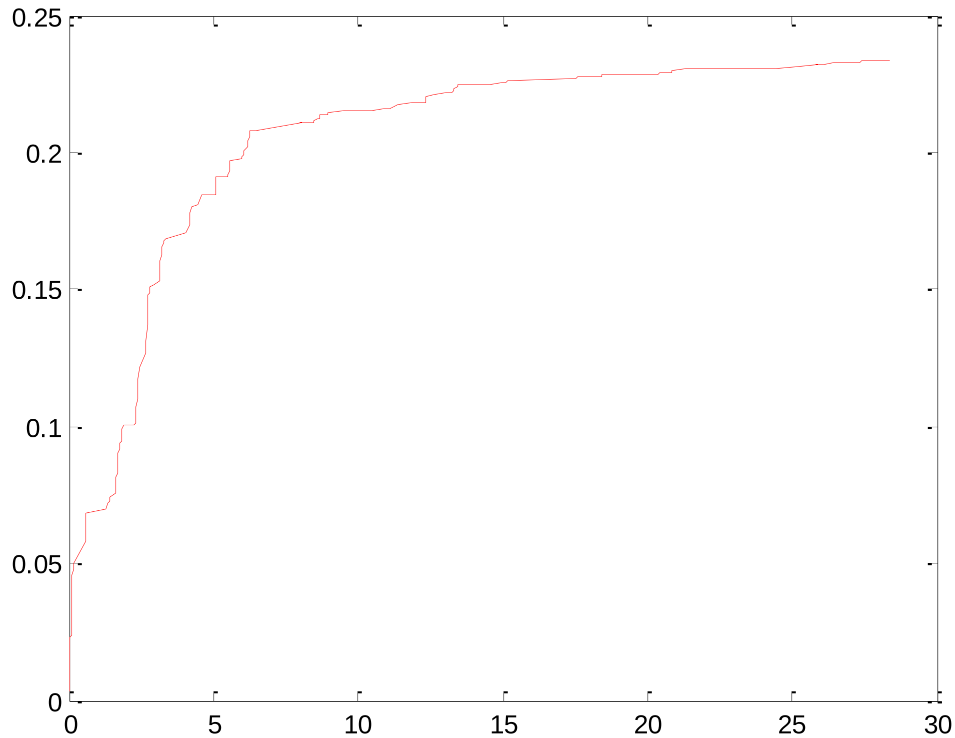

Fig. 11 Success Probability Pruned by Monte Carlo Simulation vs. Time

F. Some Remarks and Conclusion

The method of using Dynamic Programming above has been shown to be effective. This method is based on Dynamic Programming, but has certain difference with traditional Dynamic Programming:

- Since we are aiming at finding not only the longest path but also as other qualified paths, which are success states, the sequential searching method is put into use. We are not finding from one state in certain stage its longest path to the last stage, but searching from the longest to all the combination of the following path.

- Permutation is calculated which improves the efficiency to a large extent.
- The simplification process can be quite useful when the peak load is close to the total generating capacity. Although we figure out every deletion of the states by enumerating, the program can also easily select the states that will need to be deleted.
- This method is based on the sequential searching of Dynamic Programming, in which the sequencing process becomes essential.
- This method is an explorative approach to organize and truncate the state space. Further research will be needed to systemize this method.
- The stopping criterion can be the same as in the Modified Genetic Algorithm method. Simply speaking, it can just be a time t when there has been no success pruned within the time period (t-Δt, t).

Conclusions: The state space pruning technique using Dynamic Programming is efficient and effective in the searching and pruning of success states. This method is useful mostly due to:

- The sequential groups and permutations are used to simplify the identification of system states.
- Simplification techniques are used to reduce the states in stages.
- When the generation capacity varies greatly, for example, when the largest generation capacity is 1000 MW while the smallest is 100MW, this method can be especially useful for the reason that it is searching

from the changing of the smallest to the changing of the largest. Most success states lie in the beginning of the process when changing the smallest.

- Stopping criterion can be easily set, such as the duration time when no success state is pruned.

CHAPTER IV

CONCLUSIONS

The state space pruning has been shown to have faster convergence in calculation of reliability indices. Numerical and intelligent methods have been proposed in order to truncate the state space so as to have Monte Carlo Simulation sample a pruned state space with higher density of failure states.

In this thesis, two methods are proposed to prune the state space. After the state space pruning, Monte Carlo Simulation can be applied to calculate the reliability indices. Therefore, in combination with Monte Carlo Simulation, the two methods become complete adequacy assessment approaches.

In the first method, Modified Genetic Algorithm has been proposed for state space pruning. This method uses permutation numbers when pruning success states, and it changes the order of mutation and crossover within traditional GA in order to get controllable mutation rate. Using a case study, the proposed method is shown not only to have better computational efficiency but better in the selection of parameters.

A state space pruning technique using Dynamic Programming has also been proposed as the second method. In this method, the sequential groups and permutations are used to simplify the identification of system states. Other simplification techniques are also used to reduce the states in stages. With case study, it is shown that this method is efficient and effective in the searching and pruning of success states.

REFERENCES

[1] C. Singh and J. Mitra, "Composite system reliability evaluation using state space pruning," *IEEE Trans. Power Systems*, vol. 12, no. 1, pp. 471–479, Feb. 1997.

[2] L. Wang and C. Singh, "Population-based intelligent search in reliability evaluation of generation systems with wind power penetration," *IEEE Trans. Power Systems*, vol. 23, no. 3, pp. 1336–1345, Aug. 2008.

[3] N. Samaan and C. Singh, "Genetic algorithms approach for the evaluation of composite generation-transmission systems reliability worth," in *IEEE PES Transmission and Distribution Conference and Exposition*, September 2003, vol.1, pp. 113–119.

[4] N. Samaan and C. Singh, "Genetic algorithm approach for the assessment of composite power system reliability considering multi-state components," in *8th International Conference on Probabilistic Methods Applied to Power Systems*, Iowa, September 2004, pp 64–69.

[5] L. Wang and C. Singh, "Multi-deme parallel genetic algorithm in reliability analysis of composite power systems," in *IEEE Bucharest Power Tech Conference*, Romania, June 2009, pp 1–6.

[6] L. Wang and C. Singh, "Reliability evaluation of composite power systems using parallel genetic algorithms: some conceptual and simulation studies," in *IEEE PES Power Systems Conference and Exposition*, March 2009, pp. 1–7.

[7] N. Samaan and C. Singh, "Adequacy assessment of power system generation using a modified simple genetic algorithm," *IEEE Trans. Power Systems*, vol. 17, no. 4, pp. 974–981, Nov. 2002.

[8] R. Green, L. Wang, and C. Singh, "State space pruning for power system reliability evaluation using genetic algorithm," in *IEEE Power and Energy Society General Meeting*, Minnesota, July 2010, pp. 1–6.

[9] I. C. Report, "IEEE reliability test system," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-98, no. 6, pp. 2047–2054, Nov. 1979.

[10] "Final report on research project 2473-10," Technical report, EPRI, 1987.

[11] M. Pereira and N. Balu, "Composite generation/transmission reliability evaluation," *Proceedings of the IEEE*, vol. 80, no. 4, pp. 470–491, Apr. 1992.

[12] R. Billinton and R. Allen, "Power system reliability in perspective," *Electronics and Power*, vol. 30, no. 3, pp. 231-236, Mar. 1984.

[13] R. Allen and R. Billinton, *Reliability evaluation of power systems*, Plenum Press, 1984.

[14] C. Singh, *Course Note of ECEN 643: Power System Reliability*, Texas A&M University, 2009.

[15] Michael A. Trick, *A tutorial on Dynamic Programming*, Carnegie Mellon University, 1997.

VITA

Dongbo Zhao received his Bachelor of Science degree in electrical engineering from Tsinghua University, Beijing, China in 2008. He entered Department of Electrical and Computer Engineering at Texas A&M University in September, 2008. His research interests include power system reliability analysis, intelligent calculation techniques, and renewable energy and its related reliability analysis. He would like to pursue his professional career in power system field.

Dongbo Zhao can be reached at Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX 77843-3128. His email is zhaodongbo2008@gmail.com.