12-2009

A Genetic Approach To Voltage Collapse Mitigation

Dwarakesh Nallan chakravartula
Clemson University, dntula@clemson.edu

A GENETIC APPROACH
TO
VOLTAGE COLLAPSE MITIGATION

A Thesis

Presented to

the Graduate School of

Clemson University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Electrical Engineering

---

by

Dwarakesh Nallan Chakravartula

December 2009

Accepted by:

Dr. Adly. A. Girgis, Committee Chair

Dr. Elham B. Makram

Dr. Richard E. Groff

ABSTRACT

The objective of this thesis is to provide an efficient and accurate corrective solution to a system that is on verge of voltage collapse. This thesis describes, in detail, the development of an optimization scheme that aims to alleviate power system instability and voltage collapse condition based on the principles of an evolutionary approach called *Genetic Algorithm*. The state of a system is determined using a voltage stability identifier termed *Collapse Proximity Index (CPI)* and the critical loading condition is identified. Applying principles of Genetic Algorithm, the critical system is brought back to a stable operating region. The sequential procedure and application of this scheme is primarily discussed in this thesis.

The thesis is structured to include theoretical discussion of the Collapse Proximity Index, development of a Genetic Algorithm – based solution to the voltage collapse problem and its simulated implementation on a test system, along with result analysis and suggestions for future development. Conclusions are drawn based on the efficiency of the application in maintaining system stability.

DEDICATION

Dedicated to my parents,

my brother Raghav,

all the graduate Power students in ECE Department,

and all those who have helped me through my two year graduate study at Clemson University

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF PLOTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

An important component of transmission planning is to examine the adequacy of a power system. Planning involves optimized methodologies in analyzing any change in network topology and system loading. With ever-increasing power demand over the recent years, transmission networks have to be utilized more efficiently than ever before. Consequently, it is important for planners to identify the loading capability limits of the system.

The determination of adequate capacity limits remains constrained as long as the behavior of system components remains uncertain. When a heavy loading condition stresses a power system, there is a substantially different response to system parameters than compared to that of a non-stressed system. System voltage instability increases as the transmission system becomes more broadly loaded [1].

Under such critical loading, a relatively small disturbance or additional load perturbation may cause a complete system upset and lead to a system collapse.

In addition to the local system breakdown, large areas of the interconnected system may also be affected by the small perturbation.

Numerous instances from the past highlight the importance of transmission planning for pre-determined overloaded conditions. The August 14th 2003 Blackout in US-Canada region [2] can be attributed to overloading of the system as a secondary consequence to a generator outage (Eastlake 5) and a fault on one of the lines (345 kV) due to excessive strain and sag, with the assumption that system voltage remained stable. Inadequate planning and lack of periodical contingency analysis resulted in a complete blackout for a number of regions in North-eastern United States and regions of Canada. The voltage levels dropped severely resulting in a cascading effect and formation of numerous power system islands.

The blackout in Italy on September 28th 2003 [2] also portrays the closeness of a system to voltage collapse under overloaded conditions. The result of the overload was a very low system voltage in Northern Italy and consequential tripping of a number of generating stations. Despite countermeasures being implemented, such as load shedding, the loss of generation made it impossible for the system to restore back to its stable state, leading to a complete blackout.

The 2003 September 23rd blackout in Sweden and Denmark [2] is another clear example of system voltage collapse due to overloading in some parts of the system. An unscheduled drop in generation from a nuclear station resulted in system overloading, ultimately leading to voltage collapse and a system blackout.

1.2 Overview of Voltage Collapse

1.2.1 Voltage Stability

IEEE Power System Relaying Committee report [3] defines Voltage Stability as the ability of a system to maintain voltage such that when load admittance is increased, load power will increase thereby making power and voltage controllable.

Two most important conditions for a stable system voltage profile, as prescribed by Hill et al [4] are that:

- System voltages must lie within an acceptable band
- The power system must be in a voltage regular operating point

CIGRE [5] defines voltage stability in a general way similar to other dynamic stability problems. According to its definition, a power system at a given operating state and subject to a given disturbance is voltage stable if voltages near loads approach pre-equilibrium values.

Voltage stability is classified based on a system's dynamic behavior. A common classification of voltage stability is as follows:

- **Small Disturbance voltage stability**: A power system at a given operating state is small disturbance stable if following any small disturbance, its voltages are identical to or close to their pre-disturbance equilibrium values.
- **Large disturbance voltage stability**: A power system at a given operating state and subject to a given large disturbance is large disturbance voltage stable if the voltages approach post-disturbance equilibrium values.

1.2.2 Voltage Instability

According to CIGRE's definition, Voltage instability is the absence of voltage stability, resulting in progressive voltage decrease or increase. A power system at a given operating state and subject to a given disturbance is said to be unstable if voltages near loads are far away from their pre-disturbance equilibrium values.

A system enters a state of voltage instability when a disturbance, increase in load, or system changes causes voltage to drop quickly or drift downward, and operators and automatic system control fail to halt the decay. The voltage decay may take just a few seconds or minutes. If the decay continues uninterrupted, voltage collapse will occur.

1.2.3 Voltage Collapse

Voltage collapse is a phenomenon observed when a heavily loaded system is disturbed by a load perturbation or a small disturbance. Voltage Collapse, according to the IEEE definition [3] is the process by which voltage instability leads to loss of voltage in a significant part of the system. Under a voltage collapse situation, the post-disturbance voltage values do not reach their pre-disturbance equilibrium conditions.

The major symptoms of voltage collapse as specified by the IEEE Power System Relaying Committee are:

- Low voltage profiles
- Heavy reactive power flows
- Inadequate reactive support
- Heavily loaded system

1.2.4 Relation between Power and Voltage (P-V & Q-V curves)

A two bus equivalent system of a three phase power system [1] is considered as shown in Fig 1.1.

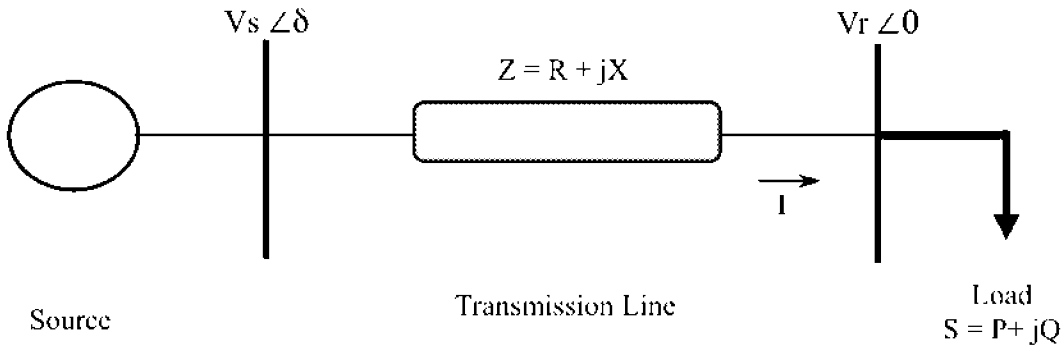


Fig 1.1

Two – Bus Equivalent System

The power transfer equation from Bus 1 to bus 2 is obtained as:

$$P_{SR} = |V_S|^2 G - |V_S||V_R|G * \cos(\delta) + |V_S||V_R|B * \sin(\delta) \qquad (1.1)$$

$$Q_{SR} = |V_S|^2 B - |V_S||V_R|B * \cos(\delta) - |V_S||V_R|G * \sin(\delta) \qquad (1.2)$$

The complex Power expression can be written as:

$$S_D = P_D (1 + j \tan \delta) \qquad (1.3)$$

With respect to the load at the receiving-end Bus, power demand is defined by the power transferred between the two buses:

$$\left.\begin{array}{ll} P_D = P_{SR} = |V_S||V_R|B * \sin(\delta) & \text{`(1.4)} \\ Q_D = Q_{SR} = |V_S|^2 B - |V_S||V_R|B * \cos(\delta) & (1.5) \end{array}\right\} \text{Neglecting transmission line conductance}$$

Assuming the constant ß and equating the expressions for P_D and Q_D, an expression relating Voltage and the Power demand at the load bus is as follows:

$$(|V_R|^2)^2 + [(2 P_D ß / B) - |V_S|^2] |V_R|^2 + (P_D^2/B^2) [1 + ß^2] = 0 \qquad (1.6)$$

Solving the quadratic equation with respect to the receiving-end voltage:

$$|V_R|^2 = (|V_S|^2)/ 2 – (P_D ß / B) \pm [|V_S|^4/4 – (P_D / B)((P_D / B) + ß |V_S|^2)] \wedge 0.5 \qquad (1.7)$$

Assuming that the sending end voltage is $|V_S| = 1.0$ pu, the expression for the receiving-end voltage with respect to the Power demand is obtained as:

<u>Relating V_R and P_D:</u>

$$|V_R|^2 = (1 – ß * P_D \pm [1 – P_D (P_D + 2 ß)] \wedge 0.5) / B \qquad (1.8)$$

Similarly, an expression relating the reactive power demand and receiving-end voltage can be obtained by replacing the active power demand from the above expression in terms of Q_D and ß.

<u>Relating V_R and Q_D:</u>

$$|V_R|^2 = (1 – Q_D \pm [1 – (Q_D / ß) ((Q_D / ß) + 2 ß)] \wedge 0.5) / B \qquad (1.9)$$

The above-defined expression is useful in obtaining the P-V and Q-V plots.

A set of P-V curves for a stable system is obtained as shown:

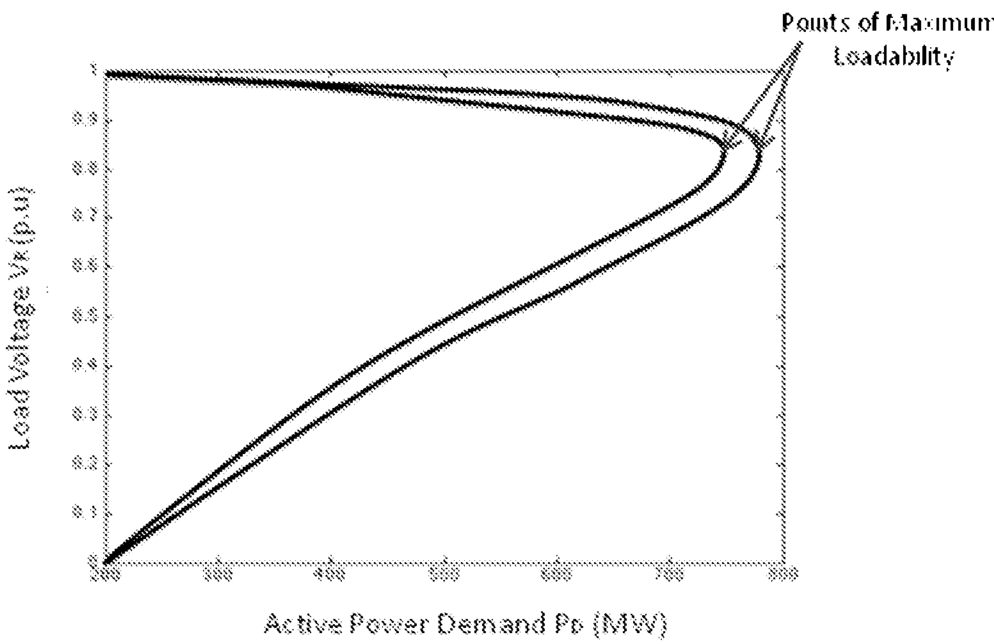


Fig 1.2

P-V curves for a stable system

The Q-V curve for a stable system is obtained as shown:

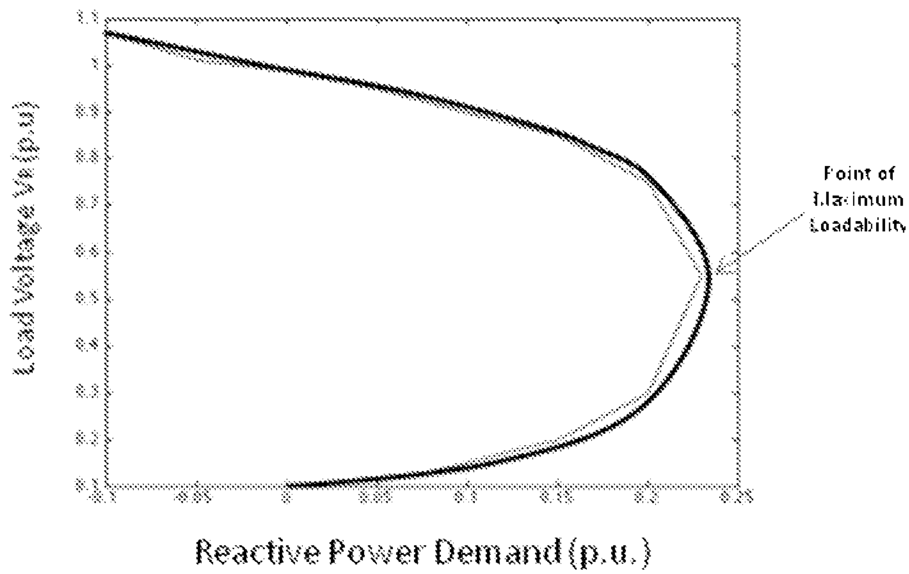


Fig 1.3

Q-V curves for a stable system

As shown in figures 1.2 and 1.3, a system would be in its stable operating region as long as the pu Voltage with respect to the pu power transferred is above the critical point (Point of Maximum Loadability – PML).

When the voltage at the selected bus goes below a pre-defined criterion, then the transfer at which this occurs is the Low Voltage transfer limit for that bus. Ignoring the low voltage and continuing to increase transfer would eventually bring the curve to a point where the system collapses called the **Critical Point**.

When a system reaches a condition where the voltage profile extends beyond its critical point, system voltage collapse is imminent.

Maximum Loadability refers to the condition where the load at the receiving end-bus reaches its maximum permissible value beyond which the system would observe a voltage collapse situation. The point of maximum loading capacity can be determined by either solving the 1^{st} derivative of the P-V equation or by evaluating the load flow Jacobian Matrix singularity. A detailed theoretical explanation to on this topic can be found in Chapter 2, Section 2.2 on Index formation.

1.2.5 Load Model

In analyzing voltage instability, it is important to consider the network under various voltage profiles. Voltage Stability depends on the level of current drawn by the loads. The level of current drawn by the loads can depend on the voltage seen by the loads.

Therefore, voltage instability analysis requires a model of how the load responds to voltage variations. The system should supply the loads at all times. Consequently, the system must manage all load-voltage dependencies without restraints. All types of electrical loads behave differently. One possible way to describe the static voltage-power relation is to use the relations [6]:

$$P = P_o [\, p_1 (V/V_o)^2 + p_2 (V/V_o) + p_3 \,] \tag{1.10}$$

$$Q = Q_o [\, q_1 (V/V_o)^2 + q_2 (V/V_o) + q_3 \,] \tag{1.11}$$

Where the subscript ‘o’ indicates the initial operating condition.

The load model is composed of three components:

- Constant impedance component (p_1, q_1)
- Constant current component (p_2, q_2)
- Constant power component (p_3, q_3)

Since voltage instability causes voltage decline, alleviation of voltage instability results if demand reduces with voltage decline.

1.3 Review of Available Techniques in Voltage Collapse Identification

In order to understand the degree of voltage instability in the system, a number of static analysis based voltage stability indicators or indices have been developed over the years.

1.3.1 Voltage Stability Indices based on Power flow analysis

A number of voltage stability studies were developed based on power flow analysis, concentrating on Jacobian matrix singularity and load flow feasibility. The paper presented by B.Gao et al [7] concentrated on voltage stability assessment of a system using modal analysis. The measure of voltage instability was identified using Eigen values and associated eigenvectors of a reduced Jacobian matrix.

A problem identified with the proposed theory was that steady state equations used in determining the system state became singular at the bifurcation point resulting in numerical instabilities in the proposed method when close to the collapse point.

Y.Tamura et al [8] and P.Kessel et al [9] described indices based on load flow feasibility studies, concentrating on identifying the relationships between voltage instability and multiple load flow solutions in a power system. The authors of the latter defined an indicator L with a range between 0 and 1 using the basic load flow information. Though the proposed indicator had advantages with respect to simplicity and expressiveness, the authors limited its use to situation where thermal limits were violated or protective devices were tripped off rather than focusing on incremental loading of a system and planning analysis of voltage stability.

C.A.Canizares et al [10] and P.A.Lof et al [11] proposed their theories based on the use of Jacobian matrix singular values. One of the proposed methods included a reduction of load flow Jacobian with respect to the critical bus of the system. As computation-time consumption in these earlier proposals was a major factor, a fast method to calculate minimum singular value and corresponding singular vectors that utilized the sparsity of the power flow Jacobian matrix was also proposed. The major disadvantage associated with these models was the difficulty in accurately identifying system voltage stability margins.

1.3.2 Voltage Stability Indices based on local measurements

Taking into consideration, the disadvantages among models based on Power-flow feasibility, direct measurement voltage stability indices involving bus voltage and sensitivity factors based on dynamic analysis have been widely implemented in protective devices.

Local measurements and non-linear aspects of voltage stability have been given greater importance over the recent years [12] - [16]. Khoi Vu et al [15] proposed a data-processing method to estimate the proximity to voltage collapse using local measurements of bus voltage and load current. The proposed theory calculated the strength of the transmission system relative to the bus. The collapse point was identified based on the local load and its closeness to the strength calculated. The primary disadvantage in implementing the proposed theory was the complexity in initiating remedial control actions based on local measurements, as the optimal pickup values of these measurement-based indices were difficult to determine.

With emerging wide spread real time data transmission through SCADA and synchrophasors, dynamic stability indices have also been proposed [17], [18], [19]. The platform of these dynamic stability indices is based on the possibility of building Wide Area Monitoring Systems (WAMS) and high-speed communication networks.

WAMS help in taking snapshots of the power system variables where the synchronized phasor measurement units are installed and help in developing wide area stability assessment and protection applications for early detection and prevention of potential voltage instabilities. The application of these dynamic indices is concentrated only for real-time operations and would not be helpful in the planning phase of a power system.

1.4 Control Actions for Voltage Collapse Mitigation

With the accurate identification of system stability margin for a given loading condition, it is essential in implementing voltage stability-enhancing techniques to tackle instability issues.

Over the years, several studies within the industry have determined a number of practical control actions that can be initiated in order to mitigate voltage instability. On a broader scale, these include

- Addition of reactive power compensators
- Changing transformer tap settings
- System reinforcement
- Coordinating relays and control
- Load shedding

IEEE Power System Relaying Committee has put forward a detailed description of the above mentioned control actions in its report on Voltage Collapse Mitigation [3]. With reference to Voltage Collapse mitigation techniques, T.V. Cutsem [20] has highlighted the most practical and widely used control actions being implemented in the industry. A tabulated form of various control actions depending on the mode of application can be observed as shown in Table 1.1

Table 1.1

Control Actions based on modes of application for Voltage Collapse Mitigation

System Reinforcement	New Transmission lines Construction of generating stations with low pf Series compensation Shunt compensation SVCs
Devices and Control	Line-drop compensation in AVRs Control of Generator step-up transformer taps Automatic Shunt compensation switching
Operational Planning	Voltage Security Assessment
Real-time operation	Voltage profile monitoring Generator Rescheduling
System Protection schemes	HVDC modulation Emergency Load Shedding

1.5 Genetic Algorithms and Voltage Collapse Mitigation

In order to determine the best-fit control action for a given voltage collapse condition, it would be practical to use one of the many optimization techniques used in present day technology. Genetic Algorithm (GA) is a preferred choice among different optimization techniques suited for this application.

Genetic Algorithm is stochastic search technique used in computing to determine the exact or approximate solution to an optimization problem. GAs are evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection and crossover.

This thesis concentrates on the applications of Genetic Algorithms in determining the appropriate combination of control actions to be taken in bringing a critically loaded system back to its stable operating condition. A modified voltage instability indicator termed as Collapse-Proximity Index (CPI) is developed and used in applying the evolutionary algorithm into voltage collapse mitigation study.

The thesis can be summarized as follows:

Chapter 2 concentrates on the theory, development and implementation of the Collapse-Proximity Index, with its application on two test systems.

Chapter 3 sheds light on the basic principles of Genetic Algorithms along with the various processes involved in the evolution of an optimized solution.

The chapter also covers, in depth, the application of Genetic Algorithms in Voltage Collapse mitigation, including all the steps in utilizing CPI to obtain the best-fit control actions for voltage stability mitigation.

Chapter 4 describes the implementation of the developed algorithm on a real-time test system with simulations to portray the applications and advantages of the proposed algorithm. Final remarks and future recommendations are included in the Conclusion Section of Chapter 4.

CHAPTER 2

VOLTAGE COLLAPSE PROXIMITY INDEX

In order to determine the voltage stability of a system, an index has to be formulated which would facilitate the computation of accurate system stability identification. Taking into consideration the different advantages and disadvantages of the earlier proposed indices, a modified index termed as *Collapse Proximity Index (CPI)* is developed. The modified index is used as an identifier in determining the appropriate control actions for a system using the principles of Genetic Algorithms.

2.1 Equivalent system

Equivalent systems [24] are useful in two circumstances:

- To allow larger areas of major interconnected systems to be represented in studies
- To achieve improved computational speed in simulations by removing buses and branches that influence system behavior, but are not of specific interest.

2.1.1 Background on Equivalent system

Every large interconnected power system can be categorized into three different categories: Internal system, Boundary system and External system as shown in Fig 2.1. The external system is that part of the system that needs to be equivalenced. The internal system is that part of the system that is under consideration.

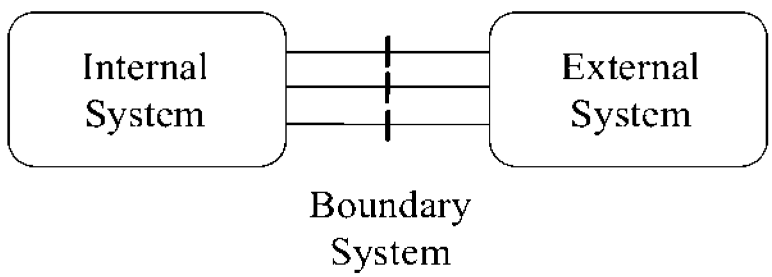


Fig. 2.1

Categorization of a large interconnected power system

The internal system can be classified into three sub-categories:

- Load Bus Sub-system: Includes buses that are connected to loads
- Source Bus sub-system: Includes buses that are connected to generators
- Tie Bus sub-system: Includes buses that have neither loads nor generators

2.1.2 Two-Bus Equivalent system representation

A simple two-bus equivalent system model is taken into consideration for formulation of the Voltage Collapse-Proximity index, as shown in Fig 1.1. The two-bus system consists of a source with voltage magnitude Vs. A load with active power P, reactive power Q is supplied from the source through a transmission line of line impedance Z (Z = R + jX). The load bus voltage magnitude is taken as Vr.

2.2 Index Development

The load demand on the receiving-end bus can be expressed as a relation between Load bus voltage V_R and the current I flowing through the transmission line from source to load [15], [17].

$$S = P + jQ = V_R\, I^* \tag{2.1}$$

The expression for current flowing through the line can be expressed in terms of the voltage drop and the transmission line impedance Z as:

$$I = (V_S\angle\delta - V_R\angle 0) / (R + jX) \tag{2.2}$$

2.2.1 Power Voltage relation

Substituting for the current in 2.1:

$$P + jQ = V_R\, [(V_S\angle\delta - V_R\angle 0) / (R + jX)]^*$$

$$= V_R\, [(V_S \cos(\delta) + j\, V_S \sin(\delta) - V_R)]^* / (R - jX)$$

$$= V_R [(V_S \cos(\delta) - V_R) R + V_S \sin(\delta) X] / (R^2 + X^2)$$

$$+ j V_R [(V_S \cos(\delta) - V_R) X - V_S \sin(\delta) R] / (R^2 + X^2) \quad (2.3)$$

Separating real and imaginary parts for 2.3:

$$P = V_R [(V_S \cos(\delta) - V_R) R + V_S \sin(\delta) X] / (R^2 + X^2) \quad (2.4)$$

$$Q = V_R [(V_S \cos(\delta) - V_R) X - V_S \sin(\delta) R] / (R^2 + X^2) \quad (2.5)$$

To determine the maximum transferrable power from source to the load, it is necessary to obtain a relationship between the sending-end and receiving end voltages, V_S and V_R with respect to the load P, Q.

2.2.2 Expression for V_R

Squaring and adding equations 4 and 5:

$$P^2 + Q^2 = V_R^2 / (R^2 + X^2)2 [[(V_S \cos(\delta) - V_R) R + V_S \sin(\delta) X]^2 + [(V_S \cos(\delta) - V_R) X - V_S \sin(\delta) R]^2]$$

Separating V_R to obtain a quadratic expression:

$$V_R^2 = ((V_S^2 / 2) – (QX + PR) \pm \surd[(V_S^4/4) – (QX + PR) V_S^2 – (PX – QR)^2])^{0.5} \quad (2.6)$$

From the above equation, the term in the square root can be extracted out as:

$$C = [(V_S^4/4) – (QX + PR) V_S^2 – (PX – QR)^2] \quad (2.7)$$

2.2.3 Maximum Transferrable Power

The maximum transferrable power can be obtained from the expression C obtained in (2.7) [17].

For a solution to be obtained for V_R from (2.6), the part of the expression within the square root should be positive.

(i.e.) $C \geq 0$ (2.8)

$$[(V_S^4/4) - (QX + PR)\,V_S^2 - (PX - QR)^2] \geq 0 \quad (2.9)$$

For maximum power to be transferred between the source and the load for the two-bus equivalent system, the value of C should not exist, or:

$C = 0$

$$[(V_S^4/4) - (QX + PR)\,V_S^2 - (PX - QR)^2] = 0 \quad (2.10)$$

Under the condition described in (2.10), the load would be

$S_{MAX} = P_{MAX} + jQ_{MAX}$

This transforms (2.10) as:

$$[(V_S^4/4) - (Q_{MAX}X + P_{MAX}R)\,V_S^2 - (P_{MAX}X - Q_{MAX}R)^2] = 0$$

<u>For Maximum Active power demand:</u>

$$[(V_S^4/4) - (QX + P_{MAX}R)\,V_S^2 - (P_{MAX}X - QR)^2] = 0$$

Solving for V_S^2,

$2V_S^2 = (QX + P_{MAX}R) \pm \sqrt{[(QX + P_{MAX}R)^2 + (P_{MAX}X - QR)^2]}$

Extracting P_{MAX} from the above equation:

$2P_{MAX}X^2 = 2QRX - V_S^2R + V_S \sqrt{[(R^2 + X^2)(V_S^2 - 4QX)]}$

$$\mathbf{P_{MAX} = (QR / X) - (V_S^2R + V_S \sqrt{[(R^2 + X^2)(V_S^2 - 4QX)]}) / 2X^2} \tag{2.11}$$

The above expression derives a relationship between the Maximum Active Power demand with respect to reactive power demand and the source voltage of the two-bus system.

<u>For Maximum Reactive power demand:</u>

$[(V_S^4/4) - (Q_{MAX}X + PR)\, V_S^2 - (PX - Q_{MAX}R)^2] = 0$

Solving for V_S^2,

$2V_S^2 = (Q_{MAX}X + PR) \pm \sqrt{[(Q_{MAX}X + PR)^2 + (PX - Q_{MAX}R)^2]}$

Extracting Q_{MAX} from the above equation:

$2Q_{MAX}R^2 = 2PRX - V_S^2X + V_S \sqrt{[(R^2 + X^2)(V_S^2 - 4PR)]}$

$$\mathbf{Q_{MAX} = (PX / R) - (V_S^2X + V_S \sqrt{[(R^2 + X^2)(V_S^2 - 4PR)]}) / 2R^2}$$

The above expression derives a relationship between the Maximum Reactive Power demand with respect to active power demand and the source voltage of the two-bus system.

2.2.4 Relating P_{MAX} and P through CPI

The maximum transferrable active power demand determines the upper limit of the load flow that would be possible for the given system parameters. If a condition is reached where the active power demand exceeds the maximum transferrable power for the given system loading condition, a system voltage collapse is imminent.

As a measure of accurately identifying the proximity of the system to such a voltage collapse condition, the ***Collapse Proximity Index (CPI)*** is defined.

The Collapse Proximity Index can be expressed as:

$$\mathbf{CPI = P_{MAX} / P} \tag{2.12}$$

2.3 Properties of Collapse Proximity Index

The Index defined above has the following prominent properties:

- The ratio of PMAX to P represents the extent to which the system is stressed. If current loading condition is approximately equal to the Maximum transferrable active power demand:

  $$\text{CPI} \approx 1$$

  $$\text{PMAX} / \text{P} \approx 1 \qquad \Rightarrow \qquad \text{PMAX} \approx \text{P}$$

  The above expression represents the boundary condition for system stability. The system is said have a **Boundary Value (Margin B)** of Voltage Stability.

- If the current system loading condition were to be greater than the maximum permissible active power demand, the Collapse Proximity Index value would reduce below the boundary value of 1.

$$CPI < 1$$

$$PMAX / P < 1 \quad \Rightarrow \quad PMAX < P$$

The system would now move to an unstable state, leading to voltage collapse. In other words, the system would be in the **Voltage Collapse Region (Region C)**.

- On the other hand, if the Index shows a value much greater than 1, it represents the current loading condition of the system to be much less than the permissible maximum active power that can be transferred from the source to the load.

$$CPI >> 1$$

$$PMAX / P >> 1 \quad \Rightarrow \quad PMAX >> P$$

As the power currently being transferred is well within its maximum limit, the power system would be in a stable condition during the existence of the above said condition. In other words, the system would be in the **Stable Operating Region (Region A)** in terms of system voltage stability.

- The different operating regions based on the Collapse Proximity Index can be portrayed as shown in Region of Orientation, Fig 2.2:

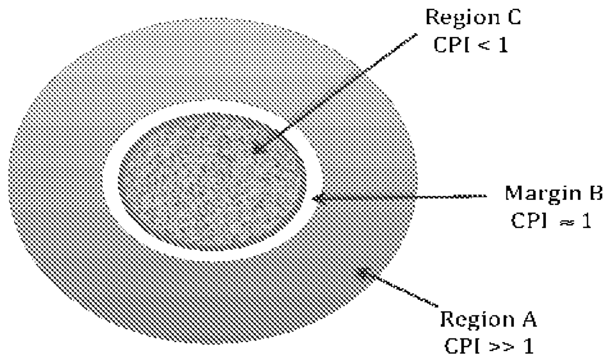


Fig 2.2

Regions of CPI Orientation

2.4 Similarities with other Indices

The Collapse Proximity Index is a modified version of existing indices. As the maximum power demand for a two-bus equivalent system forms the basis for the formulation of the index, it can be compared to a lot of other indices that have been developed on the same platform.

The index formulated in [17] is also based on the maximum power transferred in a two-bus equivalent system.

The index is devised as:

$VSI = \min(P_{margin}/P_{Max}, Q_{margin}/Q_{Max}, S_{margin}/S_{Max})$

where P_{margin}, Q_{margin}, S_{margin} are the load margins calculated as a difference between the maximum permissible loading and the actual loading.

The modified CPI derived in this chapter is a close extension to the above-defined VSI as both the indices are completely based on the maximum transferrable power theory. The modified CPI provides a more accurate and broader variation in its values, ranging between 10 and 1.00 as the system loading reaches its critical loading condition.

The technique proposed in [14], utilizing a Stability Monitoring And Reference Tuning (SMART) Device, concentrates on the use of local measurements to estimate the proximity of the system to voltage collapse. The developed device calculates the strength of the transmission system relative to the bus. The collapse point is identified based on the local load and its closeness to the strength calculated.

The proposed CPI and the SMART Device voltage collapse identifier follow similar patterns in voltage collapse point identification as both indices have a decreasing index trend with increase in system loading. Just as in the case of the CPI, the identifier proposed in [14] portrays a weakened system as the index reaches a value of 1.00. At the voltage collapse point, the value of the Thevenin equivalent impedance (Z_{Thev}) is equal to the magnitude of the apparent impedance (Z_{App}) making the index value equal to 1.00. This represents a condition similar to the one described in Section 2.3 and Fig. 2.2.

2.5 Advantages

The following lists the advantages of implementing CPI for determining system closeness to voltage collapse:

- *Accurate System State projection:*

  As the proposed CPI shows a greater range-variation for the three different system states – Stable, Marginally Stable, Unstable – the accuracy of system collapse prediction is high and accurate. The index provides a closer and precise look at the degree of closeness to the voltage collapse point by determining the threshold and the marginal limits. As previously described, the index accurately portrays the system state to be in critical condition as it moves closer to the 1.00 mark.

- *Simpler Implementation:*

  With complete formulation of the index based on basic principles of power flow and two-bus equivalent systems, it remains simple and easy to understand and implement in different system studies. The fundamental background in determining the closeness of the system to the voltage collapse point is based on the basic principles of identifying the maximum power demand that the system can take in. This simple yet powerful base helps in better understanding of the index and improved applications to it.

- *Easier application of remedial actions*

  As described, the optimal pickup values of the CPI are easier to determine. This helps in improved applications of remedial actions, like the ones described in the latter sections of this thesis.

- *Extensive use in Transmission Planning Studies:*

  The utilization of the two-bus equivalent system reduction in determining the closeness of the system to voltage collapse in itself underlines the importance of its extensive use in Transmission system planning operations. The index can be used not just in determining the system state for a given loading condition, but can also be used in identifying the maximum loading capability for a given bus based on the present loading conditions and implementation of appropriate control actions. This can be a very important tool in load forecasting, as it would help transmission planners in observing the load trend and making certain that the system load incrementation does not reach beyond its maximum capability limit.

- *Possible applications in system dynamics:*

  With the application of synchrophasors and real time monitoring equipments, time-synchronized measurements can be obtained and utilized in determining the current system state and closeness of the system to voltage collapse.

2.6 Index Implementation on Test Systems

The proposed index is applied on two test systems. The degree of closeness to voltage collapse condition is determined by observing the value of the index and its underlying region according to Fig 2.2.

2.6.1 Test systems

Two test systems are considered for implementing the index and observing its effectiveness in determining a voltage collapse situation.

A PSS/E test system consists of 23 buses, including 6 generators. The test system includes of a number of loads, transformers and capacitor banks. The single line representation of the test system is represented as shown in Fig 2.3.

An IEEE 39 Bus New England Test system is also used to check for voltage collapse-identifying situations using the proposed index. This test system consists of 10 generating units along with a number of loads and transformers. The single line diagram for the 39-bus test system is shown in Fig 2.4.

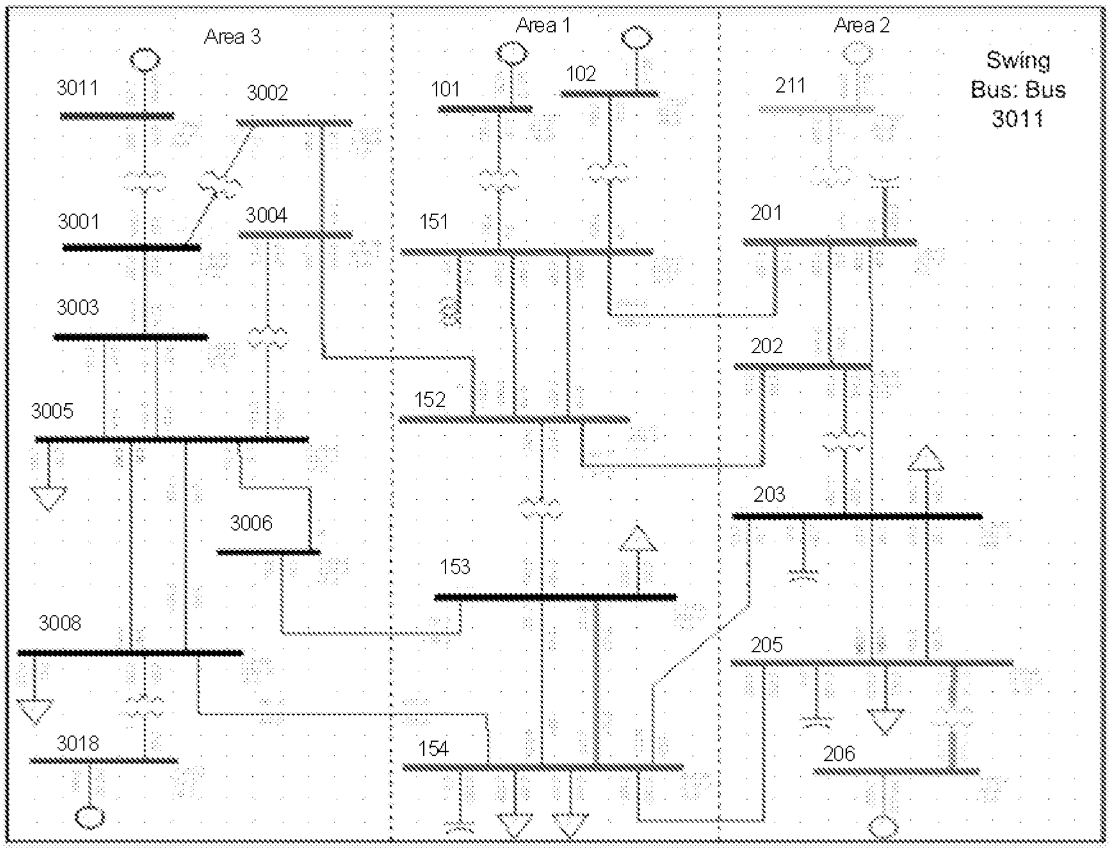


Fig 2.3

Single line representation of

PSS/E ‘SAVNW’ test system

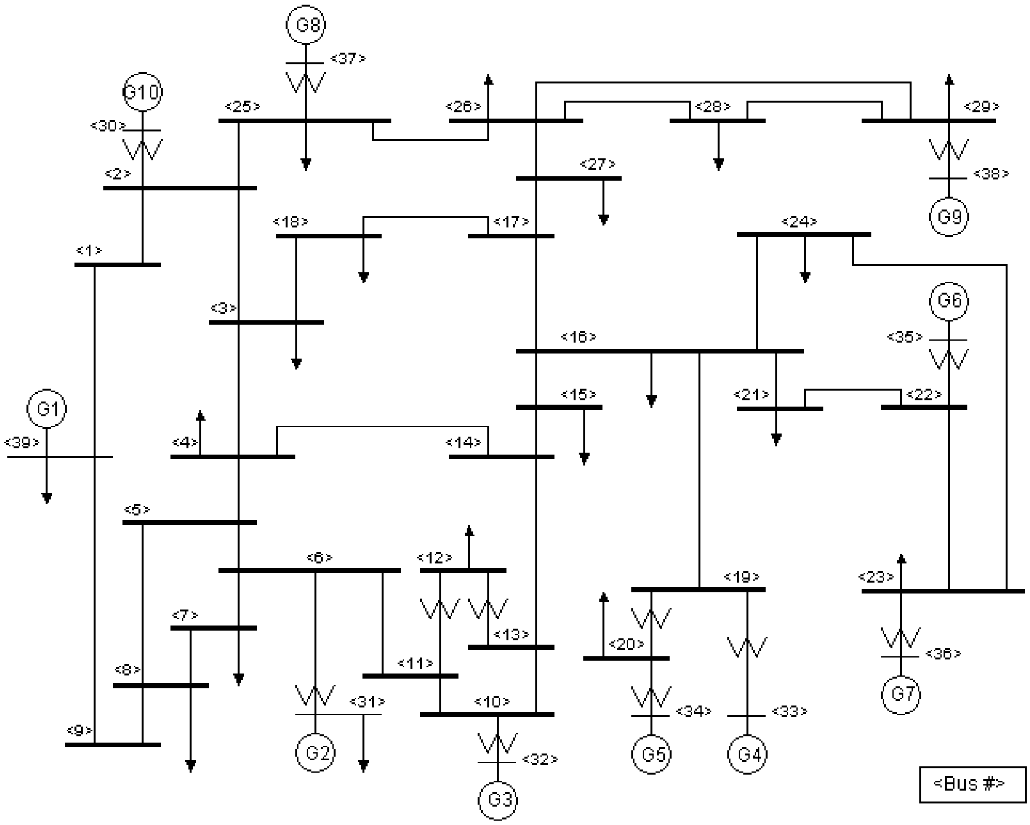


Fig 2.4

Single line representation of

IEEE 39 Bus New England test system

2.6.2 Implementation on PSS/E test system

In order to determine the effectiveness of the index in determining a voltage collapse situation, a load bus from the PSS/E test system is considered. Load incrementation is performed at this bus and the system is observed for voltage instability.

2.6.2.1 Steps in Index evaluation

The following steps are performed to identify the critical loading condition at a specific bus on the IEEE test system using the proposed index:

- *Identification of the test load bus:*

  The load bus to be monitored is identified as Bus 203 with a base-case loading of 300 MW, 150 MVAR, as shown in Fig 2.5.

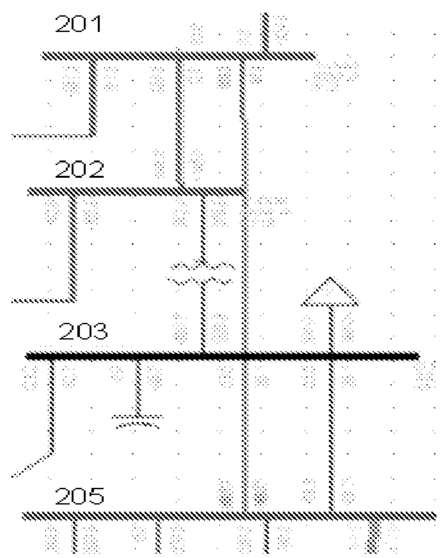


Fig 2.5

Load Bus 203 (PSS/E system) Identification

The reasons for selection of Bus 203 for simulation analysis are

- Lightly loaded bus under normal operating conditions
- Bus voltage well within the upper and lower limits
- Closeness to a central district

- *System reduction to a two-bus equivalent:*

  The system is reduced to a two-bus equivalent behind the load Bus 203; system swing bus being Bus 3011.

- *Calculation of P_{MAX} for different loading conditions*

  The value of P_{MAX} is obtained based on the expression derived in 2.11 of Section 2.2.3.

- *Load increment*

  Load is incremented by approximately 60 MVA.

- *Calculation of CPI*

  The Collapse Proximity Index for each loading condition is calculated based on the expression described in (2.12) of Section 2.2.3.

- *Observing the region of orientation*

  The Collapse Proximity Index is observed in the region of orientation, as described in Figure 2.2. The region of orientation portrays a distinct variation in CPI as loading at Bus 203 approaches 220% of its base case loading condition.

- *Determining the critical loading point*

  The critical loading condition, based on the orientation of the indices is determined. It can be observed that the critical loading condition is reached when the load is 220% of the base case loading at Bus 203 for the PSS/E test system.

2.6.2.2 Tabulation of observed CPI and Inferences

The following table (Table 2.1) represents the value of the Collapse Proximity Index for the corresponding loading condition at Bus 203 of the PSS/E SAVNW test case. Table 2.2 represents the load bus voltage with respect to different loading condition.

The results obtained from Table 2.1 clearly indicate the existence of *critical loading condition beyond 220%* and *voltage collapse situation beyond 240%.* It can be determined from the Collapse Proximity Index data that as the index value approaches 1.00, the system moves from a stable state to a region of critical/marginal stability.

Table 2.1

Collapse Proximity Index Table for Bus 203 (PSS/E system)

P Load (MW)	% of Base case Loading	CPI
300	100	3.0421
360	120	2.441
420	140	2.0083
480	160	1.6804
540	180	1.422
600	200	1.2118
660	***220***	***1.1018***
720	**240**	**0.8855**

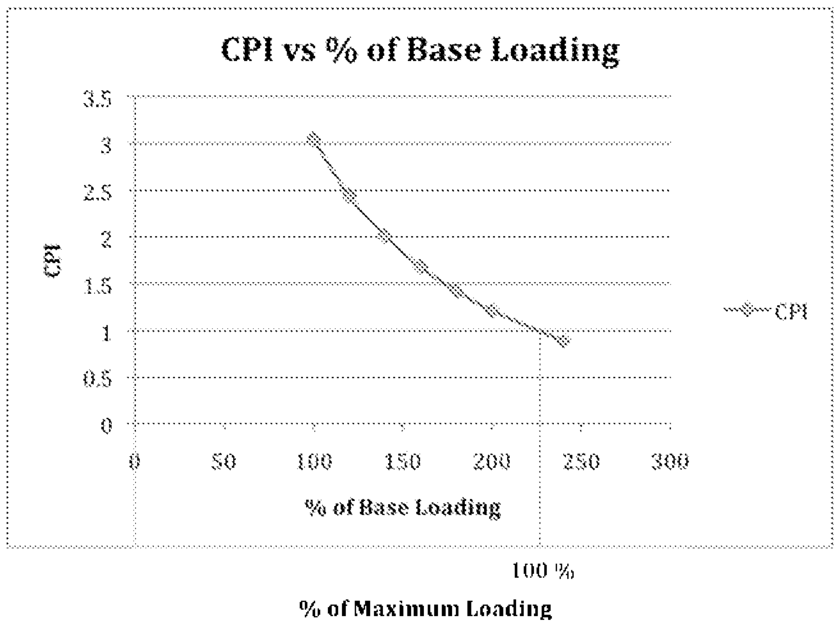
From Table 2.2, it is clear that beyond the loading of 660MW at Bus 153, the bus voltage V_R shows a sharp drop to 0.14pu form a value of 0.92p.u. This clearly indicates the usability of the index in determining the state of the system as it moves to a region of instability resulting in voltage collapse.

Table 2.2

Voltage and Active Power demand comparison for Bus 203

P Load (MW)	V_R
300	0.9665
360	0.9556
420	0.9438
480	0.931
540	0.9169
600	**0.9013**
720	**0.0254**

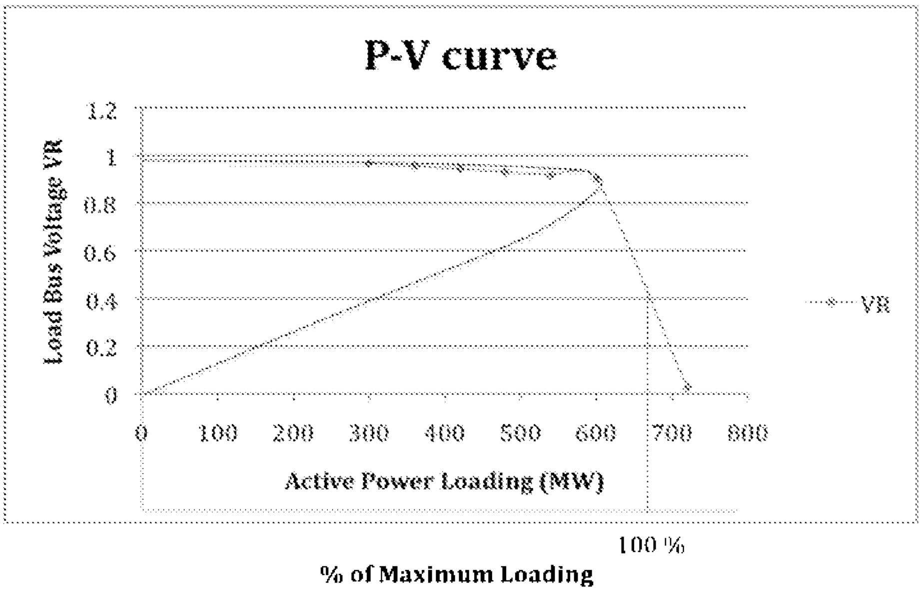
The following plot shows the trend observed for the Collapse Proximity Index with respect to the load increment. It can be observed the region between 200% and 220% of base case loading refers to the critical loading region.



Plot 2.1

CPI vs % of Base loading for Bus 203

The corresponding P-V curve for the above loading is obtained as shown in Plot 4.2.



Plot 2.2

P-V curve for Bus 203

2.6.3 Implementation on IEEE New England test system

The effectiveness of the index in determining a voltage collapse situation is ascertained by incrementing the load at a bus for the IEEE 39 Bus New England Test System.

2.6.3.1 Steps in Index evaluation

The following steps are performed to identify the critical loading condition at a specific bus on the IEEE test system using the proposed index:

- *Identification of the test load bus:*

  The load bus to be monitored is identified as Bus 23 with a base-case loading of 247.5 MW, 84.6 MVAR, as shown in Fig 2.5.

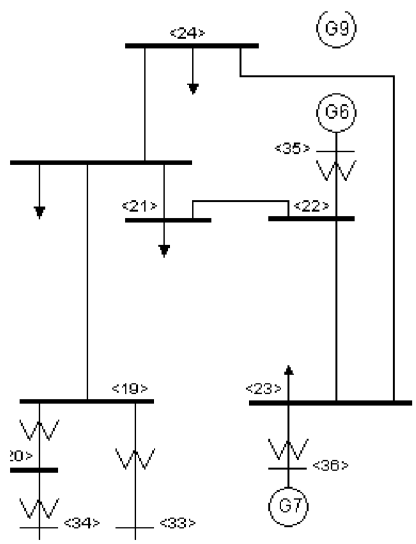


Fig 2.6

Load Bus 23 (IEEE system) Identification

  The reasons for selection of Bus 23 for simulation analysis are as described in Section 2.6.2.1.

- *System reduction to a two-bus equivalent:*

  The system is reduced to a two-bus equivalent behind the load Bus 23; system swing bus being Bus 39.

- *Calculation of P_{MAX} for different loading conditions*

  The value of P_{MAX} is obtained based on the expression derived in 2.11 of Section 2.2.3.

- *Load increment*

  Load is incremented by 20 % of base-case loading.

- *Calculation of CPI*

  The Collapse Proximity Index for each loading condition is calculated based on the expression described in 2.12, Section 2.2.3.

- *Observing the region of orientation*

  The Collapse Proximity Index is observed in the region of orientation, as described in Figure 2.2. The region of orientation portrays a distinct variation in CPI as loading at Bus 23 approaches 220% of its base case loading condition.

- *Determining the critical loading point*

  The critical loading condition, based on the orientation of the indices is determined. It can be observed that the critical loading condition is reached when the load is 230% of the base case loading at Bus 23 for the 39-bus test system.

2.6.2.2 Tabulation of observed CPI and Inferences

The following table (Table 2.3) represents the value of the Collapse Proximity Index for the corresponding loading condition at Bus 23 of the IEEE 39-bus test case.

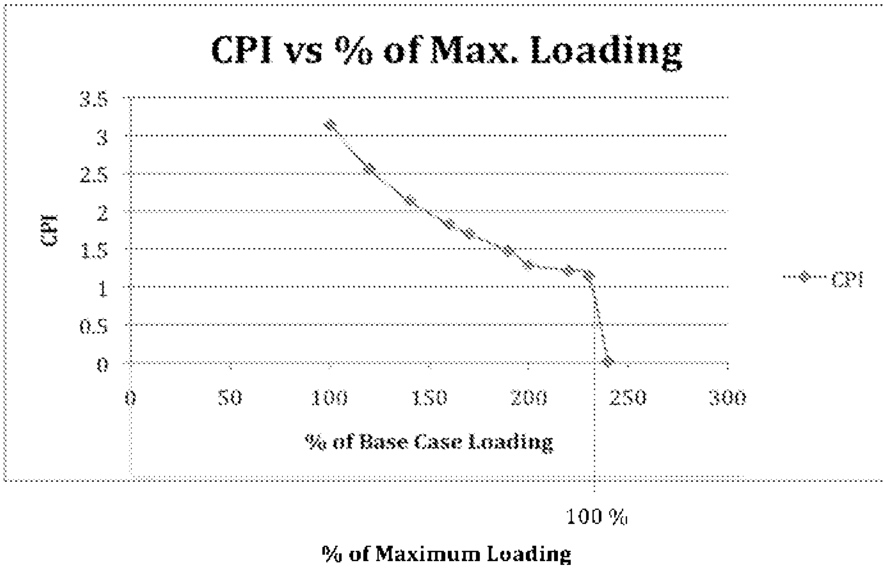
The results obtained from Table 2.3 clearly indicate the existence of *critical loading condition beyond 230%* and *voltage collapse situation beyond 240%*. It can be determined from the Collapse Proximity Index data that as the index value approaches 1.00, the system moves from a stable state to a region of critical/marginal stability.

Table 2.3

Collapse Proximity Index Table for Bus 23 (IEEE system)

P Load (MW)	% of Base-Case Loading	CPI
247.5	100	3.1423
297	120	2.5578
346.5	140	2.139
396	160	1.8236
420.75	170	1.6933
470.25	190	1.4729
519.75	200	1.2933
544.5	220	1.2153
569.25	***230***	***1.1438***
59400	**240**	**0.0108**

The following plot (Plot 2.3) shows the trend observed for the Collapse Proximity Index with respect to the load increment. It can be observed the region between 230% and 240% of base case loading refers to the critical loading region.



Plot 2.3

CPI vs % Base Loading for Bus 23

CHAPTER 3

GENETIC ALGORITHM – OVERVIEW AND APPLICATION

3.1 Introduction

Genetic algorithms (GAs) are stochastic global search and optimization methods that mimic the metaphor of natural biological evolution. GAs operates on a population of potential solutions applying the principle of survival of the fittest to produce successively better approximations to a solution. At each generation of a GA, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and reproducing them using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals from which they were created, just as in natural adaptation.

Genetic Algorithms are based on the Theory of Evolution - 'survival of the fittest' [21]. In nature, individuals that are more fit are more likely to breed and pass their characteristics on to future generations. Genetic Algorithms retain their identity from nature, modeling complex and difficult-to-solve problems as genetic objects. These algorithms are inspired by Darwin's theory about evolution. Solution to a problem solved by genetic algorithms is evolved.

In Genetic Algorithms, an initial population is built using individuals representing random solutions. Each subsequent population that is built uses the previous population as a base - taking the more fit individuals to breed better solutions.

Genetic Algorithms breed solutions. An initial population is built using individuals representing random solutions. Each subsequent population that is built (each subsequent generation) uses the previous population as a base - taking the more fit individuals to breed better solutions.

These algorithms have been used in the past to help solve very complex problems not easily solved using standard, problem-specific methods.

3.1.1 Terminology used in Genetic Algorithms

Certain expressions used in the GA have been derived from their biological roots [21]. Some of the most common terms used in this thesis with reference to Genetic Algorithms are as follows:

- Individual - Any possible solution to the applied problem
- Population - Group of all individuals randomly generated for optimization
- Search Space - All possible solutions to the problem
- Chromosome – Blueprint (coding) of an individual

- Trait - Possible aspect of an individual
- Locus - The position of a gene on the chromosome
- Fitness – Degree of acceptability of an individual
- Allele - Possible settings for a trait
- Selection, Crossover, Mutation – Operators used on chromosome population based on Fitness

A clear representation of the above used vocabulary is shown as follows in Fig 3.1:

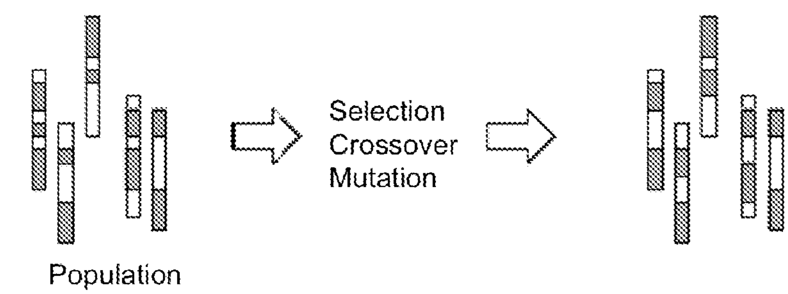


Fig 3.1

Genetic Algorithm Terminology

3.1.2 Characteristics of Genetic Algorithms

Genetic algorithms (GAs) differ from more traditional optimization techniques with the following characteristics:

- GAs use objective function information to guide the search, not derivative or other auxiliary information, making it a global search optimization technique.
- Coding of parameters is utilized in Genetic Algorithms to calculate the objective function in guiding the search rather than utilizing the parameters themselves. This helps in maintaining the originality of the given parameters while manipulating with their coded counterparts for a best-fit solution.
- GAs search through many points in solution space at one time, not a single point. The broadened search space facilitates in greater accuracy and greater optimization.
- GAs use probabilistic rules not deterministic rules in moving from one set of solution to the next.

3.2 Steps Involved in Genetic Algorithms

Genetic Algorithms require a definite set of steps to be followed in order to optimize a given problem. A pictorial representation providing an overview of the steps can be observed in Fig 3.2 as follows:

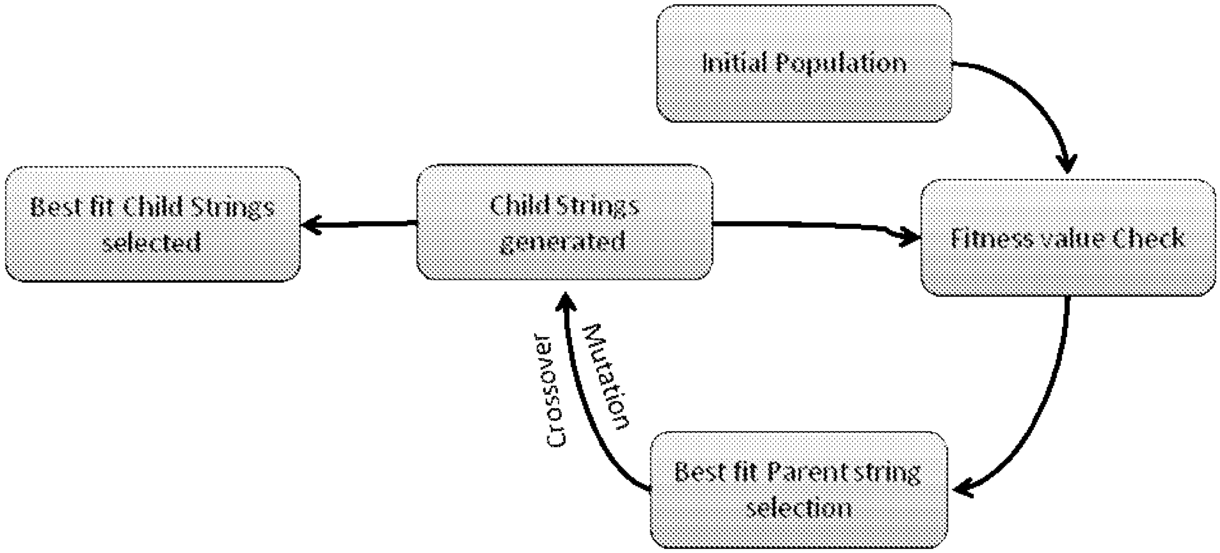


Fig 3.2

Genetic Algorithm Process

The following sequence of steps help yield an optimized solution:

- *Randomly generated initial population strings:*

A given search space is selected to randomly generate a *Population* for the given optimization problem. The Population consists of all the *Individuals* that can be generated within the given search space. These individual members are also termed as *Problem Variables*.

- *Individual Encoding:*

Encoding forms the basis for initiating a Genetic Algorithm process for an optimization problem.

The randomly generated problem variables (individuals) are 'Encoded' or in other words, converted to suitable strings. The reason for conversion is due to the underlining principle of genetic algorithms that they do not modify the authenticity of the generated individuals but perform manipulations on these encoded strings. Encoding is further classified into four categories

- Binary Encoding
- Permutation Encoding
- Value Encoding
- Tree Encoding

Of these, Permutation, Value and Tree encoding procedures are operation and problem dependent. A brief description of Binary Encoding procedure is as follows.

Binary Encoding

Binary encoding is the most common encoding used in practice. Initial research in Genetic Algorithms started with binary encoding and has been widely used primarily due to its relative simplicity. The binary strings that are generated are termed as *Chromosomes* as they represent the biological components of a gene in a binary form.

As the name suggests, every chromosome formulated through binary encoding consists of a string of bits **- 0** or **1**.

Binary encoding gives many possible chromosomes even with a small number of alleles. However, a major drawback that this coding encounters is that the binary form is often not natural for many problem variables and hence requires corrections to be made after crossover and/or mutation operations.

- *Evaluating Fitness value*

Fitness value forms the primary basis is selecting the best-fit chromosomes from the population. The fitness value for each chromosome is obtained based on a problem specific expression called *Fitness Function*.

A fitness function is a particular type of objective function that quantifies the optimality of a chromosome in a genetic algorithm so that the particular chromosome may be ranked against all the other chromosomes. An ideal fitness function correlates closely with the algorithm's goal, and yet may be computed quickly.

- *Selecting the Best set of parent strings*

Based on the fitness value obtained for each chromosome, the *Best-Fit Individuals* are selected for further GA operations. The best-fit chromosomes are also termed as *Optimal Chromosomes* for their ability to approach an optimal solution to the problem. Optimal chromosomes, or at least chromosomes which are closer to being optimal, are allowed to breed and mix their datasets by any of the several techniques, producing a new generation of chromosomes that would be even better fit when compared to the parent strings.

There are a number of selection processes that are utilized and implemented in GA. The most widely used selection procedures are:

- Roulette Wheel Selection
- Rank Selection
- Steady State Selection
- Elitism

a. Roulette Wheel Selection

Parent strings are selected according to their fitness. The better the chromosomes are, the more chances they have to be selected. A roulette wheel is where all the chromosomes in the population are placed. The size of the section in the Roulette wheel is proportional to the value of the fitness function of every chromosome - the bigger the value is, the larger the section is.

The following algorithm can describe this process:

1. **Sum**: Calculate the sum (S) of fitness of all chromosome in a population
2. **Select:** Generate random number (r) from the interval ***(0,S)***
3. **Loop:** Analyze (Rotate) the population and the fitness sum from ***0*** to sum ***S***. When the sum ***S*** is greater than ***r***, stop and return the current chromosome.

The step **1** is performed only once for each population.

For example, a sample set of chromosomes with the following fitness values and corresponding % of the total fitness is considered as shown in Table 3.1

Table 3.1

Sample Chromosome set with fitness values

Chromosome	Fitness Value	% of Total
1	6.82	31
2	1.11	5
3	8.48	38
4	2.57	12
5	3.08	14

It can be observed from the above table that *Chromosome 3* has the highest fitness value while *Chromosome 2* has the least. The percentage fitness values are used to configure the Roulette Wheel. Fig 3.3 highlights that *Chromosome 3* has a segment equal to 38% of the total area. The Roulette Wheel sections indicate the clear share of the total fitness for the given sample parent set.

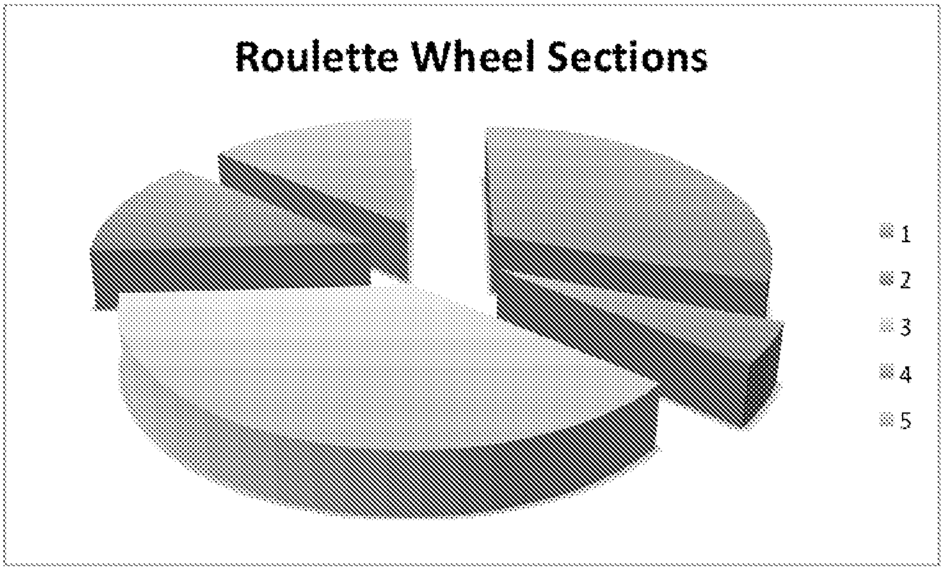


Fig 3.3

Roulette Wheel Sections

The number of times the roulette wheel is spun is equal to size of the population. As can be seen from the way the wheel is now divided, each time the wheel stops, the fitter individuals have the greatest chance of being selected for the next generation compared to the other chromosomes. This can be observed in Fig 3.4 as shown below.

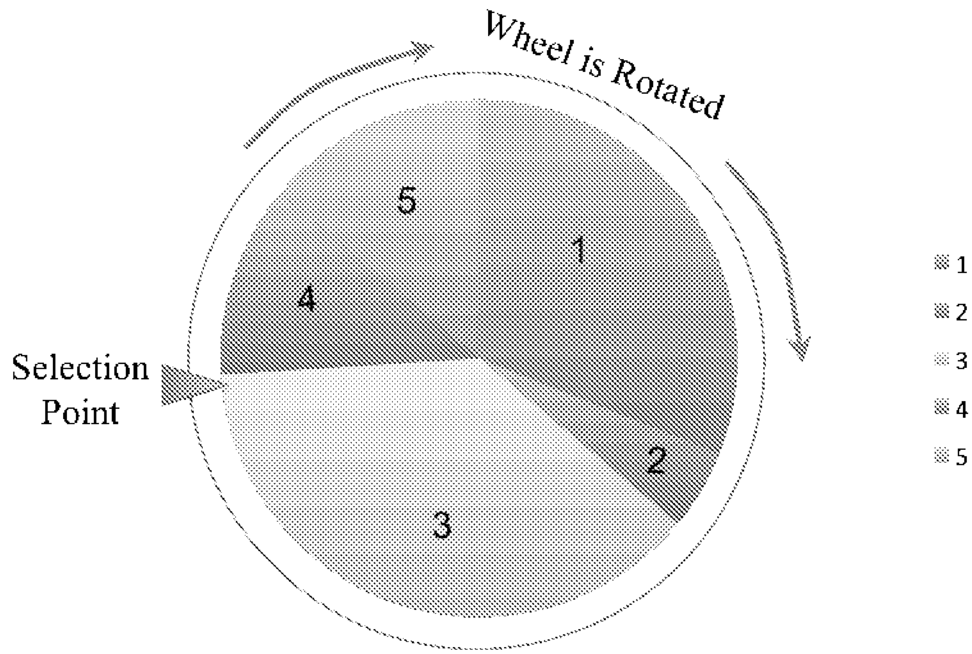


Fig 3.4

Roulette Wheel Selection Process

b. Rank Selection

Roulette Wheel Selection will have problems when there are greater differences in fitness values among various individual population elements. Rank selection ranks the population first and then every chromosome receives fitness value determined by this ranking. This selection process ensures that all chromosomes have a chance to be selected. The primary disadvantage of this procedure is the fact that this method could lead to a slower convergence. This is because the best chromosomes do not differ much with respect to their fitness values from the rest.

c. Steady State Selection

The main principle applied in this selection process is that a large number of chromosomes survive to the next generation while the un-fit get neglected.

d. Elitism

When creating a new population by the processes of crossover and mutation, there is a high probability that the best chromosome is lost in the milieu of operations. Elitism is the process of selecting better individuals, or more specifically, selecting individual with a bias towards the better ones. Elitism is a method that first copies the best chromosome (or few better chromosomes) to the new population. The rest of the population is constructed using the above-mentioned techniques like Roulette Wheel, Rank or Steady State Selection. Elitism can rapidly increase the performance of GA as it prevents a loss of the best-found solution.

- *Creating new strings by applying crossover and mutation operators*

The final important step in the Genetic Algorithm optimization is performing operations to obtain the child strings from the evolved best-fit parent strings. These operations include *Selection, Crossover and Mutation* - terms largely based on the biological evolution theory. These operations yield new child strings that carry the best attributes of both the parent strings, resulting in a better-fit child population.

A fitness value check of all the individuals from the child string population brings out the best-fit child string, which undoubtedly is the most optimum solution to the objective function.

a. Crossover Operation

Every individual of a population is defined by its genetic information - stored in nature using a DNA strand. With the help of the Crossover operator, certain genes from parent chromosomes are selected to create new offsprings, called child strings.

The primary purpose of the crossover operator is to get genetic material from the previous generation to the subsequent generation. When two individual strings undergo crossover operation, the resulting individuals' DNA reflects some of the information from each of the parent strings.

After having been selected, two individual chromosomes swap genetic material to create 'offsprings', called child strings. The idea is that, through this swapping of material, even two relatively average individuals can create even more fit offspring.

Crossover Classifications

Based on the methodology applied to generate child strings from two parent binary strings, the crossover operation can be classified into four categories as explained below.

a.1 Single point crossover

In this type of operation, only one crossover point exists. Binary string from the beginning of the chromosome to the crossover point is copied from the first parent while the rest is copied from the other parent.

The operation of a single point crossover can be explained from the example shown below in Fig 3.5

11001011 + 11011**111** = **11001111**

Fig 3.5

Single Point Crossover Example

a.2 Two-point crossover

As indicated in the name, two crossover points are selected here. Binary string from the beginning of the chromosome to the first crossover point is copied from the first parent, the part from the first to the second crossover point is copied from the other parent and the rest is copied from the first parent again.

The operation of a Two-point crossover can be explained from the example shown below in Fig 3.6

110010**11** + 11**0111**11 = **11011111**

Fig 3.6

Two – Point Crossover Example

a.3 Uniform crossover

There is no specific number of crossover points in this classification. Bits are randomly copied from the first or from the second parent string to form the child string.

a.4 Arithmetic crossover

Arithmetic operations including AND, OR, NAND & NOR operators are the basis for generation of child strings from parent chromosomes using this crossover classification.

Crossover Probability

Crossover probability determines the frequency of performing a crossover operation. If there is no crossover, offspring (child stings) are exact copies of the parent strings. If there is crossover, offspring are made from parts of both parent's chromosome.

- If crossover probability is **100%**, all offspring are generated through crossover operations.
- If the probability is **0%**, a whole new generation is made from exact copies of chromosomes from the parent population.

b. Mutation Operation

Over time, all of the individuals remaining in the population may have lost a specific attribute. Mutation allows for the reintroduction of attributes, by randomly altering the characteristics of an individual.

The purpose of mutation in GAs is to allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. In other words, mutation prevents failing of all solutions in population and presents a local optimum solution to the problem. The new offspring child strings can be changed randomly using Mutation. For binary encoding, a few randomly chosen bits can be interchanged from 1 to 0 or from 0 to 1.

The following individual shows the effects of mutation:

BEFORE: 1101101001101110
AFTER: 1101100001101110

Fig 3.7

Mutation Process

Mutation Probability

Mutation probability determines how often parts of chromosome will be mutated. If there is no mutation, offspring are generated immediately after crossover (or directly copied) without any change. If mutation is performed, one or more parts of a chromosome are changed.

- If mutation probability is 100%, the entire binary structure of the chromosome is changed
- On the other hand, if the probability is 0%, no part of the chromosome is changed.

3.3 General applications of Genetic Algorithms

With a complete generic overview of the basic structural organization in implementing Genetic Algorithms, it makes it necessary to understand their applications in real-time problems.

Genetic Algorithms have successfully been applied to a wide range of optimization problems. On a more specific note, Genetic Algorithms (GAs) have been implemented with an illustrious reputation in solving *combinatorial optimization problems* – problems in which optimization fortifies a fundamental solution in addition to an intermediate solution. The use of Genetic Algorithms in a number of these special optimization problems has produced improved solutions than most of the other conventional methods.

3.3.1 Voltage Stability Specific Application

With reference to Power system voltage stability, a few theories have been proposed on varied topics, involving contingency ranking [22], [23], system robustness measure and economic impact analysis. A more viable combinatorial optimization problem appears in mitigating voltage collapse condition appearing on a system.

The primary objective of voltage collapse mitigation is to use appropriate control actions to bring back system stability.

Before finalizing on the necessary mitigatory measures, it is essential to determine the collapse loading condition and identify the critical point of the system for given system parameters. As a result, critical point identification forms an intermediate problem that needs to be handled before a solution to voltage collapse mitigation is determined.

The need to optimize a fundamental problem in addition to an intermediate issue classifies it as a combinatorial optimization problem with Genetic Algorithm as a more suitable optimization approach. The technique proposed in this chapter concentrates on bringing out the optimizing capabilities of Genetic Algorithms in determining the appropriate combination of control actions to be taken in bringing a critically loaded system back to its stable operating condition. The intermediate solution that it utilizes would be the critical loading point determined by the Collapse Proximity Index (CPI) described in Chapters 2 of this thesis.

The Genetic Algorithm based combinatorial optimization approach to voltage collapse mitigation can be represented as shown in Fig 3.8.

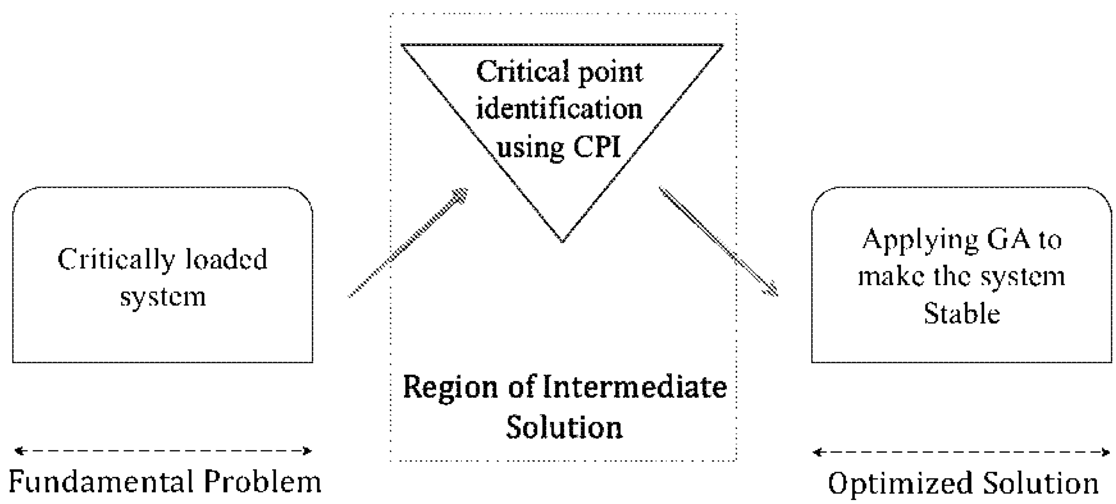


Fig 3.8

GA Based Combinatorial Optimization Approach to Voltage Collapse Mitigation

3.3.2 Integrating CPI and Genetic Algorithms for Voltage Collapse Mitigation

The following Fig 3.9 represents schematically, the operation flow in using the formulated Collapse Proximity Index as an intermediate solution in generating optimized control actions based on the principles of Generic Algorithms to mitigate a voltage collapse condition. Integrating the index along with Genetic Algorithms makes it a combinatorial optimization problem.

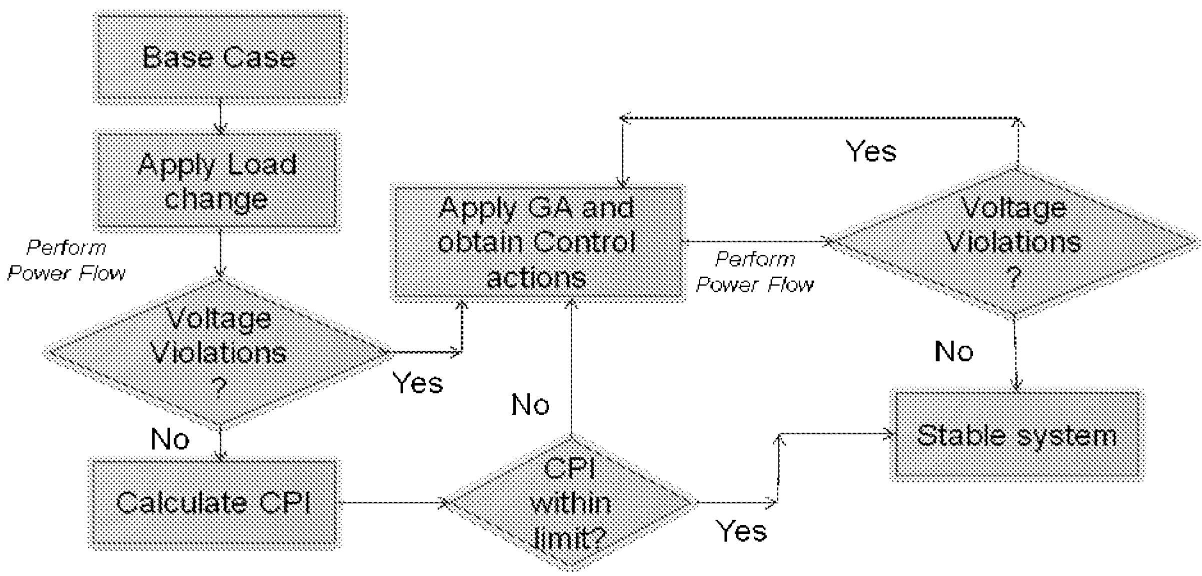


Fig 3.9

Operation Flow for CPI Integrated GA Approach

The entire process leading to Voltage collapse mitigation, as shown in the above figure, can be categorized into three sections:

- Load Incrementation
- Critical Point Identification
- Voltage Collapse Mitigation

a. Load Incrementation

In order to identify the critical loading point for a given system at a load bus, the most efficient procedure is to increment the load by a defined margin and observe the system operation.

Load Incrementation is performed starting from 100% loading condition of the base-case load at a particular load bus with load increment of the order of 10% - 20% of the actual (base case) active power load value. The defined load increment pattern helps in identifying the critical loading point efficiently and accurately. The value of Incrementation depends on the closeness to the critical loading point as defined in by the Collapse Proximity Index.

b. Critical Point Identification using CPI

As mentioned in detail in Chapter 2, the Collapse Proximity Index provides an accurate indication of the critical loading point for a load at a particular bus in a system. The identification of the Critical loading condition forms the Intermediate solution to the Combinatorial Optimization Problem.

c. Voltage Collapse Mitigation using GA

Incorporating the principles of Genetic Algorithms, the best-fit combination of earmarked control actions are generated, helping push the system away from a voltage collapse condition.

The control actions that can be utilized to mitigate voltage collapse have been identified and discussed in detail in Chapter 1, Section 1.4. As each power system has varied system parameters, the control actions selected for implementation are problem-specific and cannot be generalized.

3.4 Sequence of Steps in Applying Genetic Algorithms

The following algorithm presents an outline of all the steps implemented using Genetic Algorithms to mitigate a voltage collapse situation:

1. *Population Size determination:*

   Before initiating a Genetic Algorithm process, the population size of the randomly generated strings has to be assumed. Population size would vary depending on the accuracy and the time-consumption requirements for a given problem. As the given combinatorial optimization technique requires higher accuracy and lesser time consumption, a population size of ***'20'*** is selected for computation.

2. *Appropriate control action identification:*

   In order to initiate the GA process, the control actions required for mitigating a collapse situation have to be identified. As specified in Chapter 1, Section 1.4, a number of control actions can be implemented based on system parameters and economic liabilities. Some of the most commonly implemented control actions include *Transformer tap settings* and *Capacitor bank settings*.

3. *Initial Population generation:*

   Based on the selected control actions, an initial population of identified population size is generated. Individual population is generated for each control action. All individuals in the population are randomly generated, but are maintained within the specified boundary values for each control action.

4. *Population Encoding:*

   Taking into consideration, all the advantages associated when comparing to other encoding techniques, *Binary Encoding* is selected to convert all population individuals into binary strings called *Chromosomes*.

5. *Fitness Value determination:*

   One of the most vital steps in Genetic Algorithms is to determine the fitness value of individuals based on the Objective (Fitness) function.

As the collapse mitigation problem is a combinatorial optimization issue, an intermediate solution based on the Collapse-Proximity Index (CPI) is utilized in the fitness function.

Hence, for this optimization problem, the Genetic Algorithm is implemented with respect to the following fitness function f:

$$\boldsymbol{f(\mathrm{i}) = 1 / [\ 1 + \mu * \mathrm{CPI}\ (\mathrm{i})\ ]} \tag{3.1}$$

where:

i = Individual from the generated Population

CPI = Collapse Proximity Index

μ = Precision Identifier

Population member ***i*** represents one set of combination of different control actions randomly generated in Step 3. As each combination refers to a distinct system configuration, the load flow and consequential equivalent circuit reduction would result in a distinct *Collapse Proximity Index.*

The importance of *Precision Identifier* μ is to help improve the accuracy of the fitness value calculated.

The range of μ is limited for a simple reason that a very high value would result in an extremely small fitness value while smaller values of μ would yield indistinguishable fitness values. A value of **0.01** has been identified as a more favorable value for μ. The fitness value calculated for each population member is stored in an array of size equal to the total population size.

6. *Selection of Best-fit chromosomes:*

Roulette Wheel Selection scheme is implemented in identifying the best-fit chromosomes from the population. As Roulette Wheel selection is simpler to implement and provides a greater probability of selecting strings with higher fitness values more often than those that are less fit, the child strings to be generated are expected to be of better attributes than with those implemented on other schemes.

The three steps in Roulette Wheel Selection process – Sum, Select and Rotate – are followed and the best-fit strings are isolated from the rest of the population for further operations. An overview of all the steps performed in the Roulette Wheel Selection Process is as follows:

i. The fitness values of all the individual population members are stored.

ii. The population counter is initialized to $i = 0$ and incremented by counter $i = i + 1$

iii. The selection counter is initialized as $j = 0$ and the cumulative sum (fitness total) as $S= 0$

iv. A random number X is generated

v. Selection counter is incremented as $j = j + 1$

vi. The cumulative sum of fitness is calculated as

$$S= S + f(i)/ \sum f(i)$$

vii. Till the value of $X >= S$, Step (vi) is repeated.

viii. When the above condition is satisfied, the individual is selected and stored as SEL_i.

ix. The whole process from Step (iii) is repeated for the entire population of chromosomes.

x. SEL represents the best-fit chromosomes selected from the population.

7. *Generation of Crossover site:*

Before performing Crossover operation on the best-fit strings, it is necessary to identify the *Crossover Site* – the position at which crossover between two parent strings has to be performed.

In order to generate crossover sites for parent strings, the maximum string size of the parent string has to be taken into consideration. It should be noted that the maximum position of crossover couldn't exceed or equate to the total binary string length.

For example, considering a binary string A & B as shown below for a crossover operation:

A – 1000100101001001

B – 1011110110001011

As the maximum string length is 16,

Crossover Site < 16

Also, a pair of the best-fit chromosomes would have one common crossover site.

Taking all these factors in view, random number initiation is implemented in obtaining crossover sites for each pairs of best-fit parent binary strings.

Hence, for a best-fit chromosome population of size N, N/2 Crossover sites are generated.

8. Crossover Operation:

For the crossover operation, the *crossover probability* for this Genetic Algorithm optimization problem is taken to be *100%.* In other words, the entire string of the child string is formed by crossover operation. The ideal crossover type that can be effectively implemented in this combinatorial optimization problem is the *Single-Point Crossover*.

As the binary string length of each chromosome is not constant over different populations, Single point crossover is more advantageous here, making it simpler and efficient for programming and comprehending.

Crossover is performed between pairs of chromosomes in each parent string population. It is to be noted that crossover is performed between pairs within the same population only. As each population represents a randomly generated array of a particular control action value, crossover cannot be performed between two individuals from two different populations (two different control action elements).

Upon crossover, a new population of binary chromosomes, called *Child Strings* is generated. These child strings have the best attributes of both the parent strings.

9. *Child String fitness evaluation after Crossover Operation:*

While the arrays of newly formed child strings are in their binary forms, there is a need to convert these binary strings to their decimal equivalents in order to evaluate their fitness.

The binary-to-decimal conversion yields a new set of individual elements representing various combinations of control actions in each population. In order to calculate the fitness value for each population member, respective control action combinations are incorporated into the power system. The two-bus equivalent system produces a specific Collapse Proximity Index (CPI) for each control action combination and this Child string – based CPI is utilized in the fitness function.

Fitness value for each of the population member is calculated using the expression from 3.1. All the fitness values are tabulated for best-fit child string selection.

10. *Repeated Crossover:*

From the above fitness value evaluation, if the fitness values of at least 50% of the child strings from a population are less than or equal to the fitness value of the best-fit parent string from the parent population, multiple crossover operations are performed.

Repeated crossover is performed on the child strings until the fitness value evaluation yields a population of child strings of which 50% or more are more fit than compared to their best-fit parent string. Repeated crossover greatly increases the possibility of obtaining the best-fit child chromosomes as the search space is limited and possibility of local minima stagnation is drastically reduced.

The Genetic Algorithm based for this specific optimization problem restricts the maximum repetitions in crossover operation to 10. If the requirement for a repetitive crossover operation exceeds beyond 10, mutation operation is performed.

11. *Mutation Operation:*

Mutation is not implemented for all population strings. Depending on the fitness values obtained for the child strings, the applicability of mutation operation is decided.

If the fitness values of all the child strings in a population generated by ten repetitive crossover operations are less than or equal to that of the best-fit parent string for that population, mutation operation is performed.

If on the other hand, all child strings for a population have fitness values greater than that of the best-fit parent string for that population, mutation is not required and the best-fit child string among the lot is selected as the optimized solution.

If, upon fitness value analysis, it has been determined that mutation operation is inevitable, a *mutation probability* of 1% (0.01) is considered for the operation. The reason for low mutation probability is due to the fact that a higher probability would turn the process into a primitive random search. A low mutation probability helps prevent the child string population from stagnating at local optima.

As binary encoding is put into operation, *Flip-Bit mutation process* is performed on the child strings as shown in Fig 5.11 of Chapter 5. The flip bit process inverts the value of a chosen gene, changing bit 0 to 1 and vice-versa.

12. *Child String Fitness Evaluation after Mutation Operation:*

Mutation yields child strings that are more optimized and fit than their predecessors. The fitness of these better-optimized strings is calculated by repeating the fitness evaluation procedure as before.

Upon identification of the highest fitness value from the array, the corresponding mutated child string is extracted out and the decimal values for the different control actions are generated. The control action values obtained are the <u>*best-fit optimized control actions*</u> that would bring the system back to its stable operating condition.

If mutation operation is not performed, the most optimized control actions are extracted from identification of best-fit child strings obtained after crossover (or repeated crossover) operations.

A flow chart representing the sequence of events implemented for the above described Combinatorial Optimization Problem based on the principles of Genetic Algorithm is as shown in Fig 3.10.

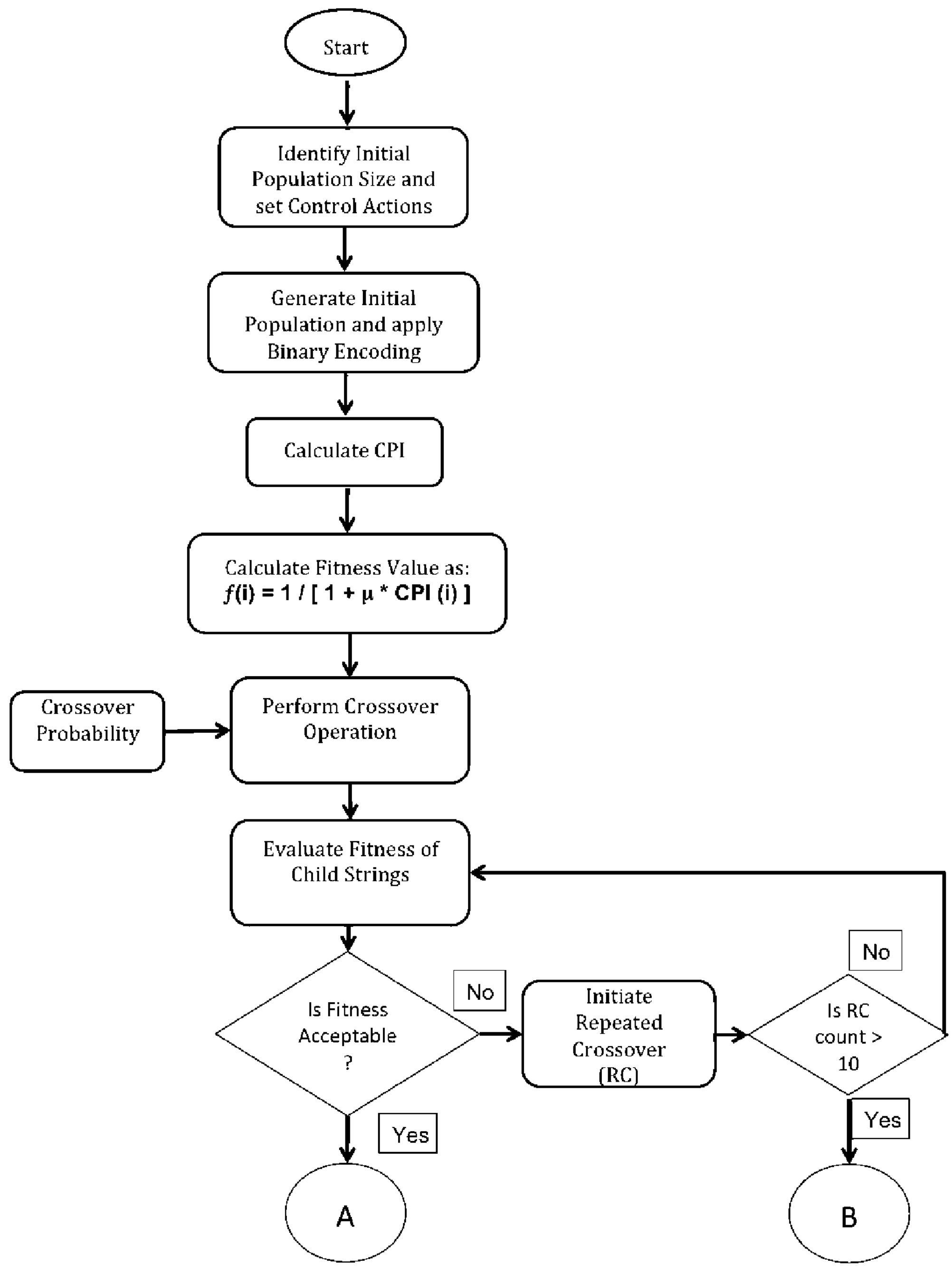


Fig 3.10 (continued on Pg. 76)

Flowchart for Application of Genetic Algorithms

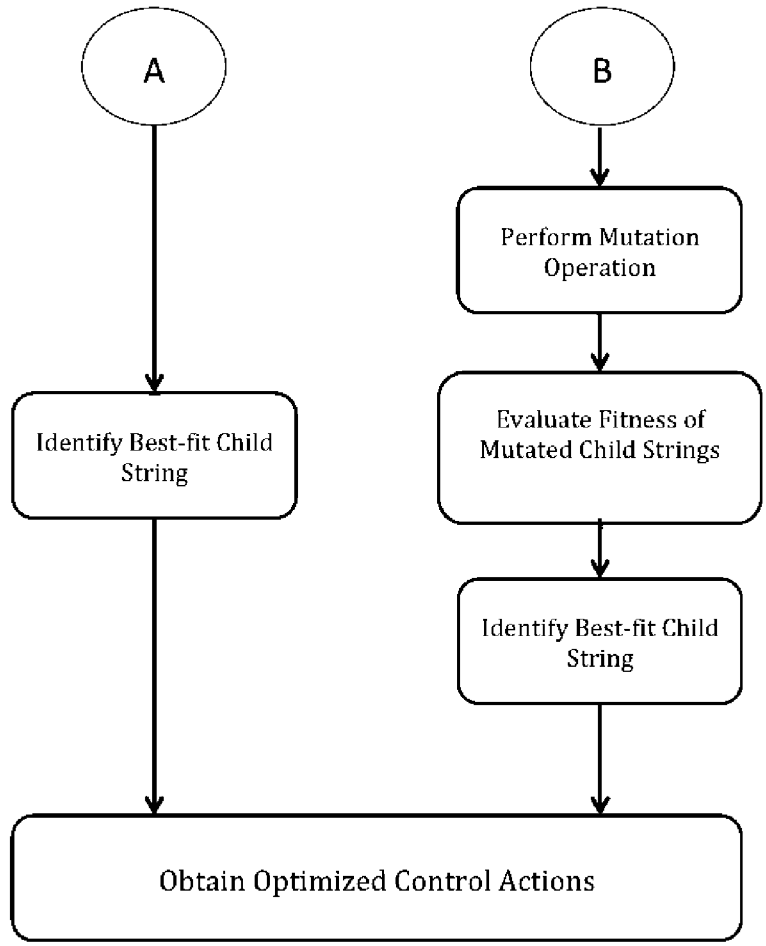


Fig 3.10

Flowchart for Application of Genetic Algorithms

3.5 Software Programs Employed

In order to implement the Genetic Algorithm-based approach on a system with critical loading conditions, a combination of Power system related software programs is utilized. The list of software packages employed in program development are as follows:

- Siemens PTI's PSS/E (Version 31) [24]
- Mathworks' MATLAB (Version R2007a) [25]
- MATPOWER (Version 3.2) [26]

An overview of all the Softwares utilized for this scheme can be observed in Appendix C (Pgs. 121).

CHAPTER 4

TESTING THE GA-BASED VOLTAGE COLLAPSE MITIGATION SCHEME

4.1 Introduction

The Genetic Algorithm – based combinatorial optimization technique developed in Chapter 3 is simulated on a test system. The critical loading point of the system with respect to a load bus is identified and an appropriate combination of control actions is generated to bring the system out of the voltage collapse condition using the principles of Genetic Algorithms.

4.1.1 PSS/E New England test system – System under consideration

A PSS/E sample system named ‘SAVNW – New England Test System’ is considered to implement the proposed optimization technique. Siemens Power Technologies International has developed the system as a sample system based on a New England power system model.

Some key features of the test system are as shown below:

- The given system is a 3-area, 4-zone, 23 bus system which includes 21.6kV, 230kV and 500kV bus subsystems

- There are 6 generating units in the system, with a maximum generation of up to 800MW
- The system consists of 23 transmission lines connecting 18 load buses and 6 generator buses. 2 multi-section lines are also put in place to connect a hydro station to a 230kV sub-station
- Bus 3011 is considered to be the swing bus of the system
- 8 PQ loads are connected to the system with loads ranging between 200MW to 1200MW depending on the proximity to the central district
- The system also consists of 11 two-winding transformers
- 5 Capacitor shunt banks are placed at strategic locations on the system

A single-line representation of the above mentioned system could be observed in Fig 2.3. The system network data is tabulated and can be observed in Tables A-1 to A-4, Appendix A.

4.1.2 Normal Operation Conditions for the system

The test system mentioned above shows no Voltage violations in its normal operating condition. The system convergence is met in one iteration. The p.u voltage values for the system under Normal Operating conditions can be observed in Table A-1 in Appendix (Pg 107).

4.2 Load Bus to be examined

In order to effectively implement the proposed GA-based control action procedure for voltage collapse mitigation, it is necessary to identify the load bus that encompasses the following attributes:

- Lightly loaded under normal operating conditions
- Bus voltage well within the upper and lower limits
- Closeness to a central district

Taking all the above-mentioned factors into consideration, ***Load Bus 153*** is selected for load increment and voltage collapse mitigation simulation. Load Bus 153 displays the following features to form an ideal test load point:

- Bus 153 has a steady-state loading condition of 200 MW, 100 MVAR, one of the lowest compared to the rest of the loads in the system, making it a highly likely test bus for the load increment simulation.
- The normal operating voltage at Bus 153 is 0.9930 p.u, well within the lower voltage limit of 0.94 p.u and upper limit of 1.08 p.u.
- The location of the bus is one of the most important features for its selection. Load Bus 153 is situated in the Midtown region of Area 1. The closest bus to Bus 153 is the Bus 154, situated in Downtown region of Area 1.

From transmission planning perspective, this makes Bus 153 an ideal bus for addition of load when a situation arises where load expansion in the area is inevitable. This raises more creditability in selecting Bus 153 for the load increment simulation.

A clearer understanding of the advantages of Bus 153 as a test point for the GA-based simulation can be obtained from Fig 4.1, depicting the location of Bus 153 in Area 1.

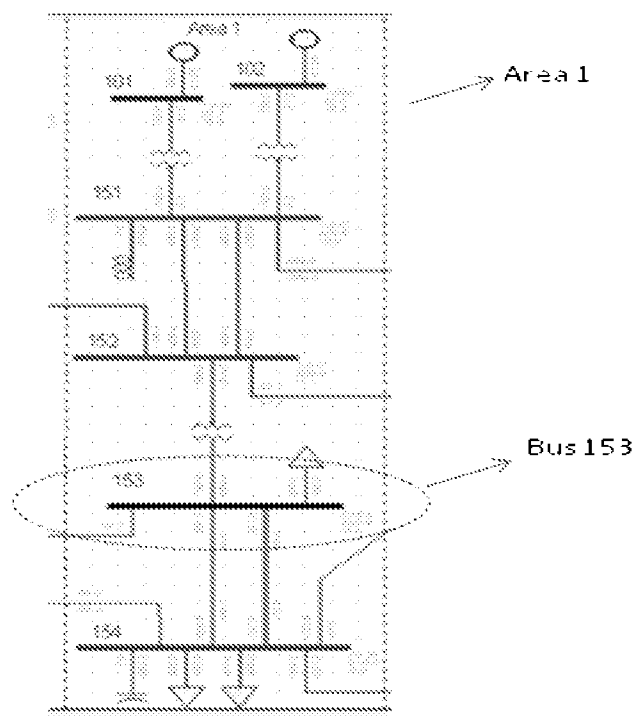


Fig 4.1

Load Bus 153 (PSS/E system) Identification

4.3 Loading Capability Analysis for Bus 153

4.3.1 Simulation

The maximum loading capability of Load Bus 153 is determined as an intermediate solution before optimization is applied for voltage collapse mitigation. Collapse Proximity Index is utilized in determination of the critical point of loading, beyond which, the system moves to voltage instability region. The following steps are included in determination of the critical loading point:

- *System reduction to a two-bus equivalent:*

  The entire system is reduced to a two-bus equivalent behind load Bus 153. System swing bus at Bus 3011 is considered to be the equivalent system source. The two-bus equivalent is obtained by repeated Kron reduction of the entire system. The equivalent system is observed as shown in Fig 4.2:

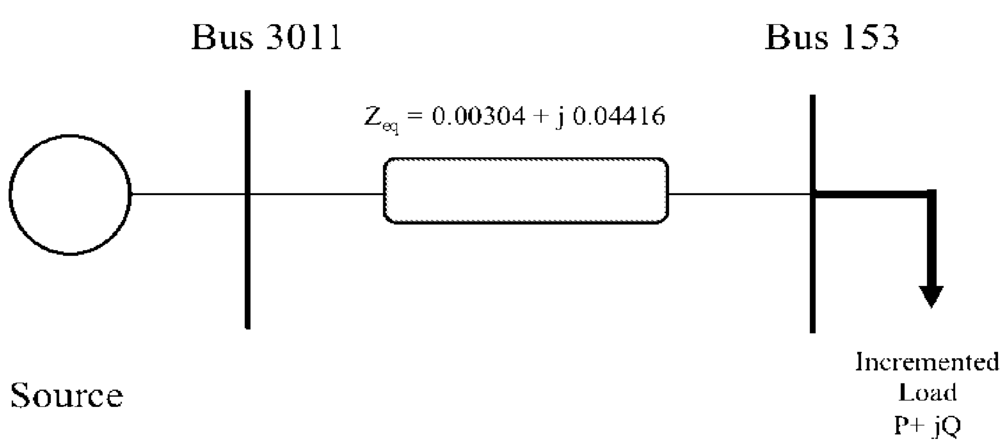


Fig 4.2

Single line representation of the PSS/E 'SAVNW' test system with respect to Bus 153

- *Load increment*

The load at Bus 153 is incremented by approximately 20 MVA. For every load increment, the value of P_{MAX} is calculated for CPI calculation based on Equation 2.11 in Chapter 2.

- *Calculation of CPI*

The Collapse Proximity Index is calculated based on the expression described in 2.12 in Chapter 2. The value of the index is tabulated for each loading condition based on the P_{MAX} calculated as shown in Table 4.1. The voltage variations at Load Bus 153 observed with respect to the loading increment is also tabulated as shown in Table 4.2.

Table 4.1

Collapse Proximity Index Table for Bus 153 (PSS/E system)

P Load (MW)	% of Maximum Base-Case Loading	CPI
200	100	5.27734
240	120	4.269395848
280	140	3.582853636
320	160	3.066368032
360	180	2.663210144
480	240	1.850730342
600	300	1.351804197
620	310	1.285003386
660	***330***	***1.164931925***
680	*340*	*1.100242129*

Table 4.2

Voltage and Active Power demand comparison for Bus 153

P Load (MW)	V_R
200	0.993
240	0.9887
280	0.9841
320	0.9794
360	0.9744
480	0.9582
600	0.9393
620	0.9358
660	***0.9158***
680	*0.1427*

4.3.2 Result Analysis

- *Observing the region of orientation*

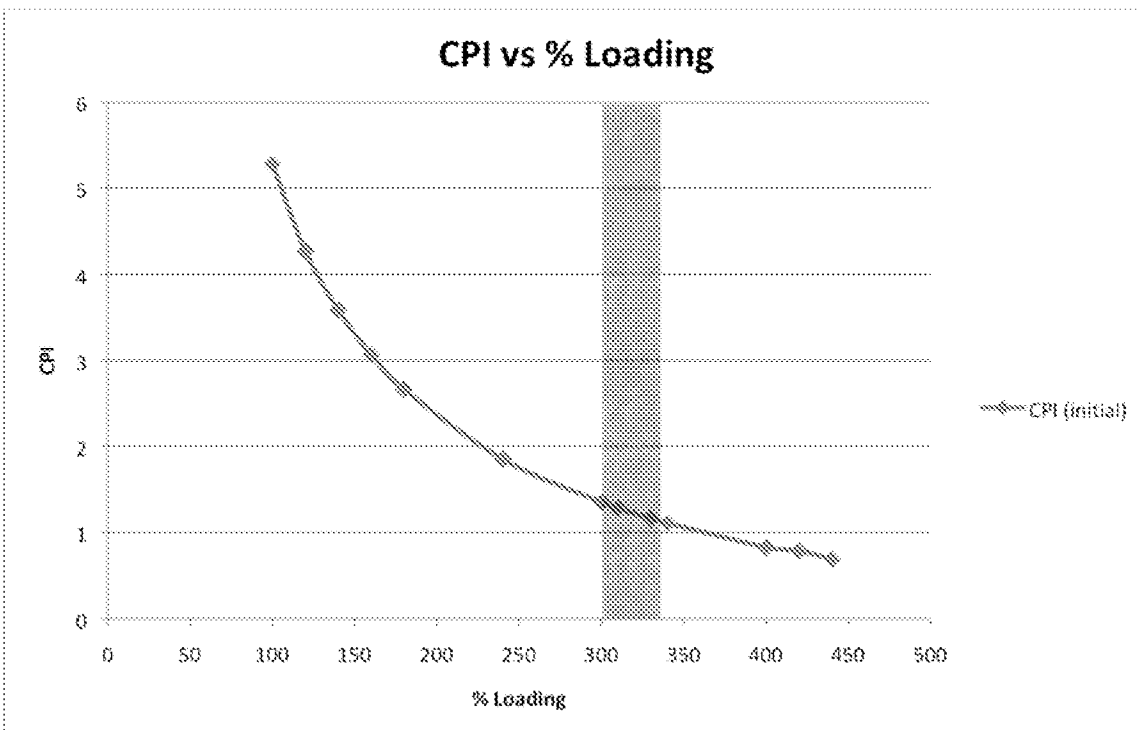
  The Collapse Proximity Index is observed in the region of orientation, as described in Figure 2.2. The tabulated CPI values for different loading conditions determined are observed and their trend is noted. The region of orientation portrays *a distinct variation in CPI* as loading at Bus 153 approaches 300% of its base case loading condition.

- *Determining the critical loading point*

  The critical loading condition, based on the orientation of the indices is determined.

It can be observed that the <u>*critical loading condition*</u> is reached when the load is <u>***330%** of the base case loading at Bus 153*</u> for the PSS/E test system.

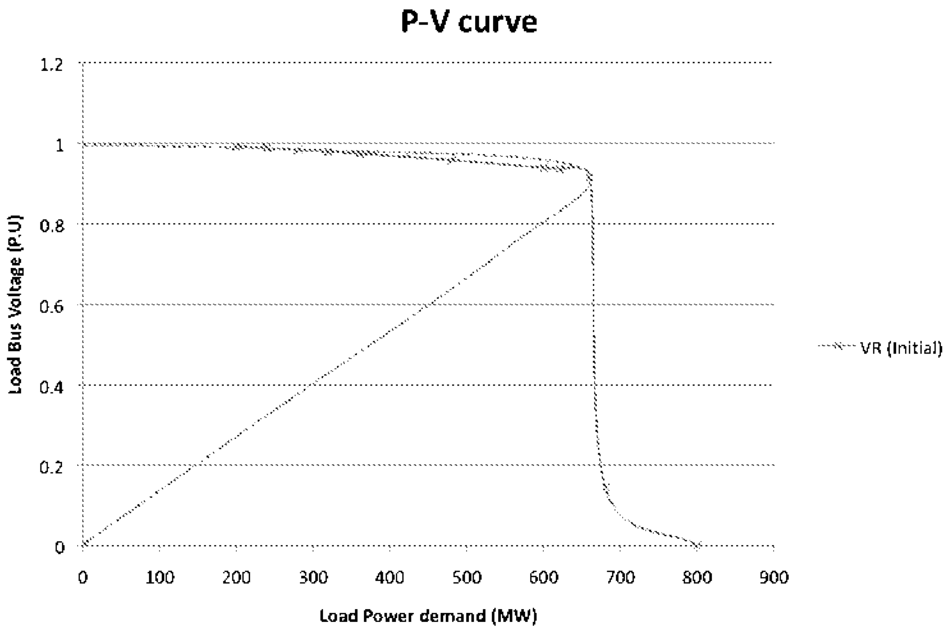
The following plot (Plot 4.1) shows the trend observed for the Collapse Proximity Index with respect to the load increment. It can be observed the region between 300% and 330% of base case loading, shown as a shaded region in the plot, refers to the <u>*critical loading region*</u>.



Plot 4.1

CPI vs % Loading for Bus 153 (PSS/E System)

The corresponding P-V curve for the above loading is obtained as shown in Plot 4.2.



Plot 4.2

P-V curve for Bus 153

The P-V curve clearly shows that *beyond a loading of 660 MW*, the system voltage drops drastically, leading to a voltage collapse condition.

4.4 Voltage Collapse Mitigation using Genetic Algorithms

The application of Genetic Algorithms into the voltage collapse mitigation problem is a combinatorial optimization technique, as observed from Fig 3.12, Chapter 3.

The sequence of steps involved in mitigating voltage collapse condition at Bus 153 from the test system, thereby improving the voltage profile, by the application of Genetic Algorithms is as follows:

1. *<u>Population Size determination:</u>*

   As the given combinatorial optimization technique requires higher accuracy, a population size of ***'20'*** is selected for computation.

2. *<u>Appropriate control action identification:</u>*

   A number of control actions can be implemented based on system parameters and economic liabilities. The PSS/E SAVNW New England test system consists of Capacitor banks and transformer taps, two of the most commonly used control actions for voltage collapse mitigation. Load shedding is an alternative but is consciously shunned as it negates the eventual objective of the optimization scheme – increasing the loading capability of the system.

   Taking all the factors into consideration, *Transformer tap settings* and *Capacitor bank settings* are considered as the two sets of control actions to be implemented.

3. *<u>Initial Population generation:</u>*

   Based on the two selected control actions, an initial population of with a population size of 20 is generated for each control action.

All individuals in the population are randomly generated, but are maintained within the specified boundary values for each control action.

The boundary limits for

- Transformer tap position – 1 to 33
- Capacitor Bank settings – 0 MVAR to 1000MVAR

4. *Population Encoding:*

Binary Encoding is utilized in converting all population elements into binary strings called chromosomes.

A few of the converted populations strings are obtained as shown in Tables 4.3 and 4.4 for both the control actions (transformer tap settings, capacitor bank settings) respectively.

As there are 11 Transformer tap settings and 5 Capacitor bank settings to be monitored, the initial population binary arrays are of the size [20 x 66] and [20 x 50] respectively. Sample structures of encoded chromosomes representing all transformer tap and capacitor bank settings are shown in Fig 4.3 and Fig 4.4 respectively.

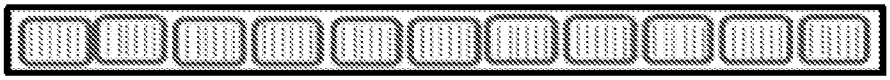


Fig 4.3

Encoded Chromosome structure for Transformer Tap setting

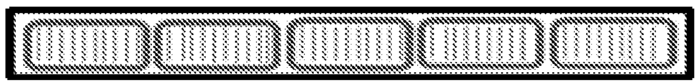


Fig 4.4

Encoded Chromosome structure for Capacitor Bank setting

Tabular representation of Control Actions:

Table 4.3

Initial randomly generated Population for Transformer Tap Positions

Decimal Values		Binary Equivalent					
27		0	1	1	0	1	1
30		0	1	1	1	1	0
5		0	0	0	1	0	1
31		0	1	1	1	1	1
21		0	1	0	1	0	1
4		0	0	0	1	0	0
10		0	0	1	0	1	0
19		0	1	0	0	1	1
32		1	0	0	0	0	0
6		0	0	0	1	1	0

Table 4.4

Initial Randomly generated Population for Capacitor Bank Settings

Decimal Values		Binary Equivalent									
36		0	0	0	0	1	0	0	1	0	0
110		0	0	0	1	1	0	1	1	1	0
448		0	1	1	1	0	0	0	0	0	0
480		0	1	1	1	1	0	0	0	0	0
328		0	1	0	1	0	0	1	0	0	0
381		0	1	0	1	1	1	1	1	0	1
43		0	0	0	0	1	0	1	0	1	1
449		0	1	1	1	0	0	0	0	0	1
462		0	1	1	1	0	0	1	1	1	0
63		0	0	0	0	1	1	1	1	1	1

5. *Fitness Value determination:*

Using the expression from 3.1, Chapter 3, the fitness values of all individual elements from the two population spaces are calculated. As the collapse mitigation problem is a combinatorial optimization issue, an intermediate solution based on the Collapse-Proximity Index (CPI) is utilized in the fitness function.

Population member ***i*** from the fitness function represents one set of combination of the two control actions randomly generated in Step 3. As each combination refers to a distinct system configuration, the load flow and consequential equivalent circuit reduction result in a distinct *Collapse Proximity Index.*

The value of the precision identifier μ is taken to be **0.01** for the fitness function.

The fitness value calculated for each population member is stored in an array of size equal to the total population size. A fitness value table representing values of population strings from a random generation are shown in Table 4.5.

Table 4.5

Fitness values of randomly generated Population Strings

Population Number	Fitness Value
1	0.9874
2	0.9849
3	0.9866
4	0.9864
5	0.9853
6	0.9856
7	0.9842
8	0.9856
9	0.9869
10	0.9841
11	0.9854
12	0.9851
13	0.985
14	0.9853
15	0.9854
16	0.9876
17	0.9858
18	0.9865
19	0.9848
20	0.986

6. *Selection of Best-fit chromosomes:*

   Taking into considerations all the advantages associated with it, *Roulette Wheel Selection* scheme is implemented in identifying the best-fit chromosomes from the population. All the steps mentioned under Roulette Wheel Selection in Section 3.2, Chapter 3 are performed and the best-fit strings are selected.

7. *Generation of Crossover Site:*

   Crossover Sites are generated for all the best-fit parent strings in order to perform crossover operation. A crossover site is commonly generated for a pair of parent strings of the population.

   As mentioned earlier, the maximum position of crossover cannot exceed or equate to the total binary string length. In this optimization problem, two populations of best-fit parent strings exist, each with different binary string length.

   Binary String length for

   - Population representing Transformer Tap positions:

     6 (for each tap position, totaling to 66 for 11 transformer tap positions)

   - Population representing Capacitor Bank Setting:

     50 (combined strings for 5 capacitor banks, each of length 10 bits)

Hence, the crossover site for the best-fit parent strings from the two populations should not exceed the following limits:

- For Transformer Tap position strings : Maximum Crossover Site – 5
- For Capacitor bank setting strings : Maximum Crossover Site – 49

8. *Crossover Operation:*

The *crossover probability* for this Genetic Algorithm optimization problem is taken to be *100%*. Crossover is performed between pairs of chromosomes in each parent string population depending on the position of the crossover point.

As there are two population sets of best-fit parent strings - each representing a particular control action, crossover is performed between pairs of parent strings within the same population only. Also, care is taken to perform crossover individually for each pairs of sub-parent strings from the transformer tap position population, similar to a multi-point crossover operation. This can be explained clearly from an example shown below in Fig 4.5.

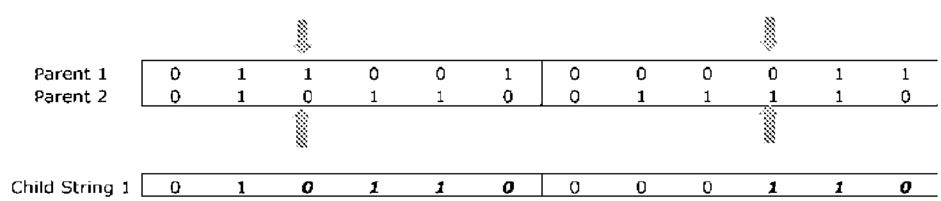


Fig 4.5

Multi-point Crossover between two parent strings

Two tap positions of two best-fit parent strings are shown in the above figure. Crossover operation is performed at locations 3 and 4 respectively on the parent strings. The child string generated is shown containing the traits of the two parent strings in both its sub-strings.

9. *Child String Fitness Evaluation and Repeated Crossover:*

The newly formed child strings, which are in their binary forms, are converted into their decimal equivalents in order to facilitate fitness evaluation.

Fitness values of these strings are determined using the fitness function described in 3.1, Chapter 3. On analysis, it can be observed that there is a need of applying repeated crossover operations till the best-fit individual chromosome is obtained.

Repeated crossover is performed on the child strings until the fitness value evaluation yields a population of child strings of which 50% or more are more fit than compared to their best-fit parent string.

As the best-fit child string is determined in the process of repeated Crossover operations, the necessity of mutation operation is eliminated.

The complete list of tabulated parent and child string populations along with their fitness values and crossover sites can be observed in Appendix B, tables B1-B8.

The MATLAB program developed to generate the desired control actions using Genetic Algorithms for Voltage Collapse Mitigation can be observed in Appendix D.

4.5 Simulation Results and Observations

The above implemented Genetic Algorithm – based Voltage Collapse Mitigation Scheme for the PSS/E 'SAVNW' New England test system generates a combination of control actions that reduce the system instability considerably.

4.5.1 Best Fit Control Actions

On performing repeated Crossover operations and evaluating the fitness values on the generated child stings, a combination of best-fit control actions is obtained.

The two sets of Control Actions implemented in this scheme are:

- Transformer Tap settings – for 11 transformers in the system and
- Capacitor Bank settings – for 5 shunt capacitors attached to the system

A Best-fit combination of two control actions generated using Genetic Algorithms is obtained as shown in Table 4.6.

Table 4.6

Best-fit Control Action combination obtained using GA

Best Transformer Tap Settings
21
18
7
2
27
34
15
26
4
12
28

Best Capacitor Bank Settings
732
265
539
185
741

In order to ensure that the implementation of these control actions into the system does not violate any of the system voltage conditions, a power flow analysis on system is performed, taking into consideration, the loading condition of 340% (Voltage Collapse point) at Bus 153.

4.5.2 Power flow with Control Actions at Collapse Point of 340% loading

The power flow beyond base-case critical loading proves the effectiveness of this scheme by demonstrating the following results:

- ***No Voltage collapse*** is observed at 340% loading at Bus 153 on the system
- ***No voltage limits are violated*** in the entire system with the control actions put into effect

- System stability is maintained
- Collapse Proximity Index (***CPI***) is at a ***higher value*** (>1.1)

The following table, Table 4.7 depicts the improvement in system stability at 340% loading condition with the addition of control actions generated from a Genetic Algorithm run.

Table 4.7

Comparison of System Parameters with and without control actions

	At 340% Loading at Bus 153		
	No Control Actions		**With Control Actions**
Voltage at Bus 153	0.1427		0.94433
Critical loading reached	Yes		No
Voltage Violations in rest of the system	Yes		No
Collapse Proximity Index	1.1092		1.3664

4.5.3 CPI Analysis with control actions-implemented system

The Collapse Proximity Index variation with respect to % Load Increment at Bus 153 for the PSS/E test system under normal operating conditions can be observed in Plot 4.1. With the implementation of the GA – generated control actions on the system, there is bound to be a variation in the Index values when compared to those obtained in the absence of control actions.

The GA – based control actions are incorporated into the system and load incrementation is performed in a manner similar to the procedure specified in Section 4.4.1.

It can be observed that the new Collapse Proximity Indices for different load incrementations with the new system parameters have higher values than their predecessors. A comparative study marking the differences in the old and the new values can be achieved from the following Table 4.8:

Table 4.8

Comparison of CPI values with and without control actions

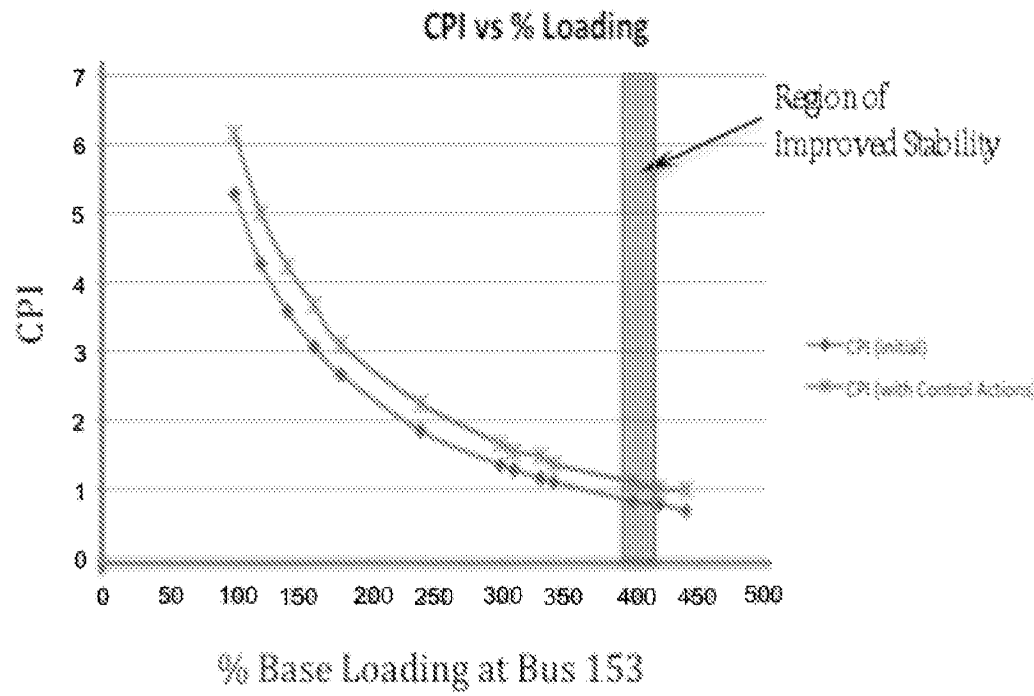
P Load (MW)	% of Base Case Loading	CPI (initial)	CPI (with Control Actions)
200	100	5.27734	6.1543
240	120	4.269395848	5.0034
280	140	3.582853636	4.2332
320	160	3.066368032	3.6714
360	180	2.663210144	3.0988
480	240	1.850730342	2.2438
600	300	1.351804197	1.6587
620	310	1.285003386	1.5476
660	***330***	***1.164931925***	1.4872
680	***340***	***1.100242129***	1.3664
800	***400***	0.827804642	***1.1109***
840	420	0.7872	1.0122

It can be clearly observed from the above table that system stability improves to a great extent with inclusion of GA – generated control actions. The system, which was *previously in a critical operating state at 330% loading*, now remains *stable till 400%* of base case loading is reached.

4.5.3.1 New Critical Loading Condition

From Table 4.8, it can be pointed out that the *new critical loading point* has shifted from 330% to ***400%***.

A graphical comparison of the two Collapse Proximity Indices reveals a more detailed region of critical stability. (Plot 4.3)



Plot 4.3

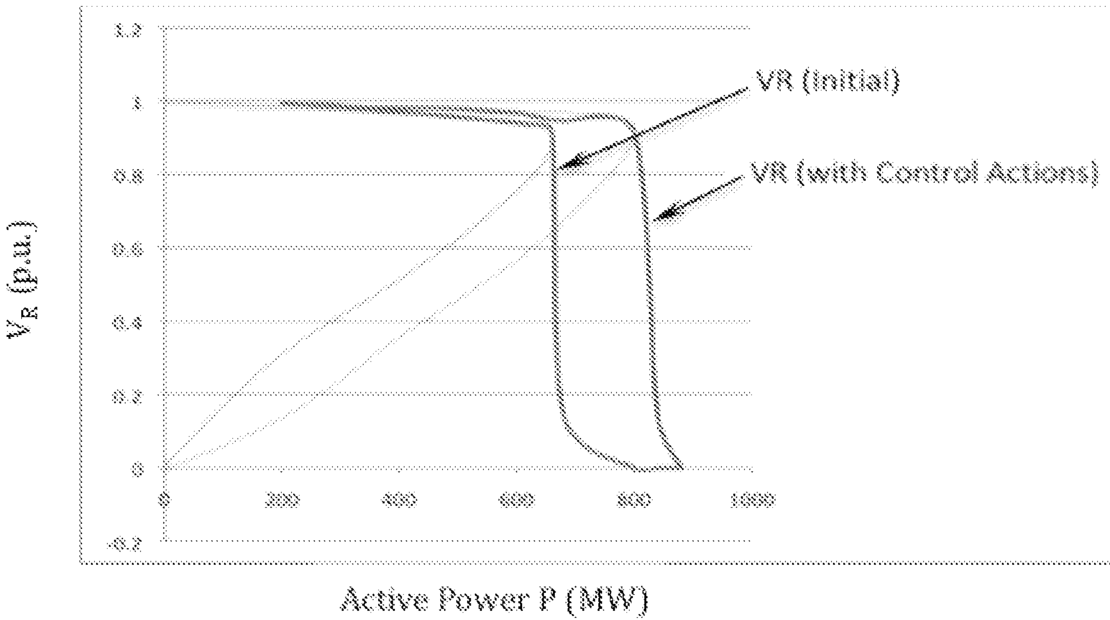
CPI vs % Loading for Bus 153 (PSS/E System) with Control Actions generated using GA

It can be observed the region between 390% and 415% of base case loading, shown as a shaded region in the plot, refers to the *critical loading region*.

4.5.4 P-V curve Analysis

The P-V curve for the PSS/E New England test system without control actions can be compared with the system, which has control actions implemented. The distinction can be observed in Plot 4.4 as shown below.

P – V Curve Comparison



Plot 4.4

P-V curve comparison for Bus 153 with Control Actions generated using GA

In the above plot, V_R represents the receiving end Voltage at Bus 153, as shown along the vertical axis. The horizontal axis represents the active load demand at the load bus.

The plot provides a comparative study of the effect of GA – generated control actions on the P-V curves for the test system. The plot points a ***noticeable shift in the nose of the P-V curve towards right***, indicating an extended stability region.

As explained in the previous section, the *stability region now extends to 400% loading* beyond which critical system stability is observed.

This is a perceptible improvement in stability of the system as, in the absence of control actions, the system shows a critical loading region beyond 330%, with a collapse point of 340%.

4.5.5 Inference – Advantages of the developed scheme

The Genetic Algorithm – based combinatorial optimization technique, focusing on voltage collapse mitigation, provides a comprehensive solution to system instability.

A number of advantages can be associated with the proposed scheme. Some of the practically important advantages are:

- *Shift in Critical Point of Voltage Instability:*

  The Implementation of the proposed scheme on a test system portrays the *shift of the critical point* to a <u>higher loading condition</u> – indicating an improved loading condition to the system.

- *Improved P-V profile:*

  An increased system Loadability results in an improved P-V profile. The scheme implements control actions that help in <u>*increasing the area under stable operating region.*</u>

- *Optimized Control Actions:*

  The <u>*Genetic Algorithm procedure ensures that optimized control actions*</u> are selected for the bringing the system back to stable operating condition.

- *Highly beneficial from Planning perspective:*

  The proposed scheme provides an improved loading capability of a load bus, thereby helping transmission planners in analyzing load sharing and load expanding possibilities. From the test system utilized and the results obtained, it can be observed that the load bus under consideration (load bus 153) can be expanded for incorporating future loads that the area (Area 1) can expect.

- *Flexibility of the Algorithm:*

  The algorithm has been designed and formulated in a layout such that it can be readily applied on any system, taking into consideration, the format of loading system data and varying control action requirements.

4.6 Conclusion and Recommendations

Voltage Collapse and system instability have been some of the most critical, yet difficult-to-handle issues faced by system planners in transmission planning and operation. This thesis provides an efficient and accurate corrective solution to a system that is on verge of voltage collapse.

Voltage Collapse Mitigation requires implementation of appropriate control actions to bring back system stability. The approach proposed in this thesis concentrates on bringing out the optimizing capabilities of a stochastic global search technique called ***Genetic Algorithm*** in determining the appropriate combination of control actions to be taken in bringing a critically loaded system back to its stable operating condition.

A ***Collapse Proximity Index*** to determine the critical loading point for a load bus in a system is formulated as an intermediate solution in generating optimized control actions based on the principles of Genetic Algorithms for voltage collapse mitigation. Integrating the index along with Genetic Algorithms makes it a *combinatorial optimization problem* – a complex classification of problem in which optimization provides a fundamental solution, taking into consideration, the effect of the intermediate solution.

The proposed scheme for Voltage Collapse mitigation is put to test on a three-area test system.

The conclusions drawn from the test results vindicate the accuracy and efficiency of the Genetic Algorithm – based method.

Integrating Collapse Proximity Index with the Genetic Algorithm scheme provides a comprehensive solution to the system instability problem. An apparent shift in the critical loading point indicates an improved loading condition as a result of implementing the GA – based scheme.

The scheme implements control actions that help in increasing the area under stable operating region observed in the P-V curve representation, portraying an improved voltage profile for the load bus. The developed algorithm's flexibility and its generic organization make it an expedient tool for voltage collapse mitigation.

From a power system planner's perspective, the designed technique would not only determine the appropriate control actions to mitigate a critical stability situation, but would also assist in analyzing load-expanding and load-sharing capabilities in an area based on the area's forecasted load data.

A recommendation for future research on this thesis would be to widen the control action search space to facilitate the inclusion of a number of other practically applied control actions including system reinforcement and possibly, load shedding.

Another interesting extension to the thesis would be to assimilate the proposed GA – based mitigation scheme with an improved load forecasting technique, based on the principles of Genetic Algorithm. The load-forecasting tool could be utilized to predict expected loading conditions at a bus and the proposed collapse mitigation scheme could be implemented to improve system stability at these load levels.

APPENDIX

A

PSS/E ‘SAVNW’ New England Test Case

- Bus Data

Table A – 1
System Bus Data

BUS	CODE	AREA	VOLTAGE (P.U.)	ANGLE
101	2	1	1.02	16.55
102	2	1	1.02	16.55
151	1	1	1.0119	10.89
152	1	1	1.0171	-1.12
153	1	1	0.993	-3.24
154	1	1	0.9389	-9.89
201	1	2	1.04	6.16
202	1	2	1.0088	-1.32
203	1	2	0.9665	-6.92
204	1	2	0.9787	-3.73
205	1	2	0.949	-9.18
206	2	2	1.0236	-2.97
211	2	2	1.0404	12.92
3001	1	5	1.0298	-1.37
3002	1	5	1.0279	-1.83
3003	1	5	1.0233	-2.25
3004	1	5	1.0165	-3.43
3005	1	5	0.9948	-5.18
3006	1	5	0.994	-3.79
3007	1	5	0.9637	-8.54
3008	1	5	0.9586	-9.05
3011	3	5	1.04	0
3018	2	5	1.0218	-4.08

PSS/E 'SAVNW' New England Test Case

- Branch Data

Table A – 2
System Line Data

FROM BUS	TO BUS	LINE R	LINE X	LINE B	RATE A	RATE B	RATE C
151	152	0.0026	0.046	3.5	1200	1300	1
151	152	0.0026	0.046	3.5	1200	1300	1
151	201	0.001	0.015	1.2	1200	1300	1
152	202	0.0008	0.01	0.95	1200	1300	1
152	3004	0.003	0.03	2.5	0	0	1
153	154	0.005	0.045	0.1	300	350	1
153	154	0.006	0.054	0.15	300	350	1
153	3006	0.001	0.012	0.03	0	0	1
154	203	0.004	0.04	0.1	200	250	1
154	205	0.0003	0.0033	0.09	600	660	1
154	3008	0.0027	0.022	0.3	400	440	1
201	202	0.002	0.025	2	1200	1300	1
201	204	0.003	0.03	2.5	1200	1300	1
203	205	0.005	0.045	0.08	200	250	1
203	205	0.005	0.045	0.08	200	250	1
3001	3003	0	0.008	0	0	0	1
3002	3004	0.006	0.054	0.09	0	0	1
3003	3005	0.006	0.054	0.09	0	0	1
3003	3005	0.006	0.054	0.09	0	0	1
3005	3006	0.0035	0.03	0.07	0	0	1
3005	3007	0.003	0.025	0.06	0	0	1
3005	3008	0.006	0.05	0.12	0	0	1
3007	3008	0.003	0.025	0.06	0	0	1

- Transformer Data

Table A – 3
System Transformer data

FROM	TO	R	X	RATE A	RATE B	RATE C	TRANSFORMER TAP
101	151	0.0003	0.0136	1250	1350	1750	0.9848
102	151	0.0003	0.0136	1250	1350	1750	0.9848
152	153	0	0.005	2500	3000	3500	1
201	211	0.0007	0.0212	800	1000	1120	1.015
202	203	0.0004	0.0162	800	1040	1200	1
204	205	0.0003	0.015	800	1040	1200	1
205	206	0.0003	0.0133	900	1080	1350	1.015
3001	3002	0.0003	0.015	800	1040	1200	1
3001	3011	0.0002	0.01	1300	1560	1820	1.015
3004	3005	0.0004	0.0162	800	1040	1200	1
3008	3018	0.0002	0.085	150	200	250	1.015

PSS/E 'SAVNW' New England Test Case

- Load Data

Table A – 4
System Load data

BUS	P LOAD	Q LOAD	SHUNT G	SHUNT B
101	0	0	0	0
102	0	0	0	0
151	0	0	0	600
152	0	0	0	0
153	660	330	0	0
154	1000	800	0	300
201	0	0	0	300
202	0	0	0	0
203	300	150	0	50
204	0	0	0	0
205	1200	700	0	300
206	0	0	0	0
211	0	0	0	0
3001	0	0	0	0
3002	0	0	0	0
3003	0	0	0	0
3004	0	0	0	0

APPENDIX

B

Table B – 1

Initial Random Parent Population for Capacitor Bank Setting

Cap. Bank 1	**Cap. Bank 2**	**Cap. Bank 3**	**Cap. Bank 4**	**Cap. Bank 5**
677	426	957	948	466
427	479	916	48	791
68	244	430	322	96
376	875	318	57	911
711	456	573	636	992
704	943	540	92	645
464	137	700	609	560
823	166	274	852	953
678	1017	902	250	737
244	782	669	705	960
167	25	846	975	261
424	17	418	608	374
345	171	446	743	613
408	920	445	534	604
425	349	752	118	295
956	364	38	590	784
672	750	985	297	136
870	957	212	912	990
247	933	668	15	408
991	843	434	303	361

Table B – 2

Initial Random Parent Population for Transformer Tap Settings

TAP 1	**TAP 2**	**TAP 3**	**TAP 4**	**TAP 5**	**TAP 6**	**TAP 7**	**TAP 8**	**TAP 9**	**TAP 10**	**TAP 11**
27	22	15	25	12	6	4	29	26	19	22
30	2	13	9	28	27	32	21	13	10	13
5	29	26	17	20	11	2	12	8	25	27
31	31	27	24	19	18	26	17	14	7	18
21	23	7	30	31	6	27	14	4	23	12
4	26	17	32	10	20	29	3	5	7	31
10	25	15	19	25	9	3	8	32	13	29
19	13	22	5	25	22	14	5	32	21	19
32	22	24	5	13	23	9	7	19	26	21
32	6	25	9	19	25	27	8	2	3	20
6	24	10	28	3	15	15	14	8	31	7
33	2	23	9	2	3	31	2	12	26	10
32	10	22	27	18	8	7	30	28	17	16
17	2	6	9	26	31	9	32	2	15	8
27	4	4	31	31	6	5	17	2	15	28
5	28	17	12	5	28	5	17	6	11	7
14	23	32	7	19	18	29	12	22	17	8
31	11	12	9	16	33	20	30	25	17	6
27	32	20	21	2	3	19	13	22	27	8
32	2	8	16	12	15	5	4	15	27	15

Table B – 3

Initial Collapse Proximity Index

Collapse Proximity Index
5.2271
4.2694
3.5837
3.068
2.6654
1.853
1.3551
1.2896
1.1696
1.1143
0.8326

Table B – 4

Capacitor Bank 1 with initial binary population equivalent

Decimal Value	**Binary Equivalent**									
677	1	0	1	0	1	0	0	1	0	1
427	0	1	1	0	1	0	1	0	1	1
68	0	0	0	1	0	0	0	1	0	0
376	0	1	0	1	1	1	1	0	0	0
711	1	0	1	1	0	0	0	1	1	1
704	1	0	1	1	0	0	0	0	0	0
464	0	1	1	1	0	1	0	0	0	0
823	1	1	0	0	1	1	0	1	1	1
678	1	0	1	0	1	0	0	1	1	0
244	0	0	1	1	1	1	0	1	0	0
167	0	0	1	0	1	0	0	1	1	1
424	0	1	1	0	1	0	1	0	0	0
345	0	1	0	1	0	1	1	0	0	1
408	0	1	1	0	0	1	1	0	0	0
425	0	1	1	0	1	0	1	0	0	1
956	1	1	1	0	1	1	1	1	0	0
672	1	0	1	0	1	0	0	0	0	0
870	1	1	0	1	1	0	0	1	1	0
247	0	0	1	1	1	1	0	1	1	1
991	1	1	1	1	0	1	1	1	1	1

Table B – 5

Transformer Tap 1 with initial binary population equivalent

Decimal Value	Binary Equivalent					
27	0	1	1	0	1	1
30	0	1	1	1	1	0
5	0	0	0	1	0	1
31	0	1	1	1	1	1
21	0	1	0	1	0	1
4	0	0	0	1	0	0
10	0	0	1	0	1	0
19	0	1	0	0	1	1
32	1	0	0	0	0	0
32	1	0	0	0	0	0
6	0	0	0	1	1	0
33	1	0	0	0	0	1
32	1	0	0	0	0	0
17	0	1	0	0	0	1
27	0	1	1	0	1	1
5	0	0	0	1	0	1
14	0	0	1	1	1	0
31	0	1	1	1	1	1
27	0	1	1	0	1	1
32	1	0	0	0	0	0

Table B – 6

Crossover Positions for Parent Strings in a repeated Crossover Operation

Crossover Points												
Transformer Tap Population												**Cap. Bank Population**
5	4	3	0	5	2	3	5	2	5	4		20
5	4	3	0	5	2	3	5	2	5	4		20
0	5	5	5	3	4	2	5	4	4	3		0
0	5	5	5	3	4	2	5	4	4	3		0
3	0	2	3	5	1	2	0	3	5	3		14
3	0	2	3	5	1	2	0	3	5	3		14
4	5	5	5	2	5	1	4	3	3	1		34
4	5	5	5	2	5	1	4	3	3	1		34
1	3	2	0	2	5	1	5	3	5	2		4
1	3	2	0	2	5	1	5	3	5	2		4
1	2	5	5	5	5	1	2	4	2	5		18
1	2	5	5	5	5	1	2	4	2	5		18
5	4	0	5	2	4	2	2	5	3	1		23
5	4	0	5	2	4	2	2	5	3	1		23
4	4	2	4	3	0	4	2	5	2	0		45
4	4	2	4	3	0	4	2	5	2	0		45
1	2	3	4	1	2	1	0	2	5	1		49
1	2	3	4	1	2	1	0	2	5	1		49
4	2	5	3	5	5	3	4	2	4	2		48
4	2	5	3	5	5	3	4	2	4	2		48

Table B – 7

Fitness Values for Child strings generated from a repeated Crossover Operation

Population Individual	Fitness Value
1	0.9856
2	0.9853
3	0.9862
4	0.9852
5	0.9861
6	0.9853
7	0.9852
8	0.9856
9	0.987
10	0.9857
11	0.9864
12	0.9857
13	0.9864
14	0.986
15	0.987
16	0.9857
17	0.987
18	0.987
19	0.987
20	0.987

Table B – 8

Best – fit Child Strings

Best-fit Child Strings from Repeated Crossover Operation	
Best - fit Cap. Setting	**Best - fit Tx. Tap Setting**
732	21
265	18
539	7
185	2
741	27
	34
	15
	26
	4
	12
	28

APPENDIX

C

C.1 Overview of PSS/E

Since its introduction in 1976, the Power System Simulator for Engineering (PSS/E) tool has become the most comprehensive, technically advanced, and widely used commercial program of its type. It is widely recognized as the most fully featured, time-tested and best performing commercial program available.

PSS/E is utilized for load Incrementation and identification of the critical operating point based on the Collapse Proximity Index (CPI).

C.2 MATLAB overview

MATLAB is a numerical computing environment and fourth generation programming language. Built around the MATLAB language, MATLAB as an application resembles an advanced version of Object oriented Programming, with similar features of other OOP languages like C++, Java and Python.

MATLAB is extensively employed for scripting the entire Genetic Algorithm program along with inclusion of Power system operations from MATPOWER. With data generated from PSS/E and MATPOWER accumulated into MATLAB workspace, a Genetic Algorithm-based program is scripted to generate the best-fit control action set.

C.3 MATPOWER application

MATPOWER is a package of MATLAB M-files for solving power flow and optimal power flow problems. It is intended as a simulation tool for researchers and educators that is easy to use and modify. MATPOWER is designed to give the best performance possible while keeping the code simple to understand and modify.

MATPOWER is utilized to run power flow solutions and perform two-bus equivalent system reductions for CPI calculations based on updated *Child String* sets generated from Genetic Algorithms. With a number of repetitive child string generations, it is imperative to have an in-built power flow program within the MATLAB environment. MATPOWER is employed along with the GA script to satisfy this purpose.

APPENDIX

D

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                          %
%                                                          %
%    A Genetic Approach to Voltage Collapse Mitigation     %
%                                                          %
%                    Dwarakesh Nallan                      %
%                                                          %
%                                                          %
%                                                          %
% Main Program:                                            %
%                                                          %
%                                                          %
%                                                          %
% This program can be classified into two parts:           %
%                                                          %
%                                                          %
%                                                          %
%      Part 1 - Collapse Proximity Index calculation       %
%               and Collapse Point Identification          %
%                                                          %
%      Part 2 - Application of Genetic Agorithms to        %
%               generate control actions                   %
%                                                          %
%                                                          %
% System and Load Bus:                                     %
%                                                          %
%                                                          %
% The PSS/E 23-Bus 3-area test system is utilized          %
%                                                          %
% Load Bus 153 (Area 1) is taken into consideration        %
%                                                          %
% Function Called:                                         %
%                                                          %
%      'function_bestfit_childstring' called for           %
%      Best-fit Child String Generation                    %
%                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%--------------------------- PART 1 -----------------------%
%---------------------- CPI calculation ------------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear all
```

```
% The P Q load increment for Bus 153
load('savnw_153bus_PQVangle2')

% System Parameters
Vs=1.04;
Vsvect=complex(1.04,0);
R=0.00304;
X=0.04416;
Zthev=complex(0.00304,0.04416);
Zthevmod=abs(Zthev);


% Calculation of Collapse Proximity Index (Table 4.1) for
Bus 153
for i=1:11
    P_i=P(i);
    Q_i=Q(i);

    % Pmax Calculation
    Pmax= ((Q_i*R)/X) - ((Vs^2)*R)/(2*(X^2)) +
(Zthevmod*Vs*((Vs^2)-(4*Q_i*X))^0.5)/(2*(X^2));

    % CPI Calculation
    CPI_1(i)=Pmax/P_i;

    % CPI vs % Loading Plot
    plot(CPI_1)
end
CPI_1;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
----------------------------- PART 2 ----------------------------------------------%
%--------------------- G A Process --------------------------------------------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Critical loading is determined to be 330% beyond which
system becomes unstable.

% Power flow calculation at 330% loading using MATPOWER
Programming

disp('Mat Power Calculation');
[MVAbase,bus_new, gen_new, branch_new] =
runpf('psse_savnw_bus153ld330');
clc
```

```
% ---------------------------------------------------------------%
% Two conrol actions that are Modeled are                        %
|                                                                %
%         - the capacitor bank settings                          %
|                                                                %
%         - transformer tap settings                             %
|                                                                %
% Total number of transformer taps = 11                          %
|                                                                %
% Total number of shunt capacitaors = 5                          %
% ---------------------------------------------------------------%


% ** Step 1 - Initial Population generation and Binary
Encoding **%
% Generating random population for Taps 1 to 11
clear ytap1;
ytap1=ceil(rand(20,11)*33);
for i=1:20
    for j=1:11
        if ytap1(i,j)==1
            ytap1(i,j)=2;
        end

    end
end

tap1bin=zeros(20,11);
clear tap1;
tap1=zeros(20,66);

% Converting into a binary form
for i=1:20
    for j=1:11
        str=dec2bin(ytap1(i,j));
        tap1bin(i,j)=str2num(str);
        aa = tap1bin(i,j)/100000;
        if aa>=1
            tap1(i,(6*(j-1))+1)=1;
        end
        bb = mod(tap1bin(i,j),100000);
        bb1=bb/10000;
        if bb1>=1
            tap1(i,(6*(j-1))+2)=1;
        end
        cc = mod(bb,10000);
        cc1=cc/1000;
```

```
        if cc1>=1
            tap1(i,(6*(j-1))+3)=1;
        end
        dd = mod(cc,1000);
        dd1=dd/100;
        if dd1>=1
            tap1(i,(6*(j-1))+4)=1;
        end
        ee = mod(dd,100);
        ee1=ee/10;
        if ee1>=1
            tap1(i,(6*(j-1))+5)=1;
        end
        ff = mod(ee,10);
        ff1=ff/1;
        if ff1>=1
            tap1(i,(6*(j-1))+6)=1;

       end
    end
end
ytap1;
tap1;

% Generating random population for Capacitor Banks 1 to 5
clear bz;
clear b11;
clear b1;
bz=zeros(20,10);
b11=zeros(20,50);
b1=zeros(20,50);
clear yb1;
yb1=zeros(20,5);

for count=1:5
    count;
    clear bz;
    bz=randn(20,10)<0
    for i=1:20
        x=1;
        for j=((count-1)*10)+1:((count-1)*10)+10
            b1(i,j)=bz(i,11-x);
            x=x+1;
        end
    end
    % Converting into a decimal form
    j=0;
    for i=1:20
        pow=0;
```

```
        for j=1:10

            pow=pow+(bz(i,j)*2^(j-1));
        end
        yb1(i,count)=pow;
    end
end




% ********** Step 2 - Fitness Value Calculation ***********

% Testing capacitor bank and transformer tap new settings -
% - transferred into an excel file and running power flow
using MATPOWER

% Loading and duplicating
load('psse_savnw_data_bus1531d330')


% Run power flow with the new values using MATPOWER
    baseMVA = 100;
    [MVAbase, bus_new_changed_1, gen_new_changed_1,
branch_new_changed_1] = runpf('psse_savnw_bus1531d330_mod');

% CPI calculation
Pmax_reduced = ((3.30*R_reduced)/X_reduced) -
((Vs^2)*R_reduced)/(2*(X_reduced^2)) +
(Z12_red_mod*Vs*((Vs^2)-
(4*3.30*X_reduced))^0.5)/(2*(X_reduced^2));
    CPI_1_reduced(popnum)=Pmax_reduced/6.60;

% Fitness value Calculation
    fitness_list(popnum)=1/(1+(0.01*CPI_1_reduced(popnum)));
    fitness_total=fitness_total+fitness_list(popnum);
end
clc
CPI_1_reduced;
fitness_list;


% **** Step 3 - Selection of best fit parent strings ******

% Roulette Wheel Selection
j=1;
x=0.05;
s=0;
clear sel_1;
```

```
for i=1:20

    s=s+(fitness_list(i)/fitness_total);
    while x>=s
        j=j+1;
        s=s+(fitness_list(i)/fitness_total);
    end
    sel_1(i)=j;
    if sel_1(i)>20
        sel_1(i)=20;
    end
    if sel_1(i)<=0
        sel_1(i)=1;

    end
    s=0;
end
sel_1;

% Selecting the Best fit strings
i1=0;
i2=0;
clear popsel_1;
popsel_1=zeros(1,20);

for q=1:19
    while rand<0.9
        q1=q+1;
        i1=1+round(q*rand);
        i2=1+round(q1*rand);
        popsel_1(q)=i1;
        if popsel_1(q)==0
            popsel_1(q)=popsel_1(q)+1;
        end
        popsel_1(q1)=i2;
        if popsel_1(q1)==0
            popsel_1(q1)=popsel_1(q1)+1;

        end
    end
end
popsel_1;

y=20;
clear popsel1_1;
for i=1:20
    popsel1_1(i)=popsel_1(y);
    if popsel1_1(i)==0
```

```
        popsel1_1(i)=popsel_1(i)+1;

    end
    y=y-1;
end

popsel1_1;
clear sel1_1;
tab1=[sel_1
      popsel1_1];
for i=1:20
    xx=popsel1_1(i);
    if xx>20
        xx=20;
    end
    sel1_1(i)=sel_1(xx);
end
sel1_1;

clear bnew1;

% Generating new 'b' value (Shunt capacitor Settings)
for i=1:20
    yy=sel1_1(i);
    for zz=1:50
        bnew1(i,zz)=b1(yy,zz);
    end
end
b1;
bnew1;
clear tapnew1;

% Generating new 'tap' values (Transoformer Tap Settings)
for i=1:20
    yy=sel1_1(i);
    for zz=1:66

        tapnew1(i,zz)=tap1(yy,zz);
    end
end
tap1;
tapnew1;


% ********** Step 4 - Crossover operation ****************

% Calculating the list of crossover locations for parent
string pairs
i1=1;
```

```
for i=1:10

    while rand<0.9 && rand>0
        i1=1+round(csb*rand);
        if i1>=49
            i1=49;
        end
        csbsingle_1(i)=i1;
    end
end

% Generating Capacitor Bank child strings using crossover
operation
for i=1:2:20
    locb1=csiteb1(i);
    for j=1:locb1
        b1child(i,j)=bnew1(i,j);
        b1child(i+1,j)=bnew1(i+1,j);
    end
    for j=locb1+1:50
        b1child(i,j)=bnew1(i+1,j);
        b1child(i+1,j)=bnew1(i,j);
    end
end

% Calculating the list of crossover locations for parent
string pairs

% - 'csitetap1' gives the position for each parent string
(double of cstapsingle_1)
for col=1:11
    i=1;
    for j=1:10
    csitetap1(i,col)=cstapsingle_1(j,col);
    i=i+1;
    csitetap1(i,col)=cstapsingle_1(j,col);
    i=i+1;
    end
end
csitetap1;

% Generating Tx. tap child strings using crossover operation
for col=1:11
    for i=1:2:20
    loctap1=csitetap1(i,col);
    for j=(6*(col-1))+1:(6*(col-1))+loctap1
        tap1child(i,j)=tapnew1(i,j);
        tap1child(i+1,j)=tapnew1(i+1,j);
    end
```

```
    for j=(6*(col-1))+loctap1+1:(6*col)
        tap1child(i,j)=tapnew1(i+1,j);
        tap1child(i+1,j)=tapnew1(i,j);

    end
    end
end
csitetap1;
tapnew1;
tap1child;
% Decimal conversion for the child strings
% Calculating ybchild
j=0;
yb1child=zeros(20,5);
for count=1:5
    for i=1:20
        pow=0;
        exp=0;
        for j=(10*(count-1))+10:-1:(10*(count-1))+1
            pow=pow+(b1child(i,j)*2^exp);
            exp=exp+1;
        end
        yb1child(i,count)=pow;
    end
end


% ******* Step 5 - Fitness Value for Child Strings ********

% Testing capacitor bank and transformer tap new settings -
% - transferred into an excel file and running power flow
using MATPOWER

% Fitness value Calculation
fitness_list_child(popnum)=1/(1+(0.01*CPI_1_redcued_child(po
pnum)));

fitness_total_child=fitness_total_child+fitness_list_child(p
opnum);

    % - CPI check
    if CPI_1_redcued_child(popnum) > 1.25*CPI_1(10)
        popnum_best=popnum_best+1;
        CPI1child_best(popnum_best,1)=popnum_best;

CPI1child_best(popnum_best,2)=CPI_1_redcued_child(popnum);
```

```
        ytap1child_best(popnum_best,:)=
ytap1child(popnum,:);
        yb1child_best(popnum_best,:)=yb1child(popnum,:);

    else if CPI_1_redcued_child(popnum) >= 1.1667*CPI_1(10)
&& CPI_1_redcued_child(popnum) <= 1.25*CPI_1(10)
            if bus_new_changed_1(:,8) <= 1.09
            popnum_best2=popnum_best2+1;
            CPI1child_best2(popnum_best2,1)=popnum_best2;

CPI1child_best2(popnum_best2,2)=CPI_1_redcued_child(popnum);
            ytap1child_best2(popnum_best2,:)=

ytap1child(popnum,:);

yb1child_best2(popnum_best2,:)=yb1child(popnum,:);
            end
    end
    end
end


% *************** Step 6 - Repeated Crossover ************

% Checking for need for Repeated Crossover operations
if popnum_best > 0
    sort(CPI1child_best,2,'descend');
    CPI1child_best(1,1);
end
if popnum_best2 > 0
        popnum_best2;
end

popnum_best_2=0;
popnum_best2_2=0;

if popnum_best2 == 0
% Calculating the list of crossover locations for parent
string pairs
i1=1;
for i=1:10
    while rand<0.9 && rand>0
        i1=1+round(csb*rand);
        if i1>=49
            i1=49;
        end
        csbsingle_1(i)=i1;
    end
end
```

```
% Generating Capacitor Bank child strings using crossover
operation

for i=1:2:20
    locb1=csiteb1(i);

    for j=1:locb1
        b2child(i,j)=b1child(i,j);
        b2child(i+1,j)=b1child(i+1,j);
    end
    for j=locb1+1:50
        b2child(i,j)=b1child(i+1,j);

    b2child(i+1,j)=b1child(i,j);
    end
end

% Calculating the list of crossover locations for parent
string pairs
for col=1:11
    i=1;
    for j=1:10
    csitetap1(i,col)=cstapsingle_1(j,col);
    i=i+1;
    csitetap1(i,col)=cstapsingle_1(j,col);
    i=i+1;
    end
end
csitetap1;

% Generating Tx. tap child strings using crossover operation
for col=1:11
    for i=1:2:20
    loctap1=csitetap1(i,col);
    for j=(6*(col-1))+1:(6*(col-1))+loctap1
        tap2child(i,j)=tap1child(i,j);
        tap2child(i+1,j)=tap1child(i+1,j);
    end
    for j=(6*(col-1))+loctap1+1:(6*col)
        tap2child(i,j)=tap1child(i+1,j);
        tap2child(i+1,j)=tap1child(i,j);
    end


    end
end
```

```
% *** Step 7 - Fitness Value for Repeated Child Strings ***

% Fitness value Calculation
fitness_list_child(popnum)=1/(1+(0.01*CPI_2_redcued_child(po
pnum)));

fitness_total_child=fitness_total_child+fitness_list_child(p
opnum);

    if CPI_2_redcued_child(popnum) > 1.25*CPI_1(10)
        popnum_best_2=popnum_best_2+1;
        VSI2child_best(popnum_best_2,1)=popnum_best_2;

VSI2child_best(popnum_best_2,2)=CPI_1_redcued_child(popnum);
        ytap2child_best(popnum_best_2,:)=
ytap2child(popnum,:);
        yb2child_best(popnum_best_2,:)=yb2child(popnum,:);


    else if CPI_2_redcued_child(popnum) >= 1.1667*CPI_1(10)
&& CPI_2_redcued_child(popnum) <= 1.25*CPI_1(10)
        if bus_new_changed_1(:,8) <= 1.09
        popnum_best2_2=popnum_best2_2+1;
        VSI2child_best2(popnum_best2_2,1)=popnum_best2_2;
VSI2child_best2(popnum_best2_2,2)=CPI_2_redcued_child(popnum
);
        ytap2child_best2(popnum_best2_2,:)=
ytap2child(popnum,:);
        yb2child_best2(popnum_best2_2,:)=yb2child(popnum,:);
        end
        end
end
end
end
clc

% ******* Step 8 - Best fit Child String Selection ********

popnum_best2_3 = popnum_best2_2;
brk_count = 0;
while popnum_best2_3 == 0
    load('psse_savnw_data_bus1531d330')
    brk_count = brk_count + 1;

    % Calling function for Best - Fit_child string (Step 9)
    [popnum_best2_3, ytap2child_best3, yb2child_best3,
     b3child, tap3child] = function_bestfit_childstring
      (b2child, tap2child, bus, gen, branch, areas);
    b3child = b2child;
```

```
    tap3child = tap2child;
    if brk_count == 7
        popnum_best2_3 = 1;
        ytap2child_best3 = ytap10child;
        yb2child_best3 = yb10child;
    end
    if popnum_best2_3 > 0

        break;
    end
end

% ********** Step 10 - Result Display ********************

% Displaying the final desired outputs
disp ('------------------------------')
disp (' Collapse Proximity Index - Initial ')
disp(' ')
CPI_1
disp(' ')
disp ('------------------------------')


disp ('------------------------------')
disp (' Child Strings generated from the initial Crossover
operation ')
disp(' ')

yb1child
ytap1child

disp ('------------------------------')
disp (' Collapse Proximity Index - FInal ')
CPI_2_redcued_child
disp ('------------------------------')
disp (' BEST-FIT Child Strings generated from the repeated
Crossover operation ')
disp ('Best - fit Tap Setting')
ytap2child_best3
disp (' Best - fit Capacitor Bank Settings ')
yb2child_best3
disp ('------------------------------')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                          %
%  A Genetic Approach to Voltage Collapse Mitigation      %
%                                                          %
%                  Dwarakesh Nallan                        %
%                                                          %
% Sub program:                                        %
%         - Calculates the Best-fit Child String from     %
%           repeated Child String Population               %
%                                                          %
%         - Sends the Best fit Child String Set to Main   %
%           program                                   %
%                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [popnum_best2_3, ytap3child_best2, yb3child_best2,
b3child, tap3child] = function_bestfit_childstring (b2child,
tap2child, bus, gen, branch, areas)

disp('Control passed into function')

Vs = 1.04;
dels = 0;
popnum_best2_3 = 0;

VSI_1_1(10) = 1.1143;
ytap3child_best2 = zeros(1,11);
yb3child_best2 = zeros(1,5);


% ********** Step 9 a - Best-fit Crossover Operation *******

% Generation of Crossover site for Shunt Capacitor parent
strings
% - 'csb' is the max position for crossover
csb=49;
csbsingle_1=zeros(1,10);

% Calculating the list of crossover locations for parent
string pairs
i1=1;
for i=1:10
    while rand<0.9 && rand>0
        i1=1+round(csb*rand);
        if i1>=49
            i1=49;
        end
        csbsingle_1(i)=i1;
```

```
    end
end

% - 'csiteb1' gives the position for each parent string
(double of csbsingle_1)
i=1;
for j=1:10
    csiteb1(i)=csbsingle_1(j);
    i=i+1;
    csiteb1(i)=csbsingle_1(j);
    i=i+1;
end

% Generating Capacitor Bank child strings using crossover
operation
for i=1:2:20
    locb1=csiteb1(i);

    for j=1:locb1
        b3child(i,j)=b2child(i,j);
        b3child(i+1,j)=b2child(i+1,j);
    end
    for j=locb1+1:50
        b3child(i,j)=b2child(i+1,j);
        b3child(i+1,j)=b2child(i,j);
    end
end

% Generation of Crossover site for Tx. Tap parent strings
% - 'csa' is the max position
clear cstapsingle_1
cstap=5;
i1=1;
for col=1:11
% 'col' keeps a count of the col of a.. ie a1, a2...
    for i=1:10
        while rand<0.9 && rand>0
        i1=1+round(cstap*rand);
        if i1>=5
            i1=5;
        end
        cstapsingle_1(i,col)=i1;
        end
    end
end

% Calculating the list of crossover locations for parent
string pairs
```

```
% - 'csitetap1' gives the position for each parent string
(double of cstapsingle_1)
for col=1:11
    i=1;
    for j=1:10
    csitetap1(i,col)=cstapsingle_1(j,col);
    i=i+1;
    csitetap1(i,col)=cstapsingle_1(j,col);

    i=i+1;
    end
end
csitetap1;

% Generating Tx. tap child strings using crossover operation
for col=1:11
    for i=1:2:20
    loctap1=csitetap1(i,col);
    for j=(6*(col-1))+1:(6*(col-1))+loctap1
        tap3child(i,j)=tap2child(i,j);
        tap3child(i+1,j)=tap2child(i+1,j);
    end
    for j=(6*(col-1))+loctap1+1:(6*col)
        tap3child(i,j)=tap2child(i+1,j);
        tap3child(i+1,j)=tap2child(i,j);
    end
    end
end

% Decimal conversion
% Calculating ybchild
j=0;
yb2child=zeros(20,5);
for count=1:5
    for i=1:20
        pow=0;
        exp=0;
        for j=(10*(count-1))+10:-1:(10*(count-1))+1
            pow=pow+(b3child(i,j)*2^exp);
            exp=exp+1;
        end
        yb3child(i,count)=pow;
    end
end
% Calculating ytapchild
j=0;
for count=1:11
    for i=1:20
```

```
        pow=0;
        exp=0;

        for j=(6*(count-1))+6:-1:(6*(count-1))+1
            pow=pow+(tap3child(i,j)*2^exp);
            exp=exp+1;
        end
        ytap3child(i,count)=pow;
    end
end


% ** Step 9 b - Fitness Value for Repeated Child Strings **

% Testing capacitor bank and transformer tap new settings -
% - transferred into an excel file and running power flow
using MATPOWER

% Duplicating
bus_changed=bus;
gen_changed=gen;
branch;
branch_changed=branch;
areas_changed=areas;
fitness_list=zeros(20,1);
fitness_total_child=0;

% Replacing the orignial with new tap and cap. values
for popnum=1:20
    branch_changed(24,9)=1-(ytap3child(popnum,1)/330);
    branch_changed(25,9)=1-(ytap3child(popnum,2)/330);
    for i=3:7
        branch_changed(23+i,9)=1+(ytap3child(1,i)/330);
    end
    branch_changed(31,9)=1-(ytap3child(popnum,8)/330);
    for i=9:11
        branch_changed(23+i,9)=1+(ytap3child(popnum,i)/330);
    end
    bus_changed(3,6)=-yb3child(popnum,1);

    bus_changed(6,6)=yb3child(popnum,2);
    bus_changed(7,6)=yb3child(popnum,3);
    bus_changed(9,6)=yb3child(popnum,4);
    bus_changed(11,6)=yb3child(popnum,5);

        % Copying into new '.mat' file
    delete ('psse_savnw_data_bus1531d330_mod');
```

```
    save psse_savnw_data_bus1531d330_mod bus_changed
gen_changed branch_changed areas_changed;

    % Run power flow with the new values using MATPOWER
    baseMVA = 100;
    [MVAbase, bus_new_changed_1, gen_new_changed_1,
branch_new_changed_1] = runpf('psse_savnw_bus1531d330_mod');

    % CPI calculation
    % - Formulating the 2 bus equivalent system using Krone
Reduction
    [rows_bus, cols_bus]=size(bus_new_changed_1);
    bus_new_changed_1=sortrows(bus_new_changed_1,1);
    for i=1:rows_bus
        comp_matrix(i,1)=bus_new_changed_1(i,1);
        bus_new_changed_1(i,1)=i;
        comp_matrix(i,2)=bus_new_changed_1(i,1);
    end
    [rows_branch, cols_branch]=size(branch_new_changed_1);
    branch_new_changed_2=zeros(rows_branch,11);

branch_new_changed_2(:,1:11)=branch_new_changed_1(:,1:11);
    for i=1:rows_branch
        for j=1:rows_bus
            if branch_new_changed_2(i,1)==comp_matrix(j,1)
                branch_new_changed_2(i,1)=comp_matrix(j,2);
            end
        end
        for j=1:rows_bus
            if branch_new_changed_2(i,2)==comp_matrix(j,1)
                branch_new_changed_2(i,2)=comp_matrix(j,2);

            end
        end
    end
    [Ybus, Yf,
Yt]=makeYbus_psse_savnw_bus1531d330_withnewval(baseMVA,
bus_new_changed_1, branch_new_changed_2);
    Ybus;
    for i=1:rows_bus

        for j=1:rows_bus
            Ybus_1(i,j)=Ybus(i,j);
        end
    end
    Ybus_1;
    Ybus_new=Ybus_1;
    for i=2:rows_bus
```

```
        Ybus_new(1,i)=Ybus_1(22,i);
        Ybus_new(i,1)=Ybus_new(1,i);
    end
    for i=2:rows_bus
        Ybus_new(22,i)=Ybus_1(1,i);
        Ybus_new(i,22)=Ybus_new(22,i);
    end
    for i=1:rows_bus
        Ybus_new(2,i)=Ybus_1(5,i);
        Ybus_new(i,2)=Ybus_new(2,i);
    end
    for i=1:rows_bus
        Ybus_new(5,i)=Ybus_1(2,i);
        Ybus_new(i,5)=Ybus_new(5,i);
    end
    Ybus_new(2,2)=Ybus_1(5,5);
    Ybus_new(5,5)=Ybus_1(2,2);
    Ybus_new(2,5)=Ybus_1(5,2);
    Ybus_new(5,2)=Ybus_1(2,5);
    Ybus_new;

    % Krone reduction
    K=Ybus_new(1:2,1:2);
    L=Ybus_new(1:2,3:23);

    M=Ybus_new(3:23,3:23);
    Ybus_reduced=K-(L*inv(M)*transpose(L));
    Z12_reduced=-(Ybus_reduced(1,2))^-1;
    R_reduced=real(Z12_reduced);
    X_reduced=imag(Z12_reduced);
    Z12_red_mod=abs(Z12_reduced);

    % - CPI Calculation
    Pmax_reduced = ((3.30*R_reduced)/X_reduced) -
((Vs^2)*R_reduced)/(2*(X_reduced^2)) +
(Z12_red_mod*Vs*((Vs^2)-
(4*3.30*X_reduced))^0.5)/(2*(X_reduced^2));
    CPI_2_redcued_child(popnum)=Pmax_reduced/6.60;

    % Fitness value Calculation

fitness_list_child(popnum)=1/(1+(0.01*CPI_2_redcued_child(po
pnum)));
fitness_total_child=fitness_total_child+fitness_list_child(p
opnum);

    if CPI_2_redcued_child(popnum) >= 1.1667*VSI_1_1(10) &&
CPI_2_redcued_child(popnum) <= 1.25*VSI_1_1(10)
```

```
        if bus_new_changed_1(:,8) <= 1.10
        popnum_best2_3=popnum_best2_3 + 1;
        VSI2child_best2(popnum_best2_3,1)=popnum_best2_3;

VSI2child_best2(popnum_best2_3,2)=CPI_2_redcued_child(popnum
);
        ytap3child_best2(popnum_best2_3,:)=
ytap3child(popnum,:);
yb3child_best2(popnum_best2_3,:)=yb3child(popnum,:);
        end
    end
end
clc
end
```

BIBLIOGRAPHY

[1] Anderson, P. M.; Fouad, A. A. “Power System Control and Stability.” *Institute of Electrical and Electronics Engineers.* John Wiley & Sons, Inc., 2003.

[2] Yuri V. Makarov, Viktor I. Reshetov, Vladimir A. Stroev, Nikolai I. Voropai. “Blackouts in North America and Europe: Analysis and Generalization.” *Power Tech, IEEE Russia,* June 2005.

[3] IEEE Power System Relaying Committee, Substation Protection Subcommittee, Working Group K12, “Voltage Collapse Mitigation.” *Report to IEEE Power System Relaying Committee*, Final version, December 1996.

[4] Hill, D. J., Hiskens, I. A., “Load recovery in voltage stability analysis and control.” *Bulk Power System Phenomena III, Voltage Stability, Security and Control*, pp. 579-595, ECC, Davos, August 1994.

[5] CIGRÉ Task Force 38-02-10, “Modeling of Voltage collapse Including Dynamic Phenomena.” 1993.

[6] Lindén, K., Segerqvist, I., “Modeling of Load Devices and Studying Load/System Characteristics.” *Technical report - Chalmers University of Technology*, No. 131L, Sweden, 1993.

[7] Gao, B., Morison, G.K., Kundur, P., Ontario Hydro, Ont. "Voltage stability evaluation using modal analysis." *IEEE Transactions on Power Systems,* Vol. 7, No. 4, November 1992.

[8] Y. Tamura. H. Mori and S. Iwamoto, "Relationship Between Voltage Instability and Multiple Load Flow Solutions in Electric Power Systems." *IEEE Transaction* Vol. PAS-102. No. 5, pp. 1115-1125, May 1983.

[9] P. Kessel. and H. Glavitch, "Estimating the Voltage Stability of a Power System." *IEEE Transaction on Power Delivery*, Vol. PWRD- 1, No. 3. pp. 346-354, July 1986.

[10] Canizares, C.A., De Souza, A.C.Z., Quintana, V.H.Dept. of Electrical & Computer Eng., Waterloo Univ., Ont., "Comparison of performance indices for detection of proximity to voltage collapse." *IEEE Transactions on Power Systems,* Vol. 11, No. 3. pp. 1441-1450, August 1996.

[11] Lof , P.A., Smed, T., Andersson, G., Hill, D.J., "Fast Calculation of a Voltage Stability Index." *IEEE Transactions on Power Systems*, Vol. 7, No. 1, February 1992.

[12] Taylor, C. W., "Power System Voltage Stability." McGraw-Hill, 1994.

[13] Proceedings of Bulk Power System Voltage Phenomena, "Voltage Stability, Security and Control." Davos, Switzerland, Aug. 1994.

[14] Vu, K., Liu, C., C., Taylor, C. W., Jimma, K. "Voltage Instability: Mechanisms and Control Strategies." *Proceedings of IEEE, Special Issue on Nonlinear Phenomena in Power Systems*, pp. 1442-1455, Nov. 1995.

[15] Vu, K., Begovic, M.M., Novosel, D., Saha, M.M., "Use of local measurements to estimate voltage-stability margin." *IEEE Transactions on Power Systems*, Vol. 14, No. 3, August 1999.

[16] Yabe, K., Koda, J., Yoshida, K., Chiang, K.H., Khedkar, P.S., Leonard, D.J., Miller, N.W., "Conceptual designs of AI-based systems for local prediction of voltage collapse." *IEEE Transactions on Power Systems*, Vol. 11, No. 1, February 1996.

[17] Yanfeng Gong, Schulz, N., Guzman, A., Mississippi State Univ., "Synchrophasor-Based Real-Time Voltage Stability Index." *IEEE Power Systems Conference and Exposition*, 2006.

[18] Ken Kato, Chairman - External Network Modeling Task Force, "External Network Modeling – Recent Practical Experience." *IEEE Transactions on Power Systems*, Vol. 9, No. 1, February 1994.

[19] Raymond R. Shoults, William J. Bierck, Jr., "Buffer System Selection of a Steady-State Selection External Equivalent Model for Real-Time Power Flow using an Automated Sensitivity Analysis Procedure." *IEEE Transactions on Power Systems*, Vol. 3, No. 3, August 1988.

[20] Thierry Van Cutsem, "Voltage Instability: Phenomena, Countermeasures, and Analysis Methods." *IEEE Transactions on Power Systems*, Vol. 88, No. 2, February 2000

[21] David E Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning." Addison-Wesley, 1989.

[22] Kothari, D.P., Dhillon, J.S., "Power System Optimization." PHI Learning Pvt. Ltd., 2004.

[23] Momoh, J. A., Dias, L. G., Adapa, R., "Voltage Stability Assessment and Enhancement Using Artificial Neural Networks and Reactive Compensation." *Proceedings of The International Conference on Intelligent Systems Applications to Power System*, Florida, 1996.

[24] PSS/E Version 31, "User Manual", Siemens PTI, 2008

[25] MATLAB Version 4.6, "Full Software Help", Mathworks, 2007

[26] MATPOWER Version 3.2, http://www.pserc.cornell.edu/matpower, 2008