DATA MINING AND GRAPH THEORY FOCUSED SOLUTIONS TO SMART GRID CHALLENGES

BY

SUDIPTA DUTTA

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Doctoral Committee:

Professor Thomas J. Overbye, Chair
Professor Peter W. Sauer
Assistant Professor Alejandro Domínguez-García
Professor David Nicol
Dr. James Weber, PowerWorld Corporation

ABSTRACT

The Smart Grid represents a transition of the power and energy industry into a new era of improved efficiency, reliability, availability, and security, while contributing to economic and environmental health. However, several challenges must be addressed for real-life implementation of Smart Grids. Demonstrating the effectiveness of data mining and graph theory in solving some of these problems is the motivation of this dissertation.

One of the key challenges in taking advantage of what the Smart Grid offers is to extract information from volumes of power system data accumulated by a suite of new sensors and measurement devices. Data presents unprecedented potential of developing better understanding of the underlying system. Handling "data explosion" in power systems and mining it for information is hence a critical challenge, necessitating the development of sophisticated algorithms. To address this need, a particular instance of power system data, namely transient stability data is studied. Generator frequencies in a large power system are analyzed with data mining techniques to extract information such as groups of coherent generators. An effective visualization method based on "spark-lines" is also presented answering a long-time question of how to best display time-varying power system data. Spark-lines are automatically placed on a geographical map of the system employing methods of graph drawing. Developed methods detected abnormal behavior in two generators of the system which was caused by errors in the generators' simulation models that were previously undetected and subsequently corrected. This brings out the power of the developed methodology.

Another important aspect of the Smart Grid is to enable integration of large quantities of renewables such as wind power. This requires installation of large wind farms and in turn availability of advanced methods for designing wind farms. The electrical collector system is the single most important element of a wind farm after the wind turbines, and its optimal design is necessary for optimal wind farm operation. However, there is a need for algorithms to automatically design optimal wind farm collector systems. This represents the second problem addressed in this dissertation. A graph-theoretic approach has been applied to design an optimal wind farm collector system with minimum total trenching length. Clustering techniques have also been found extremely useful in handling specific design constraints. Application of the developed methods generated designs with significantly lower costs compared to an actual real-world wind farm.

The third and final challenge addressed is reliably integrating large quantities of wind power into the system. Inherent problems of variability of wind power can be overcome by developing better wind power forecasting methods and incorporating energy storage units such as batteries. A least squares estimation based short-term wind power forecasting method has been presented. Additionally, methods have been developed to determine optimal storage capacity required and optimal generation commitment for a wind farm with on-site energy storage. Both methods have been found to be extremely sensitive to the statistical properties of wind and load forecast data.

In summary, this work applies tools, techniques, and concepts from the areas of graph theory and data mining to address three critical challenges of real-life implementation of Smart Grids. It is anticipated that the work presented in this dissertation will encourage future research in application of graph theory and data mining to other Smart Grid challenges.

ACKNOWLEDGMENTS

I would like to extend my heartfelt gratitude first and foremost to my Ph.D. advisor, Professor Thomas Overbye, for his continual guidance and support throughout my time at the University of Illinois. I consider myself extremely fortunate to have had the opportunity of working with him and learning from him.

I would also like to thank all of my committee members for their interest, involvement, and encouragement of my work over the past years. Additionally, several people I have worked with both at the University of Illinois and in other places have inspired me, and who I want to thank. Specifically, I would like to extend my sincere gratitude to Professor Peter Sauer, who besides being on my thesis committee has been a mentor to me and with whom I have had the opportunity of teaching ECE 330 at University of Illinois. Professor Anjan Bose of Washington State University, Dr. Ratnesh Sharma at NEC Labs America, Dr. Jinjun Xiong, and Brian Gaucher at IBM Watson Research Center, Professor S. P. Singh, my M.S. advisor at the Institute of Technology, BHU, India are some of the other people who have trained me at different stages of my academic career.

I would like to thank Dr. Lalit Bahl for the Joan and Lalit Bahl fellowship, and financial support from Global Climate and Energy Project (GCEP), Electric Power Research Institute (EPRI) and General Electric (GE) Energy. Thanks to additional financial support from the University of Illinois Ernest A. Reid Fellowship, the Grainger Foundation Power Engineering Award, and Washington State University Summer Doctoral Fellowship (NSF ADVANCE program).

Also thanks to my friends and colleagues at University of Illinois and in particular Kate Rogers Davis, Ray Klump, Angel Aquino-Lugo, Komal Shetye, Saurav Mohapatra, Soobae Kim,

Siming Guo, Chris Recio, Trevor Huchins, Robin Smith, Joyce Mast, and my ECE 333 and 330 students.

Finally, I would like to thank all of my family for their love and support. My parents Jharna and Debabrata Dutta, and my sister Dr. Sudeshna (Dutta) Tapadar have always been tremendous sources of inspiration to me. I am especially grateful to my parents-in-law Swapna and Pradip Basu. Last but not least, thanks to my husband Dr. Anirban Basu for the constant love, support, encouragement, and mentoring at every step of my Ph.D. right from taking my GRE exam to my defense, and without whom this dissertation would not have seen the light of the day.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Motivation

Electrification has been recognized as "the greatest engineering achievement of the 20th century" by the National Academy of Engineering. The present United States electric grid is a gigantic network, consisting of more than 9,200 generation units, 1 million MW of capacity, and 300,000 miles of transmission lines. Although an engineering marvel, electric power requirements of the 21st century have led to increasing complexity of grid management and hence the need to upgrade the existing energy infrastructure to the Smart Grid.

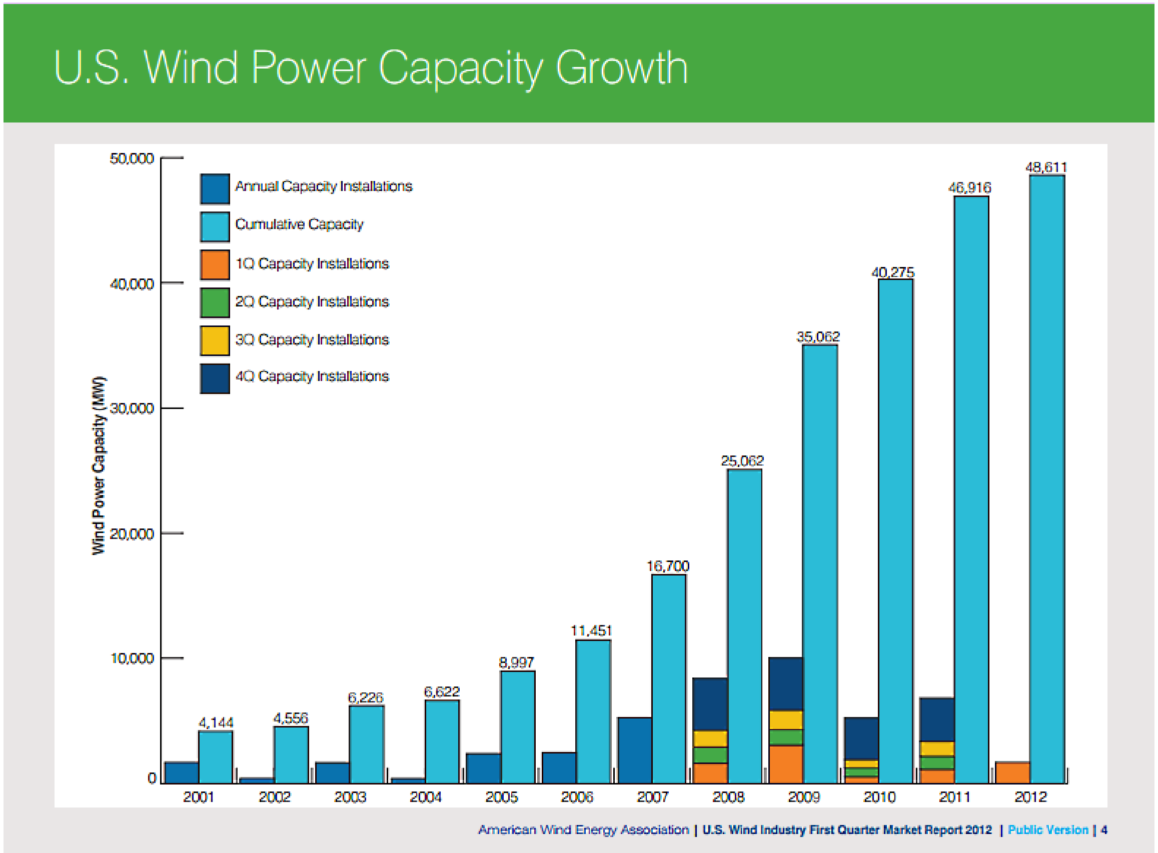
The term Smart Grid encompasses a class of technologies that enable two-way communication technology, enhanced cyber-security, handling large amounts of renewable sources of energy such as wind and solar and even integrating electric vehicles onto the grid. The principal functional characteristics that comprise the foundation of Department of Energy (DOE)'s Smart Grid program are as follows [1]:

- Self-healing from power disturbance events
- Enabling active participation by consumers in demand response
- Operating resiliently against physical and cyber attack
- Providing power quality for 21st century needs
- Accommodating all generation and storage options
- Enabling new products, services, and markets
- Optimizing assets and operating efficiently

The benefits associated with Smart Grids will be realized both at utility and consumer ends with higher energy efficiency and lower energy costs and in the overall system with improved reliability, availability, and adequacy. However, enabling the Smart Grid as envisioned is not an easy problem. Several challenges need to be addressed before the Smart Grid becomes a reality.

For example, one critical challenge is to handle increasing amounts of power system data acquired by the network of new sensors and measurement devices in the Smart Grid. As volumes of such data are becoming gigantic, terms such as "data explosion" are being used more frequently to describe the situation of power systems with respect to data. There is a lot of information embedded in this data which could enable several applications such as intelligent monitoring and control of the system. But the crucial question is: *How is this information extracted?* Another challenge involves integration of renewable sources of energy such as wind. An important Smart Grid functional characteristic is the ability to accommodate all generation and storage options which include wind power. In the United States, installed capacity of wind power has been increasing over the years as shown in Figure 1. The Department of Energy (DOE) has put forward goals of achieving 20% wind power penetration by 2030, a target that would increase the wind power installed in the United States from ~47 GW in 2011 to ~300 GW in 2030 [2, 3]. This steep growth target of ~13 GW/year is accompanied by challenges of large-scale wind power integration from planning to operations which must be addressed as a part of Smart Grid implementation as well.

The objective of this dissertation is to demonstrate how concepts and techniques from data mining and graph theory can be applied to solve some of the Smart Grid challenges. A background on each of the areas of data mining and graph theory will be provided in the next section. Also, the challenges addressed in this dissertation will be discussed in the context of graph-theoretic and data mining methods.



American Wind Energy Association | U.S. Wind Industry First Quarter Market Report 2012 | Public Version | 4

Figure 1: Installed wind power capacity in the United States (Source: AWEA)

1.2 Background

Graph theory

Graph theory is the study of graphs. A graph is an abstract representation of a set of objects called *nodes* or *vertices* in which some pairs of vertices are connected by *branches* or *edges*. A graph is often denoted by an ordered pair G = (V, E) where V is the set of vertices and E is the set of edges. An example of a graph is shown in Figure 2.

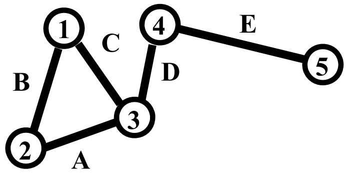


Figure 2: Example of a graph, V = {1, 2, 3, 4, 5}, E = {A, B, C, D, E}

A graph may be *undirected* or *directed* depending on whether there is a direction associated with its edges from one vertex to another. Hence Figure 2 is also an example of an undirected graph. A directed graph involving the same vertices and edges is shown in Figure 3.

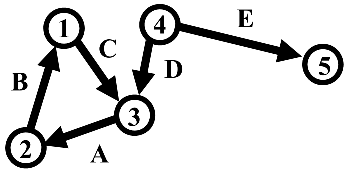


Figure 3: Example of a directed graph

From the graph-theoretic perspective, the power system is a gigantic graph of buses as vertices and transmission lines as edges. In this complex technological network, interconnectivity of thousands of buses and transmission lines ensures the transmission and distribution of electric power from generators to the loads. Because of this analogy, graph theory has found numerous applications in traditional power systems.

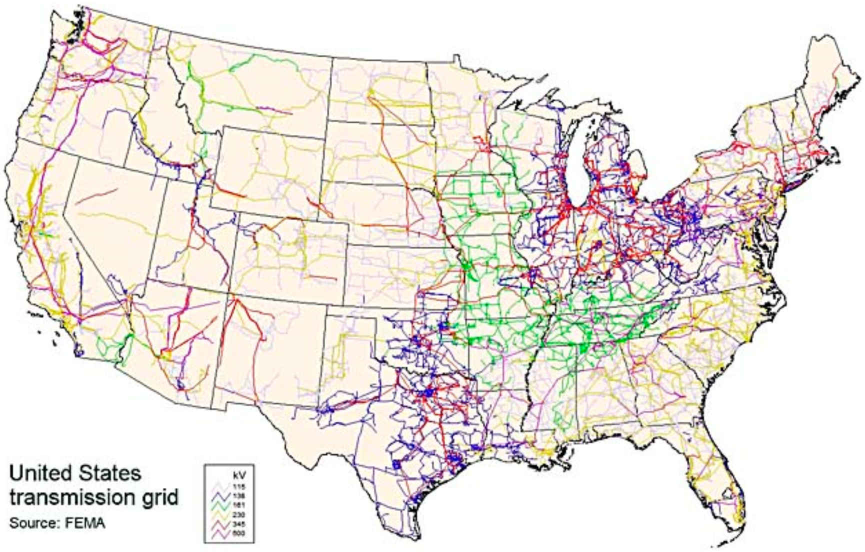


Figure 4: United States transmission grid – a complex network (Source: FEMA)

Graph-theoretic concepts are for instance very useful in topological analysis of power systems. Pai has addressed this topic in his book on application of computer techniques in power system analysis [4]. A power system network with passive elements can be represented by its graph, facilitating the computation of incidence matrices, cutset matrices, Kirchoff's current law (KCL) equations for solving for branch currents, bus impedance and admittance matrices, which in turn are required for several other power system analyses such as power flow. Suppose the graph in Figure 3 belongs to a certain passive power system network. Then in matrix form, the KCL equations for the network can be easily written as:

$$\begin{array}{ccc} & & \text{Lines} \\ & & \begin{array}{ccccc} \text{A} & \text{B} & \text{C} & \text{D} & \text{E} \end{array} \\ \text{Nodes} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \left(\begin{array}{ccccc} 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{array}\right) \end{array} \begin{bmatrix} i_A \\ i_B \\ i_C \\ i_D \\ i_E \end{bmatrix} = 0 \tag{1.1}$$

where the first matrix on the left-hand side is the incidence matrix and the second matrix on the left-hand side is the vector of line currents. Another useful reference on different power system applications of graph theory is a book by Zhu [5]. This book includes graph-theoretic applications for power flow calculations, classical economic power dispatch, security constrained economic dispatch, multi-area system economic dispatch, reactive power optimization and pricing in multi-area environment, hydro-thermal power system operation, power system state estimation, secure economic automatic generation control, automatic contingency selection, distribution network optimization, and optimal load shedding. It is beyond the scope of one dissertation to discuss all applications of graph theory in power systems. So the review provided here is of the works where the same graph-theoretic concepts have been applied as in this dissertation.

An important definition related to graphs is a *tree*, which is a graph with one and only one path between any two vertices. In other words, a graph without any loops is a tree. Hence the graph in Figure 2 is not a tree whereas the one in Figure 5 is a tree. In fact, the tree in Figure 5 is an example of a *spanning tree* since it spans all of the vertices of the graph.

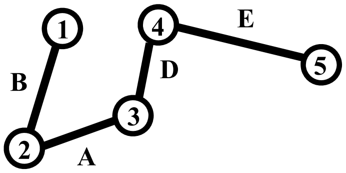


Figure 5: Example of a tree

Given a graph with a set of vertices and edges, there can be several spanning trees. For a *complete graph*, i.e. a graph (V, E) in which every pair of nodes in V is connected by an edge, the number of non-identical spanning trees is V^{V-2} as can be calculated by Cayley's Tree formula [6]. Each edge of a graph can be assigned a *weight* or *cost*, which is a number representing how unfavorable it is and can be used to assign a weight to a spanning tree by computing the sum of the weights of the edges in that spanning tree. A *minimum spanning tree (MST)* is then a spanning tree with weight less than or equal to the weight of every other spanning tree.

Spanning trees have been particularly useful in observability analysis of power system networks. A network is said to be observable if it is possible for the state estimator on it to determine the bus voltage magnitudes and angles throughout the entire network from the installed measurements. One way to determine observability of a system is by using a topological approach [7, 8, 9, 10]. A key paper in this area is by Krumpholz et al. [7] where authors have shown that a power system is topologically observable with respect to a measurement set consisting of a voltage measurement and pairs of P, Q measurements if and only if there exists a spanning tree of the system of full rank. Mori and Tzuzuki [11] proposed a minimum spanning tree based method for topological observability analysis. System observability in turn drives state estimator issues such as measurement placement. Wu's dissertation [12] describes a heuristic-based algorithm for branch flow measurement placement such that the measurements form a spanning tree. For an n-bus network, the spanning tree consists of n-1 branches. Therefore, at least n-1 branch power flow measurements are needed to make the network observable. RTUs which are capable of obtaining all measurements from a single substation are placed strategically, in a "concentrated" measurement placement scheme so that fewer RTUs are needed. Lei's thesis [13] addresses state estimator issue using graph theory and proposes a concept of contingency observability graph (COG) advancing the classical topological observability analysis, and it is proven that a power system network maintains its observability under a contingency if and only if its COG satisfies some conditions.

Another area of application of spanning trees in power system is in power distribution systems. An electrical network at the distribution level is composed of hundreds of nodes, most of which correspond to power delivery points or load points, and hundreds of branches, most of which correspond to electrical cables. The other nodes correspond to connecting points and the other branches correspond to switching busbars. In normal operation, each load point or connecting point is connected to a power delivery point through a single path. Thus, the network, when in operation, is radial and connected, i.e., the network is a spanning tree as shown in Figure 6. One of the earliest and a key paper is by Merlin and Black [14] who have presented a spanning tree based method for reconfiguring a distribution network to minimize losses. More recently several other researchers [15, 16] have also employed spanning trees and minimum spanning trees in the context of distribution system reconfiguration, planning, and design.

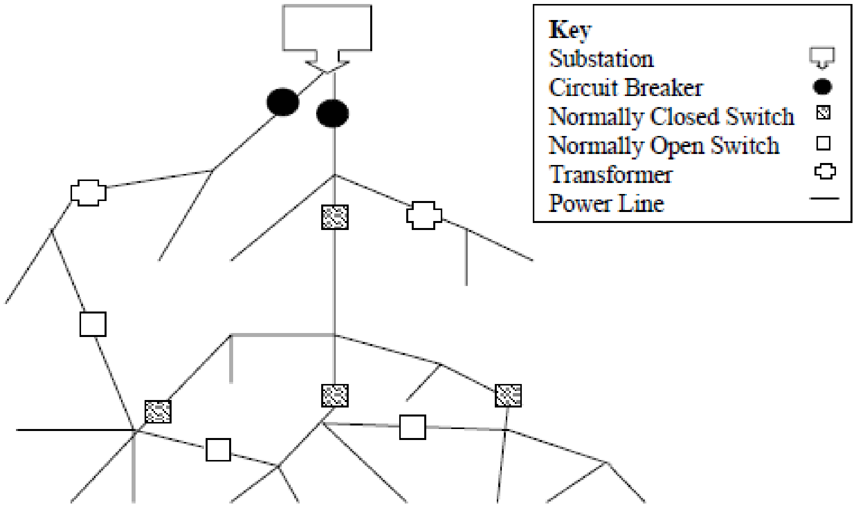


Figure 6: A radial distribution network [Source: 17]

Distribution system restoration aimed at restoring loads after a fault by altering the topological structure of the distribution network by changing open/closed states of some tie switches and sectionalizing switches in the distribution system also use spanning tree based algorithms to find candidate restoration strategies. A graph-theoretic distribution system restoration strategy that maximizes the amount of load to be restored and minimizes the number of switching operations is presented in Li's dissertation [18]. Spanning trees have also been addressed for reconfiguring shipboard power systems (SPS) [19] which supply energy to electric equipment on ships. It is critical for the system to be reconfigurable for the purpose of survivability and reliability.

Now consider a wind farm with wind turbines distributed over a geographical region and a substation which consolidates power generated by the turbines and transmits to the grid. Suppose that the wind turbines and the substation are already placed based on wind patterns, proximity to the transmission grid etc., and the question is: *How should the electrical cables which connect the wind turbines to the substation be laid out in an optimal way?* Thus the problem is that of optimal design of the wind farm collector system and is an important part of wind farm design since optimal operation of wind farms depends on it. The difficulty of a collector system design project is that given the wind turbine locations and the substation location, depending on the dimension of the wind farm, there may be thousands of feasible layout configurations to choose from. Selecting an optimal design from these choices can be a challenging task. In addition there are several design constraints to be taken into account. Hence an optimal design method is a critical need. In addition, it is necessary to automate the design process. This is a comparatively new area with a dearth of available research work. In addressing this design issue, a very interesting connection has been noticed between the wind farm collector system and *graph theory*. The problem of cable layout design for a wind farm collector system can be considered as finding a tree to meet required design characteristics in a graph G = (V, E), where V represents the set of vertices which are the locations of wind turbines and the substation, and E represents the set of branches connecting the vertices which are the connecting cables. In a wind farm with hundreds of turbines, the difficulty is in finding the optimal tree to attain a desired objective since performing an exhaustive search becomes computationally expensive. If the objective is to minimize the total cable length and hence to find a tree in the graph with minimum total length of edges, the problem directly transforms itself to finding the minimum spanning tree. Hence the challenge of wind farm collector system design presents a very interesting application of graph theory and is one of the topics considered in this dissertation.

If additional intermediate vertices and edges can be added to a graph in order to reduce the length of its spanning tree, then the resulting tree becomes a *Steiner tree* [20]. These new vertices introduced to decrease the total length of the connection are known as *Steiner points* or *Steiner vertices*. The minimum Steiner tree problem [21] comes under combinatorial optimization and the main difficulty is to place Steiner points in order to get the shortest interconnect. Hence, the problem is superficially similar to the minimum spanning tree problem which interconnects a given set of points or vertices by a network or graph of shortest total weight of edges. In fact, a minimum spanning tree is a feasible but not usually optimal solution to the Steiner tree problem.

For the Euclidean Steiner tree problem, with the geometric distance between vertices being the weight of the edges, Steiner points must have a degree of three, and the three edges incident to such a point must form three 120 degree angles. It follows that the maximum number of Steiner points that a Steiner tree can have is $N - 2$, where N is the initial number of given points. For $N = 3$, solution is given by a Steiner point located at the Fermat point of the triangle formed by the given points. Figure 7 shows the Steiner points for groups of three and four points respectively.

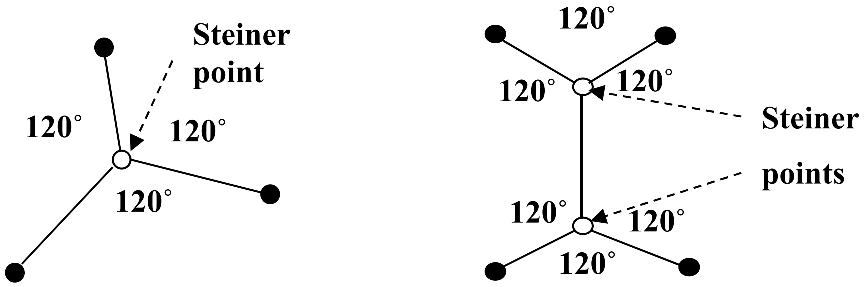


Figure 7: Steiner points for groups of three and four points

Steiner trees have been studied in power systems for routing transmission lines with minimum cost over a terrain divided into equal cost regions with the exception of certain impassable regions, such as lakes and forests, which are considered to have infinite cost by Coulston and Weisshach [22]. Miguez et al. [23] have addressed Steiner trees for optimal design of medium voltage distribution systems. A branch-exchange technique is applied to first obtain a spanning tree. This is followed by applying a heuristic Euclidean Steiner tree algorithm to improve the spanning tree design.

From the perspective of the wind farm collector system design, there is definitely the additional degree of freedom of allowing the creation of intermediate splice nodes similar to Steiner's vertices to reduce the total length obtained by the minimum spanning tree algorithm. Hence it is imperative to explore the applicability of Steiner trees in addressing this challenge and will be discussed in more detail in Chapter 3.

Graph theory also has its applications in *power system visualization*. One example is auto-generation of one-line diagrams of power systems. Drawing one-line diagrams of large power systems is a cumbersome process if done manually. Hence there is a need to automate the process. Song et al. [24] have proposed three algorithms for automatic generation of power system one-line diagrams. These algorithms use spring-embedder or force-based graph drawing for placing the buses or breakers. In [25], authors addressed a meshed system using a modified version of the Controlled Spring Embedder algorithm enhanced by the usage of physical laws and geospatial data. These applications mainly utilize the analogy of buses and transmission lines on a power system one-line with vertices and edges of a graph, and the question addressed is: *How should the buses and transmission lines be placed automatically on a display area in a way that is aesthetic with improved readability and comprehension?* Graph layout problems are a particular class of combinatorial optimization problems in which the objective is to find a layout of an input graph in such way that a certain objective cost is optimized. Several problems in a variety of areas such as network optimization, VLSI circuit design, etc. can be formulated as graph layout problems. Most of the graph layout problems are NP-complete, but, in many of their applications, feasible solutions with an "almost" optimal cost are sufficient and, thus, approximation algorithms or effective heuristics can be used.

A "good" tree drawing is aesthetically pleasing, compact, nodes are uniformly distributed, and edges between nodes do not cross. The force-based graph drawing method is a well-known method that uses laws of physics to determine an optimal configuration [26, 27, 28, 29]. In this method, it is not guaranteed that the graph is aesthetically pleasing, but a "close enough" drawing can be produced. The nodes are replaced by charged rings that repel each other and branches connecting nodes are replaced by springs. The charges make the nodes repel each other. The magnitude of the repulsion is determined by Coulomb's law which states that the force applied to a point charge by another point charge follows an inverse square law. Hooke's law can tether the nodes together to preserve the compactness criteria. The magnitude of this force is directly linear with respect to the distance between the nodes.

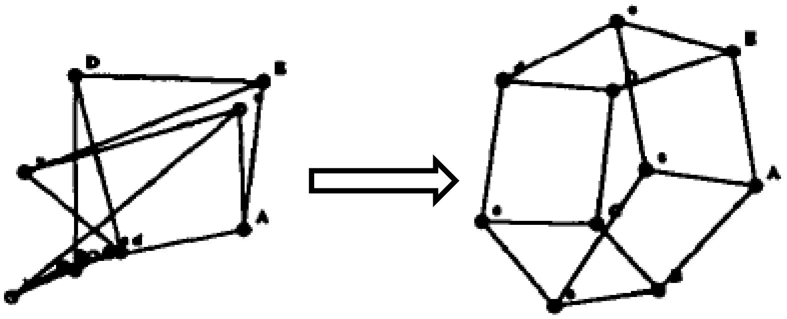


Figure 8: Force-based graph drawing (Source: [29])

Figure 8 demonstrates the application of the force-based graph drawing technique.

The force-based graph drawing technique can potentially be applied to other power system visualization applications as well. One of them is to spread out generators, transmission lines, and in general power system components which are geographically very close on a GIS map of the system. For example, displaying power system components of a substation using only the latitude longitude information will cause all of these to overlap because of physical proximity. Force-based graph layout methods can be used to attain a non-overlapping visualization in a pseudo-GIS display. The force-based graph drawing technique has been used in this dissertation once more for power system visualization, although not the traditional application of one-line diagram generation. Rather, the technique has been used to place miniature plots for visualizing transient stability data on a map and will be discussed in Chapter 2.

Data mining

Data mining is the process of extracting information from large data sets by utilizing methods from artificial intelligence, machine learning, statistics, and database systems. A fundamental idea is to discover patterns in the data and in general involves the following [30]:

- *Anomaly detection* (Outlier/change/deviation detection) which is the identification of unusual data records that might be "interesting" or contain data errors.
- *Association rule learning* or dependency modeling which finds relationships between variables.
- *Clustering* which groups "similar" objects.
- *Classification* which generalizes known structure to apply to new data.
- *Regression* that finds a function which models available data with the least error.
- *Summarization* which provides a compact representation of the data set, including visualization and report generation.

Data mining has been applied to several power system applications. Saleh and Laughton [31] described the application of clustering to decompose power networks based on load flow analysis in 1985. Use of the more formal term, namely "data mining" in power system domain, can be noted in the 1997 paper by S. Madan et al. [32] which brainstorms data mining applications in power systems. Ideas mentioned include application of data mining algorithms for classifying power system states as normal, alert, emergency, or restorative. Other ideas mentioned were developing decision trees to classify a power system as stable or unstable, discovering change of data values from previously stored ones to detect unusual patterns, load forecasting, and diagnostic expert systems for contingency analysis. Another paper published in the same year by Steel et al. [33] mentions two applications of data mining in power systems, the analysis of energy pooling and settlement data, and condition monitoring of power plants. Following this, several researchers have applied data mining in different applications. Asheibi et al. [34] have applied clustering techniques to identify classes of harmonic data from medium and low voltage (MV/LV) distribution systems. The obtained clusters are merged into super-groups by applying further data mining techniques. The ultimate goal is to find the correlation between the patterns of harmonic currents and voltages at different sites (substation, residential, commercial, and industrial) for the interconnected super-groups. Rogers and Overbye [35] have addressed identification of “load pockets” in a congested power system by applying clustering. Groups of generators are identified which gain ability to increase revenue without increasing dispatch and hence have the potential for market advantage. A paper by Mori [36] presents an overview of data mining papers in power systems.

While data mining techniques have been applied in different power system applications, these methods have gained a renewed interest in the context of Smart Grids since integration of data and information systems is one of the key advantages of the Smart Grid [37]. Over the last 10-15 years, there is an increasing trend of introduction of new sensors which is anticipated to continue in coming years [38]. There are currently several hundred Phasor Measurement Units (PMUs) already deployed across the North American grid with plans to deploy many more [39]. In the Western Interconnect, there are currently 137 PMUs installed and plans to increase that number to over 300 by end of 2012. In the Eastern Interconnect, 60 PMUs have been installed and 8 Phasor Data Connectors (PDC) have been deployed to aggregate this data. These PDCs stream their data to a super PDC at the Tennessee Valley Authority (TVA). To cover the grid adequately, it is projected that at least one third of the bulk power systems locations should be monitored by PMUs, ultimately requiring thousands of PMUs to be installed and resources for processing billions of data samples per day. Hence, one critical challenge is to handle increasing amounts of data acquired by the network of new sensors and measurement devices. As volumes of such data are becoming gigantic, terms such as “data explosion” are being used more frequently to describe the situation of power systems with respect to data. All of this data could be utilized for improved performance of the power system. However, this will require proper interpretation of the data. Analytical methods, based on the advanced ideas of statistical signal processing, pattern recognition, and intelligent controls, will increasingly become imperative. Hence there is a critical need for development of fast, robust, and intelligent algorithms that can extract important patterns and information in power system data and take advantage of what the Smart Grid offers. In looking for possible ways to address these questions, *data mining* comes up as a natural choice. These ideas have also been discussed in Rogers' dissertation [40] which addresses data mining of power system data. She has examined data mining and advanced data analysis techniques in the context of a number of specific power system applications. In particular, these applications concern the use of model data (sensitivities) to identify relationships, the data-enhanced estimation of network models, event identification from oscillation monitoring data, and dealing with the challenges of real-world data and data quality.

A specific type of data of interest in power systems is transient stability data collected over a period of time in a wide area system subject to certain disturbances. For example, consider a 16000 bus power system with 2400 generators. If there is a large disturbance in the system, frequencies in the system will oscillate and depending on the size and type of the system, a propagation of these disturbances will be noticed from one area of the system to the other. Given a data sheet with voltage and frequency measurements at all of these 16000 buses over the duration of disturbance, a pertinent question is: *What useful information can be obtained about the system just by analyzing this data?* Another question is: *How can one visualize all of this data and information embedded in it at a glance?* Since transient stability data is time-varying, there is an added challenge of showing not just the value of a variable or state at a single time point but rather to capture the variation. Typically such variations are shown by strip-charts and multi-color plots, but that leads to loss of information about the geographical location of the data source. From power system perspective, locational information is important since it helps understand how one area of the power system behaves compared to another. This is a topic that could benefit significantly by applying data mining techniques and is one of the challenges addressed in this dissertation. As will be described in Chapter 2, clustering techniques have been found to be very useful in solving this problem.

Cluster analysis is a major part of data mining. The goal of clustering algorithms is to group objects based on the information that describes them. Clusters can be considered classes, to which the clustering process automatically assigns objects. Clustering is often referred to as "unsupervised learning" since it does not require class labels to be known ahead of time. Several clustering algorithms are available in data mining literature [41]. Figure 9 shows a group of objects and one possible way of clustering them. It is important to note that there is no single way to cluster objects since the same set of objects can be classified differently according to different attributes. So selection of the appropriate clustering algorithm for a specific application is crucial.

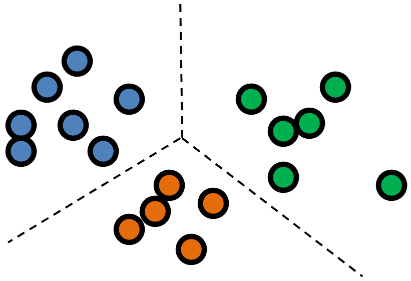


Figure 9: Objects grouped in three clusters

Clustering is a powerful tool given that it can identify groups of similar attributes or similar patterns. If there is any unusual pattern or object, it will usually be unclustered, i.e. will form a single-object cluster. Such unusual behavior could be a method for flagging down "interesting" objects and patterns. In this dissertation, this capability of clustering has been utilized in identifying an error in a simulation model of a wide area power system.

Estimation is another important technique of data mining that has been used extensively in power systems. State estimation [42, 43], which is the process of estimating power system states using real and reactive power and voltage measurements is perhaps the most popular example of this. Another common application is in estimation of parameters of power system devices and machines such as transformers, transmission lines etc. [44, 45, 46, 47, 48]. Among estimation techniques, least squares estimation is a common technique in many areas of science and engineering. This method allows computation of an approximate solution to an overdetermined system, i.e., a system with the number of equations exceeding the number of unknowns. The "least squares" solution minimizes the sum of the squares of the residuals of every equation, a residual being the difference between an observed value and the fitted value provided by a model. The most important application of least squares is in data fitting and is a simple and practical tool with numerous potential uses in power systems.

To illustrate the concept of least squares estimation, consider a set of equations expressed in matrix form where $\mathbf{x}$ denotes the vector of estimated quantities, $\mathbf{r}$ denotes residuals:

$$\mathbf{Ax} = \mathbf{b} \tag{1.2}$$

$$\mathbf{r} = \mathbf{Ax} - \mathbf{b} \tag{1.3}$$

The least squares estimation minimizes the sum of the squared residuals and results in the following estimate:

$$\mathbf{x} = \ [\mathbf{A}^{\mathbf{T}}\mathbf{A}]^{-\mathbf{1}}\mathbf{A}^{\mathbf{T}}\mathbf{b} \tag{1.4}$$

When measurements have different weights, the weighted sum of the squares of residuals is minimized instead. Thus the objective function to be minimized is formulated as follows:

$$S = \sum_{i=1}^{n} w_{ii} r_i^2 \tag{1.5}$$

Then, partial derivatives with respect to each variable to be estimated are computed. If there are k variables to be estimated, this yields k equations, the solution of which yields the k unknowns. In matrix form, the result is:

$$\mathbf{x} = [\mathbf{A}^{\mathbf{T}}\mathbf{W}\mathbf{A}]^{-\mathbf{1}}\mathbf{A}^{\mathbf{T}}\mathbf{W}\mathbf{b} \tag{1.6}$$

The weighted least squares (WLS) state estimation is an application of this method where different voltage and power measurements are assigned weights depending on the quality of the data and knowledge of the quality of devices acquiring these measurements [49, 50].

The least squares regression has also been used for forecasting power demand. A review of research works in this area has been presented by Alfares and Nazeeruddin [51]. It has been noted that several load forecasting methods involve a basic or some variant of least squares. Mbamalu and El-Hawary [52] used the following load model for applying this analysis:

$$Y_t = v_t a_t + \varepsilon_t \tag{1.7}$$

where t is the sampling time, Y_t is measured system total load, v_t is the vector of adapted variables such as time, temperature, light intensity, wind speed, humidity, day type (workday, weekend), etc., a_t is the transposed vector of regression coefficients, and $\mathcal{E}_t$ is the model error at time t. As apparent, least squares is a suitable technique for finding the regression coefficients.

In this dissertation least square estimation technique has been applied to wind power forecasting applications. A critical deterrent in the integration of large amounts of wind power with existing power grids is the inherent variability and uncontrollability of wind which can affect the reliability of the entire power system. These problems can be mitigated by developing better forecasting methods for wind. The applicability of least squares estimation technique to this topic had been addressed in Chapter 4.

Statistical properties of data can be used for extracting information about a system. Statistical methods of pattern recognition are reviewed in [53]. Hence statistical techniques are also data mining techniques. Statistical analysis of data starts by computing *averages* or *means* and *variances*. The mean value of a series of measurements is an expected value of that quantity as a random variable. The expected value of a random variable x is:

$$E(x) = \sum_{i} x_i \, p_i \tag{1.8}$$

where p_i is the probability that the random variable x takes on each value x_i. In addition to its mean, another important aspect of a distribution is the variance, which is a measure of dispersion of the data around the mean. This is the expected value of the square of the deviation of x from its mean. The variance is computed as follows:

$$V(x) = E\big(x - E(x)\big)^2 \tag{1.9}$$

The *standard deviation* is the square root of the variance and has the same units as x.

$$\sigma_x = \sqrt{V(x)} \tag{1.10}$$

The variance is also the *second moment*, and the average is the *first moment*. In general, other moments of the data can be found. The third moment is related to the “skewness” of the distribution. If all of the moments of a distribution are calculated, they provide a complete characterization.

The normal or Gaussian distribution, prominent in statistical theory and in practice, is completely characterized by its mean and variance.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right\} \tag{1.11}$$

The standard deviation σ is the distance from the location of the mean μ to the point of inflection on the distribution curve. The normal distribution is especially important as a result of the central limit theorem which states that under certain conditions, as sample size increases, the sample mean becomes normally distributed about the population mean.

The importance of studying statistical distributions comes out when dealing with probabilistic power system quantities such as load and wind forecasts. No matter how good forecasts are, there is a certain amount of uncertainty involved. Such uncertainties are crucial and should be incorporated in problem formulation. The impact of these forecast uncertainties are noted in Chapter 4 where methods are developed for coordinating energy storage with wind farm operation and optimally sizing storage units.

1.3 Dissertation overview

To summarize, this dissertation examines the application of graph-theoretic concepts and data mining techniques to address three significant challenges of the Smart Grid.

The challenge of information extraction from transient stability data by identifying patterns in the data and its visualization is addressed in Chapter 2. In essence this application captures what data mining embodies. However, not just data mining techniques, but graph drawing techniques have also been applied in the overall methodology. The procedure developed in this chapter is extremely relevant in the Smart Grid framework where there is an increasing amount of data being collected through sensors and other measurement devices.

Automatic and optimal design of wind farm collector systems for optimal operation of large-scale wind farms has been addressed in Chapter 3. This work is applicable in the planning and installation stages of wind farms and particularly relevant in the current scenario with increasing focus on integration of renewable energy such as wind power in the power grids. Clustering and graph-theoretic algorithms were found to be very useful once more in generating these designs and handling the design constraints.

Chapter 4 deals with large-scale integration of wind power and associated problems due to variability of wind. Strategies such as better wind power forecasting and integrating energy storage units such as on-site batteries with wind farms to “firm” wind power are addressed. A least squares based wind power forecasting method is presented. Also, methods are proposed to compute optimal size of a storage unit required for a wind farm to reliably meet a load and to compute optimal charge-discharge schedules for storage units in coordination with wind farm operation and determining maximum steady generation commitment that can be scheduled to be met by the combined plant. Results indicate the importance of incorporating statistical properties of load and wind forecast data in problem formulation.

Finally, conclusions and suggestions for moving forward are presented in Chapter 5. Many other Smart Grid and in general power system applications are in a position to benefit from the graph-theoretic and data mining based methods, techniques, and algorithms presented in this dissertation.

2. INFORMATION PROCESSING AND VISUALIZATION OF POWER SYSTEM TIME-VARYING DATA

This chapter presents application of data mining techniques to power system data for extracting useful information embedded in the data and presenting the information in a visually effective manner. Hence the work described in this chapter captures the true essence of data mining. The type of power system data considered is transient stability data collected over a period of time in a wide area system subject to certain disturbances. However, the analytical methods presented are also applicable to other time-varying data of power systems. There are two major contributions of the work presented in this chapter; first, methods for extracting information from the data, and second, effective visualization of the extracted information. The work presented here is anticipated to stimulate application of similar techniques and analysis on other types of power system data, thereby taking advantage of volumes of power system data collected through a network of sensors and measurement devices in the current electric grid and also in the future Smart Grid.

2.1 Motivation

In the following subsections, the motivation for this work has been detailed.

Information extraction from time-varying data

Identifying useful information from time-varying data is the first and most crucial part of this work. Consider that transient stability data is available for a wide area power system. As has been discussed in Chapter 1, this data contains a lot of information about the underlying system embedded in it. To extract this information, there is a need for developing methods for analyzing the system wide response, identify distinct patterns that characterize the overall system response and hence, help develop better understanding of the overall system. If there are any abnormal responses, these need to be detected from the set of distinct responses, since these could potentially be indicators of errors or system conditions that require attention. Considering the volume of data, all of this analysis must be done automatically. Hence there is a need for developing dedicated algorithms and methodologies for answering the question: *How does one extract information by analyzing the data?* Addressing this need is the motivation of this work.

Transient stability data from a large 16000-bus power system under the influence of disturbances has been collected. The main focus has been on the frequencies of the 2400 generators in the system. It is well known that machines in a power system can be grouped into areas based on coherency [54, 55, 56, 57, 58]. Hence the goal has been to find whether algorithms based on data mining techniques can be developed to automatically identify groups of generators with similar dynamic response and intelligently identify abnormalities such as errors in simulation models.

While data mining and clustering have been applied in power systems research for different applications such as stability studies, monitoring operating conditions, fast transient stability assessment [59, 60, 61, 62], most works consider overall stability of the system rather than inspecting the behavior of states at individual nodes. Analyzing the dynamic response is hence an important distinguishing feature of this work.

It is important to note here that an issue that comes up in analyses such as these is large data volume [63]. In fact, issues of large data volumes are being encountered in recent times more and more frequently not just in the power area but other areas as well. This has led to several research works addressing data volume and dimension reduction techniques [59, 64, 65, 66]. Data structures such as *k*-D trees can index data and speed up data processing algorithms [67]. Another method for reducing data volume proposed in the literature involves data clustering [68, 69]. In this work a K-means clustering based method for data volume reduction has been presented to show the effectiveness of clustering even in handling large data volume.

Visualization of power system time-varying data

Before considering an assessment of how time-varying information should be displayed, it is important to first briefly discuss the nature of this variation. The variation of the power system information of interest here, such as transmission line flows and bus voltage values, can be considered to be divided into four categories. In the first category are the small, zero-average, seemingly random fluctuations which are caused primarily by the switching of myriads of individual loads. These variations occur with time scales on the order of seconds to perhaps minutes and have magnitudes on the order of a few percent of the underlying load. In the second category are the slower changes driven by the diurnal, weekly, and seasonal variations in the electric load. Also in this category are the changes caused by re-dispatching of generation. Sustained rate of change on the order of several percent per minute are typical. The third and fourth categories are the changes caused by large-scale system disturbance, such as the loss of a high voltage transmission line or the tripping of a large generator. The third category contains the dynamic response of the system following such a disturbance. Frequency analysis of such disturbances [70], [71] indicates essentially all modes are faster than 0.2 Hz. Following such a disturbance the system usually returns to its new quasi-steady state operating point within a few seconds. The fourth category is then the discrete changes in these quasi-steady state values that occur following such a disturbance. For example, following a line contingency (after the oscillations have damped out within seconds) a line flow might change from 75% to 125%, and a voltage might fall from 0.98 to 0.9 p.u. The net result from a visualization perspective of these different categories of variation is that in order to fully assess the significance of the value variation it is important to show its full time signal over a reasonable period, as opposed to just showing the value and say its value one time interval prior.

Traditionally the time-variation in system information has been shown using strip-chart recorders. For example, banks of paper strip-charts are very prominent on the left sides of Figure 10 and Figure 11, while electronic strip-charts are shown on the left side of Figure 12. Clearly strip-charts have proven to be quite useful both in real-time power system operations and in post-event analysis; they are a technology that is here to stay.

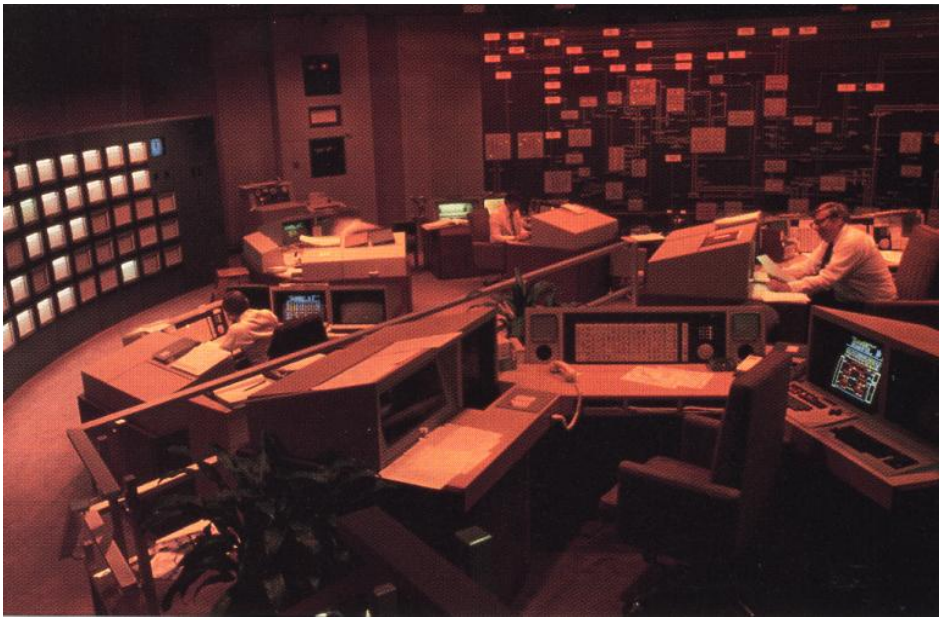


Figure 10: PSE&G control center in 1988 (Source: Figure 1 of [72])

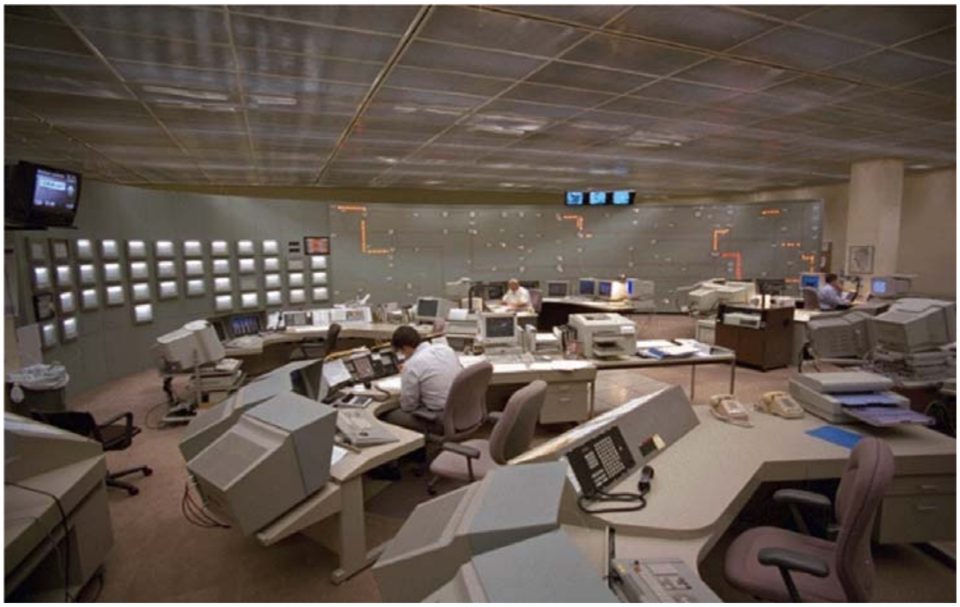


Figure 11: Commonwealth Edison control center in the late 1990s (Source: [73])

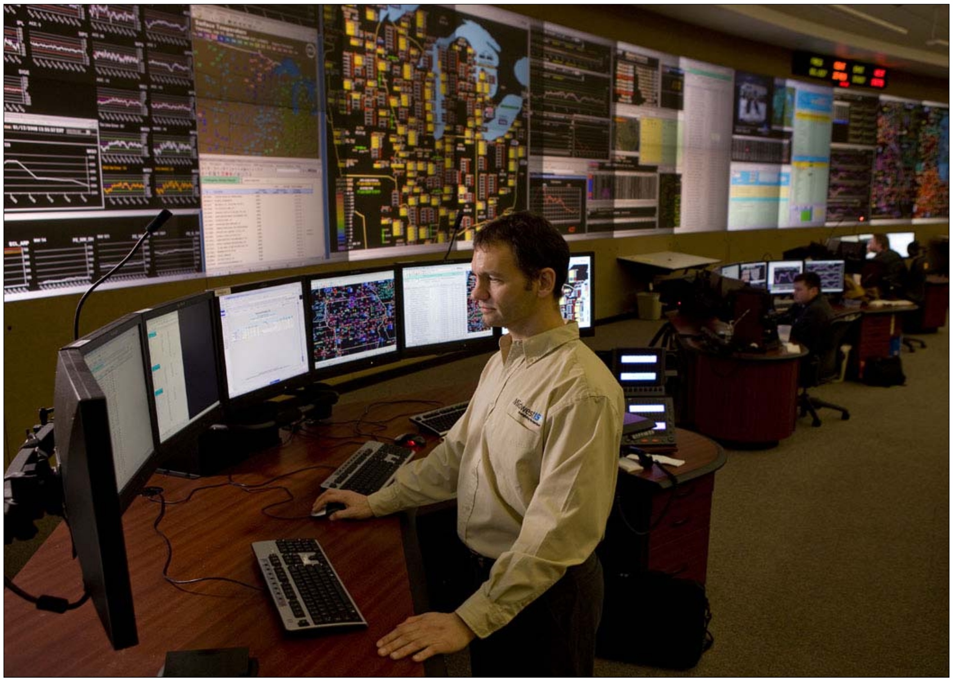


Figure 12: MISO control center in September 2009 (Source: MISO)

But traditional 2-D strip-charts have significant disadvantages in that they cannot be used to show large numbers of data points, and they cannot be used to show geographically distributed information. Of course these limitations are not significant with respect to the display of crucial system overview information like frequency and total load or generation. Multiple curves can be shown on a single display using different colors, but this becomes ineffective as the number of colors becomes higher than ten [74, p. 125].

Several solutions have been proposed to address this issue. In a technique similar to weather radar sequences, time-sequence animations can be used to show how a particular display has changed over a user-specified time period. The use of this trend playback for power system visualization is mentioned in [75], and is also a function available on the ISO-NE developed Adobe Flash display shown in Figure 13.

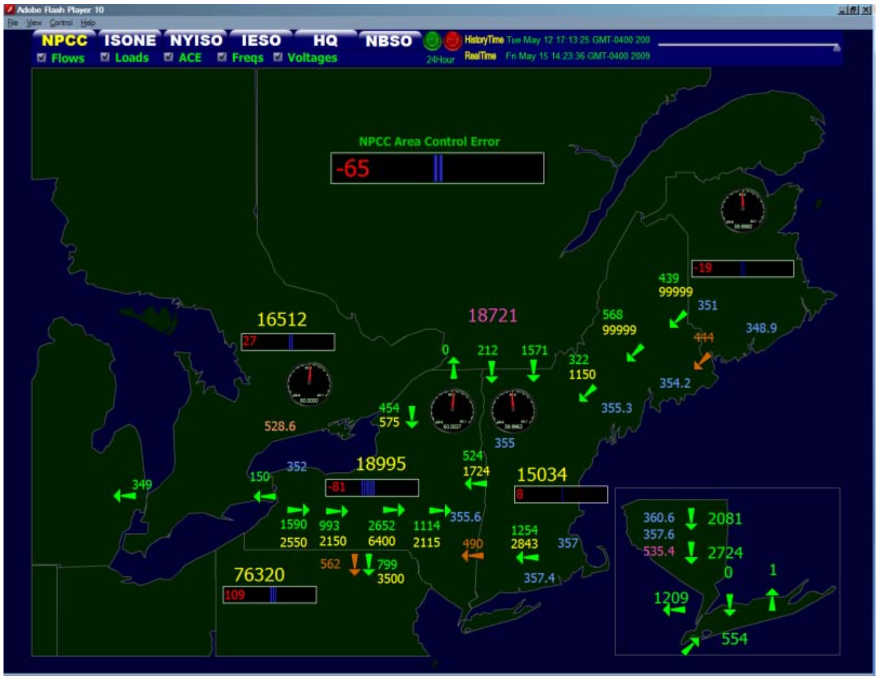


Figure 13: Adobe Flash web-based display at NE ISO (Source: [73])

This can be a very useful feature and such functionality is certainly recommended. Its major disadvantage is it requires time to see the display, so it cannot provide results at a glance.

Another possible partial solution to the issue of showing time variation of spatial data would be to use contours with bivariate color sequences. As the name implies the idea behind a bivariate color sequence is to contour data using a 2-D color-map, in which one dimension might correspond to the present value, and a second might depend on its rate of change. An example mapping might be to vary between yellow and blue in one dimension, and then between light and dark in intensity in the second. While this approach has not been applied to power system, such displays are used in other domains. However, these displays can be notoriously difficult to read ([74], p. 136). Also, in describing such displays Tufte notes, “The complexity of multifunctioning elements can sometimes turn data graphics into visual puzzles, crypto-graphical mysteries for the viewer to decode” ([76], p. 153). A sure sign a display has become a puzzle is when it requires a verbal rather than visual process to decode. More recent work is exploring the use of color coding with small graphical elements called textons, in which each represents a different numeric value [77]. An example of such a display is shown in Figure 14. Power system applications have not yet been explored.

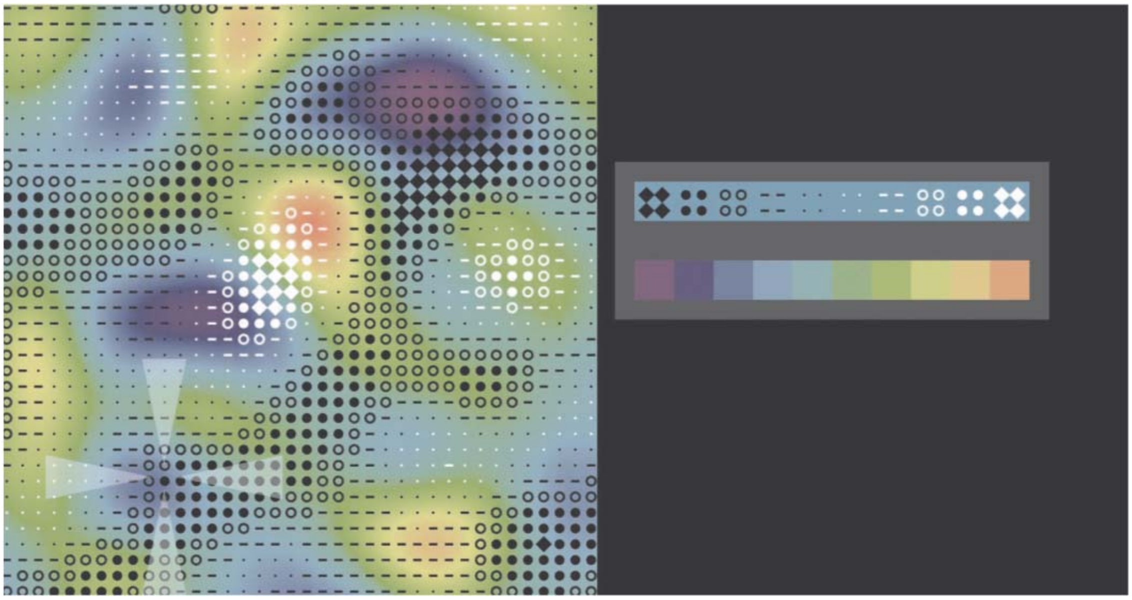


Figure 14: Contour using textons (Source: Figure 9 of [77])

Contouring is a popular method of displaying variation of data [78], but an issue with it is that this technique is good for displaying data at a single snapshot in time thus showing only the current state of the system. Time-sequence animations can be used for visualization similar to a weather radar sequence [75] but a major disadvantage is that it requires time to see the display, so it cannot provide results at a glance. In addition such plots cannot show geographically distributed information.

Another approach that has been suggested is to integrate small strip charts onto existing one-lines next to the field of interest. An example of this is shown in Figure 15. The advantage of this approach is the strip-charts are shown with good geographic context. An obvious disadvantage is to show such strip-charts for all the fields would require that they be quite small.

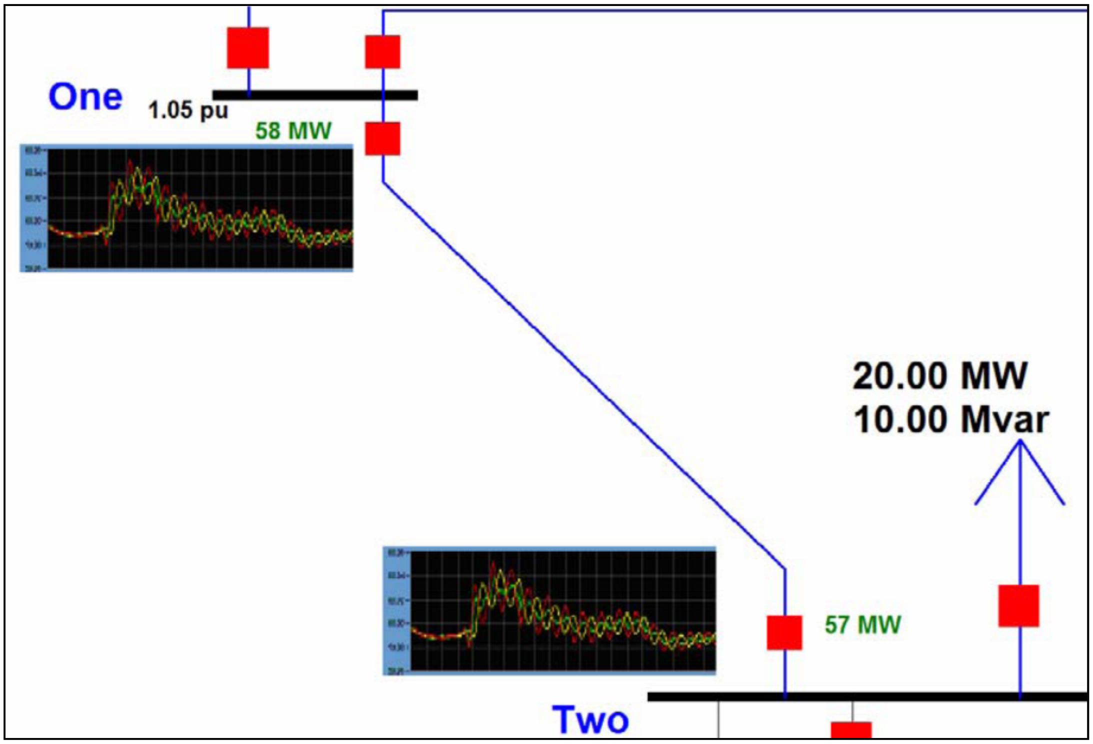


Figure 15: Geographically placed strip charts (Source: Figure 10 of [75])

One solution to this issue is use spark-lines, which have been defined by E. Tufte as intense, simple, word-sized graphics [79]. The idea of a spark-line is to show the time-variation in a signal using about the same display space as the value. Hence a spark-line is a graph without axis labels and numbers. Obviously there is a tradeoff between display space, and the amount of information shown. Spark-lines can only show data with several significant digits, but “the idea is to be approximately right rather than exactly wrong” ([79], p. 50). In a power system one-line context the x-axis time-scale could be common for all spark-lines (e.g., one hour or five minutes). The y-axis could also be implicit based on the type of value, for example between 75% and 150% for transmission line flows, and between 0.8 and 0.95 for low voltage voltages. Hence spark-lines would only need to be shown for values that are trending toward limit violations. Because of their small size spark-lines could also be embedded in tabular displays, such as showing voltage variation in the column next to the field showing the current voltage value. Use of spark-lines has been mentioned in [40] but not explored significantly in power system visualization.

In a previous work, the state-of-the art visualization techniques applied to power systems [73] have been studied and the shortcomings of existing visualization methods to present time series data of the power system have been noted. One of the recommendations of this study was that there is a need for development of effective visualization techniques to display power system time-varying data. To address this need, a visualization technique is presented for displaying transient stability data in this work using spark-lines on geographic overlays and is a contribution of this work. Thus both the issue of displaying trends and embedding it with geographic information such as those presented in the concept of Geographic Data Views [80] have been addressed.

2.2 Prior art

A method for monitoring system stability by visualizing generator oscillations on a 2-D plane with speed versus rotor position was presented [81]. This work, however, did not address the variation of system dynamics over time. Wide area frequency visualization methods have been presented in [82, 83]. The techniques used are animated event replays. Data collected from frequency disturbance recorders (FDRs) in a wide area FNET system are displayed with colored contours for every time step and played in the form of a movie as shown with the screen shots in Figure 16.

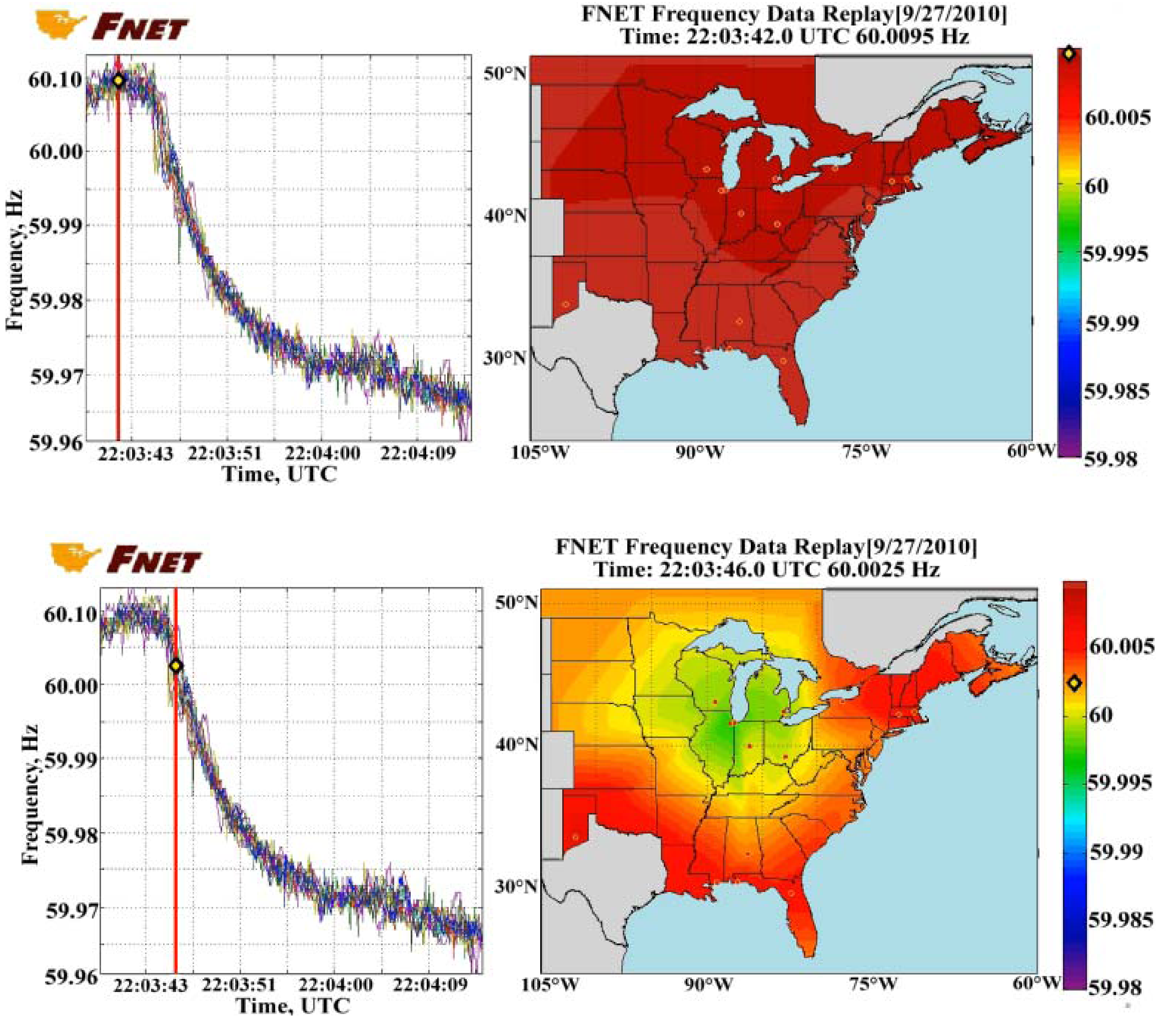


Figure 16: Screen shot of event replay of frequency data (before and after event) (Source: [82])

While contouring is in general computationally intensive, the use of graphical processor units (GPUs) for fast power system contouring allowing for near real-time usage is presented in [84]. Although certainly useful for some situations, contouring to show time-varying information requires time to show the animation loop.

Figure 17 and Figure 18 demonstrate how transient stability frequency variation is often shown for a large system. Both figures show the bus frequency response for a 16,000 bus system for twenty seconds of simulation with a large generation loss contingency occurring at two seconds. Figure 17 shows the frequency response at all of the buses in the system, while Figure 18 shows the frequency at twelve selected, primarily high voltage, buses spread throughout the system.

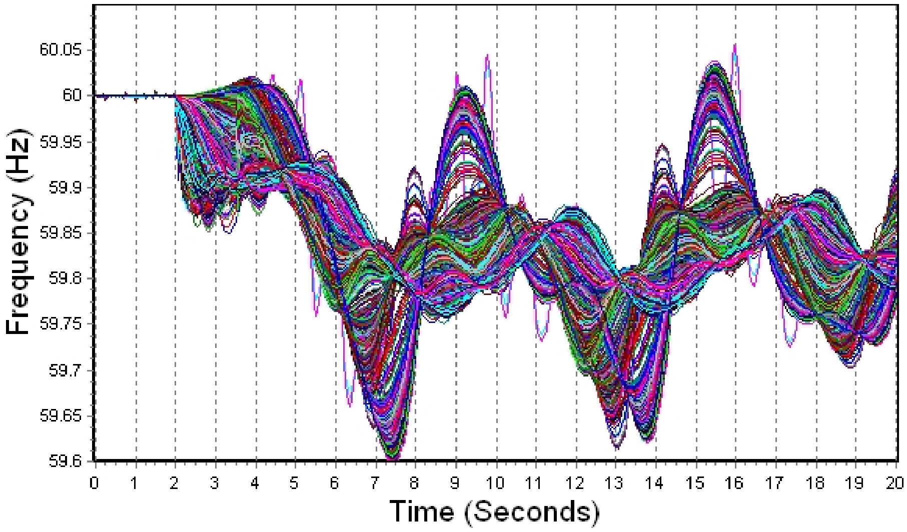


Figure 17: Frequencies at over 16000 buses in a system

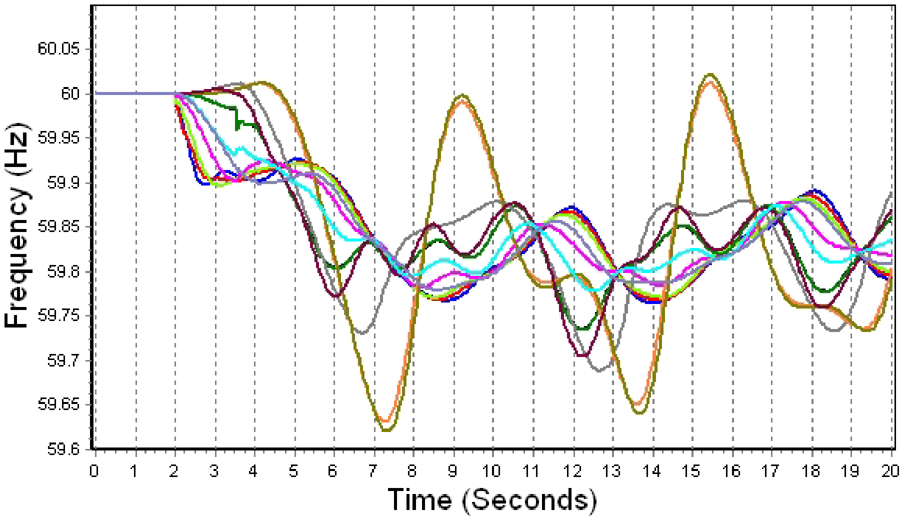


Figure 18: Frequencies at 12 selected buses

From the figures it is clear that there is coherency in the response of at least some of the buses, but it is difficult to determine the details. Obviously with 16,000 buses the purpose of Figure 17 is not to show the response at any individual location, but rather to give bounds on the overall system response. Yet even with only twelve curves in Figure 18 it is difficult to determine the individual frequency response (the individual buses are not identified on the figure due to data confidentiality concerns). Also, there is the question of how to select a small subset of buses that adequately captures the range of patterns of behavior exhibited by a large system, especially when these patterns may vary depending on the assumed contingency. The proposed methodology addresses these issues.

2.3 Methodology

The overall methodology is presented in the form of a flow chart in Figure 19.

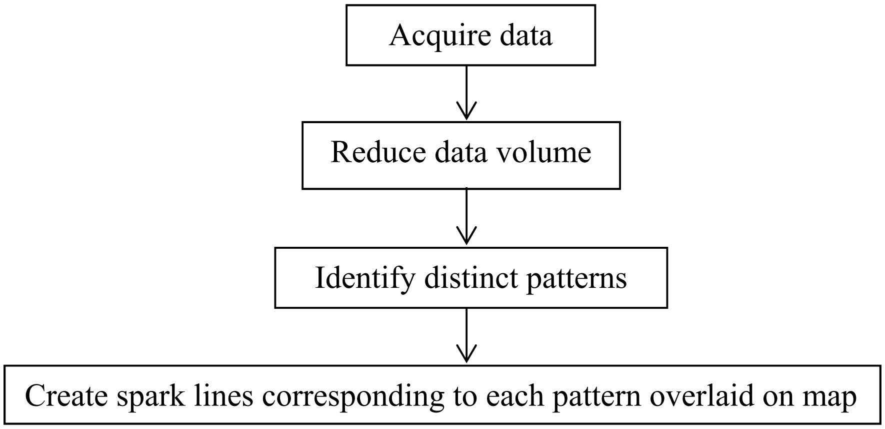

Acquire data
Reduce data volume
Identify distinct patterns
Create spark lines corresponding to each pattern overlaid on map

Figure 19: Overall methodology

Distinct patterns are identified in the data with a *Pattern identification* algorithm. A problem encountered in this process involved computational burden of processing large data sets. So a preprocessing algorithm was developed for *Data volume reduction*. To visualize the information thus extracted using the preceding two algorithms, "spark-lines" are generated which is the third step *Spark-line display* of the overall methodology. Spark-lines are overlaid on a geographical map of the power system. In the following subsections, each of these algorithms is described in detail. The term "data set" has been used frequently in this work to mean an N x τ volume of data where the number of nodes in the system is N, and the number of time points considered is τ.

Data volume reduction

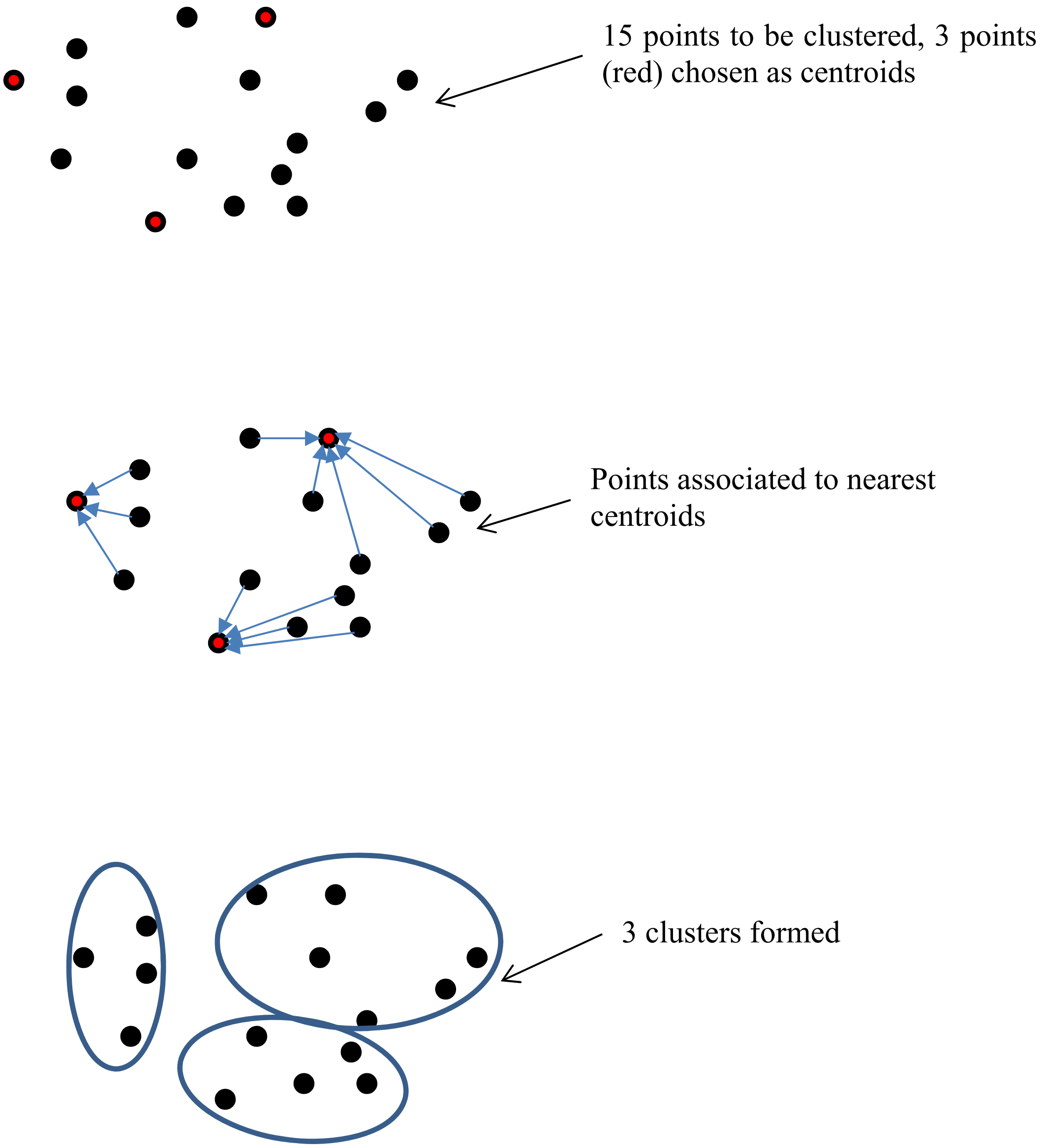
It is well known that given the size of a wide area power system, the volume of data generated by a simple transient stability run can be substantial. Analyzing such large data volumes can be computationally prohibitive. Hence this preprocessing step to reduce data volume was developed. Depending on the size of the system/data set, this step can be skipped.

Application of a K-means based clustering algorithm has been suggested as a possible method to aggregate synchro-phasor data in case of data overload [69]. Along these lines, in this dissertation the generators are clustered by identifying several patterns in the transient stability data and representing each group by one representative pattern called cluster representative.

K-means clustering is one of the simplest unsupervised learning algorithms for classifying a given data set into a certain K number of clusters fixed a priori [85, 86]. The main idea is to define K centroids, one for each cluster and assign each data point to the clusters with the nearest centroid. The problem is computationally difficult, called NP-hard, i.e. non-deterministic polynomial-time hard in computational complexity theory, meaning it cannot be solved in polynomial time. However there are efficient heuristic algorithms that are commonly employed and converge fast to a local optimum. One of the most popular heuristics for solving the K-means problem is based on a simple iterative scheme for finding a locally minimal solution. This algorithm is often called the K-means algorithm. There are a number of variants to this algorithm, and the version used in this work is known as the Lloyd's algorithm [87].

An important point to note in this iterative algorithm is that the starting centroids must be chosen carefully since different starting locations of centroids may cause the algorithm to converge to different clusters. An idea is to select centroids from the given data set in a way that they are as much as possible far away from each other. The next step is to take each point belonging to the given data set and associate it to the nearest centroid forming clusters. When all the points have been grouped, the first step is completed. At this point K new centroids are re-calculated as centers of the clusters resulting from the previous step. After this, once again the same data set points are grouped so that each one is now associated with its nearest new centroid.

This process continues till there is no change in the location of the centroids. In other words centroids do not move any more.



15 points to be clustered, 3 points (red) chosen as centroids

Points associated to nearest centroids

3 clusters formed

Figure 20: Illustration of K-means clustering

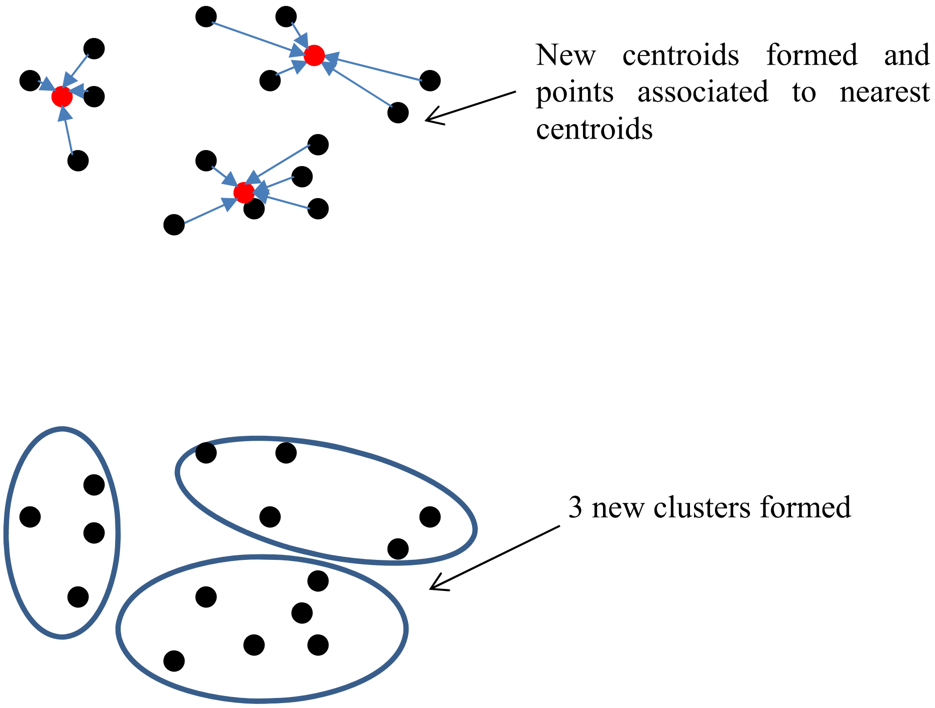

New centroids formed and points associated to nearest centroids
3 new clusters formed

Figure 20: Illustration of K-means clustering (cont.)

Figure 20 illustrates the K-means algorithm for grouping 15 objects shown with black dots into 3 clusters. Initially, three of the objects are randomly chosen as centroids, shown with red body and black outline. All other objects are assigned to the nearest centroid, forming three groups. In the next iteration, three new means or centroids are computed for each group shown with red dots. In this case, the centroids are distinct from the original set of objects. Once more, the objects are assigned to the nearest centroid forming three new clusters. The process is repeated till the centroids do not move any more.

K-means clustering algorithm is used for data volume reduction in this work with the number of clusters k chosen between 200 and 300. Although it might seem that such a procedure could fail to capture certain important information, the validity of this method and the choice of number of clusters will be justified in the discussion of results.

Comparison of multiple data series over a time window requires a similarity measure. Based on a survey of different clustering methods for time-varying data [88], in this work Euclidean distances between the time series have been used where each time series is a data point with as many dimensions as the number of time samples in the series.

Let X_p and X_q be τ-D vectors where,

$$X_n = \left[x_{n_1}, x_{n_2}, \ldots\ldots, x_{n_\tau}\right]^T \tag{2.1}$$

The Euclidean distance between these two vectors is given by:

$$d_E = \sqrt{\sum_{t=1}^{\tau}\left(x_{p_t} - x_{q_t}\right)^2} \tag{2.2}$$

Mathematically, the K-means algorithm determines a set of K cluster centers $\{\, V_i \mid i = 1, 2, \ldots, K \,\}$ where,

$$V_i = \left[v_{i_1}, v_{i_2}, \ldots\ldots, v_{i_\tau}\right]^T \tag{2.3}$$

Given a set of N nodes $\{\, X_n \mid n = 1, 2, \ldots, N \,\}$, the objective function to be minimized is:

$$\mathrm{Min}\ J(U, V) = \sum_{i=1}^{K}\sum_{n=1}^{N} u_{in} \|X_n - V_i\|^2 \tag{2.4}$$

Such that:

$$u_{in} \in \{0,1\}\ \forall\, i, n \tag{2.5}$$

$$\sum_{i=1}^{K} u_{in} = 1\ \forall\, n \tag{2.6}$$

where $\|\,.\,\|$ is the Euclidean distance measure given by equation (2.2).

The iterative solution procedure has the following steps:

- *Step 1* – Choose the number of clusters K and a small number ε for stopping the iterative procedure. Set the counter l = 0 and the initial cluster centers $V_i^{(0)}$, ∀ i, arbitrarily.

- *Step 2* – Distribute X_n, $\forall$ n to determine $U^{(l)}$ such that J is minimized. This is achieved by reassigning X_n to a new cluster that is closest to it.
- *Step 3* – Revise the cluster centers $V_i^{(l)}$, $\forall$ i.
- *Step 4* – Stop if the change in cluster centers is smaller than ε; otherwise, increment l and repeat steps 2 and 3.

The K cluster centers are cluster representative nodes (generators for the case study) and the time response corresponding to each of these characterizes its entire cluster. In the case that a cluster center is not one of the nodes in the original data set, the node closest by similarity measure to the cluster center is selected as the cluster representative node.

Pattern identification

Pattern identification forms the heart of the proposed methodology. The transient stability data from K cluster representative nodes computed in the previous step are the inputs to this stage. This stage identifies distinct patterns in the input data and selects the nodes or generators, whose responses taken together capture the behavior of the entire system. Thus a set of responses is identified that provides comprehensive understanding of the system dynamics. Referring back to Figure 17 and Figure 18, this algorithm presents a method to automatically select a small number of buses from the set of over 16,000 buses which capture the range of patterns in the system. A Quality-Threshold (QT) clustering based algorithm has been used for this purpose.

The QT (Quality Threshold) clustering algorithm [89] was originally developed to group genes into high-quality clusters. Quality is ensured by finding clusters whose diameters do not exceed a given user-defined diameter threshold. This method prevents dissimilar genes from being forced under the same cluster and ensures that only good quality clusters will be formed.

The goal of QT clustering is to form large clusters with similar expression pattern, and to ensure a quality guarantee for each cluster. Quality is defined by the cluster diameter and the minimum number of points contained in each cluster. The advantage of QT clustering algorithm compared to K-means is that in QT clustering there is no need to specify the number of clusters, as required in K-means. Also, it is not necessary that all points need to be clustered and returns the same result when run several times. However, QT is computationally more expensive than K-means.

The QT algorithm is as follows [90]:

- *Step 1* – A random point is chosen from the list of points to be clustered.
- *Step 2* – The algorithm determines which point has the greatest similarity (closest Euclidean distance) to this point. If the distance is less than the quality threshold distance which is pre-specified, then these two points are clustered together.
- *Step 3* – Other points are similarly added to this cluster. This process continues until no point can be added to this first candidate cluster without surpassing the diameter threshold.
- *Step 4* – A second candidate point is chosen.
- *Step 5* – The algorithm determines which point has the greatest similarity (closest) to this second point. All points in the list of points are available for consideration to the second candidate cluster.
- *Step 6* – Other points from the list of points that minimize the increase in cluster diameter are iteratively added to the second candidate cluster. The process continues until no point can be added to this second candidate cluster without surpassing the diameter threshold.
- *Step 7* – The algorithm iterates through all points on the selected list of points and forms a candidate cluster with reference to each point. In other words, there will be as many

candidate clusters as there are points in the list. Once a candidate cluster is formed for each point, all candidate clusters below the user-specified minimum size are removed from consideration.

- *Step 8* – The largest remaining candidate cluster, with the user-specified minimal number of points, is selected and retained as a QT cluster. The points within this cluster are now removed from consideration. All remaining points will be used for the next round of QT cluster formation.
- *Step 9* – The entire process (steps 1 to 9) is repeated until the largest remaining candidate cluster has fewer than the user-specified number of points.

The result is a set of non-overlapping QT clusters that meet quality threshold for both size, with respect to number of points, and similarity, with respect to maximum allowable diameter. Points that do not belong in any clusters will be grouped under the "unclassified" group. Figure 21 illustrates the process of clustering fourteen objects. Corresponding to each object, a candidate cluster is formed including all objects within a certain quality around it, shown by a blue circle. From the fourteen candidate clusters, the largest cluster consisting of nine objects is selected as a QT cluster, and these nine objects are no longer considered in the next iteration. Thus in the next iteration only five objects need to be considered. For each of these objects, a candidate cluster is formed and again the largest cluster consisting of four objects is finalized as a QT cluster. The remaining object is "unclassified" or forms a single object cluster.

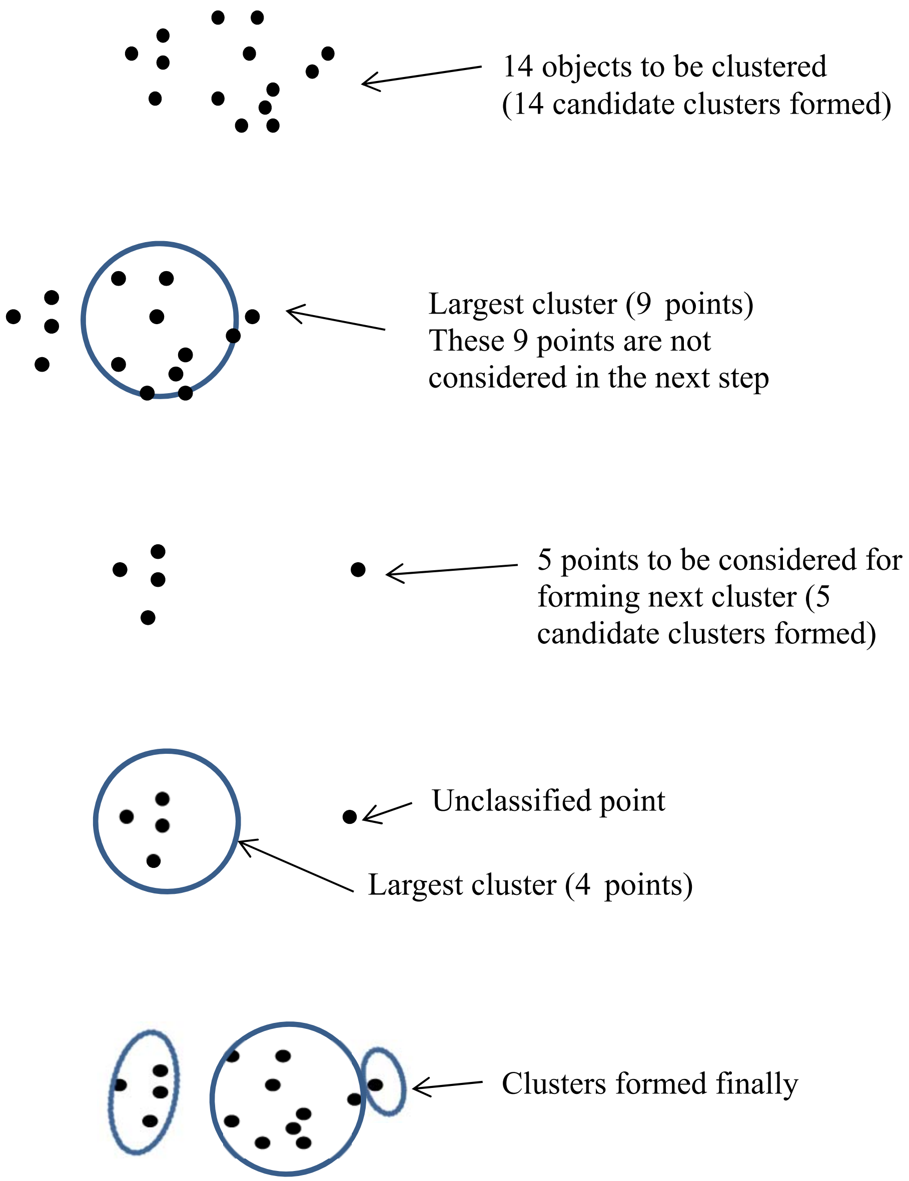

14 objects to be clustered
(14 candidate clusters formed)
Largest cluster (9 points)
These 9 points are not
considered in the next step
5 points to be considered for
forming next cluster (5
candidate clusters formed)
Unclassified point
Largest cluster (4 points)
Clusters formed finally

Figure 21: Illustration of Quality-Threshold clustering

It is important at this stage to discuss some complexity issues of the QT algorithm. Danalis et al. [91] have studied in detail the complexity of the QT clustering algorithm and found that a poor implementation of the QT algorithm could be $O(\mathrm{N}^5)$ which is prohibitively high. However, if the QT algorithm is implemented as described in this chapter, the complexity is $O(\mathrm{N}^2)$ [92] where the main source of complexity is in the computation of distances between all of N objects, or nearest-neighbor searches.

Nearest-neighbor search (NNS), also known as proximity search, similarity search or closest point search, is an optimization problem for finding closest points in metric spaces [93]. The problem is: given a set S of points in a metric space M and a query point q ∈ M, find the closest point in S to q. In many cases, M is taken to be d-D Euclidean space and distance is measured by Euclidean distance or Manhattan distance. The simplest solution to the NNS problem is to compute the distance from the query point to every other point in the database, keeping track of the "best so far." This algorithm, sometimes referred to as the naive approach, has a running time of *O*(Nd) where N is the cardinality of S and d is the dimensionality of M. There are no search data structures to maintain, so linear search has no space complexity beyond the storage of the database. Much effort has been devoted to making these searches as fast as possible and reducing the number of searches needed. One approach is the use of space-partitioning techniques. In this group of techniques, is the method involving use of a data structure called *k*-D *tree* which partitions items in *k*-D coordinate space.

Description of *k*-D trees can be found in [94] and [95]. A *k*-D tree is a binary search tree in multiple dimensions. A simple 2-D *k*-D tree for points (1,3), (6,4), (5,6), (3,9), (4,7), and (6,8) is shown in Figure 22.

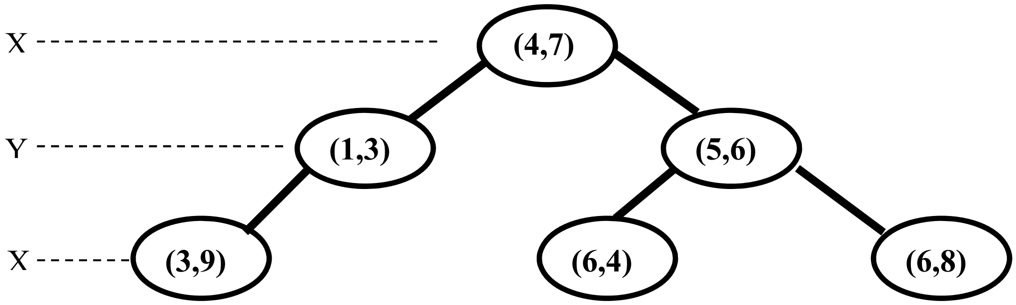


Figure 22: A *k*-D tree with 6 points

The tree is constructed recursively by cycling through each of the *k* axes of the data. At each level the data is split about the median point. Hence the tree is balanced by design. Given N points, *k*-D tree can be built with *O*(N) nodes and *O*(logN) depth in *O*(NlogN) time. By storing data in this structure, the nearest-neighbor query, i.e. the process of searching the tree and finding the point with smallest distance to the query point can be done very efficiently. Nearest-neighbor searches using *k*-D trees are discussed in detail in [95]. The average complexity for finding the nearest-neighbor is *O*(log N) [67].

The QT algorithm can be potentially speeded up by using *k*-D trees. However, it is important to note that naive search can, on average, outperform space partitioning approaches on higher-dimensional spaces [96]. In this work the dimension of the data equals the length of the time series over which dynamic frequency responses of generators are recorded that is 0-20 secs with average granularity of 0.05 secs or a time series with about 400 time points for each of the representative generators or data points. So in this case *k*-D trees are not very useful.

Compared to the K-means algorithm, QT clustering does not require specification of the number of clusters. Typical QT algorithms also have a restriction on minimal number of nodes required to form a cluster. In this work, this restriction is removed allowing formation of even single node clusters. These two features are crucial in determining the distinct patterns of the system and identifying “outliers” in the data, which tend to form single node clusters. Also, QT algorithm returns the same result when run several times. Like the previous stage, the closeness of nodes is measured by Euclidean distance.

The result is a set of non-overlapping QT clusters that meet quality threshold for similarity with respect to maximum allowable diameter. The choice of quality threshold diameter in this work is by trial and error. The problem of automating the selection of this parameter has been left as a future work. However, as the results will reveal, the choice of threshold diameter in this work is adequate. An important note is that the lower the cluster threshold diameter, the "tighter" the clusters are formed. Often this is tantamount to generation of many more clusters. Of course if all the system responses are identical, no matter how small the threshold diameter is, only one cluster will be formed.

Spark-line display

The distinct patterns identified in the previous stage are visualized with spark-lines overlaid on a map, each spark-line representing a cluster. An approach that has been suggested is to integrate small strip-charts onto existing one-lines next to the field of interest [97]. This of course requires strip-charts to be quite small. The current work leverages on this idea, displaying the transient stability information as spark-lines, and infusing these with location information by overlaying on the one-line diagram or the actual latitude-longitude map.

As discussed earlier, *spark-lines* present a historical trend of data in the space of a typical word [98]. The effectiveness of spark-lines can be explained by Figure 23, Figure 24, and Figure 25 all showing the frequency of a generator at a time instant. The simplest way to represent the frequency is in the value form shown as Figure 23. This single number can take on additional meaning when placed in context of previous frequency measurements as shown in Figure 24. Further, the current value and the time point it occurs, i.e. far right are accented in red color and shown in Figure 25. The data line thus created is called a spark-line.

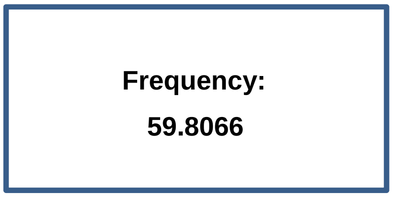


Figure 23: Display of value only

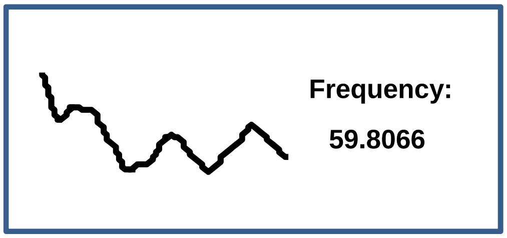


Figure 24: Display of value and trend

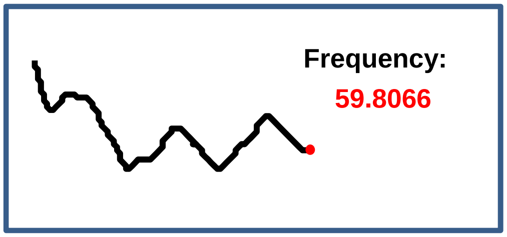


Figure 25: Display of value and trend highlighting present value

A spark-line does not have axis labels and numbers. In a power system one-line context or geographic data view, the x-axis time-scale could be common for all spark-lines, e.g. 20 seconds in a transient stability run. The y-axis could also be implicit based on the type of value, e.g. between 59.8 to 60.2 Hz for frequencies.

Spark-lines are automatically generated and laid out on a geographical map. It should be made sure that spark-lines do not overlap. This was achieved by developing overlap correction algorithms which automatically detect the occurrence of overlaps and relocate spark-lines on the available display space using force-based layout methods [29].

The force-based layout is a graph drawing method in which the nodes or vertices of the graphs are replaced by charged rings and the interconnecting edges by springs. The net force applied on a given node is the sum of the spring force applied to the node and the repelling Coulomb forces applied by all the point charges. Let the direction be given by the vector $\overrightarrow{d_{\imath,\jmath}}$ representing the distance between the ith and jth nodes ($\overrightarrow{x_\imath}$ representing the ith node's location).

$$\overrightarrow{d_{\imath,\jmath}} = \overrightarrow{x_\imath} - \overrightarrow{x_\jmath} \tag{2.7}$$

where the normalized vector representing the direction of the force is given by:

$$\widehat{d_{\imath,\jmath}} = \frac{\overrightarrow{d_{\imath,\jmath}}}{\left\|\overrightarrow{d_{\imath,\jmath}}\right\|} \tag{2.8}$$

The spring force can be given by:

$$\overrightarrow{F_{\imath,\jmath}} = -\gamma\left(\left\|\overrightarrow{d_{\imath,\jmath}}\right\| - r\right)\widehat{d_{\imath,\jmath}} \tag{2.9}$$

where γ is the spring constant, r is the length of the spring at rest, and $\overrightarrow{F_{\imath,\jmath}}$ is the force between the ith and jth node. For the point charge the repulsive force is given by

$$\overrightarrow{F_{\imath,\jmath}} = k' \frac{Q_i Q_j}{\left\|\overrightarrow{d_{\imath,\jmath}}\right\|^2} \widehat{d_{\imath,\jmath}} \tag{2.10}$$

where Q_i is the charge of the ith node and k' is the coulomb constant. After the net force on each node is computed, the velocity and location of the nodes are computed. From Newton's laws,

$$\vec{F} = ma \tag{2.11}$$

If the present velocity of a node is v_n, then,

$$v_{n+1} = ah + v_n \tag{2.12}$$

If the node has current location p_n, then its next location will be,

$$p_{n+1} = \frac{h^2}{2} a + hv_n + p_n \tag{2.13}$$

Here h represents the time steps. It is important to apply damping to the velocity over time, to prevent the structure from continuing to move in space and result in an overflow error.

Figure 26 presents the pseudocode for a typical force-based graph drawing technique.

```
set initial node velocities to zero
set initial node positions to random non-overlapping values
loop
    set total_KE = 0            // running sum of total kinetic energy (KE) over all  particles
    for this_node               // loop over every node
        set net-force = 0       // running sum of total force on this particular node

        for each other_node
            net-force = net-force + Coulomb_repulsion( this_node, other_node )
        next node

        for each spring connected to this_node
            net-force = net-force + Hooke_attraction( this_node, spring )
        next spring

        // update velocity and position
        this_node.velocity = (this_node.velocity + timestep * net-force) * damping
        this_node.position = this_node.position + timestep * this_node.velocity
        total_KE = total_KE + this_node.mass * $(\text{this\_node.velocity})^2$
    next node
until total_KE is less than some small number  // the simulation has converged
```

Figure 26: Pseudocode for typical force-based graph drawing technique

Depending on the size of the spark-line plots relative to size of the display, the spark-line plots might overlap. As such, the visualization algorithm should have the capability to automatically detect occurrence of overlaps and relocate spark-line plots on the available display space. This capability is achieved by implementing an overlap corrector algorithm. This algorithm first detects the occurrence of overlaps and then applies a force-based re-locating algorithm.

A modified version of the force-based layout algorithm has been applied in this work because of major differences in between the traditional framework and the current one. First, instead of point objects as nodes, spark-line plots need to be placed in a display area. Second, the spark-line plots themselves do not have any connection such as transmission lines in between. As such there are no sources of spring forces as per the traditional force-based layout concepts.

In running the force-based algorithm, the spark-line plots might be relocated at a position where the plot area overlaps one or more of the cluster representative nodes. This needs to be avoided as well. Hence the force-based layout algorithm is run on both spark-line reference points and the cluster representative nodes taken together with the added constraint that cluster representative nodes cannot move. In the modified force-based layout, a pseudo-Hooke's force is also added which pulls the spark-line plot reference locations back to the initial positions. This force is similar in nature to the spring force since the farther a spark-line plot is away from its original location, the greater the force acting on it pulling it toward that location.

The algorithm initially places the spark-line plots at a small offset from the geographic locations of the cluster representative nodes whose information is to be displayed. The choice of this offset, denoted by Δ will vary from display to display. This is followed by checking whether such placement causes overlaps. The lower-left corner of the plot areas are considered as the reference locations of corresponding spark-line plots and are treated similar to nodes. Dimensions of the spark-line plots are also chosen depending on the size of the display. The dimension along the x-direction and dimension along the y-direction are represented consecutively by L and H. The difference in x-coordinates and y-coordinates between any pair of nodes i and j is given by dist_x(i,j) and dist_y(i,j), respectively.

Three sets are defined. A = {spark-line reference points}, B = {cluster representative nodes}, and N = {A,B}.

Initial positions are given by:

$$p_n^{(0)} = [p_{x_n} \quad p_{y_n}] = \begin{cases} [\text{longitude} \quad \text{latitude}] & \forall n \in B \\ [\text{longitude} + \Delta \quad \text{latitude} + \Delta] & \forall n \in A \end{cases}$$

Initial velocities are given by $v_n^{(0)} = [v_{x_n} \quad v_{y_n}] = [0 \quad 0] \ \forall n \in N$.

Initially the total force on a node is given by:

$$F_n^{(0)} = [F_x \quad F_y] = [0 \quad 0] \ \forall n \in A$$

The pseudo-code for the overall algorithm is as follows:

```
// Overlap detection
If dist_x (i, j) < L and dist_y (i, j) < H  ∀ i, j ∈ A and i ≠ j
  // overlap is detected
    Go to Relocation algorithm
end

// Relocation algorithm
While $KE_{tot} > \varepsilon$   // a small number
  For  i ∈ A
      $F_i^{(t)} = F_i^{(t)} + \text{Hooke\_attraction}\left(i, p_i^{(0)}\right)$
      For  j ∈ N and j ≠ i
          $F_i^{(t)} = F_i^{(t)} + \text{Coulomb\_repulsion}(i, j)$
      End For
      // update velocity
      $v_i^{(t+1)} = \left(v_i^{(t)} + \text{stepsize} * F_i^{(t)}\right) * \text{damping}$
      // update position
      $p_i^{(t+1)} = p_i^{(t)} + \text{stepsize} * v_i^{(t+1)}$
      // update total kinetic energy
      $KE_{tot} = KE_{tot} + \theta * \left(v_i^{(t+1)}\right)^2$ // θ is a constant
  End For
End While
```

2.4 Case study and discussions

The methodology discussed in the previous section has been tested on the 16,000 bus, 2400 online generator system introduced in Figure 17 and Figure 18. A generator outage fault is applied at time two seconds which triggers a frequency disturbance that propagates across all the generators of the system. The low frequency event originating in one part of the system moves to the other part and back causing most of the generators in the system to oscillate over the simulation period of 20 seconds, the oscillations varying in phase and amplitude.

Animated contouring visualization can capture these propagations in the form of a movie. This approach has been used in [82] to visualize frequency disurbance propagation in the Eastern Interconnect simulating the 2003 blackout. As mentioned earlier, this kind of visualization requires observation of the whole movie and any individual movie frame does not contain any information of the history of the system dynamics. Furthermore, contouring is a computationally intensive visualization technique.

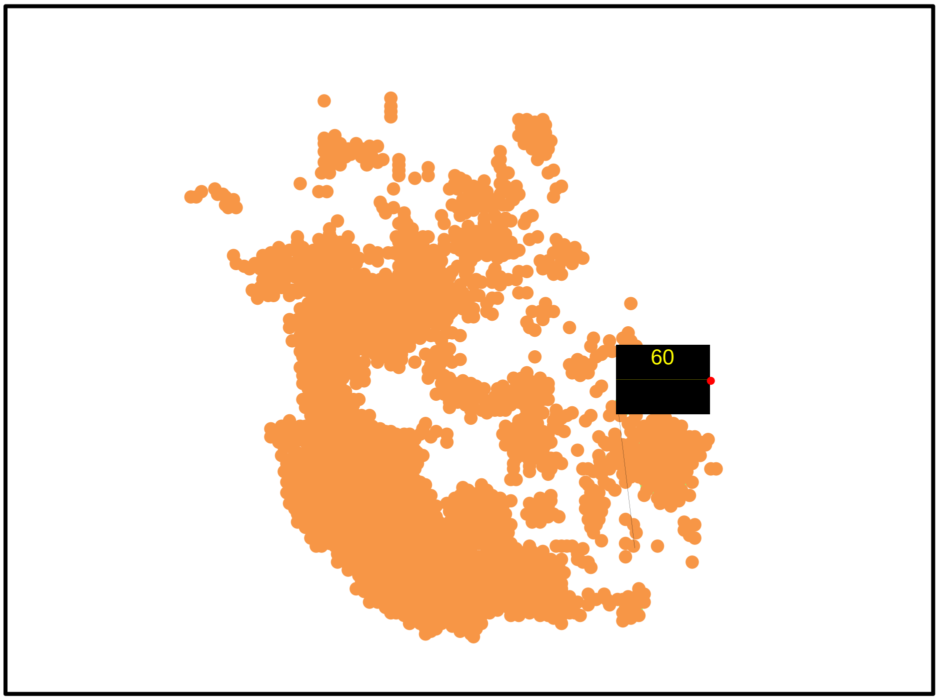


Figure 27: Frequency visualization before fault at 1.95 sec

Figure 27 and Figure 28 show the frequency distribution over the system at two time instants, just before the fault occurs at 1.95 secs and at the end of 20 secs.

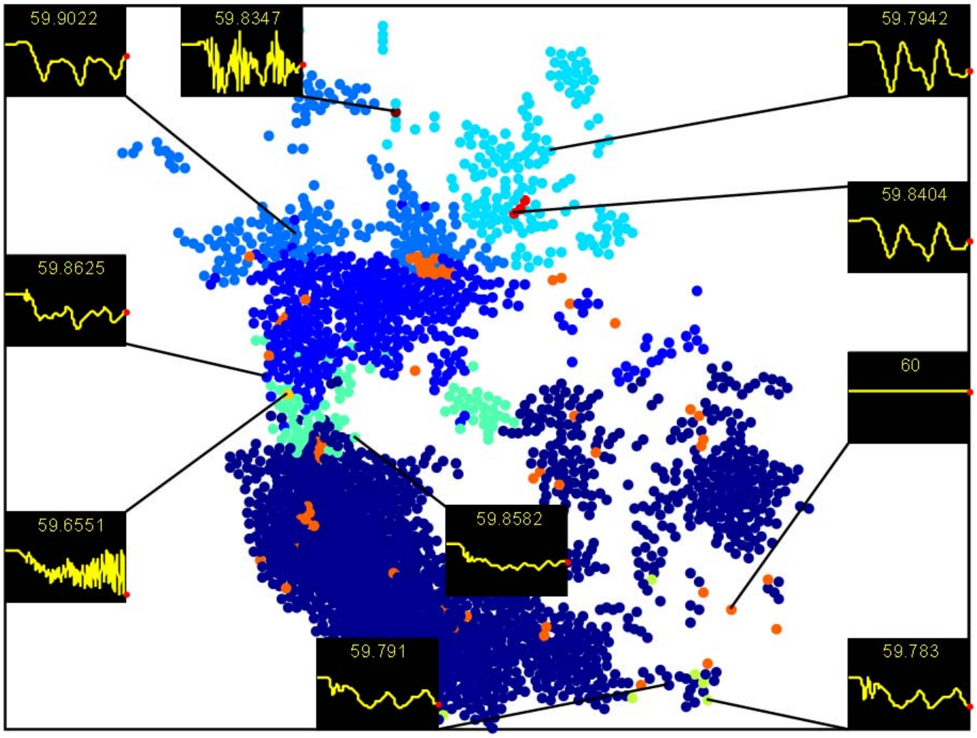


Figure 28: Frequency visualization at 20 secs

Figure 28 shows the transient stability frequency variation of the system generators over 20 seconds. All of the generators, including the cluster representatives, are displayed on a roughly geographical map of the system with different color codes corresponding to different clusters. The transient stability data corresponding to each cluster representative node is displayed by spark-lines showing the trend and the current value. In addition, lines drawn from a spark-line plot to the corresponding cluster representative node indicates the cluster to which a spark-line belongs. Ten distinct frequency response patterns are identified by the algorithm and displayed using spark-lines. The current time (i.e. 20 sec) frequency values are also displayed. Note that these include a single node cluster (in dark red color in the top row) which shows a frequency response which has significant oscillations. Another single node cluster was identified (in yellowish-green color on the left side) with oscillations increasing as time progresses. Thus two generators were automatically located by the algorithm which warrant attention.

Later these generators were carefully inspected and the cause of their abnormal behavior was found to be a limit error in their exciter models which were subsequently corrected. Interestingly, prior to the development of the technique presented here these model errors had not been noticed by the authors inspite of repeated use of the same model for other studies.

Figure 29 shows all of these 10 distinct responses on one plot. As can be seen it is difficult to determine the details even with only 10 responses in multiple colors on one plot.

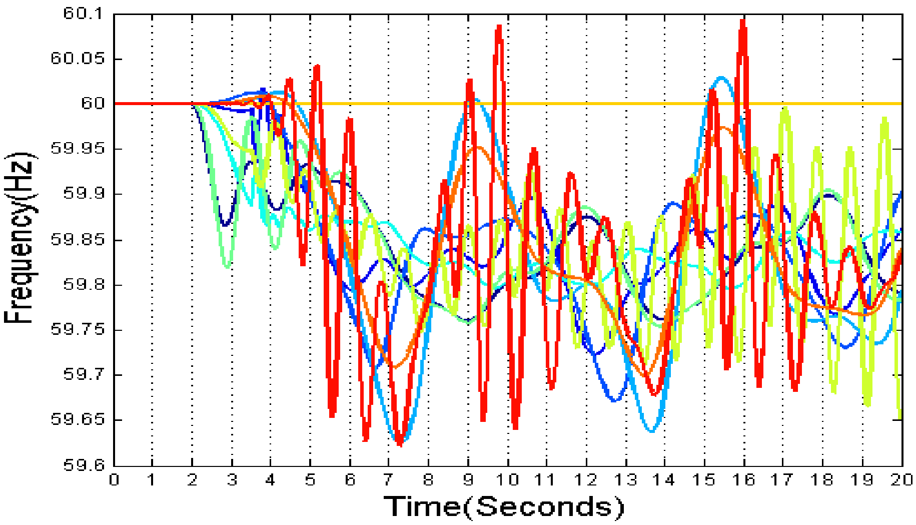


Figure 29: Frequency data from 10 generators

Figure 30 - Figure 39 show these same 10 responses in separate larger plots with each plot showing the range of frequency values of generators included in its cluster (colored in cyan). The number of generators included in each cluster is also mentioned. The close conformity of the range values with the cluster representative indicates that no important information was lost in the *Data volume reduction* process.

The running time of the algorithm analyzing frequencies of 2400 generators at over 400 time points (0-20 secs with average step size of 0.05 sec) coded in MATLAB and run on an Intel i5 2.30 GHz CPU is 4.69 secs. The running time is mainly dependent on the implementation of the clustering algorithms and can be reduced further by employing *k*-D trees [67]. The other parameters affecting the speed are the number of cluster representative points generated in the *Data volume reduction* process, quality-threshold diameter in the *Pattern identification* process, and the number of time points considered for the analysis. Although the detailed results are not presented here, the current implementation was tested on an even larger system of over 16,000 buses. Bus frequencies over 400 time points (0-20 secs with average step size of 0.05 sec) were clustered generating 100 cluster representative points and finally 10 clusters. Running time was found to be about 5 secs. Hence the proposed methodology is fast and can be further developed for real-time analysis with PMU measurements in tracking mode for example. Another extension of this work could be to enable the algorithm to automatically select the time range of data for information processing and visualization. For example, if the algorithm detects that there is no abnormal dynamic response in the system over a time window of say 20 secs, it automatically scans the data over the previous time window as well.

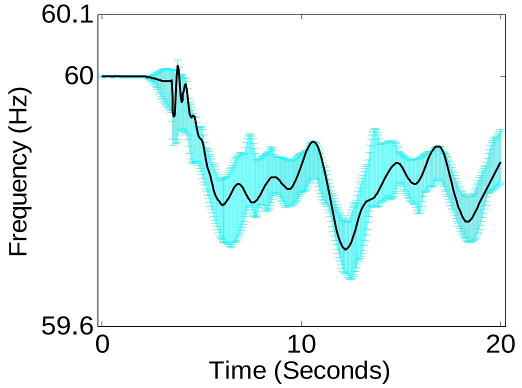


Figure 30: Cluster 1 (379 generators)

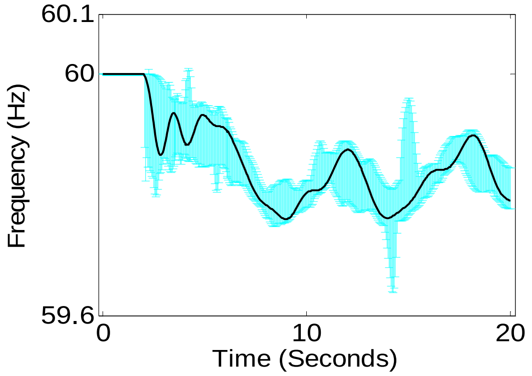


Figure 31: Cluster 2 (1518 generators)

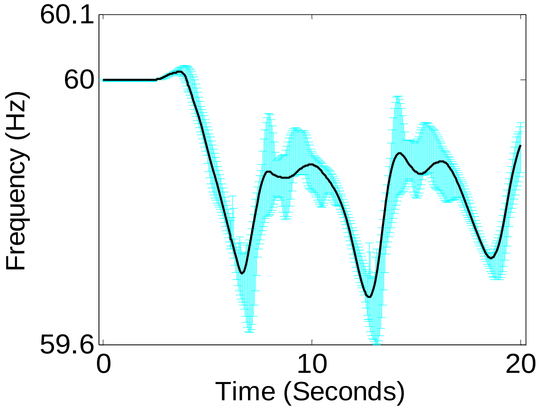


Figure 32: Cluster 3 (194 generators)

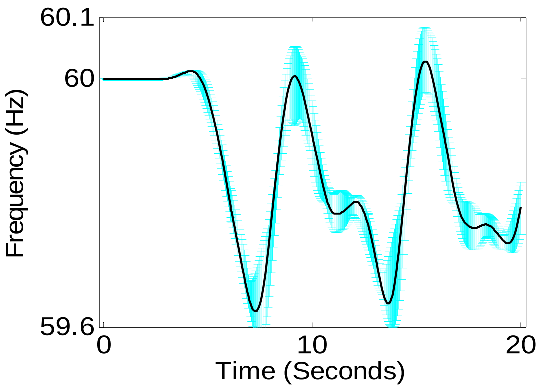


Figure 33: Cluster 4 (162 generators)

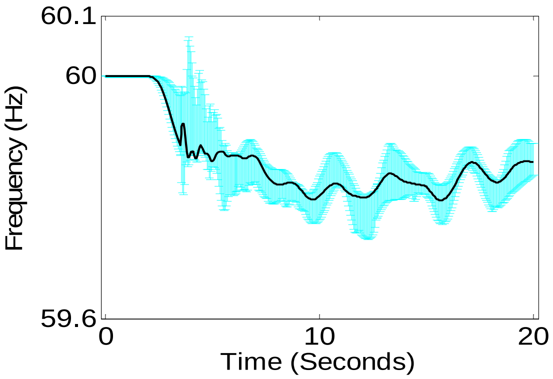


Figure 34: Cluster 5 (75 generators)

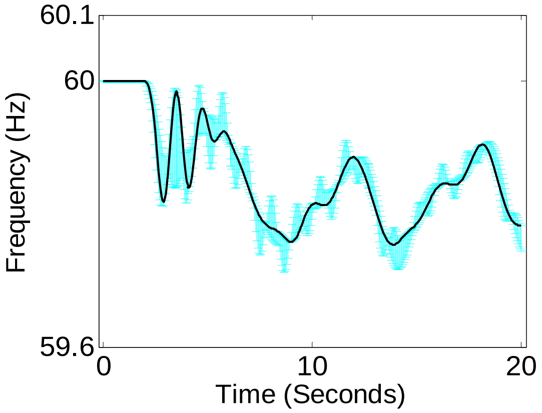


Figure 35: Cluster 6 (7 generators)

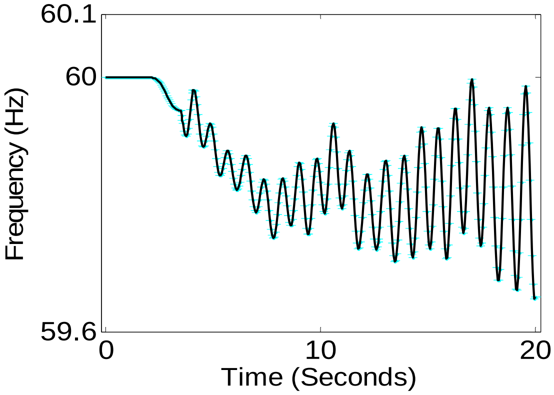


Figure 36: Cluster 7 (1 generator)

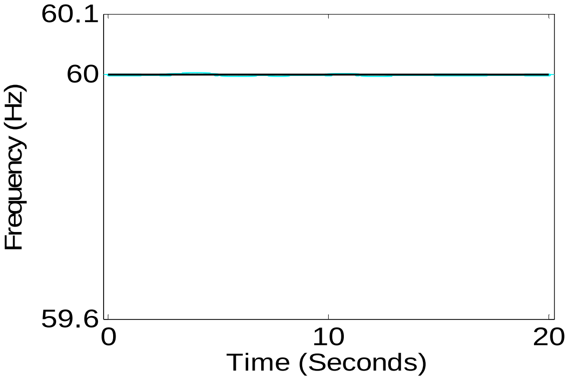


Figure 37: Cluster 8 (61 generators)

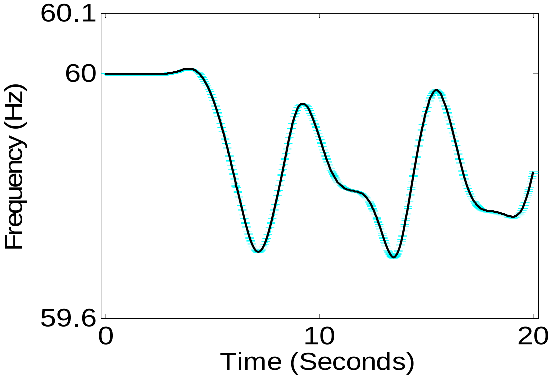


Figure 38: Cluster 9 (3 generators)

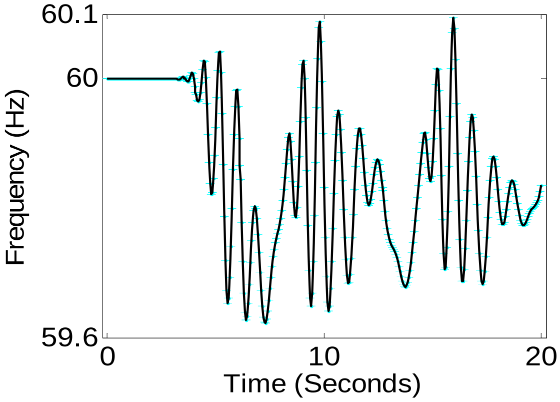


Figure 39: Cluster 10 (1 generator)

2.5 Conclusions

This work presents an important contribution by proposing a methodology for extracting important information from power system time-varying data and its visualization. Transient stability run results are the source of such data in this work. Extracted information includes abnormal dynamic response indicating some form of error or condition requiring attention. Also identified are the characteristics of the wide area power system, grouping nodes of similar response. Data volume, a problem frequently encountered in large power systems is also addressed with a method to reduce data volume without loss of information.

Another contribution is in use of spark-lines for visualizing transient stability information and their automatic placement without overlaps on a geographic map of the system. Important to note is that although the case study presented analyzes generator frequencies, the presented methodology can also be applied to other data from transient stability results such as bus voltages. The algorithms are extremely fast even when run on thousands of data points and hence can be used for real-time analysis in tracking mode with PMU measurements for example. These applications can be addressed as a future direction of the work presented here. Also demonstrated here was the effectiveness of data mining and graph drawing techniques for information extraction and visualization of power system data.

3. APPLICATION OF GRAPH THEORY AND CLUSTERING ALGORITHMS TO WIND FARM COLLECTOR SYSTEM DESIGN

A wind farm has three main components, the wind turbines that generate power from wind, the substation which transmits the power generated by turbines to the electric grid, and the electrical collector system which consists of cables, transformers, junction boxes, switchgear, and other electrical equipment that consolidate the power generated by turbine units distributed over the geographical area of the wind farm to the substation.

This chapter addresses the optimal design of wind farm collector systems. The work presented in this chapter is extremely important for planning and installation stages of large wind farms as shown in Figure 40. Due to a large number of wind farms coming up in recent times due to the focus on increasing penetration of renewable energy in our power grids, this topic is particularly relevant at the present time. It has been demonstrated that clustering techniques and graph-theoretic algorithms can be extremely useful tools in developing optimal designs also taking into consideration several design constraints.

Figure 40: Aerial view of a wind farm (Source: www.midwestenergynews.com)

3.1 Motivation

Wind farm collector systems constitute the single most important element of wind farms after the turbines and the substation. Optimal operation of wind farms depends on optimal designs of the wind farm collector systems. So there is a need to develop sophisticated methodologies to generate these designs. However, optimal design of a wind farm collector system is not an easy problem since this requires consideration of several real-life design constraints. Considerations for laying out cables in a collector system include turbine placement, terrain, reliability, landowner requirements, economics, and expected climatic conditions for the location [99]. Another consideration is the configuration of collector systems, which can be structured as the loop system or radial system depending on the desired level of collector system reliability. Typically designs of collector systems are done manually which is cumbersome and prone to errors. But with growing sizes of wind farms this manual process needs to be replaced by automatic design processes. Hence, there is a need for algorithms which automatically generate optimal collector system designs considering real-life design constraints.

The motivation for this work is to demonstrate the applicability of graph-theoretic and clustering methods and concepts to the optimal wind farm collector system design. As discussed in Chapter 1, the problem of cable layout design for a wind farm collector system can be considered as finding a tree to meet required design characteristics in a graph G = (V, E), where V represents the set of vertices or wind turbines and the substation, and E represents the set of branches or edges connecting the vertices which in this work are the connecting cables. However, in a wind farm with hundreds of turbines, performing an exhaustive search for the optimal tree is computationally expensive since there are numerous trees to be analyzed. In fact by Cayley's tree formula, the number of non-identical trees of order V is V^{V-2}. Both graph theory and data mining are rich resources with a lot of useful tools, concepts, and algorithms. So the question addressed in this work is as follows: *Can graph theory and data mining provide tools to develop algorithms for optimal collector system design?*

An added motivation for this work comes out by comparing the wind farm collector system with electrical distribution systems. Both of these have certain similarities but there are distinct differences as well. Collector systems operate at medium voltages and the substation steps the voltage up to a transmission level voltage while the input to a distribution substation is from a high voltage transmission line and the output is a number of medium voltage feeders. Also, collector system cables are mostly underground whereas distribution system lines are mostly overhead, along streets and hence more exposed to environment and inclement weather conditions with a greater failure rate compared to underground cables. Looped or meshed configurations are common in distribution systems while radial configurations are common in wind farm collector system. The area covered by a distribution system is also typically much larger compared to a collector system. The differences between distribution systems and wind farm collector systems indicate that an algorithm that generates the best layout for a distribution system may not generate the best one for a collector system. Hence arguably further research is required to develop dedicated algorithms for wind farm collector system layout design. Studies have shown that an electric utility's power distribution system can account for up to 60% of capital budget and 20% of operating costs making it a significant expense [100]. Minimizing the cost of the distribution system can be a considerable challenge, as there are thousands of feasible design options to choose from. For these reasons, a lot of research [101, 23, 102] has been focused on development of optimization algorithms to identify the lowest cost distribution configuration and hence the best design. Even with approximations, such programs can help reduce distribution costs by 5 to 10%. The similarities in the problems of distribution system optimization and wind farm collector system optimization indicates that there is scope of improvements on a similar scale by developing better designs for wind farm collector systems as well, further motivating the current work.

3.2 Prior art

There is a lot of scope in the research of optimal design of wind farm collector systems, given that this is a new area. Some related works include electric system design for an off-shore wind farm [103]. The IEEE PES Wind Plant Collector System Design Working Group has addressed issues related to the design of collector systems for wind plants. This group has summarized the important design considerations such as feeder topology, collector design, interconnect and NESC/NEC requirements for wind farms [99], described protection issues of wind farms [104], presented design guidelines based on redundancy, reliability, and economics [105], and summarized collector system design considerations including conductor selection, soil thermal properties, installation methods, and splicing [106]. A mixed-integer programming based formulation has been proposed for optimal collector system design in [107]. One of the recommendations of this work is to apply Steiner trees from graph theory literature to the collector system design problem as future research scope. Also, design constraints such as landowner requirements and limit on turbines on a feeder have not been considered. In addition, the design process is slow. Mixed-integer programming based formulation has also been developed in [108] resulting in a complex formulation and slow convergence. Recently another research group has independently approached the problem of collector system design from the graph-theoretic perspective [109]. However, Steiner trees have not been considered for design.

Also, several crucial design constraints have not been considered. These gaps in available research will be filled in with the current work.

3.3 Clustering-based design

One way to design the collector system for a wind farm is by clustering the wind turbine locations hierarchically in levels [110]. For example, consider the 22-turbine wind farm shown in Figure 41 where the dots represent the turbines and the square represents the collector substation.

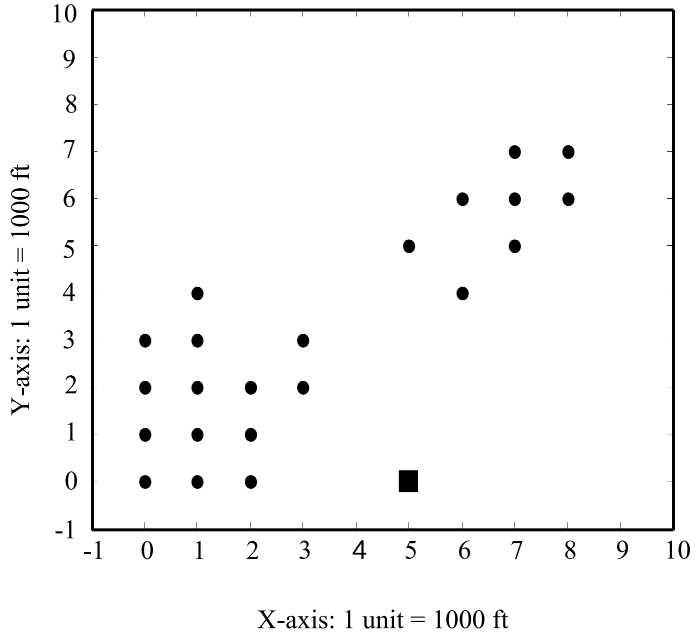


Figure 41: Wind farm with 22 turbines (dots) and one substation (square)

The locations of the wind turbine units are given by the following set of points in Cartesian coordinates: {(0,0), (1,0), (2,0), (0,1), (1,1), (2,1), (0,2), (1,2), (2,2), (3,2), (0,3), (1,3), (3,3), (1,4), (5,5), (6,4), (6,6), (7,5), (7,6), (7,7), (8,6), (8,7)}. Each unit on X and Y axes corresponds to 1000 ft. Thus, the minimum distance between any two turbine locations is 1000 ft. Also, it is assumed that the 1 MW turbines generate power at a lagging power factor of 0.8, with a maximum current per phase projected to be $\frac{1\times10^6}{3\times(34.5\times\frac{10^3}{\sqrt{3}})\times0.8} = 20.92$ A per turbine at a 34.5 kV medium voltage collector system.

At the first level, the wind turbine locations are clustered into a number of groups using Quality-Threshold clustering algorithm as described in Chapter 2. The quality threshold distance for this level is set to 2.5 units = 2500 ft. Two clusters are formed containing respectively 14 and 8 wind turbines. The turbine location in each group that is closest in distance to the substation is called the *first-level cluster representative point* for that group. Cables are laid out from each of the turbine locations of that group to the first-level cluster representative point. The resulting system is shown in Figure 42.

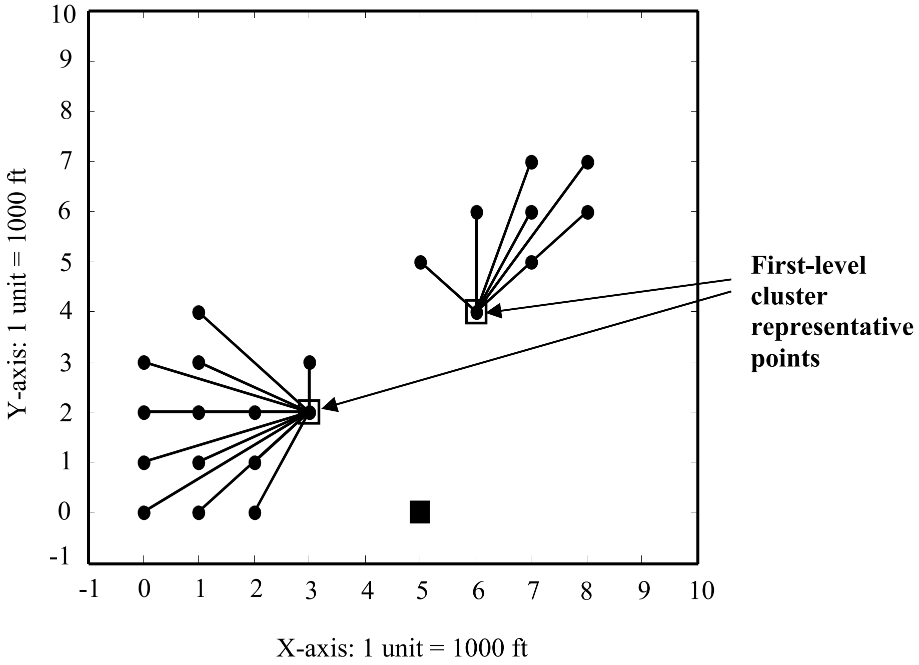


Figure 42: First-level clustering

The choice of quality threshold distance for each stage of clustering is by trial and error. A good initial guess for the first stage clustering is 2 to 3 times the minimum distance in between turbines. For every subsequent clustering stage, the quality threshold distance can be chosen as double that for the previous stage. The initial guess for the threshold distance is modified to achieve specific design targets and meet constraints such as maximum number of turbines in a cluster.

The cable size is selected from the available cable sizes provided in the Appendix to provide required ampacity to carry power. 1/0 conductor-sized cables are sufficient for this level. The total length of 1/0 conductor-sized cables is 47.3701 units per phase = 47370.1 ft per phase. Maximum losses associated with the 1/0 conductor-sized cables is 10.4 kW for all three phases which can be computed using resistance of cables.

In the next level, the first-level cluster representative points are clustered with a quality threshold distance of 4 units = 4000 ft to compute second-level cluster groups and *second-level cluster representative points* similar to the first level as shown in Figure 43. The second-level cluster representative point is marked as a diamond around a black dot. A 4/0 conductor-sized cable is needed at this level. Length of the required cable is 3.6056 units per phase = 3605.6 ft per phase. The power losses associated with this cable is 25.3 kW for three phases.

In the third and last level, a cable carries power from the second-level cluster representative point to the substation as shown in Figure 44. This cable carries power from all the turbines in the wind farm and carries a maximum current of 460.24 A per phase. The cable used for this is one with 1000 kcmil-sized conductors. The length of the required cable is 2.8284 units per phase = 2828.4 ft per phase and losses associated equal 34.6 kW for three phases.

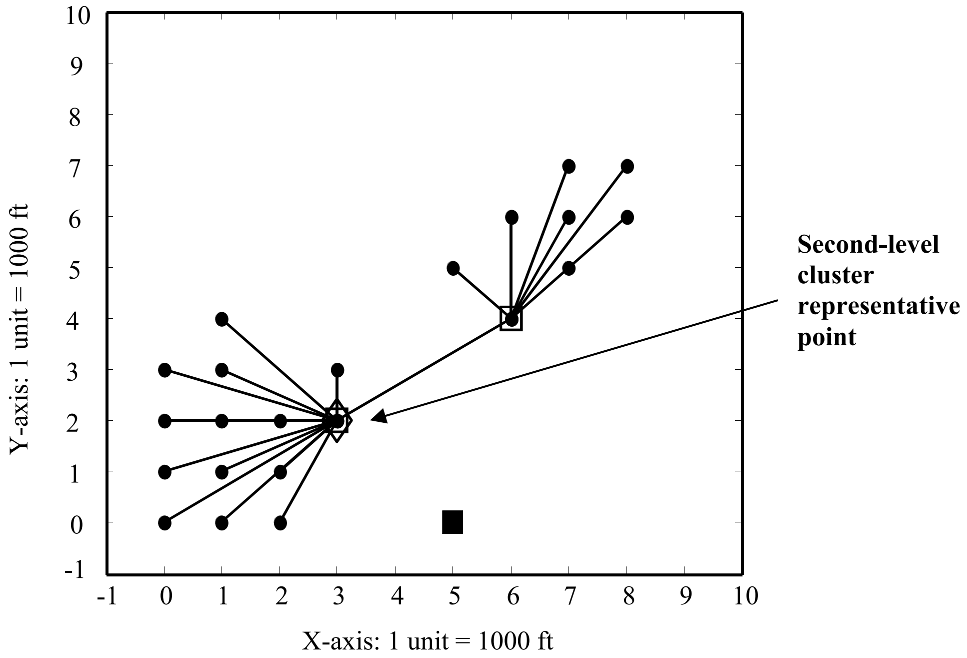


Figure 43: Second-level clustering

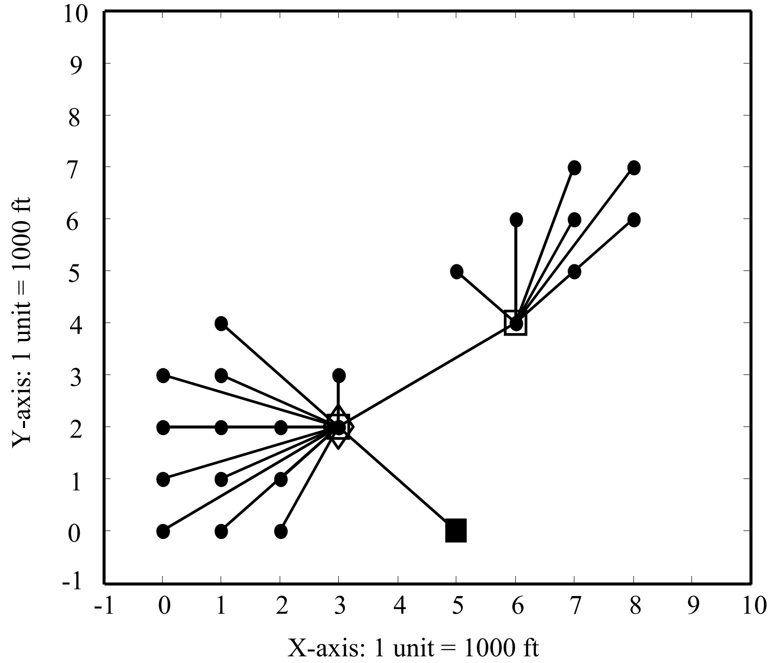


Figure 44: Third-level clustering

A conventional method of connecting wind turbines in a wind farm is the radial system or the daisy chain system. Figure 45 shows a possible radial system cable layout configuration for the wind farm considered in this work. An equivalencing process described in [111] is used to compute the currents in each cable. The cables required in this layout per phase are 24376 ft of 1/0 conductor-sized cables, 1000 ft of 4/0 conductor-sized cables, and 3000 ft of 500 kcmil conductor-sized cables.

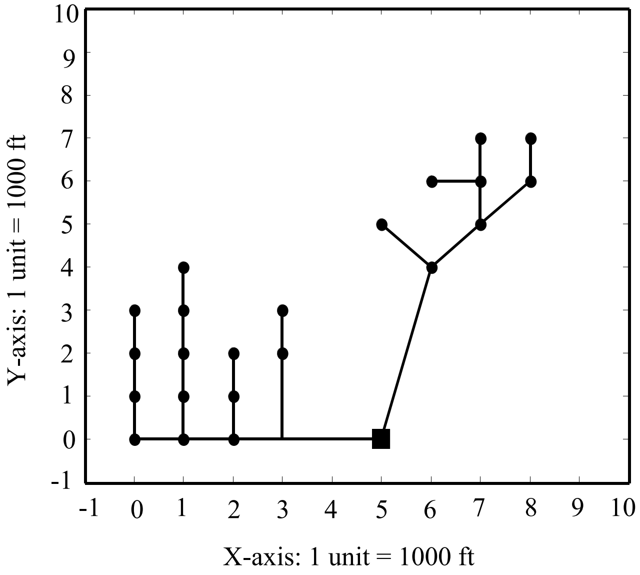


Figure 45: Radial system cable layout

A combined cluster-based and radial layout can also be considered as shown in Figure 46. The first level uses the quality-threshold clustering method followed by a radial interconnection. The total cable length required in this method per phase is 47370 ft of 1/0 conductor-sized cables, and 6943 ft of 4/0 conductor-sized cables.

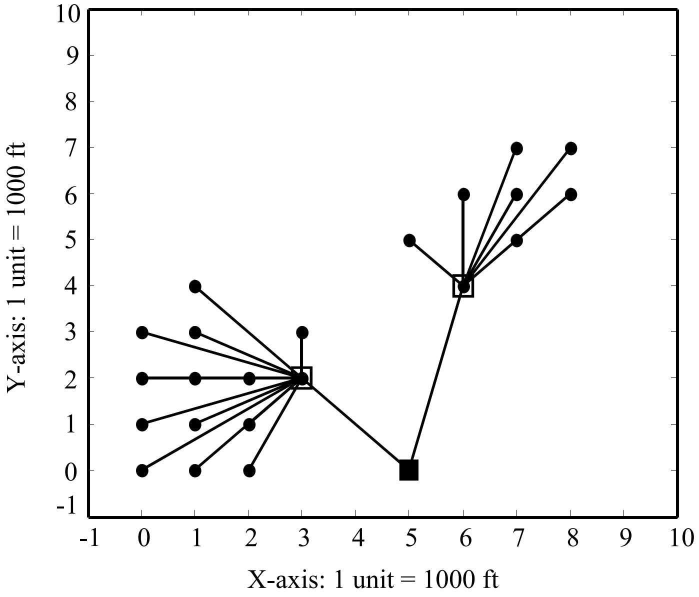


Figure 46: Mixed layout configuration

Reliability metric

To compare the reliability of the three layouts, the metric introduced is the mean number of turbines lost due to a single cable fault. The assumption is that the mean rate to a cable fault is much larger than the mean rate to a cable failure. Thus, at an instant, there can be a fault on any one cable. Let the total number of cables in each layout be N. Let the probability that a fault occurs on a cable be p = 0.1. If the number of turbines lost due to fault in cable-i is given by t_i, then the mean number of turbines lost due to a fault on a cable, μ can be given by:

$$\mu = \sum_{i=1}^{N} t_i * p * (1-p)^{N-1} = P * \sum_{i=1}^{N} t_i \tag{3.1}$$

Here P is a probability multiplier. Note that the lower the μ, the higher the reliability of the design. A comparison of the three different configurations is provided in Table 1. It is assumed that the cost of digging the land and entrenching the cables i.e. trenching costs are $15/ft. It is noted that the cluster-based methods improve the reliability of the systems, lowers power loss due to use of higher-sized cables of lower resistance compared to the pure radial system. However, the cable costs and total trenching lengths are significantly higher. This can be considered as a major disadvantage. To address this, the collector system design problem has been reconsidered from the standpoint of reducing the total trenching length [112]. This is where a graph-theoretic perspective was found to be extremely effective.

Table 1: Comparison of the layouts

Layout based on	Reliability (mean no. of turbines lost due to single cable fault)	Power loss (kW)	Total trenching length (ft)	Total cable costs ($)
Clustering	0.545	70.42	161,412	380,386
Radial system	0.916	123.9	28,376	167,880
Combined cluster-radial	0.458	99.9	54,313	306,280

3.4 Collector system design with minimum total trenching length and application of spanning trees

When the objective is to minimize the total trenching length of the collector system and hence to find a tree in the collector system graph with minimum total length of edges, the problem is relatively easy and requires a minimum spanning tree algorithm to generate the required configuration.

The minimum spanning tree (MST) algorithm

The first algorithm for finding an MST was developed by Czech scientist Borůvka in 1926. Several algorithms have since been developed and two of the most common algorithms are Prim's algorithm and Kruskal's algorithm [113, 114, 115, 116]. Both of these algorithms run in polynomial time. Prim's algorithm was developed in 1930 by Czech mathematician Jarník and later independently by computer scientist Robert C. Prim in 1957 and rediscovered by Edsger Dijkstra in 1959 [117]. Therefore it is also sometimes called the Dijkstra-Jarnik-Prim or DJP algorithm. This is the algorithm that has been used for computing an MST in this dissertation.

Three sets of branches are defined:

- Set I – The branches definitely assigned to the tree under construction (form a subtree).
- Set II – The branches from which the next branch to be added to Set I will be selected.
- Set III – The remaining branches (rejected or not yet considered).

The nodes are subdivided into two sets:

- Set A – The nodes connected by the branches of Set I.
- Set B – The remaining nodes (one and only one branch of Set II will lead to each of these nodes).

The algorithm starts by choosing an arbitrary node as the only member of Set A, and by placing all branches that end in this node in Set II. To start with, Set I is empty. From then onward the following two steps are performed repeatedly.

- *Step 1* – The shortest branch of Set II is removed from this set and added to Set I. As a result one node is transferred from Set B to Set A.

- *Step 2* – The branches leading from the node which has just been transferred to Set A to the nodes that are still in Set B are considered. If the branch under consideration is longer than the corresponding branch in Set II, it is rejected; if it is shorter, it replaces the corresponding branch in Set II, and the latter is rejected. This is followed by a return to Step I and a repetition of the process until Set II and Set III are empty. The branches in Set I form the tree required.

Figure 47 and Table 2 illustrate the algorithm on a graph of five nodes. Initially say the node 5 is selected. Thus Set A contains node 5 and Set B contains nodes 1, 2, 3, and 4. The possible branches from node 5 to the other nodes form Set II. The smallest branch in this set is (5, 3) which in the next stage is moved to the Set I. This moves node 3 to Set A. The branches from node 3 to the nodes still in Set B i.e. (3, 1), (3, 2), (3, 4) are considered. (3, 1) is compared with (5, 1). Since (5, 1) is smaller, this branch is retained. Similarly, (3, 2) is compared with (5, 2), and (3, 4) is compared with (5, 4) and found that (5, 2) and (5, 4) are smaller and hence retained as elements of Set II. These are highlighted in the table. The smallest among (5, 1), (5, 2), and (5, 4) is computed. This is found as (5, 2) and is moved to Set I. Accordingly, node 2 is moved from Set B to Set A. Next, (2, 1) is compared with (5, 1) and (2, 4) compared with (5, 4) and found that (2, 1) and (5, 4) are smaller and are retained. Between (2, 1) and (5, 4), (2, 1) is smaller so this branch is moved to Set I and accordingly node 1 is moved from Set B to Set A. Now the branches compared are (1, 4) and (5, 4), of which (5, 4) is smaller. Thus (5, 4) is added to Set I and the last remaining node from Set II, i.e. node 4 is moved to Set I.

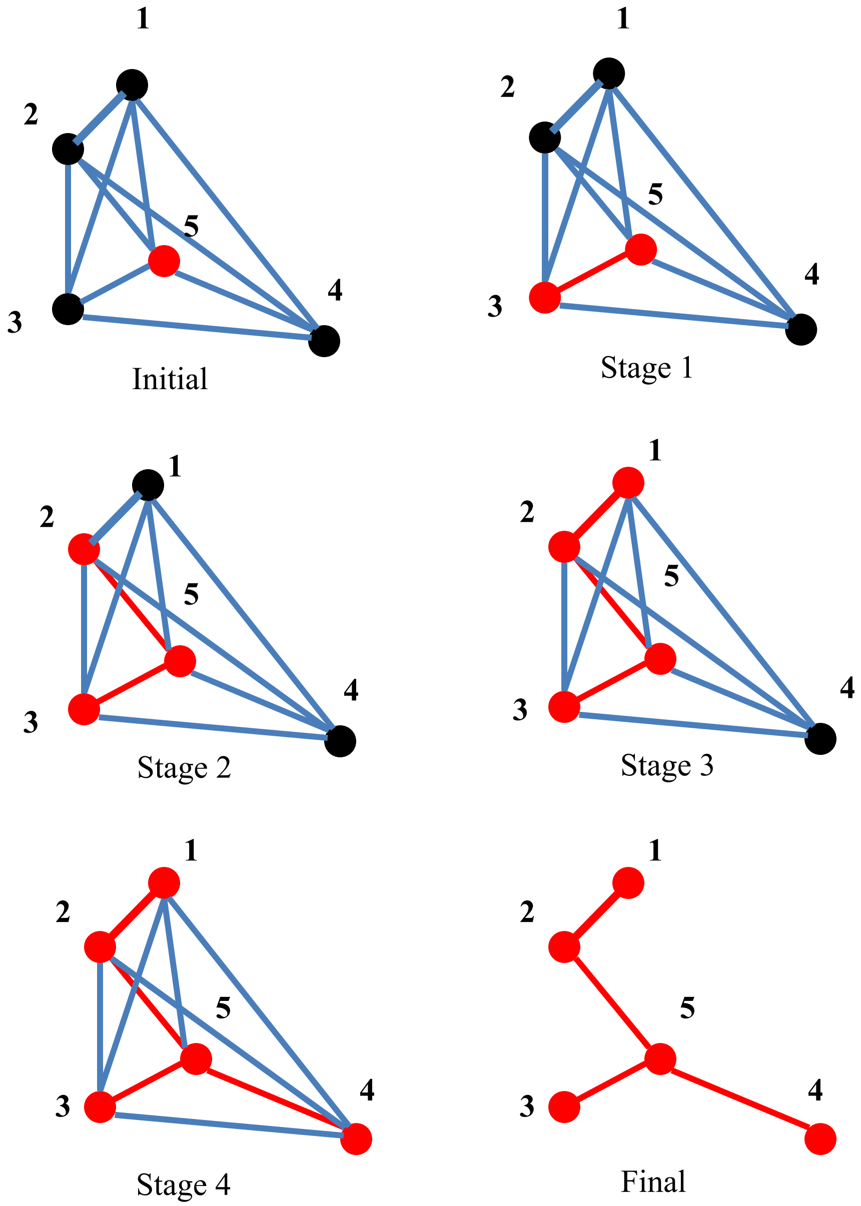


Figure 47: Illustration of the minimum spanning tree algorithm

Table 2: Illustration of the MST algorithm

Stage No.	Set I	Set II	Set A	Set B
Initial	[]	(5,1) (5,2) (5,3) (5,4)	5	1 2 3 4
1	(5,3)	(3,1) or (5,1) (3,2) or (5,2) (3,4) or (5,4)	5 3	1 2 4
2	(5,3) (5,2)	(2,1) or (5,1) (2,4) or (5,4)	5 3 2	1 4
3	(5,3) (5,2) (2,1)	(1,4) or (5,4)	5 3 2 1	4
4	(5,3) (5,2) (2,1) (5,4)	[]	5 3 2 1 4	[]

The application of the MST algorithm for collector system design with minimum total length is demonstrated with an example.

A real-life wind farm consisting of 66 wind turbines such as is common in the flat terrain of the American Midwest is considered. Figure 48 shows the locations of the 66 wind turbines with dots and the substation with a square. Figure 49 shows the actual connection diagram as it exists in the considered section of the real wind farm.

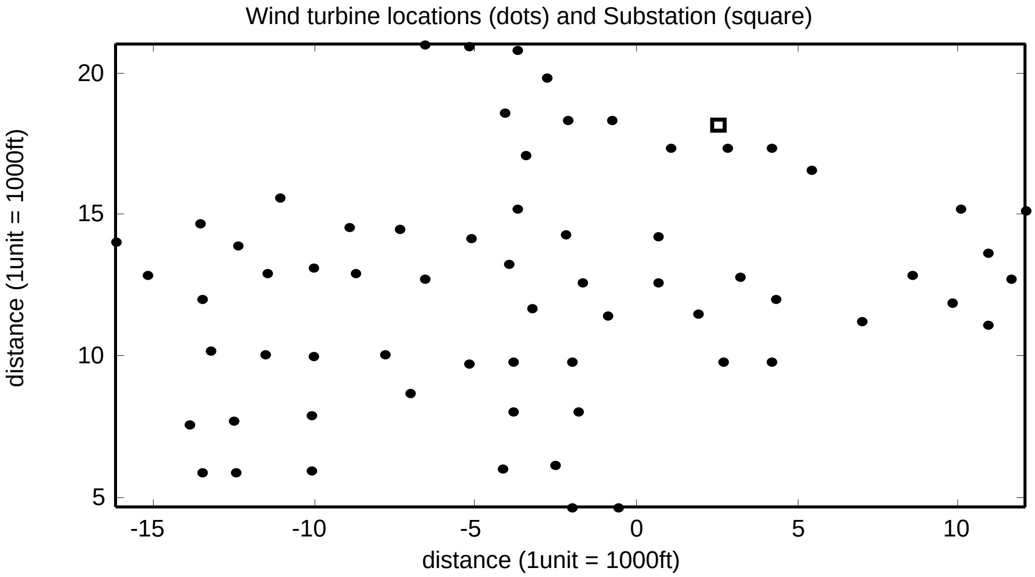


Figure 48: Location of wind turbines and the substation

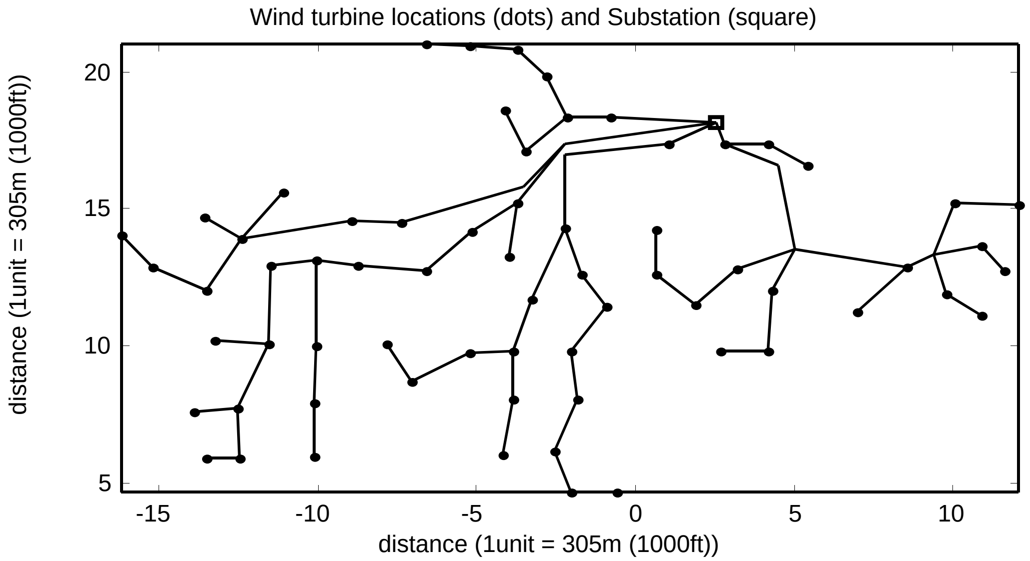

Wind turbine locations (dots) and Substation (square)
distance (1unit = 305m (1000ft))
distance (1unit = 305m (1000ft))
20
15
10
5
-15
-10
-5
0
5
10

Figure 49: Actual layout

The MST algorithm is applied on all of the turbine locations and the substation location resulting in the configuration shown in Figure 50. This layout is denoted as:

<u>*Case I*</u> *– Minimum total length configuration without any constraints*

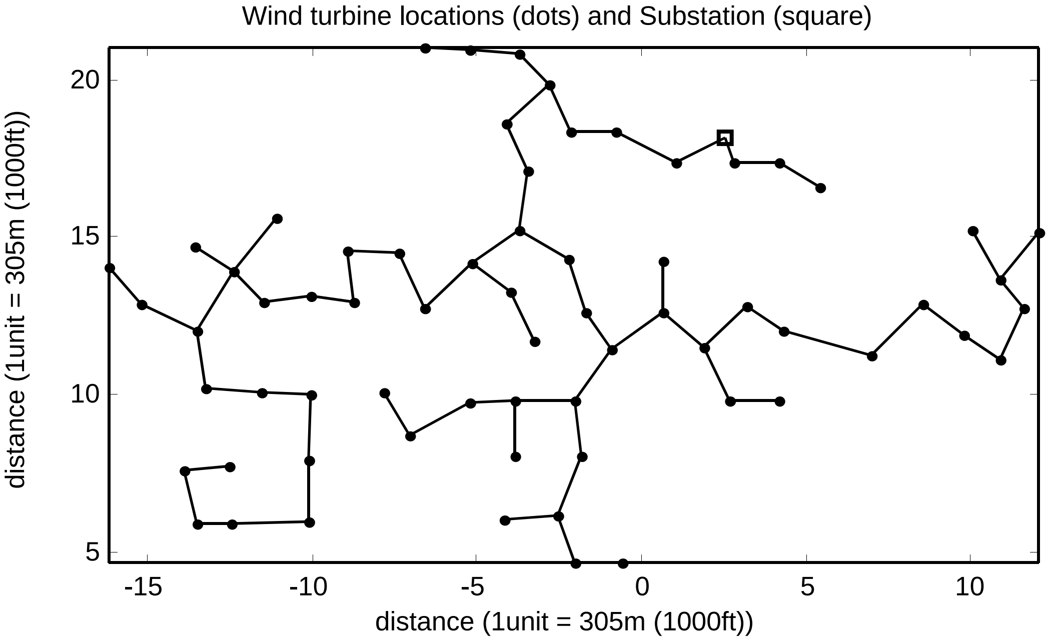


Figure 50: Layout with MST

As can be seen there are two feeders coming into the substation, one carrying the power generated from 3 turbines and the other from 66 turbines. None of the cables listed in Table 14 of the Appendix can provide the required ampacity as loading increases closer to the substation. Thus multiple circuits of cables have to be used. As a result in this layout the total cable length is larger than the total trenching length.

3.5 Collector system design with flexibility of introducing intermediate splice nodes

The MST algorithm only uses the input nodes to compute the tree. If an additional degree of freedom is introduced in the design space by allowing the creation of intermediate splice nodes similar to Steiner's vertices as introduced in Chapter 1, the total length obtained by the minimum spanning tree algorithm can be further reduced. Then the problem of optimal cable layout design to minimize the total length becomes a Steiner tree problem.

Steiner tree

The Euclidean Steiner tree problem dates back to the 17th century when Fermat proposed the problem: Find a point in a plane, the sum of whose distances from three given points is minimal [118]. This simple problem prompted over one hundred years of study before Heinen proposed a complete solution in 1834. After another hundred years the Fermat problem gained further popularity among mathematicians and received a name change. After the publishing of *What Is Mathematics?* [118] in 1941 by Courant and Robbins, the Fermat problem and its generalizations were renamed the Steiner tree problem after Steiner, a professor at the University of Berlin who made great contributions to mathematics [119]. The real-world applications of Steiner trees and their generalizations are numerous. Routing of heating and plumbing pipes inside a building, trace layout between logic gates in circuits to minimize propagation time, determining the pathway for oil or natural gas pipelines that are as short as possible while considering the terrain they cross or avoid, and other minimal networks are a few of the many examples.

In 1977, Garey, Graham, and Johnson showed that the Euclidean Steiner tree problem for general N vertices is NP-Hard [120], and hence it is not known whether an optimal solution can be found by using a polynomial-time algorithm. Several researchers have worked in the area of computing exact Steiner trees. Dave Warme, Pawel Winter, and Martin Zachariasen have made publicly available their world champion algorithm for computing optimal Steiner trees. This package, entitled GeoSteiner [121, 122], computes optimal Euclidean and rectilinear Steiner trees. There have been some other significant contributions in the area of computing exact Steiner trees as well [123, 124, 125, 126].

However, the GeoSteiner algorithm and all known exact algorithms for the Euclidean Steiner tree problem require exponential time. So the general consensus is to use heuristics and approximation algorithms. The goal is to connect the vertices by edges of minimum total length in such a way that any two points may be interconnected by line segments either directly or via intermediate points similar to but not the same as Steiner points and line segments.

In order to gauge performance of Steiner heuristics and approximation algorithms the *Steiner ratio* has been introduced. It is defined as the largest possible ratio between the total length of an MST over all vertices and the total length of a minimum Steiner tree. For the Euclidean Steiner tree problem, Gilbert and Pollak, in 1968, [127] speculated that the best obtainable ratio over all Steiner trees equals ratio for the equilateral triangle, i.e $2/\sqrt{3} \approx 1.15$. In 1992, Du and Hwang [128] presented a formal proof of this conjecture. An improvement of this order is difficult to achieve given that the problem is NP-hard. In 2002, Dreyer and Overton [129] proposed two heuristic-based algorithms for computing Steiner trees. The first algorithm is relatively faster but does not generate results as good as the other. Whereas, the second algorithm generates better results but is extremely slow, taking as long as 10 minutes on 10 points, 3 hours on 10 points, and a day on 100 points [129]. Thus each algorithm has its own advantages and disadvantages, and it is concluded that the choice of an algorithm should be dependent on the application. In this dissertation, another heuristic-based algorithm is presented that is simple and caters to the requirements of the problem of wind farm collector system design.

Heuristic algorithm for introducing splice nodes

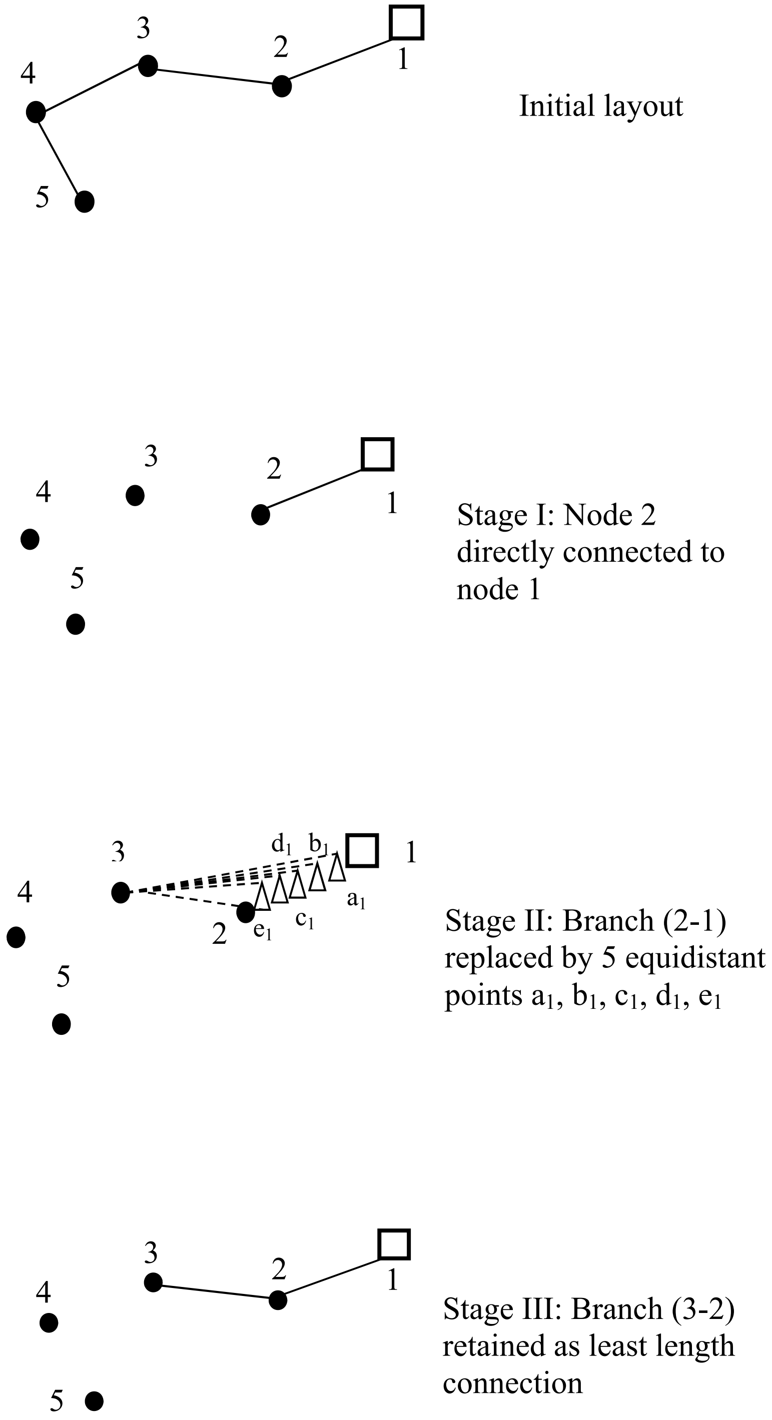
This algorithm introduces intermediate splices improving the results obtained by minimum spanning tree algorithm. The algorithm has the following steps [112]:

- *Step 1* – The algorithm is initialized by the connected graph obtained by applying minimum spanning tree algorithm on all nodes including the substation. Four sets are defined as follows: Set 1 = {the substation node}, Set 2 = {all other nodes}, Set 3 and Set 4 are null sets.
- *Step 2* – A node is selected from Set 2, called Node-A, such that it is connected to a node, called Node-B, in Set 1, as computed by the minimum spanning tree algorithm. The least distance between Node-A and the nodes in Set 1 is computed. If the least distance corresponds to Node-B, the existing branch in between Node-A and Node-B is retained and moved to Set 3. However, if the least distant node in Set 1 is not Node-B, it (the least distant node) is defined a "splice node." The splice node location is copied to the Set 4. A branch is created between the Node-A and the splice node which replaces the existing branch between Node-A and Node-B in Set 1 and is moved to Set 3.
- *Step 3* – Multiple equidistant points are created on the just moved branch to introduce new nodes. In this work, five equidistant points are chosen. The Node-A and the five newly created nodes are moved to Set 1.

Steps 2 and 3 are repeated till Set 2 is empty. This indicates that all the nodes have been considered. Finally Set 4 contains the newly created splice nodes, and Set 3 contains the branches that form a tree with original set of nodes and introduced splice nodes.

An important point to note in this algorithm is that Set 1 is initialized with the substation node. The characteristics of spanning trees having only one unique path between any pair of nodes ensures that a node from a non-empty Set 2 can always be found that has a connection with a node in Set 1. The modified tree with increased number of nodes and branches grows outward from the substation node with every iteration, adding intermediate splice nodes as necessary to reduce the total spanning length.

The process in steps 1 to 3 has been illustrated in Figure 51 and Table 3. A graph with five nodes (four turbine nodes indicated by dots and one substation node indicated by a square) is initialized by a MST algorithm. At this stage, the Set 1 consists of the substation node, i.e. node 1, and Set 2 consists of the other four nodes, i.e. nodes 2, 3, 4, and 5. At the first stage, Node-B = node 1, and Node-A = node 2, since it is the only node connected to a node in Set 1. The least distance between Node-A and a node in Set 1 corresponds to Node-B, since Set 1 contains only node 1 at this stage. Hence the branch between these is retained as shown in Stage I. This branch is replaced by five equidistant nodes as shown in Stage II and these are moved to Set 1. The Set 1 now contains the five newly created nodes, node 1, and node 2. Node 2 is removed from Set 2. Set 3 contains the branch between node 1 and node 2 and Set 4 is empty. At the next stage, node 3 is selected and distance between this node and the seven nodes in Set 1 are computed. It is found that node 2 is the least distant from node 3, so the branch between these is retained and no splice nodes introduced as shown in Stage III. This process is followed till the first splice node (indicated by a diamond) is found in Stage VII, and the Set 2 becomes empty.



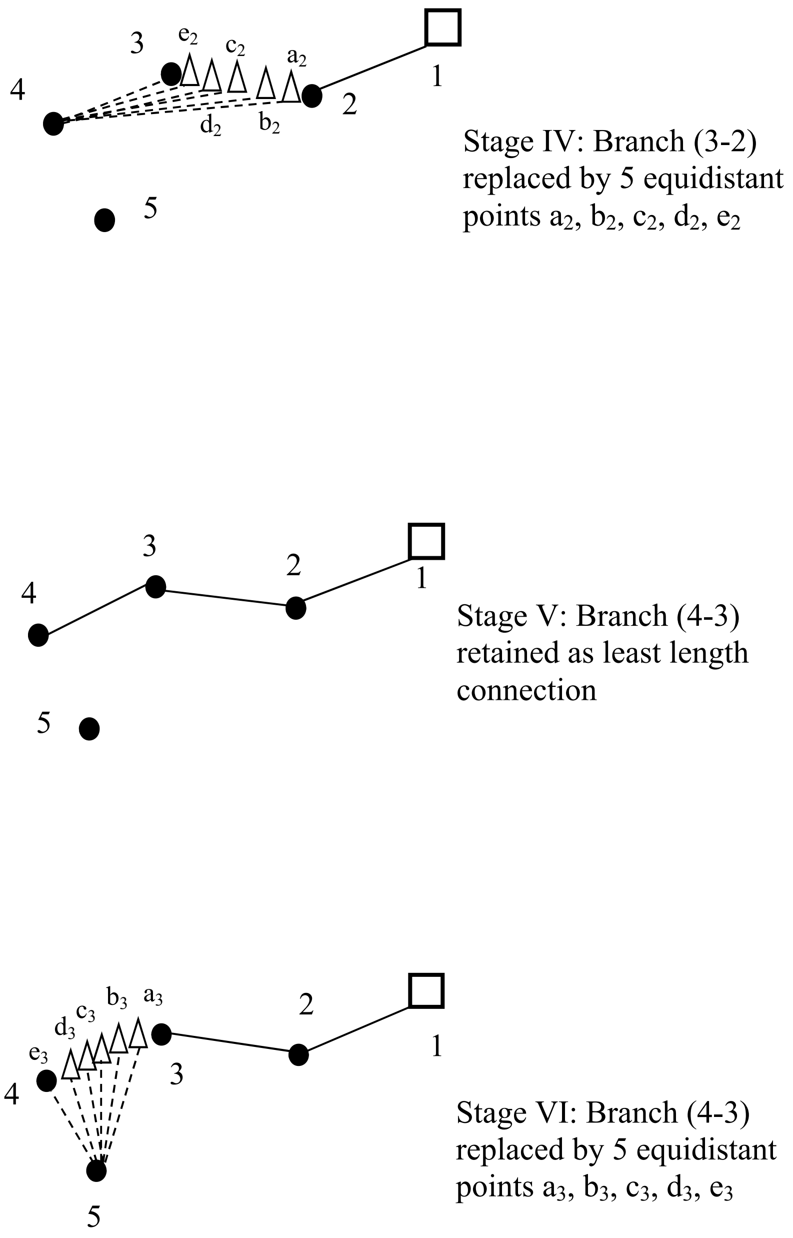
Initial layout

Stage I: Node 2 directly connected to node 1

Stage II: Branch (2-1) replaced by 5 equidistant points a_1, b_1, c_1, d_1, e_1

Stage III: Branch (3-2) retained as least length connection

Figure 51: Illustration of the splice introduction algorithm



Stage IV: Branch (3-2) replaced by 5 equidistant points a_2, b_2, c_2, d_2, e_2

Stage V: Branch (4-3) retained as least length connection

Stage VI: Branch (4-3) replaced by 5 equidistant points a_3, b_3, c_3, d_3, e_3

Figure 51: Illustration of the splice introduction algorithm (cont.)

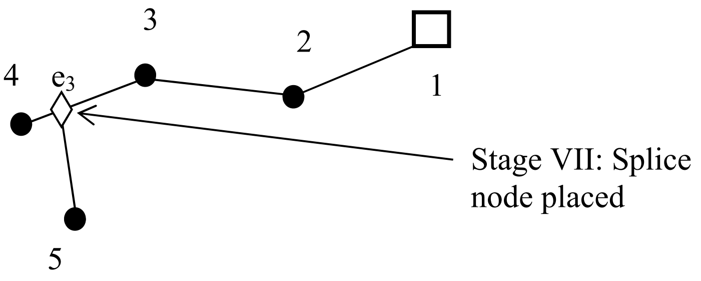


Figure 51: Illustration of the splice introduction algorithm (cont.)

Table 3: Illustration of the splice introduction algorithm – transition of the sets

Stage No.	Set 1	Set 2	Set 3	Set 4
Initial	{1} (substation)	{2, 3, 4, 5}	{[]}	{[]}
I	{1, 2}	{3, 4, 5}	{(2,1)}	{[]}
II	{1, 2, a_1, b_1, c_1, d_1, e_1}	{3, 4, 5}		
III	{1, 2, 3}	{4, 5}	{(2,1), (3,2)}	{[]}
IV	{1, 2, 3, a_2, b_2, c_2, d_2, e_2}	{4,5}		
V	{1, 2, 3, 4}	{5}	{(2,1), (3,2), (4,3)}	{[]}
VI	{1, 2, 3, 4, a_3, b_3, c_3, d_3, e_3}	{5}		
VII	{1, 2, 3, 4, 5}	{[]}	{(2,1), (3,2), (4,3), (5,e_3)}	{e_3}

The algorithm has been further illustrated with a flow chart in Figure 52.

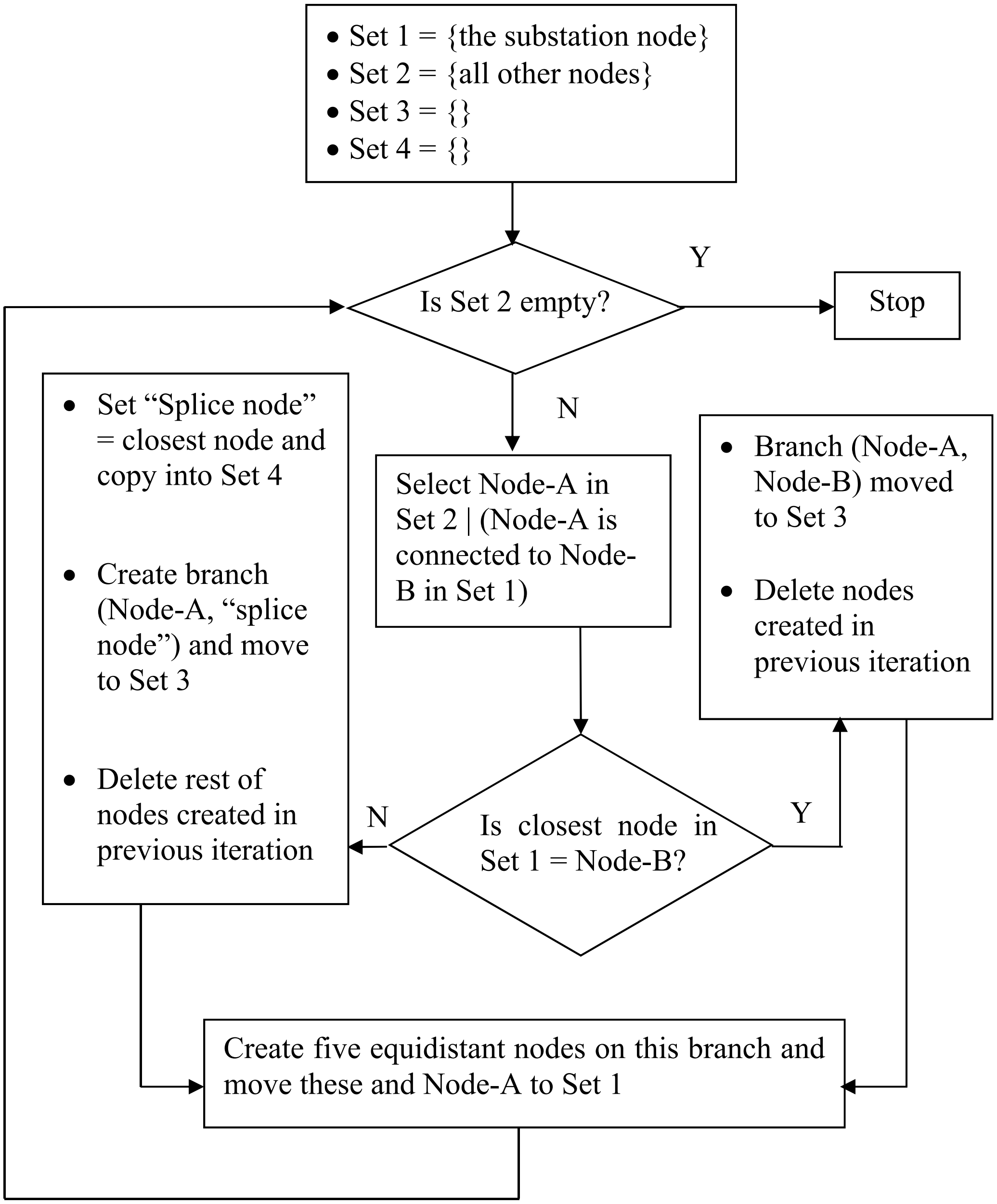

• Set 1 = {the substation node}
• Set 2 = {all other nodes}
• Set 3 = {}
• Set 4 = {}
Is Set 2 empty?
Y
Stop
N
Select Node-A in Set 2 | (Node-A is connected to Node-B in Set 1)
• Set "Splice node" = closest node and copy into Set 4
• Create branch (Node-A, "splice node") and move to Set 3
• Delete rest of nodes created in previous iteration
• Branch (Node-A, Node-B) moved to Set 3
• Delete nodes created in previous iteration
Is closest node in Set 1 = Node-B?
N
Y
Create five equidistant nodes on this branch and move these and Node-A to Set 1

Figure 52: Flow chart showing algorithm for introducing splice nodes

Applying the proposed algorithm on the wind farm results in the layout shown in Figure 53. The total trenching length decreases by a factor of about 1.01 and 8 intermediate splice nodes are introduced. This layout is denoted as follows:

<u>Case II</u> – Reduced total length configuration compared to Case I by the introduction of intermediate splice nodes without limits on the maximum number of turbines on a feeder and without trenching constraints.

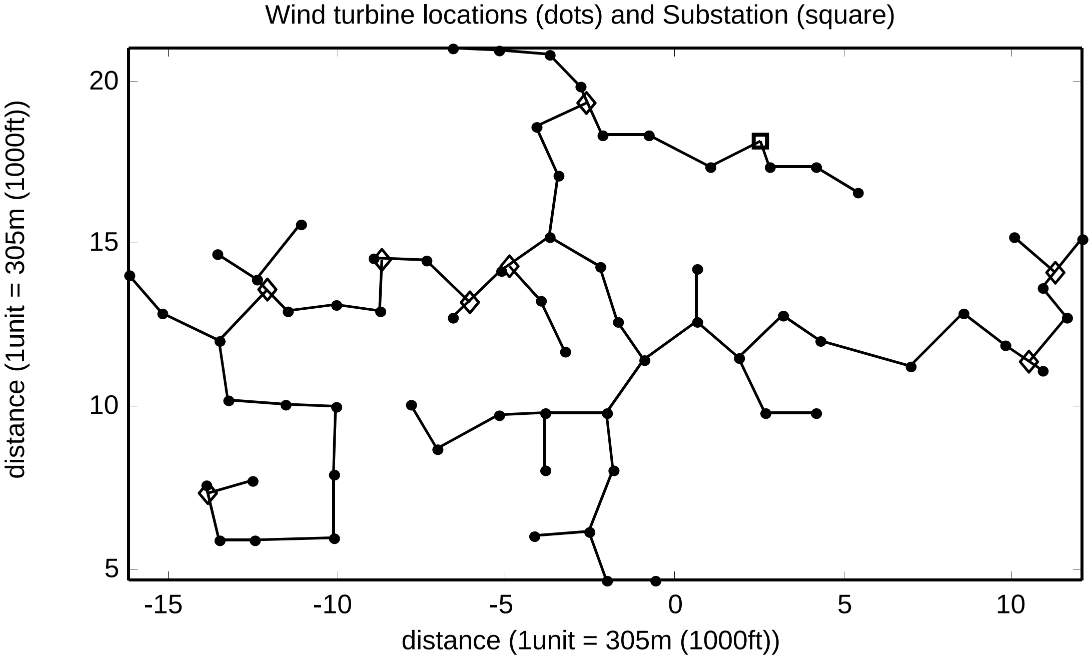


Figure 53: With introduction of splices (diamonds)

Similar to Case I (Figure 50), the total trenching length is lower than the total cable length because multiple cable circuits have to be used to provide required ampacity closer to the substation. The disadvantages with this configuration are that there are two feeders coming into the substation each connecting to 3 and 63 turbines respectively. In a practical collector system, there may be limits on the maximum number of turbines on a feeder depending on the maximum size and ampacity of available cables. To address this problem in the design, the wind turbine locations can be clustered using K-means clustering and the number of turbines in each cluster restricted to the maximum limit by an algorithm. This algorithm is described in the following section.

3.6 Applying clustering to limit maximum number of turbines on a feeder

Feeder cables have limited current carrying capacity. This limits the number of turbines that can be connected to a feeder. Assuming a limit of Nmax turbines on a feeder, the algorithm for incorporating this constraint in the automatic cable layout design is as follows [112]:

- *Step 1* – The algorithm is initialized with a value of the number of clusters required. This value is calculated by finding the ratio of the total number of turbines and Nmax.
- *Step 2* – The turbines are clustered with the K-means clustering algorithm and the size of the largest cluster is found.
- *Step 3* – If the size of the largest cluster exceeds the prespecified Nmax, the value of the number of required clusters is increased by unity and Step 2 and Step 3 are redone. If the size of the largest cluster is within limit, the value for the required cluster number is finalized and the turbine nodes are grouped according to K-means clustering algorithm.
- *Step 4* – For every cluster, the nodes corresponding to the turbines and the substation location are grouped in a set called Set A. Next the minimum spanning tree algorithm is applied on elements of Set A.

Depending on a visual inspection of the geographical distribution of wind turbines, the Step 2 can be varied to cluster the actual wind turbine locations or the radial angles of turbine locations at substation location.

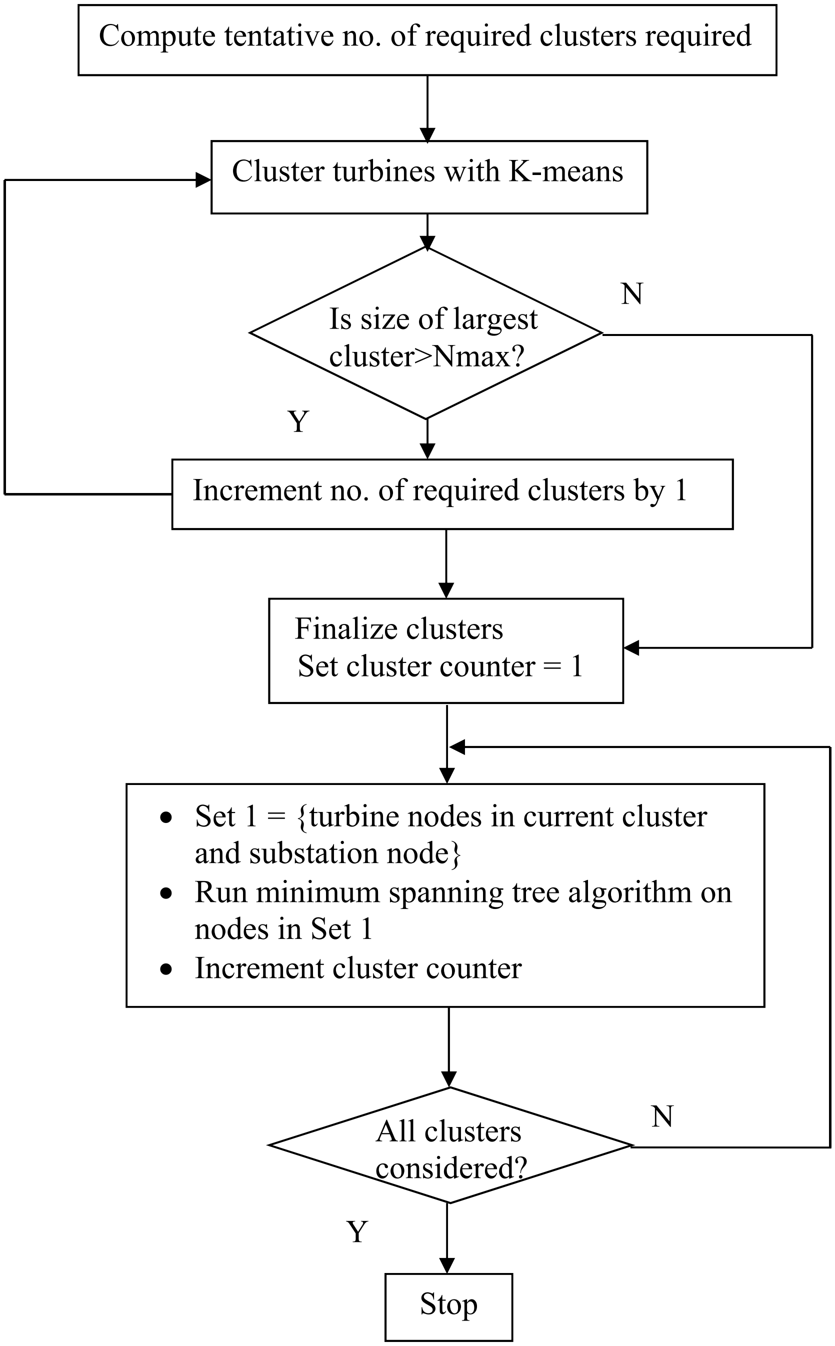

Compute tentative no. of required clusters required
Cluster turbines with K-means
Is size of largest cluster>Nmax?
N
Y
Increment no. of required clusters by 1
Finalize clusters
Set cluster counter = 1
• Set 1 = {turbine nodes in current cluster and substation node}
• Run minimum spanning tree algorithm on nodes in Set 1
• Increment cluster counter
All clusters considered?
N
Y
Stop

Figure 54: Algorithm for enforcing max. limit on no. of turbines

To ensure the convergence of the K-means clustering algorithm to the same clusters in each run, the starting centroids are selected by the algorithm in a way that the K centroids are as far away from each other as possible. For example, when the turbines are clustered by radial angles, the turbines are first sorted by increasing or decreasing angle values and then K equispaced locations are selected from this list to give the K starting centroids.

Applying this algorithm on the example wind farm results in:

Case III – Minimum total length configuration with limits on maximum number of turbines on a feeder and without introducing intermediate splice nodes or considering landowner constraints. Clustering by geographical locations of the turbines results in Figure 55. Clustering by radial angles of turbine locations at the substation results in Figure 56.

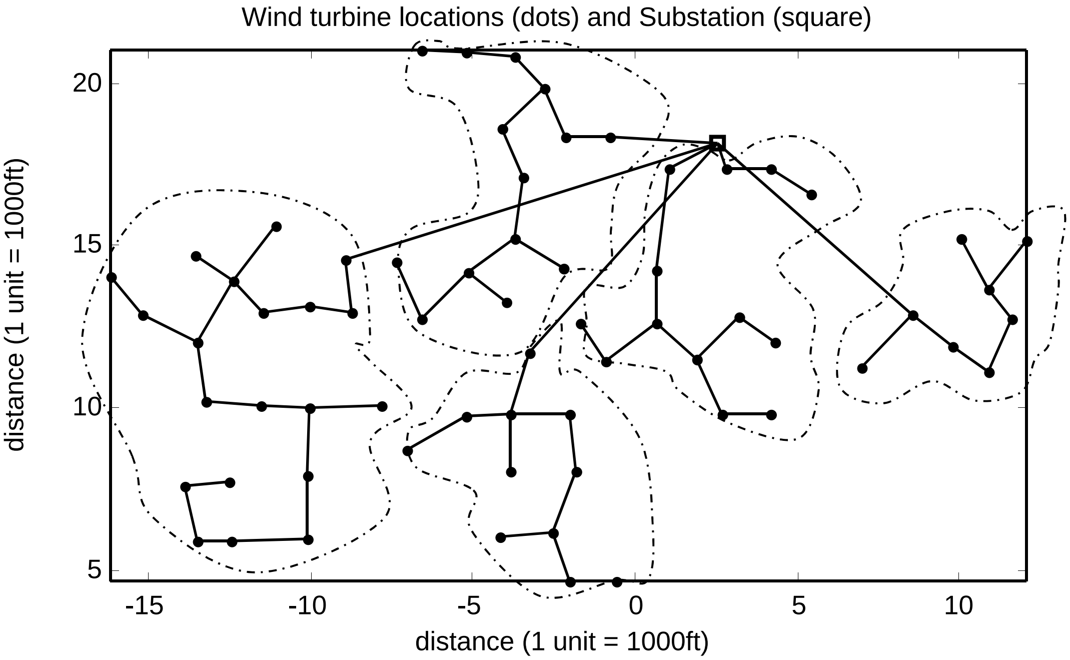


Figure 55: Clustering based on geographic locations

In each figure the dash-dot lines show the clusters. The number of clusters is determined so that the maximum number of turbines in each cluster is less than or equals 20 turbines in this work. However, the maximum limit can be varied according to the design requirement. As can be seen, in Figure 55, five clusters are formed with 20 turbines in the largest cluster, and in Figure 56, five clusters are formed with 17 turbines in the largest cluster.

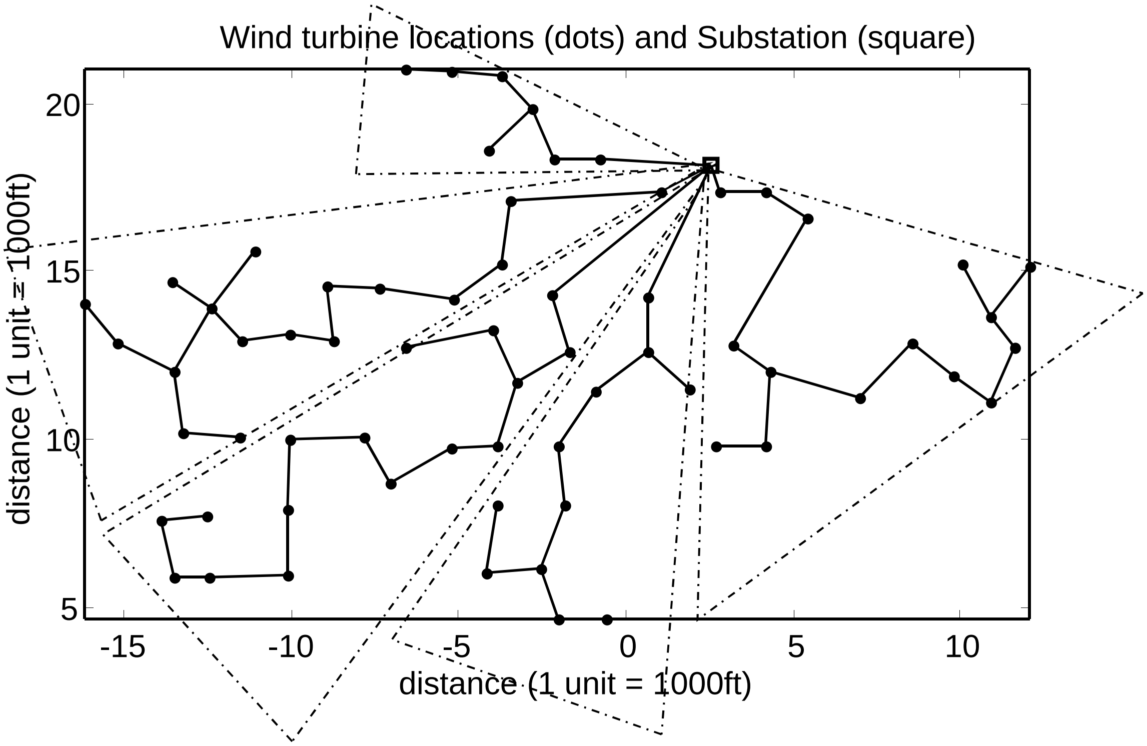


Figure 56: With limit on max no. of turbines on a feeder and radial clustering

It should be noted here that using angles subtended by turbines at substation as the criteria for clustering is an innovation that is particularly applicable for the wind farm collector system design given its radial structure. This can also be seen for the example wind farm, where the geographical distribution of wind turbine locations are such that Figure 56 (and hence a radial clustering) depicts a better choice for clustering compared to Figure 55. The disadvantage of

Figure 55 for this wind farm is that some cables connecting turbines of a cluster to the substation may pass through or very close to turbines in another cluster thus requiring application of heuristics to improve the design generated by the algorithm. However, clustering based on turbine locations can be useful in designing collector system configurations in a wind farm where the terrain and other factors lead to placement of groups of wind turbines at large distances from each other. The choice whether turbine locations or radial angles are to be clustered depends on the specific wind farm under study and can be done by a visual inspection of relative locations. Once the clusters are defined, the MST algorithm is run on individual clusters to get the least total length layout under this constraint. In both Figure 55 and Figure 56, the total cable length equals the total trenching length. This is because the maximum number of turbines on a feeder is limited. So one of the cable sizes from Table 14 (see Appendix) could be assigned to each of the cables; even those close to the substation, without violating the ampacity limits.

Applying the splice introduction algorithm along with the algorithm for limiting the number of turbines on a feeder results in Figure 57. This layout is represented as:

Case IV – Reduced total length configuration by introduction of intermediate splice nodes and considering limits on the maximum number of turbines on a feeder but without trenching constraints.

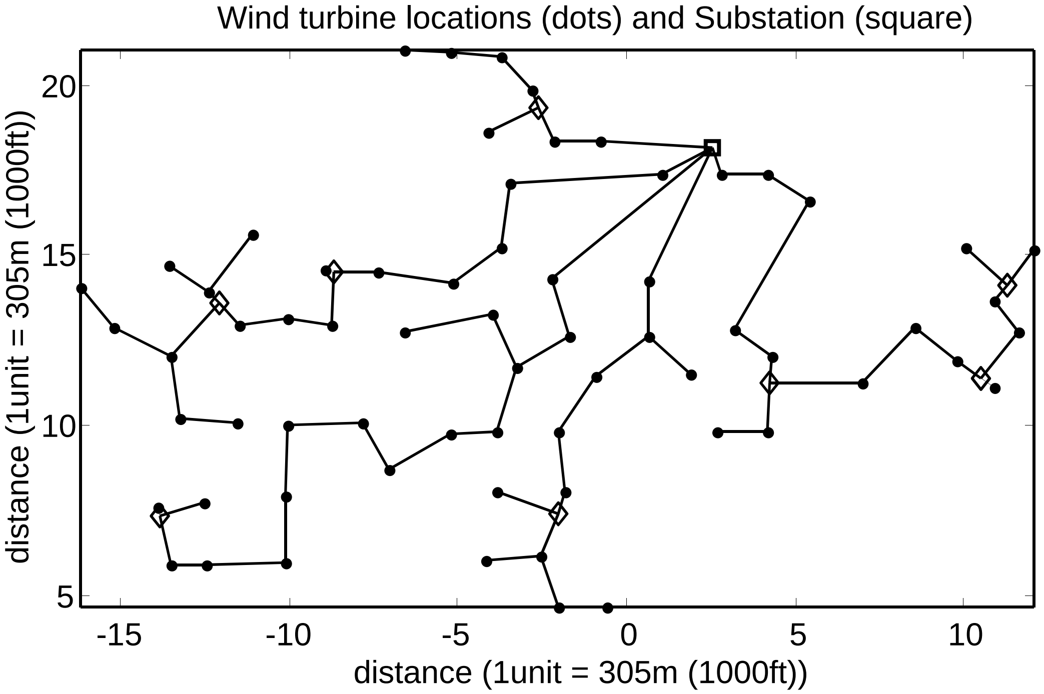


Figure 57: Algorithm with splices and restriction of max no. of turbines on a feeder

3.7 Applying graph theory to address design constraints of trenching restrictions

The geographical area of the wind farm is sometimes restricted with respect to excavating the land and/or burying cables, i.e. trenching. Such restrictions might come from the owner of the land area or be due to presence of a water body etc. The following algorithm takes into account these restrictions while designing the cable layout system. An assumption made is that the restricted areas are convex polygons and if an area is not a convex polygon, it can be approximated by one.

This algorithm is based on a modified version of the minimum spanning tree algorithm. Three sets are defined:

- Set I – The branches definitely assigned to the tree under construction (they will form a subtree),
- Set II – The branches from which the next branch to be added to Set I will be selected,
- Set III – The remaining branches (rejected or not yet considered).

Each branch in Set II has a certain length typically computed by the Euclidean distance between the nodes at the two ends of the branch. However, in this algorithm, each branch is checked for possible intersection with the polygon representing restricted area where trenching is not allowed. If such intersection is found, as shown in Figure 58, then the length of the branch is incremented to a very large value.

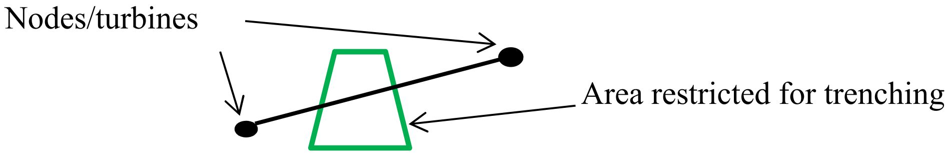


Figure 58: Branch between two nodes crossing area restricted for trenching (Distance between nodes artificially increased to a large value)

The nodes are subdivided into two sets:

- Set A – The nodes connected by the branches of Set I,
- Set B – The remaining nodes (one and only one branch of Set II will lead to each of these nodes).

The algorithm starts by choosing an arbitrary node as the only member of Set A, and by placing all branches that end in this node in Set II. To start with, Set I is empty. From then onward the following two steps are performed repeatedly.

- *Step 1* – The shortest branch of Set II is removed from this set and added to Set I. As a result one node is transferred from Set B to Set A.
- *Step 2* – The branches leading from the node which has just been transferred to Set A to the nodes that are still in Set B are considered. If the branch under consideration is longer than the corresponding branch in Set II, it is rejected; if it is shorter, it replaces the corresponding branch in Set II, and the latter is rejected. This is followed by a return to Step 1 and a repetition of the process until Set II and Set III are empty. The branches in Set I form the tree required.

This algorithm is applied on the example wind farm. The turbine locations are first clustered based on radial angles followed by applying algorithm for trenching constraints. This is followed by applying the algorithm for introducing splice nodes. The resulting configuration is shown in Figure 59. The polygons in green represent the land areas where trenching is not allowed or difficult, enforced by a large cost of laying out any possible cables crossing these areas.

Case V – Reduced total length configuration by introduction of intermediate splice nodes and considering limits on the maximum number of turbines on a feeder and trenching restrictions.

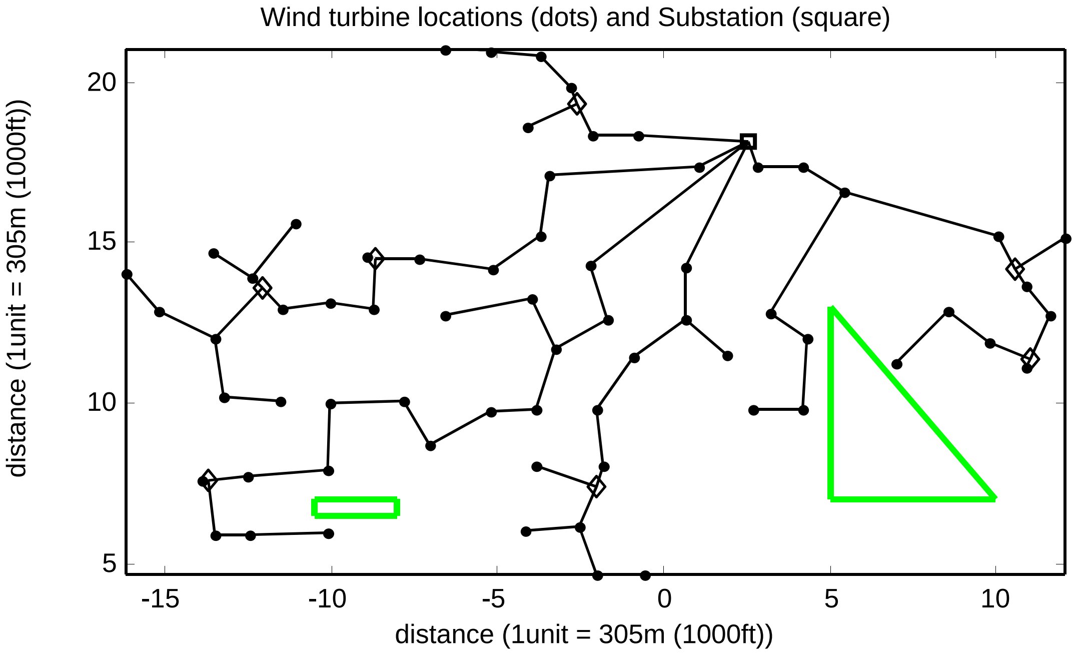


Figure 59: Layout considering trenching restrictions (restricted areas shown in green polygons)

Comparing Figure 59 and Figure 57 shows how applying the trenching restriction constraint changes the layout.

3.8 Layout obtained with same clusters as actual layout

It should be noted that the maximum number of turbines on a feeder for the actual layout (Figure 49) is 24 whereas in the layout in Figure 57, the maximum number is 17. For comparison purposes, the MST and splice introduction algorithms are applied on the same turbine clusters as the actual one, to obtain the layout shown in Figure 60:

Case VI – Reduced total length configuration by introduction of intermediate splice nodes and groups of same turbines on each feeder cable as actual layout but without considering landowner constraints.

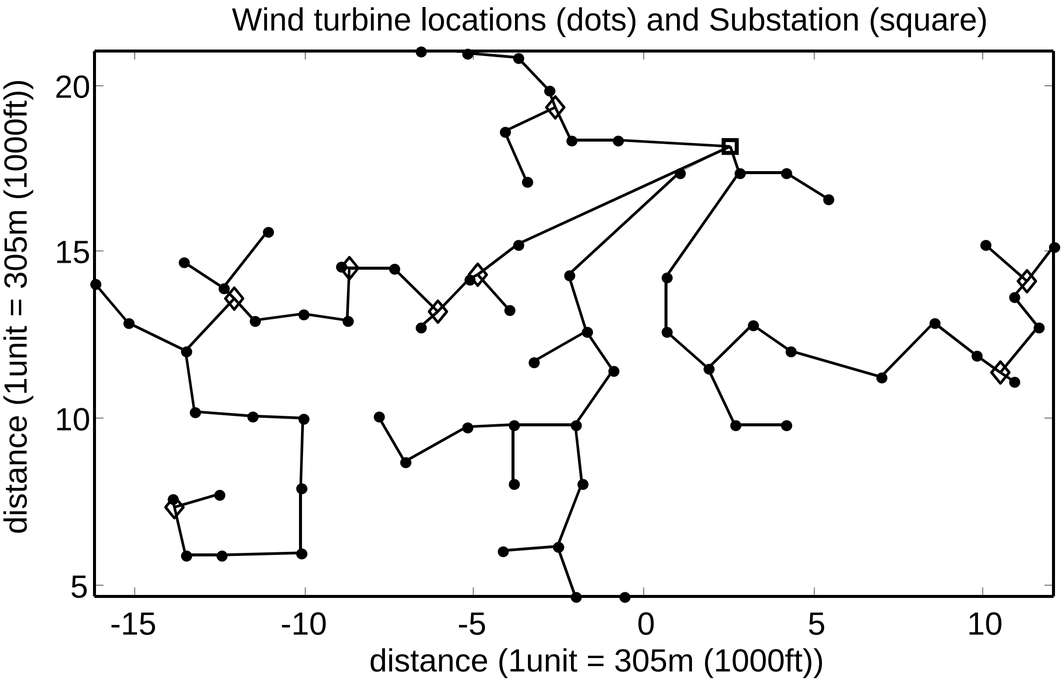


Figure 60: Layout with turbine clusters same as actual layout; intermediate splices (diamonds) introduce

The total cable length equals the total trenching length because the maximum number of turbines on a feeder is limited.

3.9 Conversion of undirected to directed graph and assigning cable sizes

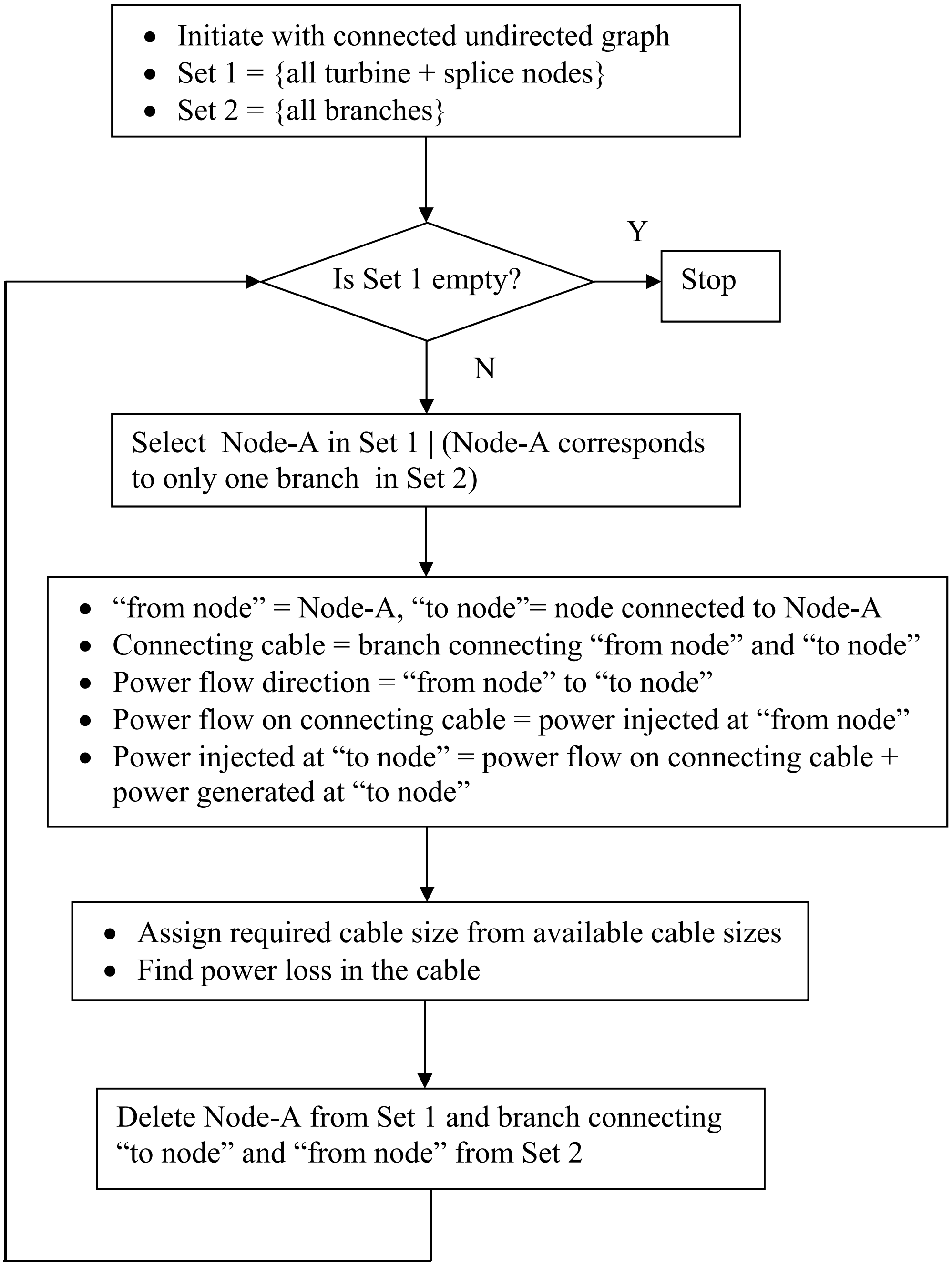
The algorithms described in the previous sections result in undirected trees. However, for the wind farm collector system design, an important concern is determining the cable sizes which depend on the ampacity of required cables which in turn depend on the actual power carried by the cable. This needs a direction of power flow on the cables and hence the undirected graphs have to be converted to directed graphs. Another algorithm has been proposed for the same. The algorithm described in this subsection converts the undirected trees to directed ones. The input to the algorithm is the connected undirected graph obtained by applying one of the previously described algorithms on substation and wind turbine locations. The output is the direction and magnitude of active power flow on the branches or cables, an assignment of cable sizes for different cables from a list of available cable sizes as shown in Table 14 (in the Appendix), and power losses on all connecting cables. The design is for the situation when all the turbines generate the rated power. Thereby the power injected by each turbine at the node corresponding to its location equals the rated power. The algorithm is as follows [112]:

- *Step 1* – Two sets are initialized. Set 1 is the set of all turbine and splice nodes. Set 2 is the set of all branches.
- *Step 2* – A node called Node-A is selected from Set 1such that Node-A corresponds to only one branch in Set 2. This step physically selects one of the terminal nodes in the graph. The Node-A is denoted a "from node." The node which the Node-A is connected to is denoted a "to node." The power flow direction is from the "from node" to the "to node." The flow on the cable is the power injected at the "from node" and the power injected at the "to node" location is incremented by the power flow on the cable. The connecting cable is assigned a cable size from continuous ampacity rating of available conductors in Table 14 (in the Appendix) and assuming a 34.5 kV medium voltage system. The resistance of the cable is computed from resistance data in Table 14 (in the Appendix). The resistance is used to find the I^2R power losses on the cable.
- *Step 3* – The Node-A ("from node") is deleted from Set 1 and the branch connecting the "to

node” and “from node” is deleted from Set 2 thus resulting in a smaller dimension graph.

- *Step 4* – The Steps 2 and 3 are repeated till the Set 1 is empty.

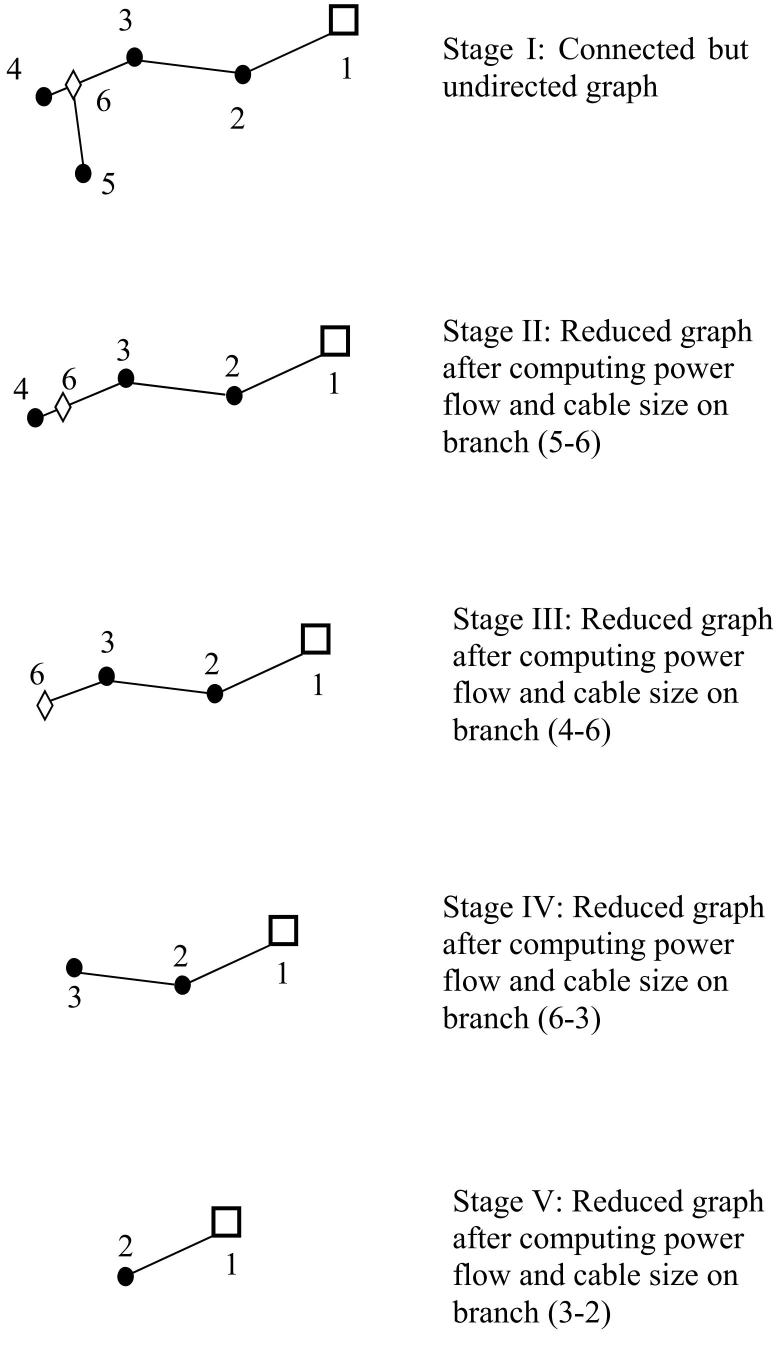
This algorithm has been illustrated with a flow chart in Figure 61.



Initiate with connected undirected graph
Set 1 = {all turbine + splice nodes}
Set 2 = {all branches}
Is Set 1 empty?
Y
Stop
N
Select Node-A in Set 1 | (Node-A corresponds to only one branch in Set 2)
“from node” = Node-A, “to node”= node connected to Node-A
Connecting cable = branch connecting “from node” and “to node”
Power flow direction = “from node” to “to node”
Power flow on connecting cable = power injected at “from node”
Power injected at “to node” = power flow on connecting cable + power generated at “to node”
Assign required cable size from available cable sizes
Find power loss in the cable
Delete Node-A from Set 1 and branch connecting “to node” and “from node” from Set 2

Figure 61: Flow chart for algorithm to convert undirected to directed graph and assign cable sizes

The algorithm is further illustrated with the Figure 62 which provides an example of a small wind farm with the substation (square) represented by node 1, turbines (dots) represented by nodes 2, 3, 4, and 5, and a splice node (diamond) represented by node 6. The transition of sets is shown in Table 4.



Stage I: Connected but undirected graph

Stage II: Reduced graph after computing power flow and cable size on branch (5-6)

Stage III: Reduced graph after computing power flow and cable size on branch (4-6)

Stage IV: Reduced graph after computing power flow and cable size on branch (6-3)

Stage V: Reduced graph after computing power flow and cable size on branch (3-2)

Figure 62: Illustration of the algorithm for converting the undirected to directed graph

Table 4: Illustration of algorithm for converting undirected to directed graph (transition of the sets)

Stage No.	From Node	To Node	Power Flow on Cable	Updates
I	5	6	P5	P6=0+P5
II	4	6	P4	P6=P6+P4
III	6	3	P6	P3=P3+P6
IV	3	2	P3	P2=P2+P3
IV	2	1	P2	P1=P1+P2

Cables are assigned depending on required ampacity on each section from available cable sizes provided in the Appendix.

3.10 Economic analysis

It is important to do an economic analysis of the design. The process for that is described here. In this work, underground cables interconnect 1 MW wind turbines and the substation at 34.5 kV. Assuming the units generate power at unity power factor, the rms current/phase is projected to be a maximum of

$$\frac{1\times10^{6}}{3\times(34.5\times\frac{10^{3}}{\sqrt{3}})\times1.0} = 16.73\ \text{A} \tag{3.2}$$

It has been assumed that the average wind speed at the wind farm location is 8.5 m/s. Assuming Rayleigh distribution, capacity factor of the turbines can be computed as [18]:

$$\mathrm{CF} = 0.087\overline{\mathrm{V}} - \frac{\mathrm{P_R}}{\mathrm{D}^2} \tag{3.3}$$

where, $\overline{\mathrm{V}}$ is the average wind speed in m/s, $\mathrm{P_R}$ is the rated power in KW of a wind turbine, D is rotor diameter in m. With rated power of each turbine unit taken as 1 MW, rotor diameter 52 m, and average wind speeds of 8.5 m/s, the capacity factor is approximately 37%. It has been assumed that the wind turbines generate either the peak power or zero. Hence, the collector system loss factor equals the capacity factor.

A levelized cost estimate for energy delivered by the wind farm has been found by computing the ratio of annual costs and the annual energy produced.

$$\text{Cost of electric energy } \left(\frac{\text{cents}}{\text{kWh}}\right) = \frac{100 * \text{ Annual cost (\$/yr)}}{\text{Annual energy (kWh/yr)}} \tag{3.4}$$

Hence, the annual energy generated is found by the product of the nameplate capacity of the wind farm reduced by the collector system losses, number of hours in a year, and the capacity factor of the wind farm.

$$\mathrm{AE} = (\mathrm{P_R} \times \mathrm{N} - \mathrm{P_{loss}}) \times 8760 \times \mathrm{CF} \tag{3.5}$$

where AE is the annual energy generated, N is the number of turbines, Ploss is the collector system losses (in kW), and the capacity factor (CF) of the wind farm.

The capital cost CC of the wind farm project is computed as:

$$CC = 1.05 \times (N \times C_T + L \times C_t + C_c) \tag{3.6}$$

where C_T is the cost of a turbine ($1× 10^6/turbine), L is the total trenching length, C_t is trenching costs of $15/ft, and C_c is the total cable costs. A multiplication factor of 1.05 is used to take into account other components of the capital cost such as site preparation, grid connections, project development, and feasibility study.

It is assumed that the wind project is financed by a loan which is 75% of the capital cost with a 7% interest rate and loan term of 20 years and the remaining by equity. The capital cost is spread out over the projected lifetime of the wind farm.

The annual cost AC in $/yr incurred is given by [18]:

$$AC = A + E + M \tag{3.7}$$

where A is the annual loan payments, E is annual return on equity, and M is operation and maintenance costs.

The annual payment on the loan is given by [18]:

$$A(\$/yr) = P\left[\frac{i(1+i)^n}{(1+i)^n - 1}\right] = P.CRF(i,n) \tag{3.8}$$

where CRF is the Capital recovery factor and P is the principle borrowed.

The remaining capital cost is met with an equity on which a 15% return is required per year. Annual return on equity (E) is given by:

$$\mathrm{E}(\$/\mathrm{yr}) = \ 0.15 * (25\% * \mathrm{CC}) \qquad (3.9)$$

The operation and maintenance costs are assumed to be 3% of the capital cost and include parts and labor, insurance, contingencies, land lease, property taxes, maintenance of transmission lines, and general and miscellaneous costs.

$$\mathrm{M}(\$/\mathrm{yr}) = 0.03 * \mathrm{CC} \qquad (3.10)$$

Finally, a levelized cost estimate for energy delivered by the wind farm has been found as follows [18]:

$$\mathrm{CEE} = \mathrm{AC}\ /\mathrm{AE} \qquad (3.11)$$

3.11 Computing the reliability of the design

In Section 3.3, a metric was introduced to measure reliability of the collector system design. This metric, namely the mean number of turbines lost due to a single cable fault is used again to develop an algorithm to compute reliability of a system. The algorithm has the following steps.

- *Step 1* – Two sets are initialized. Set 1 is the set of all turbine and splice nodes. Set 2 is the set of all branches.
- *Step 2* – A node is selected from Set 1 such that it corresponds to only one branch in Set 2. This node is named Node-A. The purpose of this step is to select one of the terminal nodes in the graph. The Node-A is denoted a “from node.” The node which it (Node-A) is connected to is denoted a “to node.” The number of turbines lost due to fault on cable (“from node”, “to node”) equals number of turbines beyond “from node.” The number of turbines beyond “to node” is incremented by unity.

- *Step 3* – Node-A (“from node”) is deleted from Set 1 and the branch connecting the “to node” and “from node” is deleted from Set 2 thus resulting in a smaller dimension graph.
- *Step 4* – The steps 2 and 3 are repeated till the Set 1 is empty.
- *Step 5* – Finally the reliability of the design is computed using equation (3.1).

This algorithm has been further illustrated with a flow chart in Figure 63.

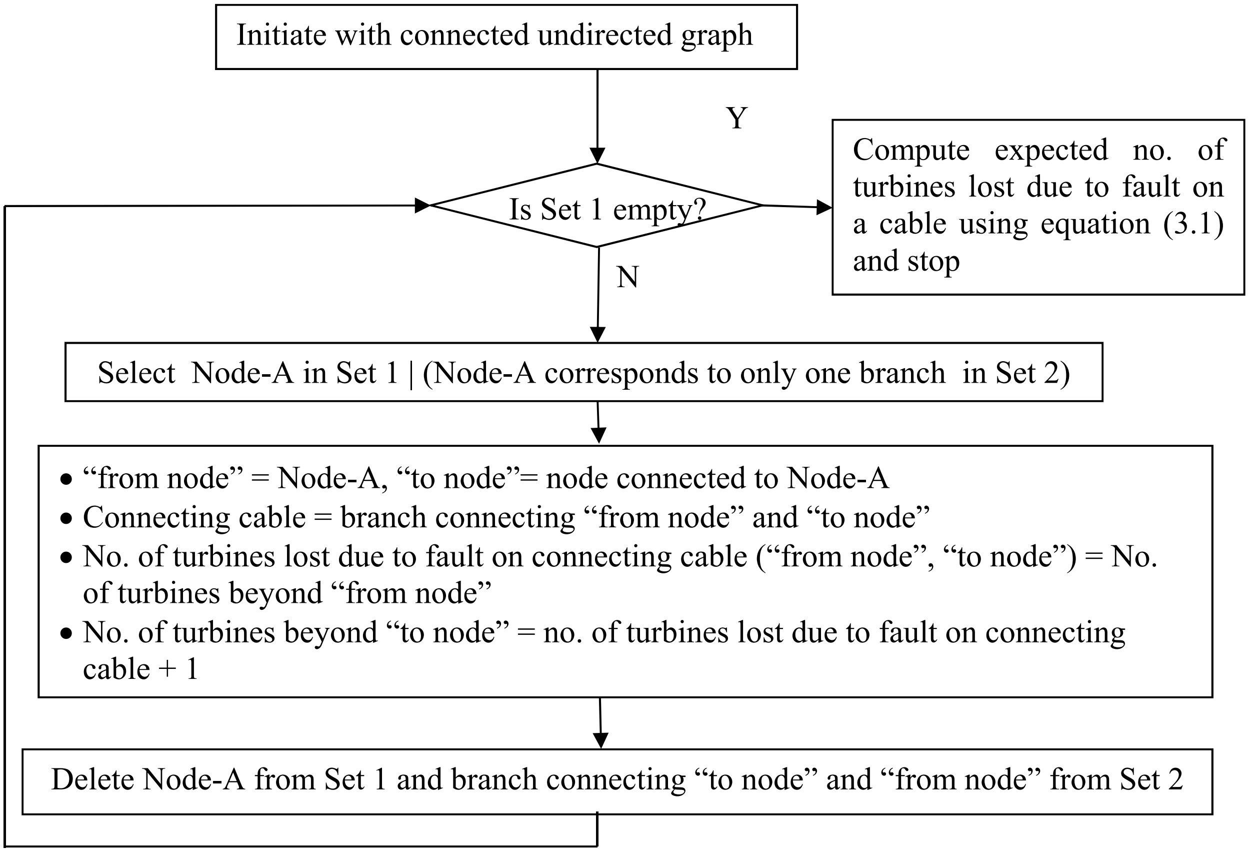

Initiate with connected undirected graph
Y
Is Set 1 empty?
Compute expected no. of turbines lost due to fault on a cable using equation (3.1) and stop
N
Select Node-A in Set 1 | (Node-A corresponds to only one branch in Set 2)
• “from node” = Node-A, “to node”= node connected to Node-A
• Connecting cable = branch connecting “from node” and “to node”
• No. of turbines lost due to fault on connecting cable (“from node”, “to node”) = No. of turbines beyond “from node”
• No. of turbines beyond “to node” = no. of turbines lost due to fault on connecting cable + 1
Delete Node-A from Set 1 and branch connecting “to node” and “from node” from Set 2

Figure 63: Computation of design reliability

3.12 Results and discussions

The results for all the cases are summarized in Table 5.

Table 5: Results summary

Layouts	Max. turbines on feeder	Total trenching length (m)	Total cable length (m)	Trenching costs (× 10^6 \$)	Cable costs (× 10^6 \$)
Case I (min. spanning tree)	63	33896	40197	1.69	2.13
Case II (splices)	63	33732	39672	1.68	2.06
Case III (normal clustering)	20	41844	41844	2.09	1.36
Case III (radial clustering)	17	39037	39037	1.95	1.29
Case IV (splices and radial clustering)	17	38870	38870	1.94	1.28
Case V (splices, radial clustering, and trenching restrictions)	17	39666	39666	1.98	1.29
Case VI (same clusters as actual layout)	24	37131	37131	1.86	1.37
Actual layout	24	43186	43186	2.16	1.44

It should be noted that adding a constraint, namely limiting number of turbines on a feeder as in Cases III (Figure 56) and IV (Figure 57) results in an increase of the total trenching length compared to Case I (Figure 50) and Case II (Figure 53). For the same reason, the total trenching length in Case VI (Figure 60) is lower than the previous layout Case III (Figure 56). Also, it is noticed that the higher the number of turbines allowed to be connected to a feeder, the greater the cable costs since more higher-sized and hence higher-cost cables are required as branch currents add up close to the substation. Furthermore, radial clustering generates better results compared to clustering by turbine locations.

The costs of energy for the different layouts are also calculated. Since the costs of energy are levelized costs, there is not a significant change between different layouts, varying between 4.6 to 4.7 cents/kWh. For example, the cost of energy in the layout with splices, radially clustered turbines and with trenching restrictions is 4.74 cents/kWh.

The reliability (expected no. of turbines lost due to fault on a cable) of the layout with splices, radially clustered turbines and with trenching restrictions (Case V) is computed to be 433*P (= 0.022 for p = 0.1).

It should be noted that the objective of the current work is to minimize the trenching lengths which is achieved. Also, with the layouts in Cases IV and VI, significant savings of respectively $380,000 and $370,000 are made compared to the actual layout in cabling and trenching. These savings are important considerations during project planning. It should also be noted that an important contribution of this work is that these layouts are generated automatically, thus saving manual labor.

3.13 Conclusions

This chapter presents applications of graph theory and data mining for generating a basic design for a wind farm collector system cable layout configuration. Several novel algorithms are developed. The first algorithm improves on a minimum spanning tree design by creating external splice locations separate from the wind turbine locations. The second algorithm applies clustering to address the constraint of a prespecified maximum number of turbines connected to a feeder cable. The third proposed algorithm addresses the constraint of trenching restrictions by modifying the spanning tree algorithm from graph theory. The fourth algorithm computes direction and magnitude of power flow on each cable, assigns cable sizes from a table of available cable sizes converting an undirected to a directed graph. Also, methods are developed for computing reliability and economic analysis of the generated designs.

Results show that the algorithms proposed can be used to generate a design that has minimum total trenching length, also taking into account constraints on the maximum number of turbines on a feeder, and trenching restrictions. The total length of the minimum spanning tree is lowered by a factor of 1.01 by the introduction of intermediate splice nodes, but this total length increases when the constraint of 20 maximum turbines on a feeder are applied. The designs generated in Cases IV and VI achieve respectively 10% and 14% reduction in total trenching length compared to the actual cable layout configuration shown in Figure 49. The major contribution of this work is in the automatic generation of a starting layout design which is optimal with respect to total length and can be modified with heuristics to incorporate specific design requirements. Also, demonstrated in this work is the applicability of a graph-theoretic framework to address the wind farm collector system design problem and use of clustering algorithms to address design constraints.

4. ESTIMATION AND CONSIDERATION OF STATISTICAL DISTRIBUTION OF DATA IN WIND POWER INTEGRATION

Wind power forecasting and integrating wind power with storage to firm outputs from wind farms are being considered critical for the large-scale integration of wind power in the Smart Grid. Incorporated with forecasting techniques and on-site energy storage, wind farms can participate in hour or day-ahead electricity markets similar to conventional power plants. The following three subsections address respectively the problems of (i) coordinating storage and stochastic wind power, (ii) sizing on-site energy storage units for energy balancing while taking into account the uncertainties of both wind and load forecasts, and (iii) a least squares based method for forecasting wind power between groups of wind farms.

Statistical properties of data, namely wind power and load forecast data are very important inputs in the formulation of all three challenges and the case studies clearly show the effect of data distribution on the result, whether it is coordinating storage and wind power or sizing on-site energy storage or wind power forecasting. In addition, the third challenge directly applies data mining technique of estimation.

4.1 Coordination of storage and wind power

Motivation and prior art

Incorporation of energy storage units with wind farms is being considered critical for wind farms to address variability in wind power generation and meeting committed generation schedules. A report by the American Institute of Chemical Engineers (AICE) on mass power storage for the grid in 2008 identified the availability of massive electricity storage as a key to making the use of renewable energy possible on a broad scale [130]. In absence of energy storage units, the dearth of generated power caused by lower than scheduled wind farm output has to be produced by more expensive thermal units, thereby increasing system operating costs. Recognizing the importance of grid level storage, some wind to storage projects are being planned and implemented across the United States [131, 132]. However, a storage unit with finite limits on maximum and minimum energy, and charge and discharge rates, behaves as a limited energy plant. This in turn limits the capability of storage units in providing required support to wind farms. Therefore, optimal charge discharge coordination of a practical storage unit with a wind farm is necessary for optimal operation of the combined system.

In the United States, wind generation is mostly a price taker participating in real-time electricity markets. The penalty charges imposed in day-ahead and hour-ahead electricity markets due to schedule deviations are therefore generally avoided by wind resources. However, according to a summary compiled by Utility Wind Integration Group [133], in markets such as PJM, for example, operating reserve deviation charges are applied on the differential between day-ahead schedules and real-time power generation levels exceeding a dead band if wind resources are self-scheduled. A wind generator participating as a “capacity resource” in MISO must also take part in day-ahead markets with imbalance charges imposed on net schedule deviations over a specified time. However, if the wind generation resource is designated as “intermittent,” then it is a price taker in the real-time market with no uninstructed deviation penalties. In ERCOT, wind generation may be exempt from charges due to deviations from submitted schedules when it is scheduled as a qualified scheduling entities’ portfolio. However, this has stimulated widespread dissatisfaction among owners of conventional generation in

ERCOT since conventional generation such as by coal and gas are imposed penalties on failing to meet committed schedules [134].

With increasing wind penetration, wind generation will have to be scheduled in forward markets for maintaining the generation-demand balance. It can be expected that wind generation will soon be operating under same policies as other conventional generation with penalty charges for schedule deviations. An energy storage unit can serve as a hedge to forecasting uncertainties and maintain wind generation schedules in day-ahead and hour-ahead markets by absorbing excess generation than forecasted, and providing energy support during periods of lower than forecasted generation.

Several technical works have described the effectiveness of integrating storage with wind. Optimal bids for day-ahead spot market have been found for a wind farm operating with a pumped-hydro storage plant by solving an optimization model [135]. The problem addressed is the bidding decisions taken by wind farm operators in a spot-market framework under uncertainty of both wind power outputs and electricity prices. Simulation results indicate that energy storage makes it possible for owners of wind power plants to take advantage of variations in the spot price, by thus increasing the value of wind power in electricity markets [136]. A dynamic programming algorithm has been used to determine the hourly trading of electricity in the spot market which maximizes the expected profit over the scheduling period. In [137], a stochastic model for the daily operation scheduling of a generation system including pumped storage hydro plants and wind power plants, where the uncertainty is represented by the hourly wind power production has been presented. However, these works focus on the maximization of profits for wind farms, and address the use of energy storage for electricity arbitrage.

In this dissertation, issues of maintaining a committed generation schedule by a wind farm over the period of one hour with the availability of generalized energy storage is addressed. It has been shown that wind generation schedule deviations can be minimized and a maximum steady schedule can be obtained that a combined wind-storage plant can serve in presence of forecasts of wind power generation.

Methodology incorporating statistical distribution of wind forecasts

The operation of the combined wind farm-storage plant can be modeled as a wind farm and a storage unit connected to a load [138, 139]. Figure 64 shows the power flows between a storage unit, wind farm, and the connected load. The load represents the committed generation of the wind farm.

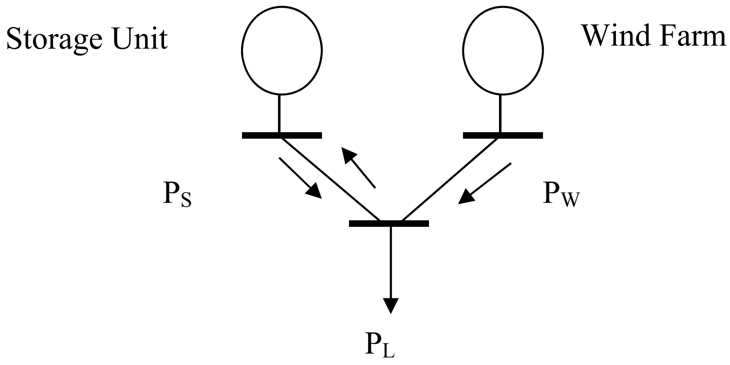


Figure 64: Electric flows between a storage unit, wind farm, and system load

Available literature documents several different technologies for storage of electric energy, some of them being batteries, compressed air storage, ultracapacitors, and SMES [140]. In this work, no specific storage technology has been addressed; rather the storage unit has been modeled as a generalized limited energy plant with limits on the rates of charging and discharging, and the maximum and minimum energy that can be stored

The operating horizon is divided into a number of intervals which can be of two types, intervals when the storage unit is discharged {k}, and intervals when the storage unit is charged {i}. Neglecting losses, the constraints under which the combined storage and wind farm operates are:

$$P_{L_k} - P_{W_k} - P_{S_k} = 0 \tag{4.1}$$

$$P_{L_i} - (P_{W_i} - P_{Wspill_i}) + (-P_{S_i}) = 0 \tag{4.2}$$

Here P_{L_j} is the load, P_{W_j} is the value of the random variable representing the power generated by the wind with known probability distribution function at interval j, and P_{S_j} is the electric power generated by the storage unit at an interval j, where j is a charging or discharging interval. It should be noted that by definition, P_{S_k}, the power generated by the storage unit at discharging interval k is positive and P_{S_i}, the power absorbed by the storage unit at charging interval i is negative. P_{Wspill_i}, is the wind power generated in excess of the load that cannot be absorbed by the storage at charging interval i. This generated power needs to be curtailed or spilled by pitching turbine blades away from the wind. Hence, equations (4.1) and (4.2) present the power balance constraints at all charging and discharging intervals. The violation of the equality constraint presented by equation (4.1) indicates deviation from committed generation schedule due to under-generation. This work aims to minimize the expected sum of these deviations over the scheduling horizon. It should also be noted that this work does not minimize the spilled or curtailed wind power denoted by P_{Wspill}, which represents the over-generation deviations over the scheduling horizon.

The intervals when no power needs to be absorbed or delivered by the storage unit, can be considered as a discharging interval with,

$$\mathrm{P}_{\mathrm{S}_\mathrm{k}} = 0 \tag{4.3}$$

Stored energy continuity equations can be formulated as follows:

$$\mathrm{E}_{\mathrm{S}_\mathrm{k}} = \mathrm{E}_{\mathrm{S}_{\mathrm{k}-1}} - \mathrm{P}_{\mathrm{S}_\mathrm{k}} * \mathrm{n}_\mathrm{k} \quad \text{at discharging interval k} \tag{4.4}$$

$$\mathrm{E}_{\mathrm{S}_\mathrm{i}} = \mathrm{E}_{\mathrm{S}_{\mathrm{i}-1}} + (-\mathrm{P}_{\mathrm{S}_\mathrm{i}}) * \mathrm{n}_\mathrm{i} \qquad \text{at charging interval i} \tag{4.5}$$

where $\mathrm{E}_{\mathrm{S}_\mathrm{j}}$ is the stored energy at the end of interval j, and n_j is the duration of interval j.

The energy stored in the storage unit is constrained by maximum and minimum storage limits $\mathrm{E}_{\mathrm{S}_{\mathrm{max}}}$ and $\mathrm{E}_{\mathrm{S}_{\mathrm{min}}}$:

$$\mathrm{E}_{\mathrm{S}_{\mathrm{min}}} \leq \mathrm{E}_{\mathrm{S}_\mathrm{j}} \leq \mathrm{E}_{\mathrm{S}_{\mathrm{max}}} \ \text{ at any interval j} \tag{4.6}$$

The maximum charge and discharge rates are also constrained by:

$$\left|\mathrm{P}_{\mathrm{S}_\mathrm{j}}\right| \leq \mathrm{P}_{\mathrm{S}_{\mathrm{max}}} \ \text{ at any interval j} \tag{4.7}$$

Discretization of wind forecasts

A typical waveform of the power output of a wind farm looks as shown in Figure 65. The proposed method requires dividing the scheduling horizon of 1 hour into 10 equal time intervals, and assumes the wind power output to be constant during each interval. This is a reasonable assumption considering that wind speeds and correspondingly wind power outputs do not vary significantly between two consecutive intervals of time, when the intervals of time are reasonably small. The wind farm power output, symbolized as P_W, is further quantized by allowing only five possible power outputs, 0 p.u., 0.25 p.u., 0.5 p.u., 0.75 p.u., and 1.0 p.u. It has been assumed that probability distribution of wind power forecasts is available over the scheduling horizon at different intervals.

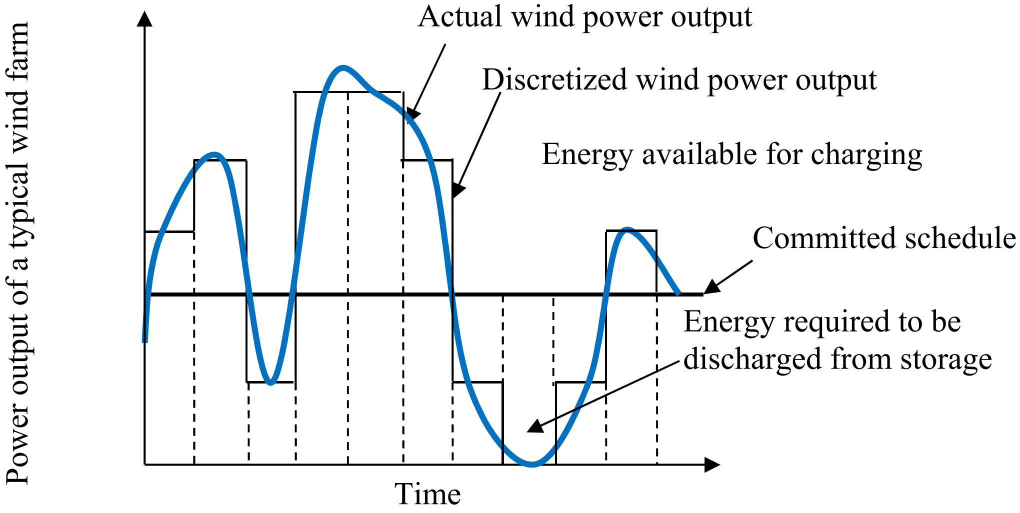

Actual wind power output
Discretized wind power output
Energy available for charging
Committed schedule
Energy required to be discharged from storage
Power output of a typical wind farm
Time

Figure 65: Wind power model

When the wind power output exceeds the committed schedule during any interval, there is an excess of power that can be used to charge the storage unit. If it is found that the amount of charging energy available exceeds the storage capacity of the storage unit, the power is spilled by pitching the turbine blades away from the wind. Similarly, when the power output is less than the committed schedule during an interval, energy is required to be discharged by the storage unit. If more energy is required than can be supplied by the storage, the combined output power fails to meet the committed schedule resulting in schedule deviations.

Assumptions

It has been assumed that the turn-around time of the storage unit is zero, i.e., the storage unit can change its mode from charging to discharging and vice versa in consecutive time intervals. The storage unit is assumed to have an efficiency of 100% and has a maximum energy storage capability of storing the maximum power that can be delivered by the wind farm over the duration of one interval of time. For simplicity, an interval of time has been defined as 1 unit. Thus, the maximum energy $\mathrm{E}_{\mathrm{S}_{\mathrm{max}}}$, which can be stored by the storage unit is 1.0 p.u. It is further assumed that the storage unit can be fully discharged, so that the minimum energy that is required to be held by the storage unit $\mathrm{E}_{\mathrm{S}_{\mathrm{min}}}$, is assumed to be 0 p.u. The initial energy $\mathrm{E}_{\mathrm{S}_{\mathrm{initial}}}$, contained in the storage unit is 0.5 p.u. The constraints on the charging and discharging rates of the storage unit are such that the storage unit can be fully charged from fully discharged condition, and vice versa, in one interval, thus $\mathrm{P}_{\mathrm{S}_{\mathrm{max}}}$ is 1.0 p.u. The power delivered or absorbed by the storage unit, P_{S}, is positive in the discharging mode, and negative in the charging mode. The committed generation at each interval, P_{L}, is also assumed to take only five possible values: 0 p.u., 0.25 p.u., 0.5 p.u., 0.75 p.u., and 1.0 p.u.

Problem formulation

The optimization problem in this work has been solved using a stochastic dynamic programming based approach. In general, a stochastic dynamic programming problem for minimization can be formulated as follows:

$$\mathrm{Min}_u V(u, X, \tilde{e}) \tag{4.8}$$

$$X_{t+1} = g(X_t, u_t, \tilde{e_t})$$

Such that

$$u_t \in \psi(X_t, \tilde{e}_t)$$

$$X \in \Omega$$

$$\tilde{e}_t \text{ is observable}$$

The decision making involves finding the optimal set of controls $\{u_1^*, u_2^*, u_3^*, \dots \dots, u_T^*\}$ that minimizes the objective under a set of constraints. Here the objective function is the sum of expected schedule deviations over the scheduling horizon. X denotes the state variables, in this case being the energy stored in the storage unit, u denotes control variables which is the amount of energy discharged from the storage unit, and $\tilde{e}$ denotes the random events that influence the state variables, the objective function or both, in this case being the wind power generated. The distribution of the stochastic variable is known, a priori. The equations of motion define the dynamic evolution of the resource. At each stage, the level of a state variable is a function of the state variable level at the previous stage, the control variable, and the realized stochastic variable. The problem is bounded by feasibility constraints. The set ψ represents the feasibility constraints for controls given the level of the state variables and the stochastic variables. The set Ω characterizes the state variable feasibility constraints. In this case it corresponds to the discretized allowable levels of energy stored in the storage unit.

The possible values of control variables, i.e. the power discharge levels at each interval of the scheduling horizon can be both positive and negative. A positive value corresponds to actual discharge from the energy storage unit and a negative value corresponds to charging of the energy storage unit. Thus, if at any interval, the energy stored in the unit is 0.75 p.u., the possible discharge levels are 0.75, 0.5, 0.25, 0, -0.25 p.u. The only negative discharge level is -0.25 p.u.

since the energy storage unit cannot absorb energy in excess of 0.25 p.u. The expected deviation is computed for every possible discharge level in an interval by summing the product of schedule deviation at that discharge level, which is a function of the stochastic wind power generation, and the probability of the wind power generation level, over all the wind power levels at that interval. At time interval t, let $\mathrm{P_{L_t}}$ denote the load, $\mathrm{P_{W_{t,m}}}$ denote one of the wind power generation levels with probability $\mathrm{Pr(P_{W_{t,m}})}$, and $\mathrm{P_{S_{t,n}}}$ denote one of the possible discharge levels. Then the schedule deviation at a particular discharge level at interval t is given by:

$$\mathrm{Dev_{t,m,n}}\left(\mathrm{P_{W_{t,m}}}, \mathrm{P_{S_{t,n}}}\right) = \max\left\{0, \mathrm{P_{L_t}} - \mathrm{P_{W_{t,m}}} - \mathrm{P_{S_{t,n}}}\right\} \tag{4.9}$$

Expected deviation at interval t for a discharge level $\mathrm{P_{S_{t,n}}}$ is given by

$$\mathrm{E_{t,n}(P_{S_{t,n}})} = \sum_{\mathrm{m}} \mathrm{Dev_{t,m,n}}\left(\mathrm{P_{W_{t,m}}}, \mathrm{P_{S_{t,n}}}\right) * \mathrm{Pr(P_{W_{t,m}})} \tag{4.10}$$

The objective is then to solve the Bellman equation:

$$\begin{aligned} \mathrm{V_t^*} &= \min\nolimits_{\mathrm{P_{S_{t,n}}}}\left\{\mathrm{V_{t-1}^*} + \mathrm{E_{t,n}}\left(\mathrm{P_{S_{t,n}}}\right)\right\} \qquad (4.11) \\ &= \min\nolimits_{\mathrm{P_{S_{t,n}}}}\left\{\mathrm{V_{t-1}^*} + \textstyle\sum_{\mathrm{m}} \max\left\{0, \mathrm{P_{L_t}} - \mathrm{P_{W_{t,m}}} - \mathrm{P_{S_{t,n}}}\right\} * \mathrm{Pr}\left(\mathrm{P_{W_{t,m}}}\right)\right\} \end{aligned}$$

and $\mathrm{V_0^*} = 0$

subject to constraints given by equations (4.3) - (4.7).

It should be mentioned here that dynamic programming is vulnerable to the “curse of dimensionality,” which means the problem can become computationally intractable with increase in size. There are several approximate approaches to deal with these problems [141]. One approach employing myopic policies has attracted a lot of attention. In this method, in each period, the objective function is optimized for that period, ignoring the potential effect on the decision in future periods. Such policies have been proven to be optimal in certain cases as well [142, 143]. In [138], the coordination of wind farm and storage was studied in a deterministic framework, with the objective to find the maximum steady generation schedule without deviations over a period of time. The problem solved using both a classical dynamic programming model and a myopic policy based method yielded the same results.

For the problem in stochastic framework, the algorithm used has been provided in Figure 66:

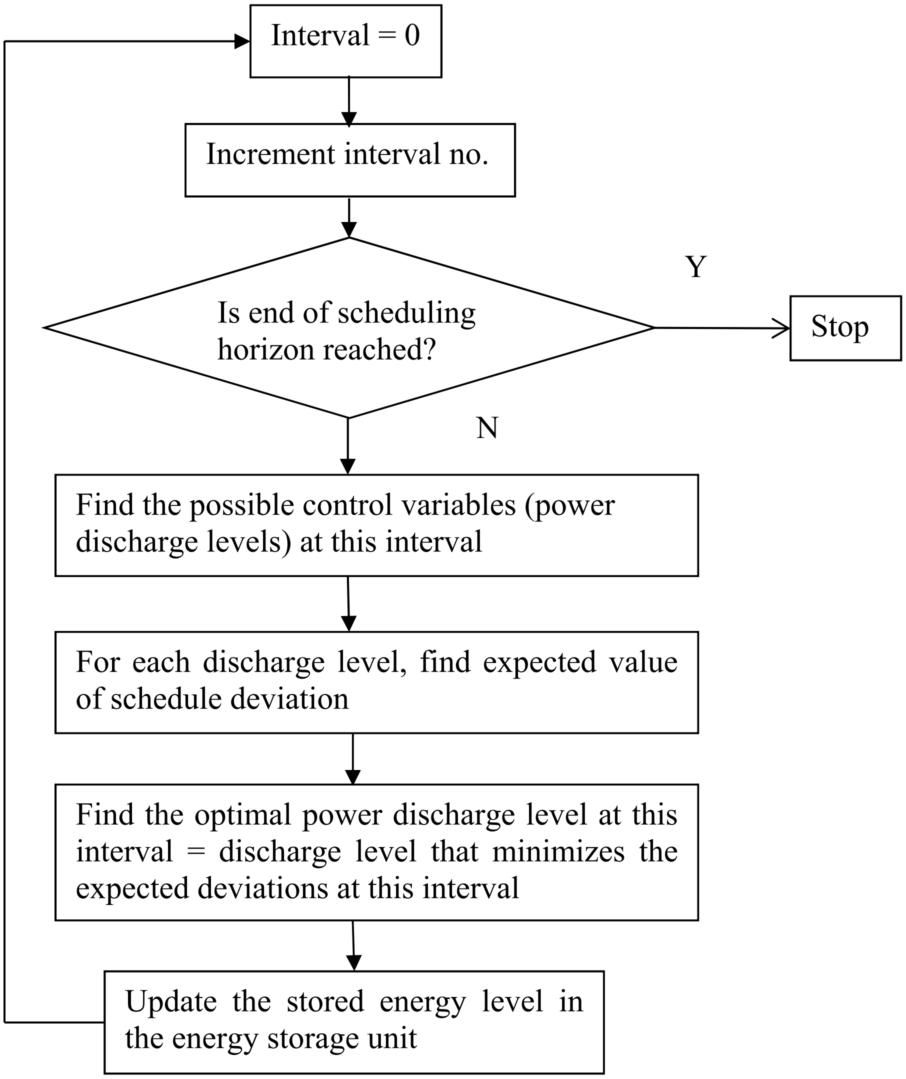

Interval = 0
Increment interval no.
Is end of scheduling horizon reached?
Y
Stop
N
Find the possible control variables (power discharge levels) at this interval
For each discharge level, find expected value of schedule deviation
Find the optimal power discharge level at this interval = discharge level that minimizes the expected deviations at this interval
Update the stored energy level in the energy storage unit

Figure 66: Optimization algorithm

Case study and results

Table 6 shows probabilities of wind power over the scheduling horizon if the forecasts are considered accurate. Hence this is a deterministic case. In Table 7, the probability distribution of predicted wind power over the same scheduling horizon is tabulated if forecasts are not accurate. This is the stochastic case. The wind power prediction in any interval in the deterministic case is approximately the mean value of wind power obtained from the probability distribution in the corresponding interval in the stochastic case.

Table 6: Probabilities of wind power generated (deterministic case)

		Wind Power generation levels (p.u.)				
		0	**0.25**	**0.5**	**0.75**	**1.0**
Intervals in Scheduling Horizon	**1**	0	1	0	0	0
	2	0	0	1	0	0
	3	0	0	0	1	0
	4	0	1	0	0	0
	5	0	0	1	0	0
	6	0	0	0	0	1
	7	0	0	0	1	0
	8	0	1	0	0	0
	9	1	0	0	0	0
	10	0	1	0	0	0

Table 7: Probability distribution of predicted wind power (stochastic case)

		Wind Power generation levels (p.u.)				
		0	**0.25**	**0.5**	**0.75**	**1.0**
Intervals in Scheduling Horizon	**1**	0.1	0.8	0.1	0	0
	2	0	0.2	0.7	0.1	0
	3	0	0	0.2	0.7	0.1
	4	0	0.6	0.4	0	0
	5	0	0.1	0.8	0.1	0
	6	0	0	0	0.3	0.7
	7	0	0	0.1	0.8	0.1
	8	0	0.9	0.1	0	0
	9	0.5	0.5	0	0	0
	10	0.1	0.8	0.1	0	0

The algorithm implemented in MATLAB programming is run for four different cases corresponding to four different steady generation commitment levels, 0.25 p.u., 0.5 p.u., 0.75 p.u., and 1.0 p.u. with initial energy stored in storage unit assumed 0.5 p.u. The resulting optimal charge discharge schedules and total deviations over scheduling horizon are as follows:

Case 1:

Committed Generation Schedule = 0.25 p.u.
Deviations in deterministic case = 0 p.u.
Expected deviations in probabilistic case = 0 p.u.

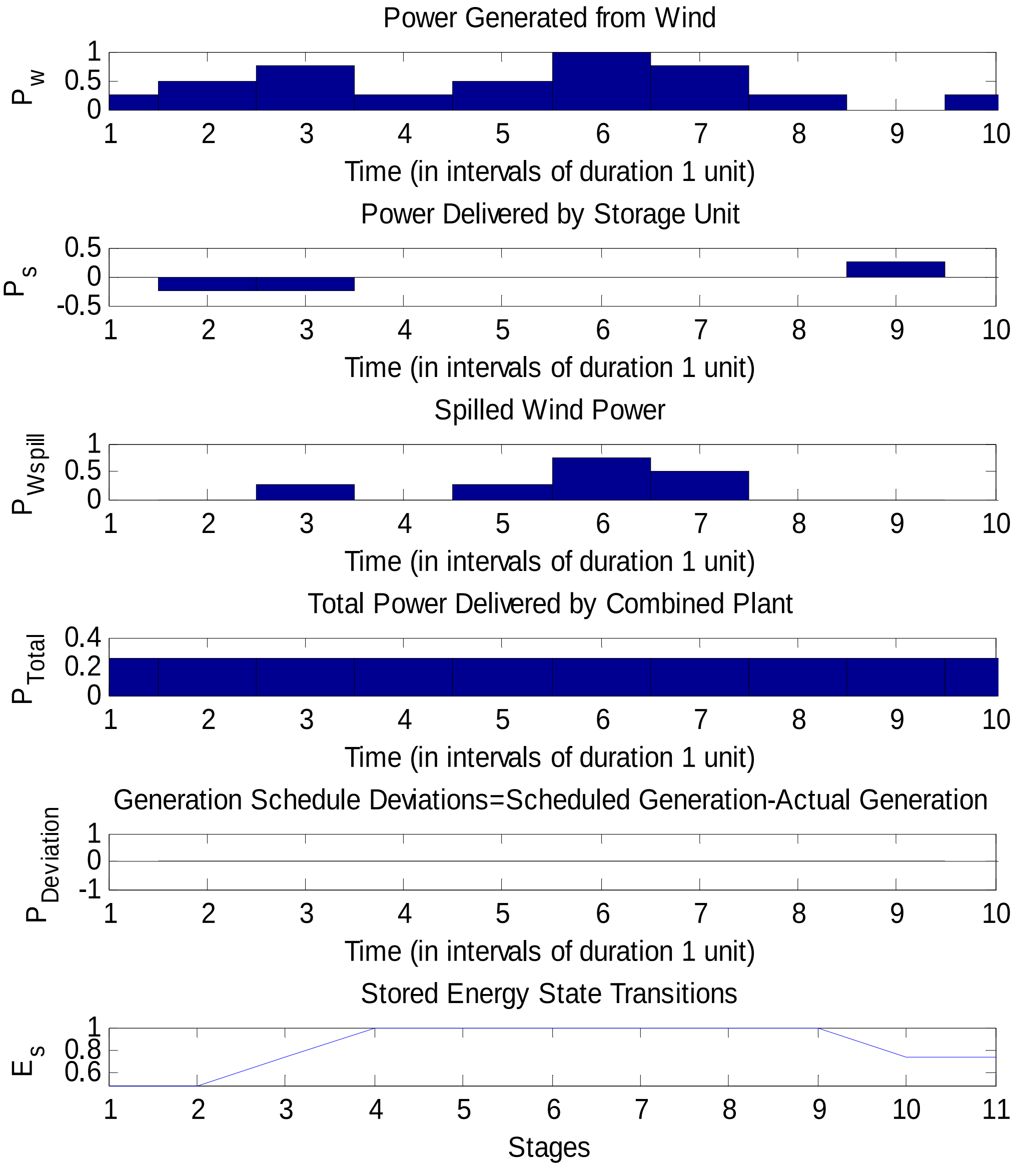


Figure 67: Deterministic case with generation commitment of 0.25 p.u.

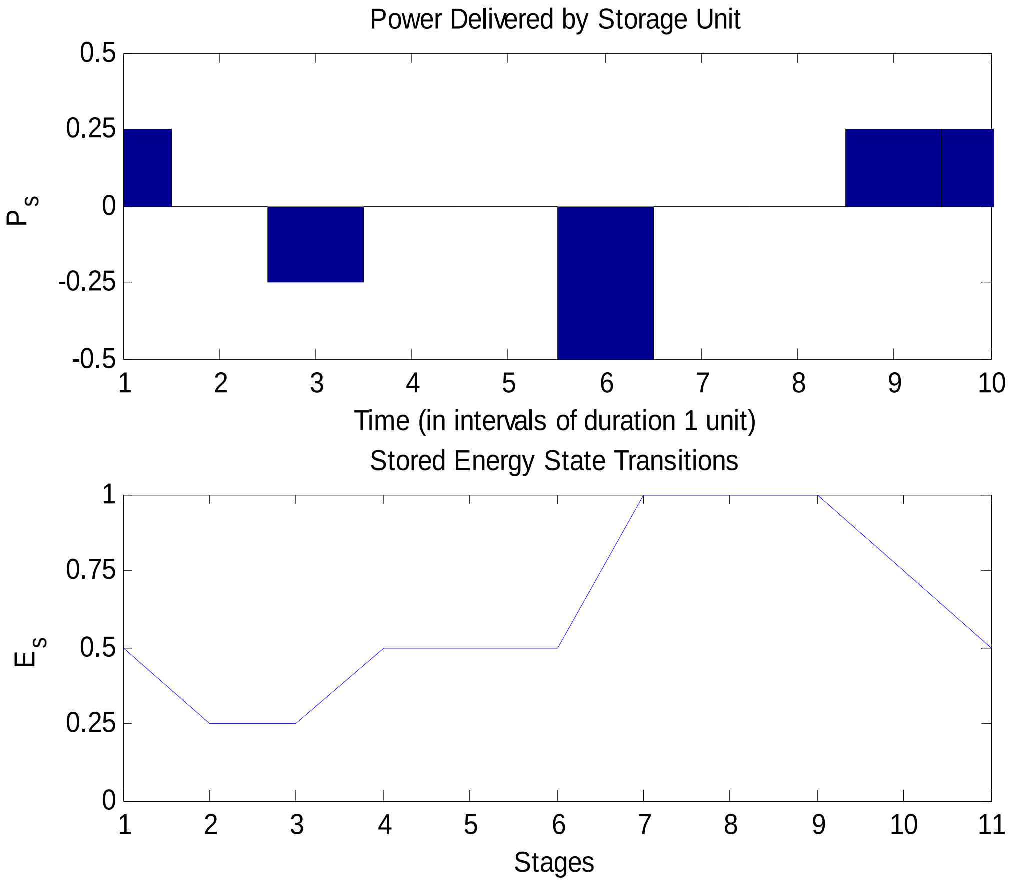


Figure 68: Probabilistic case with generation commitment of 0.25 p.u.

Case 2:

Committed generation schedule = 0.5 p.u.
Deviation in deterministic case = 0 p.u.
Expected deviation in probabilistic case = 0.85 p.u.

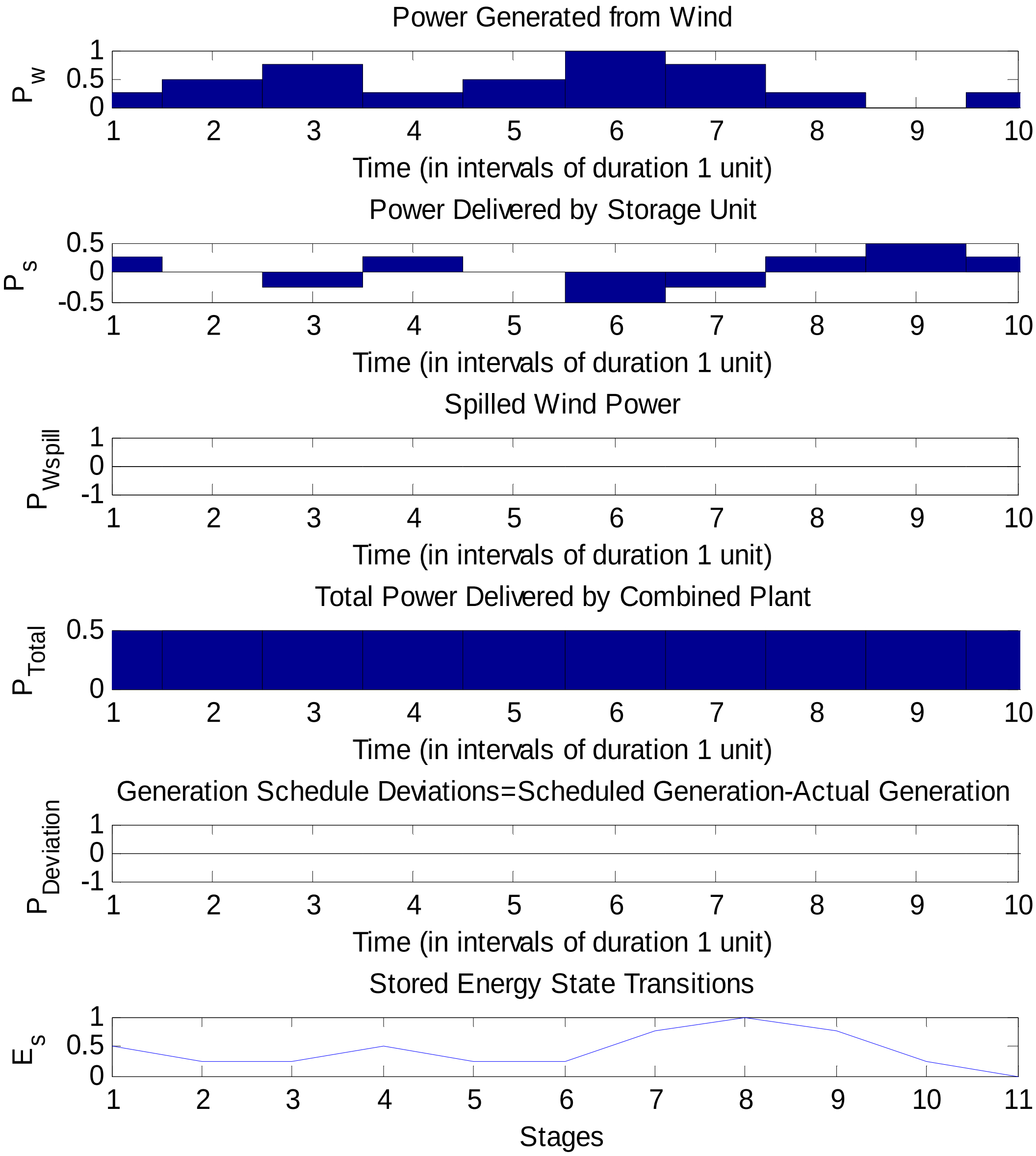


Figure 69: Deterministic case with generation commitment 0.5 p.u.

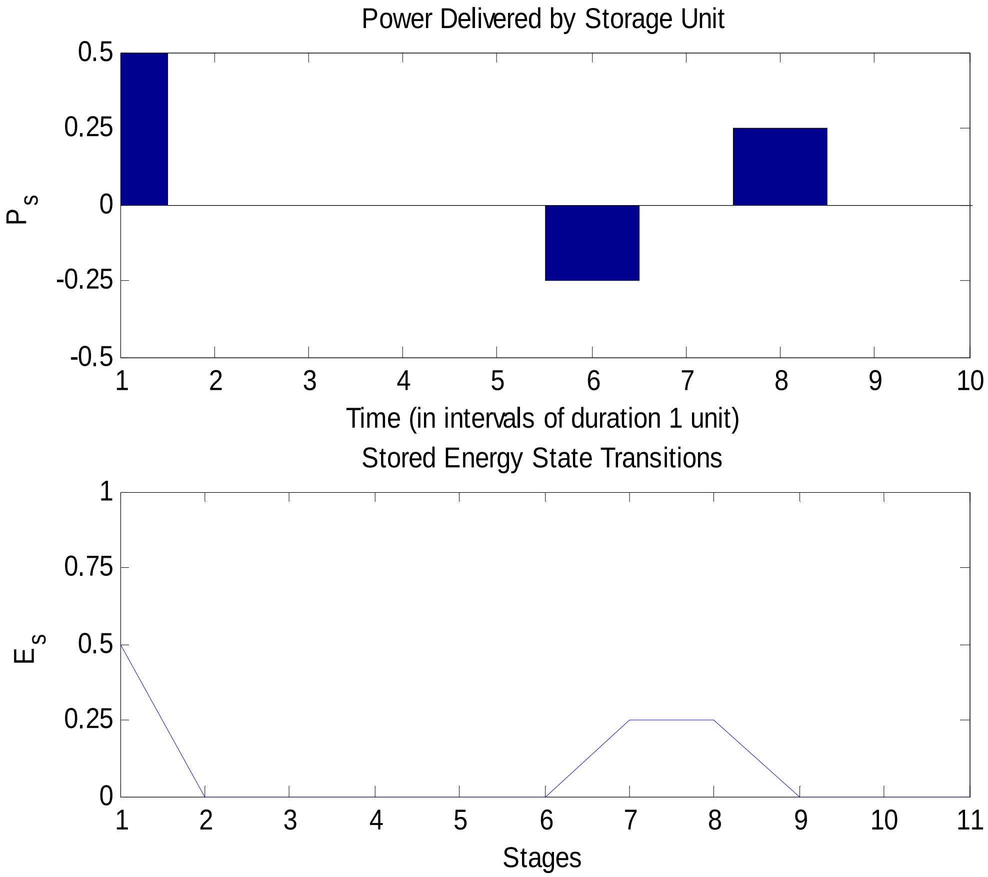


Figure 70: Probabilistic case with generation commitment 0.5 p.u.

Case 3:

Committed generation schedule = 0.75 p.u.
Deviations in deterministic case = 2.5 p.u.
Expected deviations in probabilistic case = 2.625 p.u.

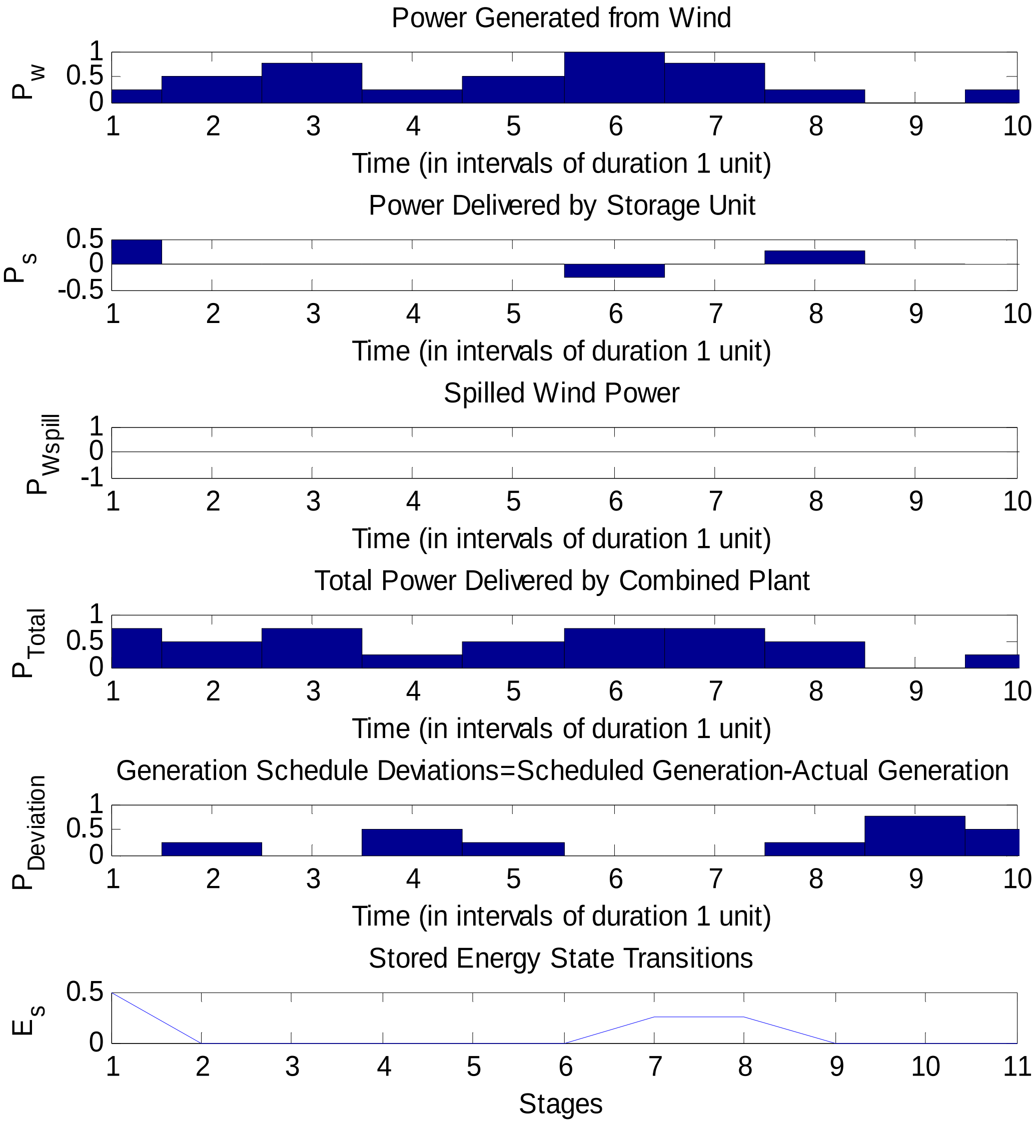


Figure 71: Deterministic case with generation commitment 0.75 p.u.

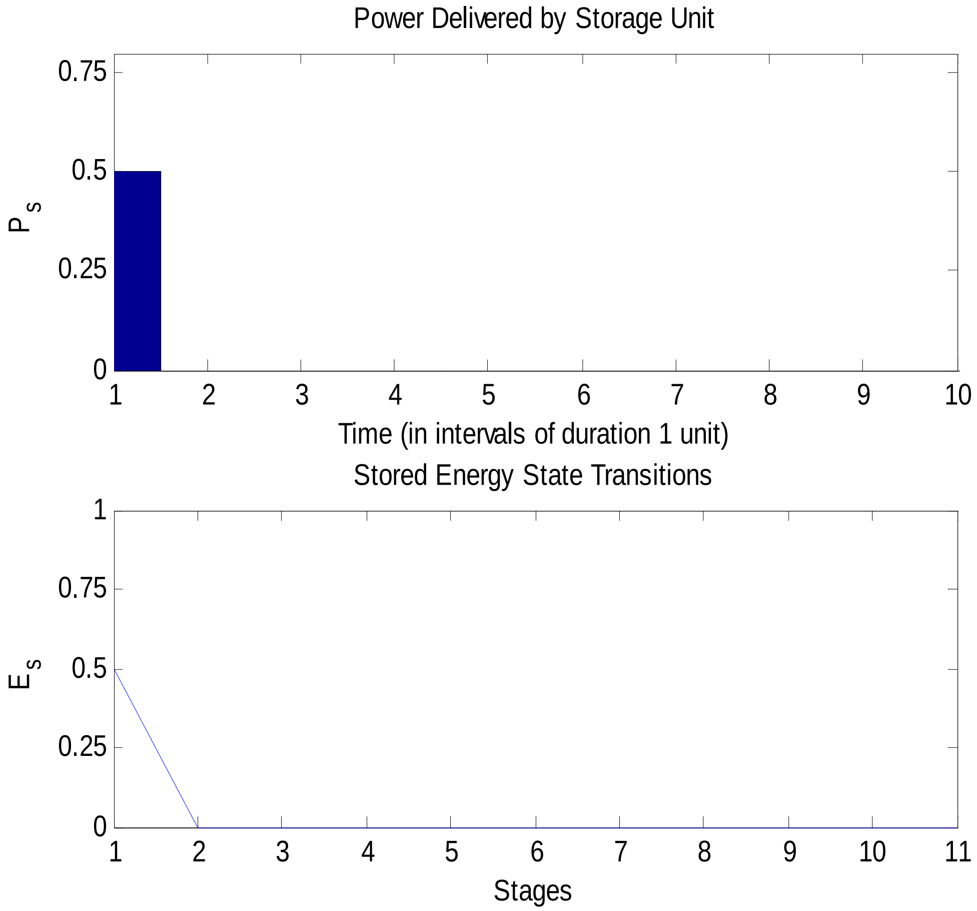


Figure 72: Probabilistic case with generation commitment 0.75 p.u.

Case 4:

Committed generation schedule = 1.0 p.u.
Deviations in deterministic case = 5.0 p.u.
Expected deviations in probabilistic case = 4.875 p.u.

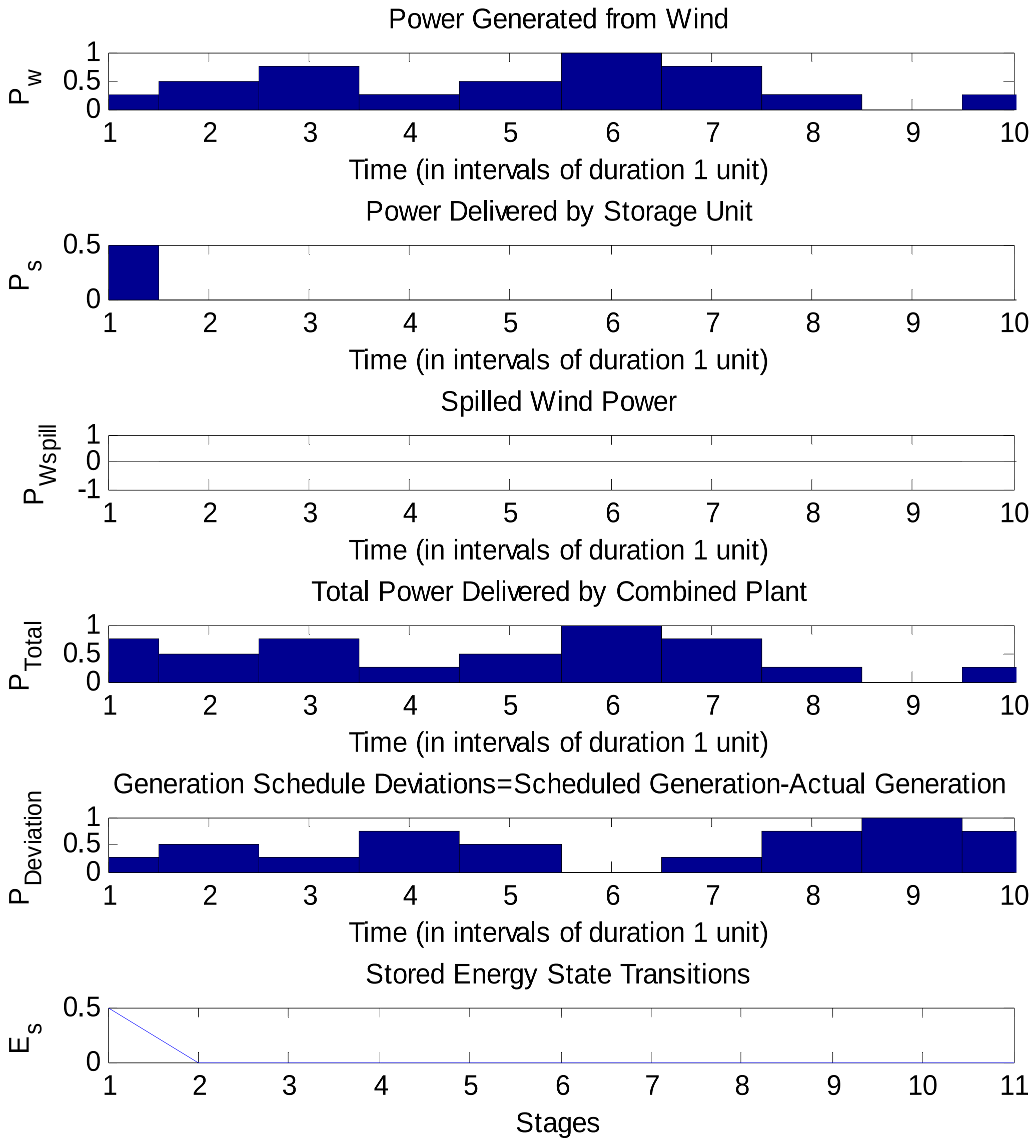


Figure 73: Deterministic case with generation commitment 1.0 p.u.

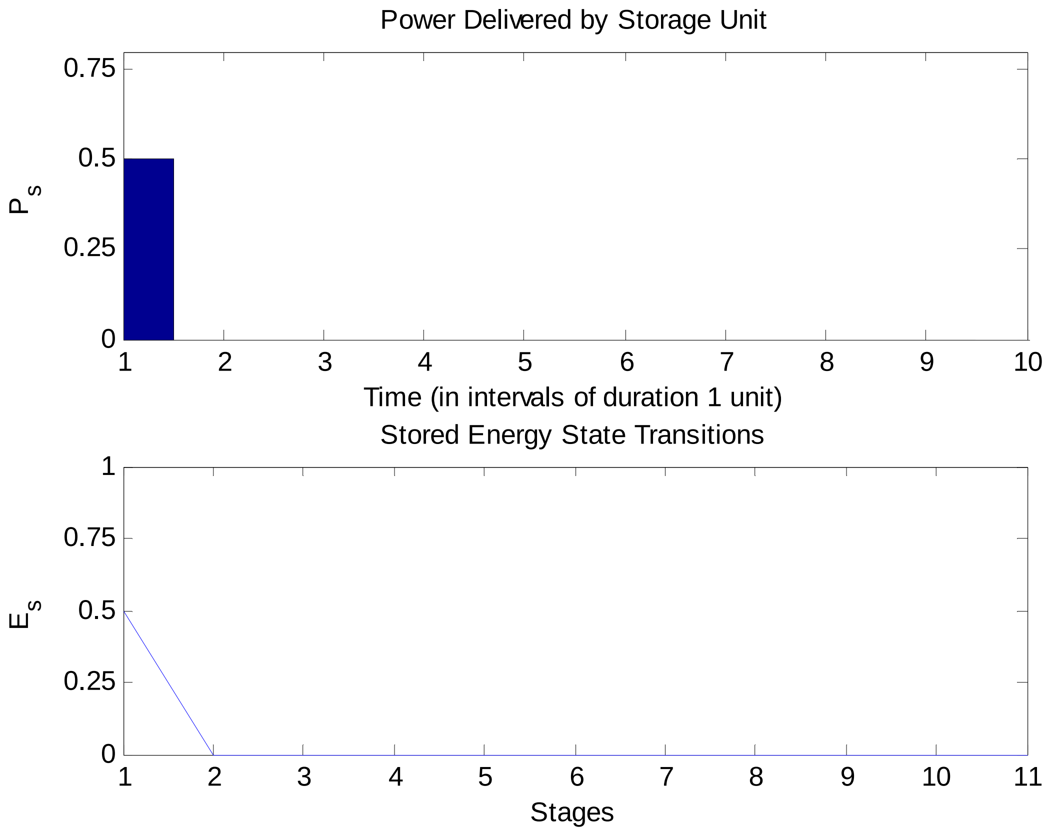


Figure 74: Probabilistic case with generation commitment 1.0 p.u.

Figure 67 through Figure 74 show respectively the results when the steady generation commitment is 0.25 p.u., 0.5 p.u., 0.75 p.u., and 1.0 p.u. It is found that in the deterministic case when wind power generated can be accurately predicted over the scheduling horizon, the maximum steady schedule that the combined wind-storage plant can meet without deviations is 0.5 p.u. However in the probabilistic case where wind power generated is not accurately known, the maximum steady schedule that can be met is obtained as 0.25 p.u. It should be noted that Case 4, where the steady generation commitment is 1.0 p.u., corresponds to the full installed capacity of the wind farm. Also, in this case, the expected deviations in the probabilistic case are less than the deviations in the deterministic case. This is because there are nonzero probability values associated with wind power generated during some intervals being greater than the deterministic values of wind power generated during the same intervals of the scheduling horizon.

Thus results show that with good short-term forecasts available, a combined wind-storage plant may guarantee meeting the committed generation schedules over at least a small period of time. The better the available forecasts, the higher is the generation commitment that can be met by wind farms coupled with energy storage units. Also, it can be clearly seen that the outcome of this analysis is dependent on statistical distribution of the wind power forecast data. This brings out the importance of incorporating the information embedded in probability distribution of wind power forecasts in the formulation.

4.2 Storage sizing for wind energy balancing applications

Motivation and prior art

Balancing supply and demand of electric power is becoming increasingly difficult with increasing wind penetration into existing power grids since both wind power and the system loads are variable and represented by forecasts. Even with state-of-the art forecasting techniques, actual wind generation can be substantially deviated from the forecasted values. In addition, the system load is also variable and needs to be forecasted ahead of time for unit commitment and system planning and operation purposes. Although daily loads follow a pattern, every forecast is associated with a certain degree of uncertainty. In a system consisting of only wind generation and load, with minimum or no connection to the grid, the task of energy balancing is extremely difficult. Incorporation of energy storage units is being considered as a possible solution to this problem. However, energy storage units till date are expensive. Hence the question that arises is, given a generation-load system, what is the optimal amount of storage required.

Different energy storage sizing requirements for different load balancing horizons such as intra-week, intra-day, intra-hour, and real time have been addressed in [144]. However, uncertainties of load and generation forecasts are not considered. In [145] authors have addressed optimal sizing of storage based on forecast uncertainties of wind power generation. However, meeting the load is not the objective here. Rather, the forecasted wind generation is treated as the bid in an electricity market, and storage serves as a hedge for the uncertainties. Also, an exhaustive approach is used to compute the energy storage requirement followed by a probabilistic study of the computed required storage sizes. In [146], a methodology on the design of a wind farm battery energy storage system to realize power dispatchability is described. The uncertainties of wind forecasts have been considered by studying the battery sizing under different forecast scenarios. However, load forecast uncertainties have not been considered here. Furthermore, the uncertainties have not been considered in the problem formulation. In [147] a methodology capable of evaluating the impact of wind generation and load uncertainties on the balancing resources requirements has been developed. However, authors have mainly considered the spinning and non-spinning reserves required on a large-scale grid-level system. Energy storage options have not been considered.

In this work [148, 149] the optimal storage size has been computed for energy balance in a system consisting of wind generation and load taking into consideration the uncertainties in forecasts of both load and the wind generation. Optimal storage size is given by the optimal energy capacity and the optimal power capacity. The minimum initial stored energy is also computed. The methodology presented can be easily modified to address a system with other renewable sources of generation such as solar power. In addition, since no specific storage technology had been considered, the methodology can be modified to consider any specific type of storage and scaled to consider a renewable-storage system at the large-scale grid level or at a small-scale system such as smart buildings. The idea is also applicable in the microgrid framework with renewable generation and with minimum or no connection to the electric grid.

Methodology incorporating statistical distribution of wind generation and load forecasts

The overall methodology can be depicted by Figure 75. It has two main stages, the optimization stage and the validation stage. The inputs to the optimization stage are load and wind generation forecasts with uncertainty specifications and outputs are estimates of optimal storage size required. These estimates are validated in the Monte Carlo simulation based validation stage.

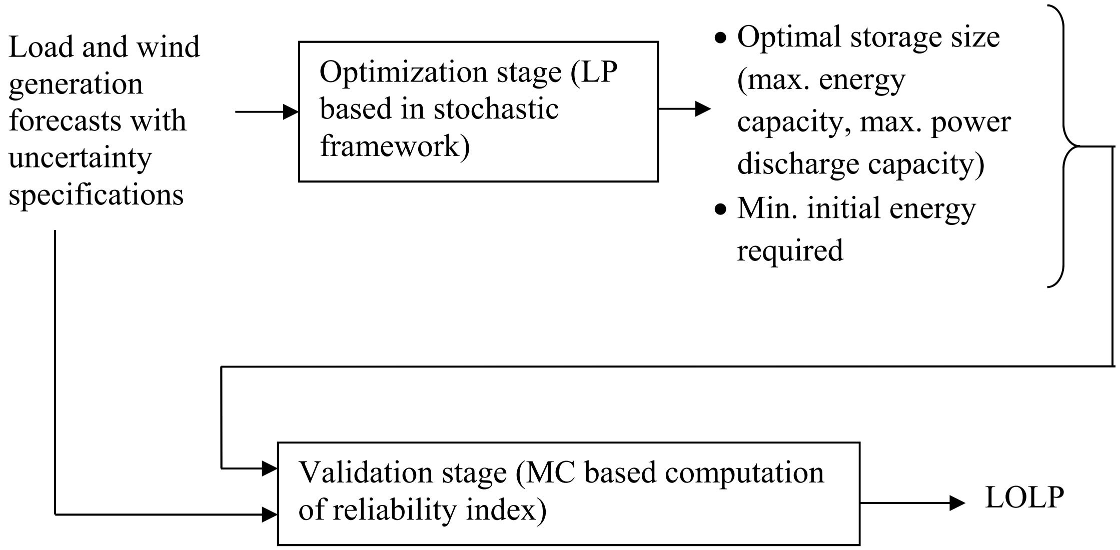

Load and wind generation forecasts with uncertainty specifications
Optimization stage (LP based in stochastic framework)
• Optimal storage size (max. energy capacity, max. power discharge capacity)
• Min. initial energy required
Validation stage (MC based computation of reliability index)
LOLP

Figure 75: Overview of methodology

Treatment of wind power and load forecast data

It has been assumed that wind power and load forecasts are available for every interval over the planning period. In addition, the forecast uncertainties quantified by confidence intervals or probability distribution of errors are also available. The probability distribution of forecast errors is taken to be Gaussian with zero mean and known standard deviation which may vary between different intervals.

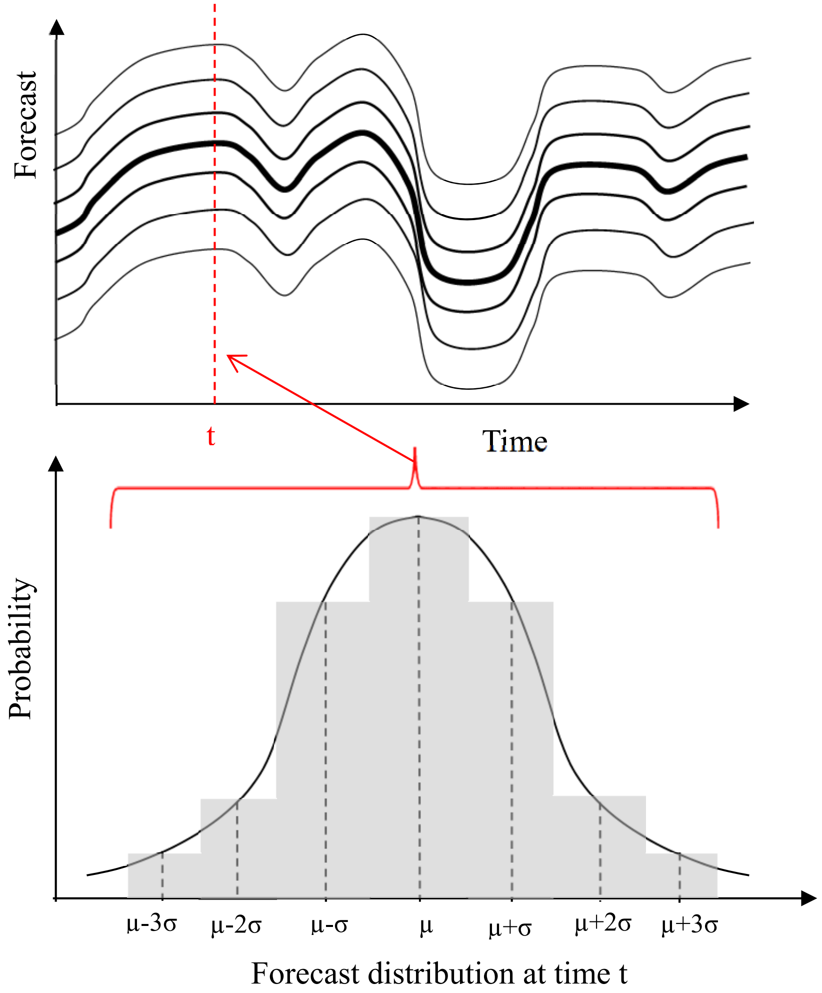


Figure 76: Wind power forecast with probability distribution of errors

Figure 76 demonstrates the discretization process for the wind power forecasts. The continuous probability distribution curve is discretized to quantize the forecasts into different levels. The process of discretization is required for the optimization formulation. In this work, the discrete levels considered are [μ-3σ, μ-2σ, μ-σ, μ, μ+σ, μ+2σ, μ+3σ] with corresponding probabilities obtained from the given probability distribution function. Here μ is the sum of the forecasted value at an interval and the mean of the forecast error. Since the forecast error is assumed to have a zero mean, μ represents the forecasted wind power. Also, σ represents the known standard deviation of the forecast error which can vary between different intervals.

A similar procedure is followed for the load forecast curve, i.e. the continuous probability distribution curve is discretized to generate discrete load levels with probabilities from the continuous probability distribution function.

Problem Formulation

First consider the optimization stage. Since the variables, namely wind generation and loads are probabilistic, a linear program is formulated in stochastic framework [150].

Minimize

$$C_{ES}.E_{S_{max}} + C_{PS}.P_{S_{max}} + C_{Einit}.E_{Sinit}+\sum_{k=1}^{N}\left[\pi \sum_{i,j} \rho_{W_{k_i}} \rho_{L_{k_j}}.\max\left\{0, P_{L_{k_j}} - P_{W_{k_i}} - P_{S_k}\right\}\right] \tag{4.12}$$

In each case the search space is restricted by the following constraints:

$$E_{S_{k+1}} = E_{S_k} - P_{S_k} \qquad \forall\, k = 1,2, \ldots .N \tag{4.13}$$

$$E_{S_{min}} \leq E_{S_k} \leq E_{S_{max}} \qquad \forall\, k = 1,2, \dots .\, N \tag{4.14}$$

$$\left|P_{S_k}\right| \leq P_{S_{max}} \qquad \forall\, k = 1,2, \dots .\, N \tag{4.15}$$

where

$$\mathrm{Prob}\left(\widetilde{P_{W_k}} = P_{W_{k_i}}\right) = \rho_{W_{k_i}} \tag{4.16}$$

and

$$\mathrm{Prob}\left(\widetilde{P_{L_k}} = P_{L_{k_i}}\right) = \rho_{L_{k_i}} \tag{4.17}$$

P_{L_k} is the load in interval k. P_{S_k} is the power discharged from the storage unit in interval k with a positive value indicating discharge and a negative value indicating charging of the storage unit. $P_{S_{max}}$ is the maximum rate of charge or discharge from the storage unit and is computed from the optimization program. $E_{S_{max}}$ is the maximum energy limit of the storage and is computed by the optimization program, $E_{S_{min}}$ is the minimum energy required in the storage unit at the start of the planning horizon and is also computed from the optimization program. Here $E_{S_{min}}$ is taken as zero, i.e. allowing deep discharge.

The capital cost of energy storage consists of an energy component, C_{ES} (\$/kWh) and a power component, C_{PS} (\$/kW). Another cost term C_{Einit} (\$/kWh) is introduced to reduce the dependence on initial stored energy of the storage unit to meet the objectives. π is a constant penalty term which is chosen to be extremely high (here taken as 40,000) to minimize the effect of the energy imbalances. This term is assumed to be the product of two components, a market price in the interval (40$/kWh), and a penalty factor over the market price for energy imbalances

(1000 p.u.). The treatment of the penalty term is similar to that taken in [135]. N is the number of intervals considered in the planning period.

The efficiency of the battery has been assumed to be 100% and no constraint has been placed on the cycle life of the battery.

Now consider the validation stage which computes the "goodness measure" of the optimal storage estimates by computing system reliability. LOLP is calculated using Monte Carlo type simulations as described in the following [151]:

- *Step 1* – Set the maximum iteration number and let the initial iteration number n = 1.
- *Step 2* – Sample the system state randomly (load level, wind generation) based on the given forecast error distribution and perform a simulation to check for a loss-of-load event. Let α_n be defined as follows:

$$\alpha_n = \begin{cases} 1 & \text{sampled scenario is loss} - \text{of} - \text{load event} \\ 0 & \text{otherwise} \end{cases} \tag{4.18}$$

Please note that the sampled scenario is the entire period under study (here a week). Thus, even with the occurrence of a single time interval (here one hour) of loss of load, the corresponding entire period under study (the whole week) is classified as a loss-of-load event. The resulting LOLP estimates the probability that a particular period (the week) will encounter at least one interval (hour) of loss of load.

- *Step 3* – Calculate LOLP, and variance of the estimated LOLP.

$$\widehat{\mathrm{LOLP}}_n = \frac{1}{n}\sum_{j=1}^{n} \alpha_j \tag{4.19}$$

$$\mathrm{V}\left(\widehat{\mathrm{LOLP}}_n\right) = \frac{1}{n}\left(\sum_{j=1}^{n} \frac{1}{n}\alpha_j^2 - \widehat{\mathrm{LOLP}}_n{}^2\right) \tag{4.20}$$

- *Step 4* – Check whether the variation $V(\widehat{\mathrm{LOLP}_n})$ is less than a specified threshold. If true or n > Nmax, stop; otherwise, n = n + 1, go to step 2.

Case study and results

The proposed methodology has been tested on a system consisting of a commercial facility which derives its energy requirements from wind power. The policy of the commercial facility is to maximize the use of "green" wind power and minimize energy purchased from the grid. Hence, the facility intends to invest in battery energy storage for energy balance. The problem is to find out the optimal size of the required energy storage unit.

It should be noted that instead of commercial load, residential or industrial loads could also have been considered. Further, the analysis presented here could be extended to grid-level renewable generation and loads with grid-level storage technologies such as pumped hydro or compressed air energy storage.

The planning horizon considered here is a week with granularity of one hour. The logic behind such a consideration is that the weekday load patterns are different from weekend load patterns, and the load patterns for the entire week repeat itself during a season. The methodology could be easily extended to incorporate longer term forecasts encompassing multiple seasons.

The optimal storage size required for the system in presence of wind power and load forecasts is obtained. Both the wind generation and load forecast errors are assumed to have a Gaussian probability distribution [152] with zero mean. The wind power forecast errors are assumed to have a standard deviation of 20% [152] of the maximum wind power generated during the week, and the load forecast errors are assumed to have a standard deviation of 2%

([153], [154]) of the peak demand. For simplicity, the standard deviations of the forecasts are considered uniform over all intervals of the period under study here. However, the formulation can also incorporate different standard deviations for different intervals. This feature is particularly important since forecast uncertainties are higher for longer-term forecasts compared to shorter-term forecasts.

The optimal storage parameters, namely the energy capacity, power capacity, and minimum initial energy required at the start of the week, are computed. For comparison, the same parameters are also computed for the system in a deterministic scenario when both the forecasts are accurate. Thus two cases are considered:

(I) Deterministic wind and deterministic load

(II) Stochastic wind and stochastic load

The energy storage unit considered for this work is a NaS battery. A NaS battery is one of the storage technologies that can be used for commercial and industrial energy management applications. However, it should be noted that instead of NaS batteries, any other battery energy storage technology can be considered using the same analysis by using the corresponding characteristics. Table 8 shows some of the typical cost specifications of a NaS battery used in such applications. Referring to these cost figures, $\mathrm{C_{ES}}$ was taken as 500$/kWh, $\mathrm{C_{PS}}$ as 3000$/kW, and $\mathrm{C_{Einit}}$ as 500$/kWh in the optimization program.

Table 8: Cost Specifications for NaS battery (From Table 4-15 in [155])

Total cost ($/kW)	3200-4000
Cost ($/kW-h)	445-555

The load considered is a commercial facility [156]. The available data being per-unitized, the base power is assumed to be 1 MW. This assumption is based on average power consumption data in a commercial building [157]. The daily load cycle for a weekday is provided in Table 9 for ease of reference. The load forecast for the week is simulated by repeating the given load curve five times for five weekdays and a 90% scaled-down load curve twice for the weekend as shown in Figure 77.

Table 9: Average hourly demand in a day for a commercial facility ([156])

Hour	Load (MW)	Hour	Load (MW)
1	0.908	13	1.355
2	0.852	14	1.338
3	0.89	15	1.37
4	0.865	16	1.385
5	0.824	17	1.426
6	0.93	18	1.403
7	1.042	19	1.261
8	1.167	20	1.217
9	1.302	21	1.1
10	1.49	22	0.999
11	1.538	23	0.961
12	1.454	24	0.878

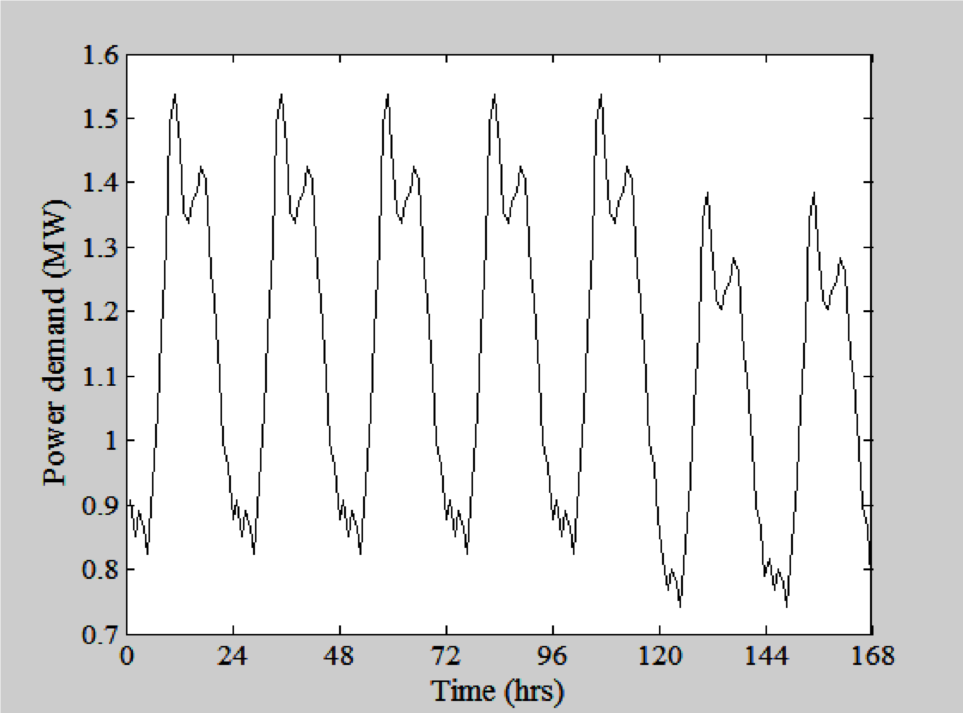


Figure 77: Load forecast for the week

Hourly wind generation data for a week has been used as wind forecasts. The data corresponds to total wind generation in the PJM RTO from Jan 20-26, 2010 [158]. The wind forecasts for a week are shown in Figure 78.

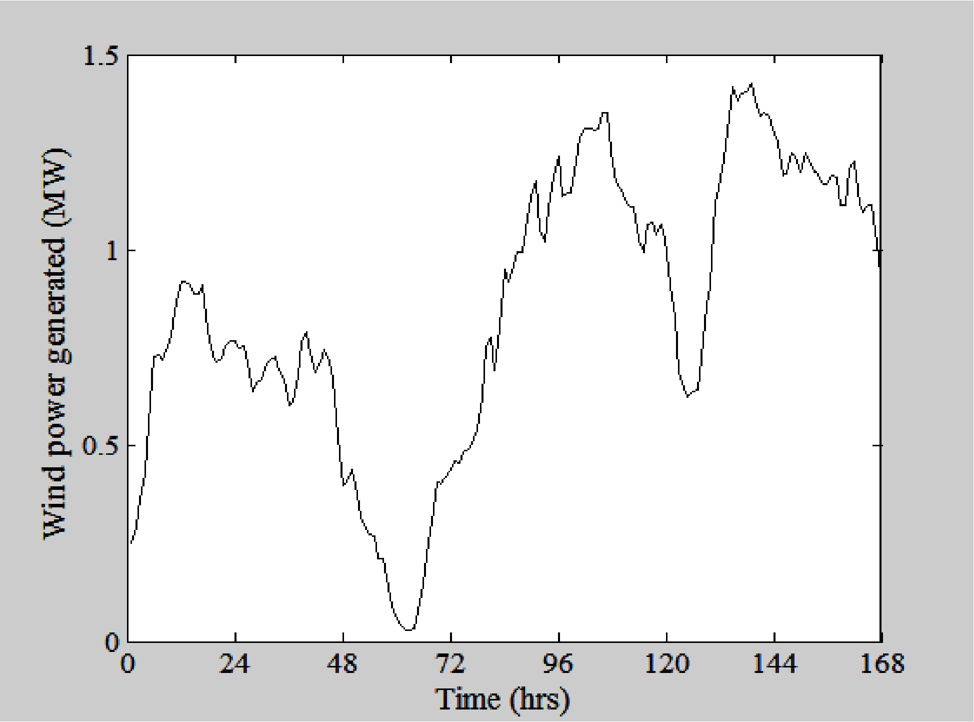


Figure 78: Wind forecast for the week

Consider the case when both the wind forecast and load forecasts are accurate. This is the deterministic case. Thus, the actual demand follows Figure 77 and actual wind power generated follows Figure 78.

With the target of minimizing under-generation, the optimal parameters are computed by the optimization program and LOLP is estimated. These are shown in Table 10. As it can be seen the system has zero LOLP, indicating that there are no events of unmet demand with the computed optimal storage size parameters.

Table 10: Deterministic case results

Condition		Min. under-generation
E_{Sinit} (MWh)		54.24
E_{Smax} (MWh)		54.24
P_{Smax} (MW)		1.45
Reliability measures for loss of load	LOLP	0
	Mean lost load frequency (hrs/week)	0
	Mean unmet energy (MWh/week)	0

Results show that in presence of accurate forecasts, the energy storage size recommended for the given system for the week under study is 54.2401 MWh and power capacity required is 1.4467 MW. Also, a minimum of 54.2401 MWh of energy needs to be stored at the beginning of the week. It is interesting to note that the total energy consumed by the load during the week (190.094 MWh) is greater than the total energy generated from wind during the week (141.7958 MWh). The difference of energy between these two quantities is the net energy deficit of the system and is found to be equal to the difference of energy stored in the storage unit at the beginning (54.2401 MWh) and the end of the week (5.9419 MWh). This fact directly follows from the law of energy conservation. The state of charge of the battery during different times is shown in Figure 79.

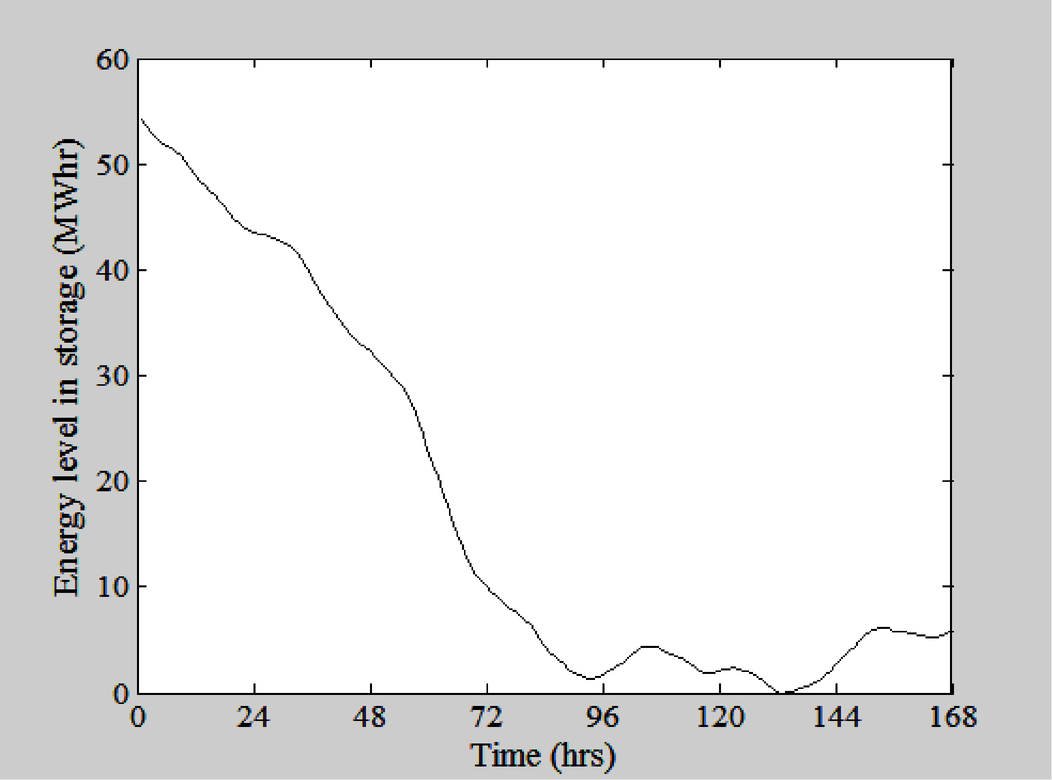


Figure 79: Energy level in storage in the deterministic scenario

The state of charge of the storage unit has a downhill slope for the most part indicating that the storage unit is in discharge mode. This is again because the total energy available from the wind is less than the total energy consumed by the load. At the 133rd hour (midway between the 5th and 6th day) the storage is completely discharged. This fact in addition to the zero LOLP indicates that the storage parameters computed by the optimization program are indeed optimal. Depending on the wind and load forecasts of the next week, the initial energy required at the beginning of the next week may or may not be different from the final energy in the storage unit at the end of the week under consideration. The balance is assumed to be taken up by the grid.

Now consider the case when the load and wind forecasts are not accurate. With the target of minimizing under-generation, the optimal storage size required is 192 MWh. With this storage size, there is no loss-of-load event. Thus, to achieve the same level of reliability in meeting load, the storage size to be invested in is approximately 4 times the size required when the forecasts are accurate. The results are shown in Table 11. These results illustrate the effect of forecast uncertainties on system reliability and performance.

Table 11: Stochastic load and stochastic wind

Condition		Min. under-generation
E_{Sinit} (MWh)		191.75
E_{Smax} (MWh)		191.75
P_{Smax} (MW)		2.17
Reliability measures for loss of load	LOLP	0
	Mean lost load frequency (hrs/week)	0
	Mean unmet energy (MWh/week)	0

Figure 80 and Figure 81 show respectively the forecasted and actual values of the load and wind generation over a week, generated by the Monte Carlo simulations with given forecast uncertainties.

If the storage size used is the same as the deterministic case, i.e. not considering the forecast uncertainties, the unmet demand is shown in Figure 82. With this storage size, there is 86% chance that a week will encounter a loss of load, and the average unmet energy is 0.95 MWh/week. Some load outage events are as high as 1.2 MW (Figure 82) which might be unacceptable. The mean loss-of-load frequency (4.142 hrs/week) can be roughly guessed from the clusters of dots in Figure 82.

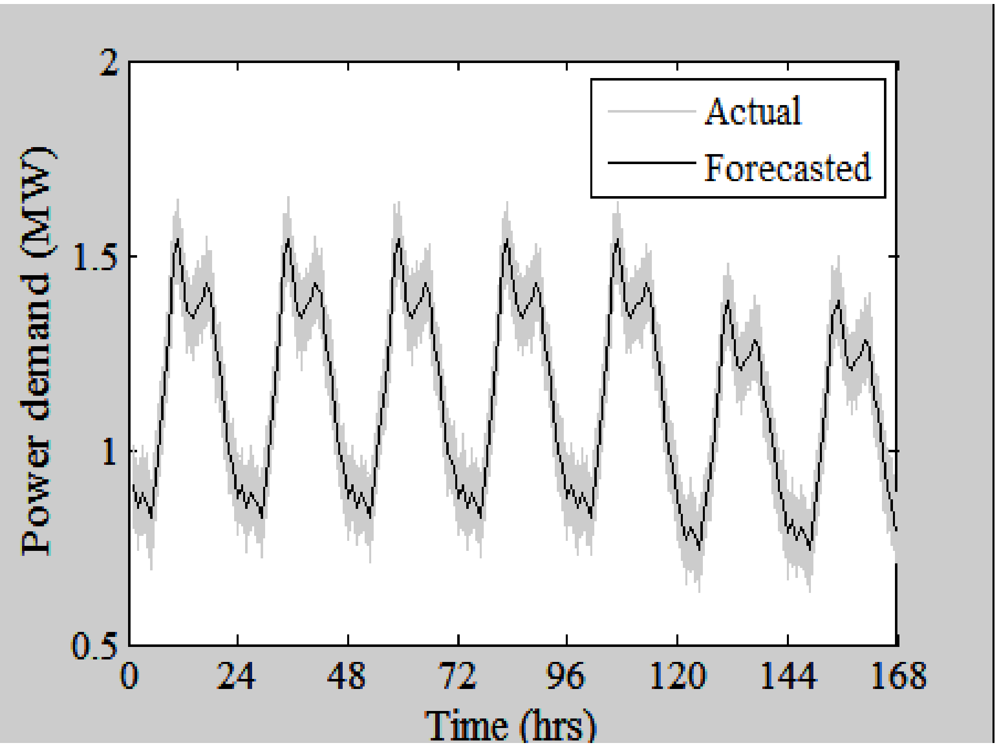


Figure 80: The actual and forecasted wind power obtained by Monte Carlo simulations

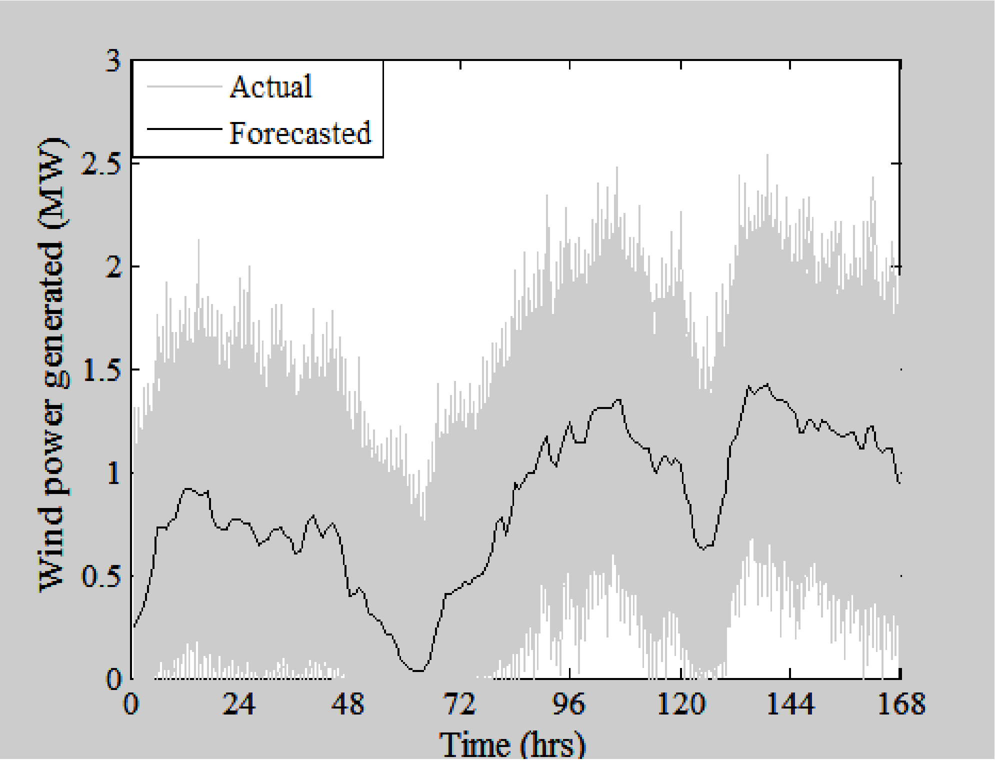


Figure 81: The actual and forecasted load obtained by Monte Carlo simulations

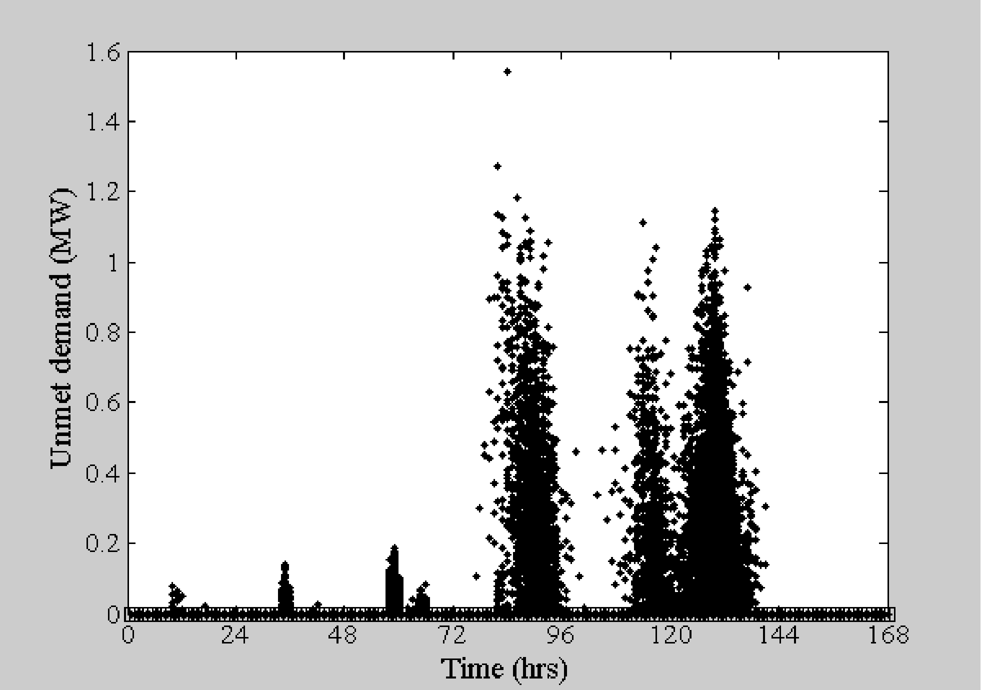


Figure 82: Unmet demand with storage parameters corresponding to deterministic case

Different weekly wind generation and load scenarios change the requirement of storage size accordingly. Hence, historic load and wind data for an extended period, such as a year should be used in a deterministic formulation to compute the optimal storage requirement for a system. However, this storage should be considered base-level storage. In the shorter term, when forecasts are available, the storage to be considered and allocated should be mobile storage technologies such as hybrid and electric cars.

The methodology presented in this work is effective in determining the optimal storage size required for energy balance purposes in a renewable generation-load system taking into account forecast uncertainties and meeting required reliability criteria. The approach is also useful in assessing performance and reliability metrics of a system with existing storage facilities. And once more, the importance of incorporating statistical properties of the forecast data into the formulation is clearly seen.

4.3 Forecasting of wind power

Motivation and prior art

Numerous studies have addressed the problem of accurately predicting the power output from wind farms both in long-term as well as short-term scenarios. Several sophisticated wind power forecasting techniques have also been developed and implemented. In [159], a bibliographical survey on the research and developments in the fields of wind speed and wind power forecasting has been presented.

Physical models based on numerical weather prediction (NWP) models developed by meteorologists for large-scale weather prediction provide good results when predicting the wind speeds in the long term, but are not as effective for short-term prediction. Several statistical models have been developed for the purpose of wind atlas preparations, prediction of wind turbine maintenance scheduling, and estimation of monthly and annual energy production at wind farm sites. These techniques based on analyzing historical wind speeds and wind directions at the sites under consideration are not effective for hourly wind speed forecasting. Time series based models are extremely popular models for wind speed and wind power forecasting. These models can surpass many other models in short-term prediction. In fact, most state-of-the-art short-term wind energy forecasts build upon this idea by taking recent data and using more sophisticated mathematical techniques to produce a forecast of the wind energy. This is very similar to the short-term load forecast, where a major predictor is past behavior. However, this causes a characteristic error of “delay.” The forecast tends to trail reality as it is strongly influenced by a persistence forecast. Alternatively, a day-ahead wind energy forecast may have the correct profile of the wind production, but shifted in time forward or backward. This would also give larger errors when the wind production is moving in fast ramps before or after it was expected. In both instances, it is up to the system operator to prepare for the wind event, and by looking at both the forecast and actual generation data, act appropriately. For example, on February 26, 2008, the Electric Reliability Council of Texas (ERCOT) had to call for an Emergency Electric Curtailment Plan (EECP) at 18:41 due to a worsening imbalance between generation and load which led to a decline in system frequency [160]. One of the major causes of this event was a large and rapid ramp-down of about 1,500 MW in 3 hours in wind generation. Post event analysis identified a lack of good wind power forecast methods, especially those for predicting wind power ramp-ups and ramp-downs.

Advanced short-term wind energy forecasts should use offsite observations to get signals about the upcoming ramping behavior. With the ideal scenario of having multiple offsite observations all being a radius of an hour away, an oncoming ramp could be more accurately forecast an hour ahead of time. Spatial correlation models take the spatial relationship of wind speeds at different sites into account. These models involve the measurement and timely transmission of wind speed values at a number of spatially correlated sites. Furthermore, since these techniques are direction dependent, collection of wind direction data is also essential for application in wind power forecasting. These models are useful when estimating wind speeds at a wind farm location using measured wind speeds at another wind farm location within a group of wind farms as depicted in Figure 83. The short-term wind power forecasting method presented here is based on these models.

With the development of artificial intelligence techniques, various new models for wind speed and power prediction are being researched. These methods include ANN, fuzzy logic methods, support vector machine, and various hybrid methods. These models have been proved effective for short-term predictions in most experiments, but these require large sets of historical data for parameter estimation and model training which can be a time consuming process.

While no single method is sufficient, most researchers use a combination of multiple methods to develop wind prediction models addressing the peculiarities of the wind speed prediction problem. A spatial correlation based wind speed prediction model has been proposed in [161] which uses the fuzzy logic method. In [162] an ANN-based method is used to predict wind speeds by spatial correlation. A fuzzy neural network based wind speed forecasting method using spatial correlation model is presented in [163], where two remote sites were chosen with the base site so that the three sites were lined along the direction of the prevailing winds. The pressure gradients, the heat transfer, the terrain landscape, and three cases of time delay were considered in the study. The proposed model exhibited superior results compared to other network models. Another model based on the ANN method and spatial correlation was proposed in [164]. The mean monthly wind speeds of reference stations were used to predict the wind speeds at target stations. A comparison of prediction results with actual data confirmed that the ANN method based on wind speed of reference stations could predict the wind speed of target stations without any topographic details or other meteorological data. In [165] a linear prediction model for wind speed time series prediction in the short term has been proposed that uses least squares to determine the coefficients of the linear model. However, spatial correlation is not taken into account.

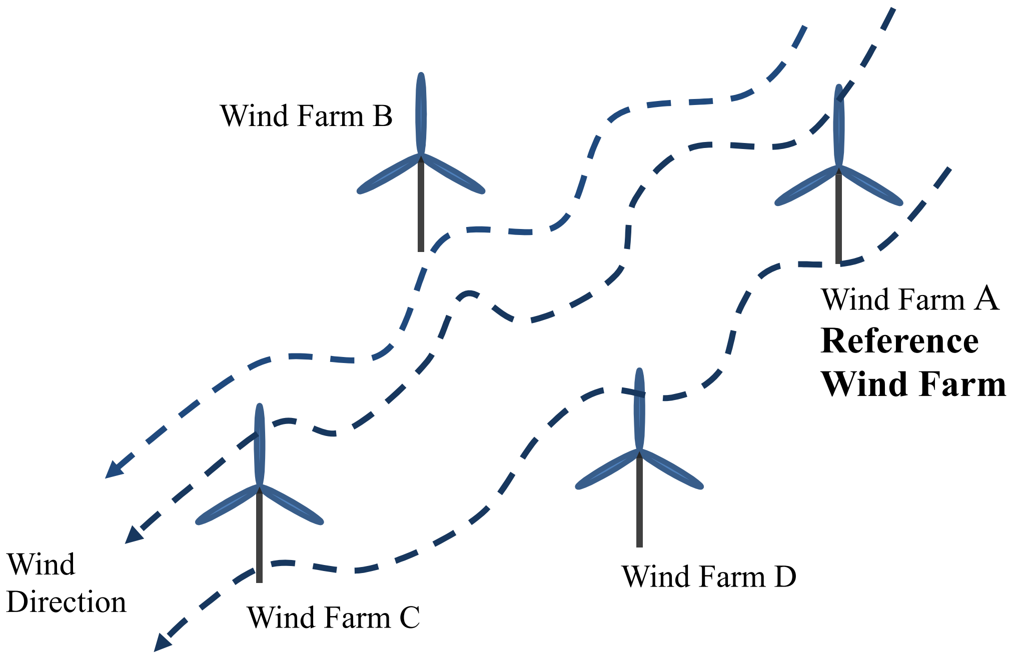


Figure 83: A group of wind farms and the wind direction

Methodology based on least squares estimation

This work presents a least squares based method that predicts wind power in the short term using the spatial correlation of wind speeds of a group of wind farms distributed over a region. The first step of the method [166] consists of identifying a reference wind farm location from a group of wind farms under study. The reference wind farm location is identified as one that results in maximum cross correlation of wind speeds with other wind farm locations at a positive time lag. A study similar to that done in [167] and [168] has been conducted for the delay at maximum cross correlation for pairs of wind farms for this purpose. The wind direction during the time interval considered can be used to corroborate the findings of the varying time lag cross correlation study. The next step is the prediction of wind speeds by the least squares method using the time lag at maximum cross correlation.

In this method, the time lag at maximum cross correlation of wind speeds between two locations is used. The wind speed predicted at the i^{th} instant at location 2, denoted by $\hat{\mathrm{v}}_{\mathrm{i}}$ is expressed as a linear function of the wind speed at location 1 at the instant "i-lag" given by the following equation:

$$\widehat{\mathrm{v}_{\mathrm{i}}} = \beta_{1_{\mathrm{i}}} + \beta_{2_{\mathrm{i}}} \mathrm{w}_{\mathrm{i-lag}} \tag{4.21}$$

The parameters of equation (4.21), $\beta_{1_{\mathrm{i}}}$ and $\beta_{2_{\mathrm{i}}}$ are determined by the relationship of the last five measurements of wind speeds at location 2 and those at location 1 delayed by the lag at maximum correlation. Matrices A and B are defined as follows:

$$\mathrm{A}_{\mathrm{i}} \triangleq \begin{bmatrix} 1 & \mathrm{w}_{\mathrm{i-5-lag}} \\ 1 & \mathrm{w}_{\mathrm{i-4-lag}} \\ 1 & \mathrm{w}_{\mathrm{i-3-lag}} \\ 1 & \mathrm{w}_{\mathrm{i-2-lag}} \\ 1 & \mathrm{w}_{\mathrm{i-1-lag}} \end{bmatrix} \tag{4.22}$$

$$\mathrm{B}_{\mathrm{i}} \triangleq \begin{bmatrix} \mathrm{v}_{\mathrm{i-5}} \\ \mathrm{v}_{\mathrm{i-4}} \\ \mathrm{v}_{\mathrm{i-3}} \\ \mathrm{v}_{\mathrm{i-2}} \\ \mathrm{v}_{\mathrm{i-1}} \end{bmatrix} \tag{4.23}$$

Where w_{j} and v_{j} are the measured wind speeds at the j^{th} instant at locations 1 and 2 respectively. This is followed by computation of the parameters β_{1_i} and β_{2_i} using the least squares method expressed as:

$$y_i = \begin{bmatrix}\beta_{1_i} \\ \beta_{2_i}\end{bmatrix} = \left[A_i{}^T A_i\right]^{-1} . A_i{}^T B_i \tag{4.24}$$

Once the wind speeds are estimated, the wind power outputs can be found from power curves of wind turbines. These curves mapping wind speeds to power outputs are typically provided by the manufacturer of the turbines.

Case study and results

Measurements of wind speeds starting from 10:10:00 AM on May15, 2006 to 11:50:00 PM on May 20, 2006, taken at intervals of 10 minutes from meteorological towers at heights of 50 m in Bureau County and Henry County have been compiled together for this experiment. The locations of these counties in Illinois are shown in Figure 84.

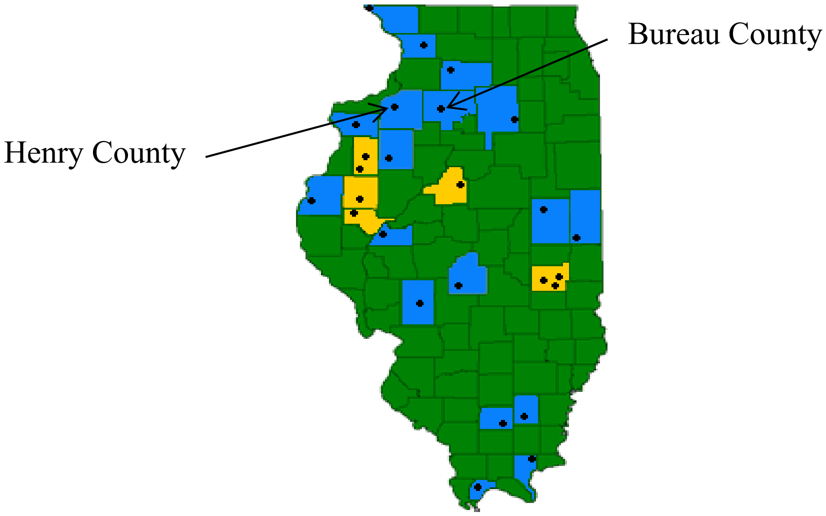


Figure 84: Location of wind data measurement stations (Source: www.illinoiswind.org)

The distance between the measurement towers at the two counties is approximately 40 miles. The cross correlation coefficients of wind speed time series using data from 10:10:00 AM on May15, 2006 to 4:20:00 AM on May 18, 2006 between the two counties at zero time lag are presented in Table 12.

Table 12: Cross correlation coefficients

	Bureau Co.	Henry Co.
Bureau Co.	1.0000	0.5214
Henry Co.	0.5214	1.0000

The cross correlation factors of the wind speed time series between the two counties were studied for varying time lags. The wind farm that yielded peak cross correlation value with wind speeds at the other location at a positive time lag was designated as the reference wind farm. In the case study conducted, it was found that the peak in the cross correlation values at Henry Co. with respect to wind speeds at Bureau Co. occurred at a positive time lag. Thus, Bureau Co. is designated as a reference location. Wind speeds at the other location were predicted using the measurement of wind speeds at Bureau Co. as the reference. Figure 85 and Figure 86 show the variation of cross correlation factors of Henry Co. with respect to Bureau Co. for varying time lags.

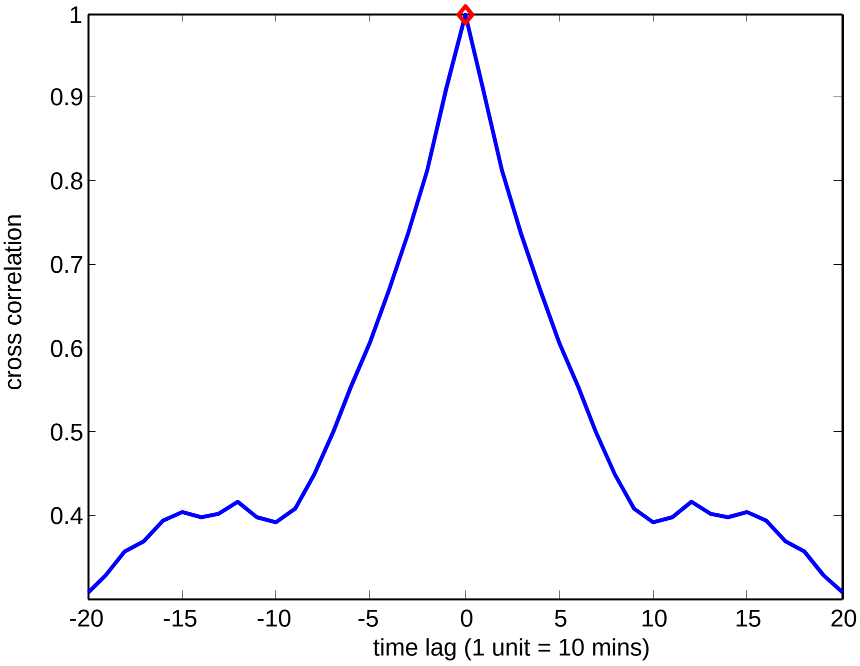


Figure 85: Cross correlation of Bureau Co. with respect to Bureau Co. for varying time lag

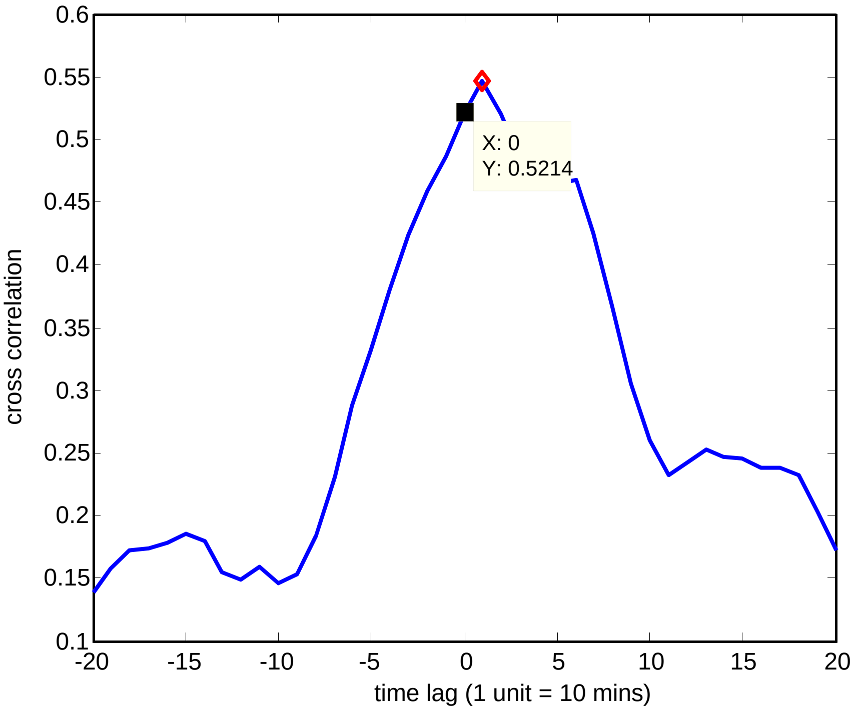


Figure 86: Cross correlation of Henry Co. with respect to Bureau Co. for varying time lag

Figure 85 shows the cross correlation of the wind speed time series measured at Bureau Co. with respect to Bureau Co. As expected, the maximum cross correlation occurs at a zero time lag. The varying time-lag cross-correlation of the wind speed time series at Henry Co. with respect to Bureau Co. in Figure 86 shows a maximum correlation for a positive time lag of +1 which equals 10 minutes. The wind direction during the period under consideration was also studied. It was found that the wind predominantly blew from the North and North-East directions during this time. Comparing the relative geographic positions of the four counties, it was found that Bureau Co. was located upwind compared to Henry Co. In the next step, wind speeds at Henry Co. were simulated from the wind speed time series at Bureau Co. as a reference using the proposed least squares based method using data starting from 4:30:00 AM on May18, 2006.

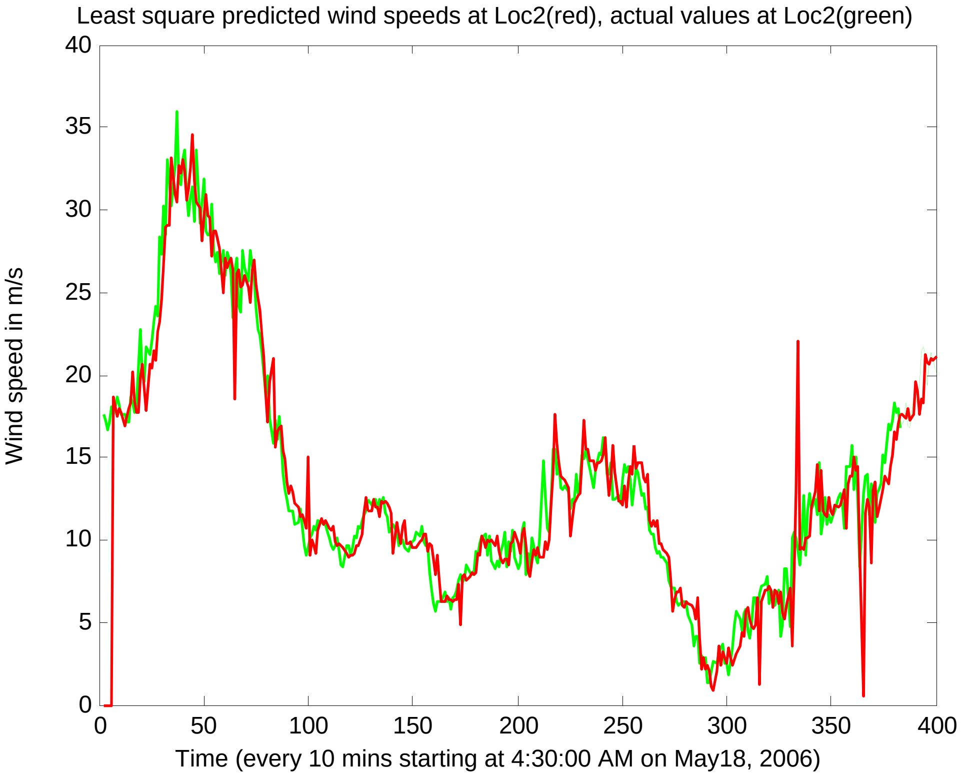


Figure 87: Estimated (in red) and measured (in green) wind speeds at Henry Co.

Figure 87 depicts the least squares predicted wind speed time series compared to the actually measured wind speed time series at Henry Co. The least squares based methodology was compared with the predictions of wind speeds using the Persistence model. For a Persistence model of order “p”, the forecasted value of wind speeds in the next time interval is given by the mean wind speeds over the last “p” intervals [169]. In most studies, a value of p = 1 is used to generate short-term wind speed predictions. This kind of model is a time series based model for wind speed prediction which assumes that the average value of wind speeds measured during an interval of time persists and remains the same over the next interval of time. However, Persistence models suffer from the drawback that the predictions of wind speeds are made without taking into account the wind speeds at nearby locations. Also, with increase in the duration of time intervals, errors in prediction increase.

To quantify the error in predictions using both the proposed least squares based model and the Persistence model of order p = 1, percentage error was defined as:

$$\text{Percentage error} = \frac{|\text{Predicted speed} - \text{Actual speed}|}{\text{Actual speed}} \times 100 \tag{4.25}$$

The average percentage errors of wind speed predictions at Henry Co. are tabulated in Table 13. It can be seen that the proposed model yields superior forecasts of wind speeds compared to the Persistence model.

Table 13: Average percentage errors

Location	Least squares model	Persistence model
Henry Co.	0.3712%	0.866%

Comparing the least squares predicted wind speeds at Henry Co. with the actual measured wind speeds, it was found that the proposed method yielded fairly accurate predictions of wind speeds in the short term. The proposed model also generated superior results compared to the Persistence model of order 1. This application demonstrates the effectiveness of least squares estimation in wind forecasting using spatial correlation information.

5. CONCLUSIONS AND FUTURE WORK

This dissertation has demonstrated the applicability of data mining and graph-theoretic algorithms and concepts in solving three problems of power systems in the era of Smart Grids, namely information processing and visualization of power system time-varying data, optimal design of wind farm collector systems, and large-scale integration of wind power into power grids. In addressing each of these problems, it has been found that several techniques and tools available in data mining and graph theory literature are extremely useful when applied directly as off-the-shelf solutions or in a slightly modified manner to suit the nature of the problem. Hence it is concluded that graph theory and data mining serve as rich resources, and it is hoped that this dissertation will pave the way for utilizing these to address other Smart Grid problems as well. The overall contribution of this dissertation is in identifying three challenges posed by Smart Grids and proposing data mining and graph theory based techniques and methodologies to solve them.

In Chapter 2, the first challenge has been addressed and a methodology has been proposed for processing power system time-varying data for important information and its visualization. Hence this work presents a direct application of data mining in power systems. Transient stability run results are the source of such data in this work. Extracted information includes abnormal dynamic response indicating some form of error or condition requiring attention. Also identified are the characteristics of the wide area power system, grouping nodes of similar response. Data volume, a problem frequently encountered in large power systems is also addressed with a clustering based method to reduce data volume without loss of information. Another contribution is in use of spark-lines for visualizing transient stability information and their graph drawing based automatic placement without overlaps on a geographic map of the system. Important to note is that although the case study presented analyzes generator frequencies, the presented methodology can also be applied to other data from transient stability results such as bus voltages. The algorithms are extremely fast even when run on thousands of data points and hence can be used for real-time analysis in tracking mode with PMU measurements for example. These applications can be addressed in a future continuation of the work presented in this dissertation.

Chapter 3 proposes novel algorithms for automatically generating an optimal design for a wind farm collector system cable layout configuration. Developed algorithms have capabilities including improving on a minimum spanning tree design by creating external splice locations separate from the wind turbine locations, addressing the constraint of a prespecified maximum number of turbines connected to a feeder cable, computing direction and magnitude of power flow on each cable and assigning cable sizes from a table of available cable sizes, addressing trenching restrictions in the design, computing reliability, and performing economic analysis of the layout. Results show that the algorithms proposed can be used to generate a design that has minimum total trenching length, also taking into account constraints on maximum number of turbines on a feeder, and trenching restrictions. The generated designs achieve ~10% reduction in capital costs compared to a real-life cable layout configuration of a wind farm. The major contribution of this work is in the automatic generation of a starting layout design which is optimal with respect to total length. For other real-life constraints, this design can be considered as a base case and heuristics can be applied to cater to specific applications. Future work will involve automating voltage studies, short-circuit studies, insulation coordination which are crucial in analyzing the design of a wind farm collector system. The graph-theoretic approach established in this dissertation is anticipated to be useful in these future extensions.

In Chapter 4 the challenge of large-scale wind power integration has been addressed. Effectiveness of maintaining committed generation schedules over the period of an hour by incorporating energy storage with wind farms has been considered. Optimal charge discharge schedules have been computed with the objective of minimizing total expected schedule deviations over the scheduling horizon of one hour. Statistical distribution of the wind forecast data is found to be one of the main factors affecting the maximum steady generation commitment that can be met by a wind farm. Probabilities of predicted wind power in different intervals of the scheduling horizon are inputs in the problem formulation. Tests on a simple wind-storage model connected to a load verified the desired objectives. The results showed that with good short-term forecasts available, a combined wind-storage plant may guarantee meeting the committed generation schedules over at least a small period of time. The better the available forecasts, the higher the generation commitment that can be met by wind farms coupled with energy storage units.

Optimal sizing of an energy storage unit for energy balancing purposes has also been considered in Chapter 4. The optimal size, characterized by optimal energy capacity and optimal power capacity of the storage unit, has been computed. In addition, the minimum initial energy required to be stored at the beginning of an operational period has also been computed. The uniqueness of this work is in taking into account uncertainties of both wind generation and load forecasts. Another contribution of this work is in the use of reliability index loss-of-load probability (LOLP) to validate the computed optimal parameters. A linear programming method has been used in a stochastic framework to solve the optimization problem and Monte Carlo based simulations have been used to compute the reliability index. A system consisting of wind generation and a commercial load have been tested with the proposed methodology considering forecast errors of the order of 20% for wind generation and 2% for the load. It was found that for meeting a zero LOLP reliability criteria, the optimal storage requirement increased by about 4 times under uncertain forecasts compared to that in the accurate forecasts scenario. Also, the storage parameters found to be optimal in accurate forecasts scenario result in higher LOLP values and hence lower reliability under uncertain forecasts. The methodology presented is hence effective in determining the optimal storage size required for energy balance purposes in a renewable generation-load system taking into account forecast uncertainties and meeting required reliability criteria. The approach is also useful in assessing performance and reliability metrics of a system with existing storage facilities.

Finally, Chapter 4 also addressed wind power forecasting methods. A least squares based methodology forecasted wind speeds using spatial correlation information and wind speeds at nearby locations. The developed methods tested with wind speeds at locations in state of Illinois achieved better forecasts compared to the persistence model for wind speed forecasting. The proposed method is also important since it can capture wind ramp ups and ramp downs effectively. Future extensions of this work can take into account information of terrain of the group of wind farms. In spite of the extensive research in the area of wind power forecasting there is not a sufficient number of prediction models that encompass the topological behavior of different geographic regions, terrains, climate zones, and situations. Therefore building accurate and robust prediction models presents an open and challenging research area.

In summary, the term Smart Grid definitely encompasses a broad area, and it is beyond the scope of a single dissertation to address all of the problems associated with its real-life implementation. Again data mining and graph theory each represent a broad and rich resource of techniques, concepts, and algorithms which could be applied to several problems of Smart Grids. The critical step is to exploit these resources properly and narrow down on the best technique to solve the Smart Grid problems. Hence, the work in this thesis has required a deep understanding of all three areas of data mining, graph theory, and power systems. An important contribution of this work is to provide a coverage of data mining and graph theory and potential uses and application areas in Smart Grids, and in general power systems. It is hoped that this thesis will motivate further extensions in the areas addressed in this dissertation and beyond. The publications associated with this dissertation have already been receiving interest in the power systems community and hence it is anticipated that this dissertation will be a useful guide and foundation for future research.

APPENDIX: AVAILABLE CONDUCTOR SIZES AND PROPERTIES

The cable layout designs are dependent on the types of cables available. So here a discussion has been provided on available conductor sizes and properties. Most cables in the wind farm collector system are the underground type. Some of the common cable sizes used in large-scale wind farms can be found in [8]. In the collector system design work it is assumed that ACSR (Aluminum Conductor Steel Reinforced) cables of only the sizes mentioned in Table 14 are available for the collector system cable layout. Cables have limits on the amount of power that can be carried by them represented by cable ampacity limits. So, at cables closer to the substation where power from turbine units gets consolidated, multiple (double and triple) circuits of cables might be needed to provide sufficient ampacity levels. The cable ampacity is based on conditions [15] that cables are installed in sand with minimum cover of approximately 1 m, load factor is 100%, and maximum ambient earth temperature is 20 deg C. The dc resistances of the different conductors were obtained from [15] and [16]. The ac resistances are computed from the dc resistances according to the method described in [17] and are tabulated in Table 14. Approximate cable costs are also provided.

Table 14: ACSR cable sizes, properties, and costs

Al strand conductor size	**Continuous ampacity (Amps) [15]**	**DC resistance at 25 deg C (mΩ/m) [16]**	**AC resistance at 25 deg C (mΩ/m)**	**Cost ($/m)**
1/0	150	0.5482	0.5482	28
4/0	211	0.2741	0.2741	35
500 kcmil	332	0.1161	0.1184	42
750 kcmil	405	0.0774	0.0813	85
1000 kcmil	462	0.0577	0.0633	125

REFERENCES

[1] Smart Grid, [Online]. Available: http://energy.gov/oe/technology-development/smart-grid

[2] Installed wind capacity [Online]. Available: http://www.windpoweringamerica.gov/wind_installed_capacity.asp

[3] "20% wind energy by 2030 – Increasing wind energy's contribution to U.S. Electricity Supply," U.S. Department of Energy report, July 2008, [Online]. Available: http://www.nrel.gov/docs/fy08osti/41869.pdf

[4] M. A. Pai, *Computer Techniques in Power System Analysis*, 2nd ed. New Delhi, India: Tata McGraw-Hill, 2006.

[5] J. Zhu, *Power Systems Applications of Graph Theory*. Nova Science Pub. Inc., Sep., 2009.

[6] A. Cayle, "A theorem on trees," *Quarterly Journal of Pure and Applied Mathematics*, vol. 23, pp. 376-378, 1889. [Online]. Available: http://quod.lib.umich.edu/u/umhistmath/ABS3153.0013.001/43?rgn=full+text;view=pdf

[7] G. R. Krumpholz, K. A. Clements, and P. W. Davis, "Power system observability: A practical algorithm using network topology," *IEEE Transactions on Power Apparatus and Systems*, vol. 99, no.4, pp. 1534-1541, 1980.

[8] A. Monticelli and F. F. Wu, "Network observability: Theory," *IEEE Transactions on Power Apparatus and Systems*, vol. 104, pp. 1042-1048, 1985.

[9] K. A. Clements, G. R. Krumpholz, and P.W. Davis, "Power system state estimation residual analysis: An algorithm using network topology," *IEEE Transactions on Power Apparatus and Systems*, vol. 100, no. 4, pp. 1779-1787, 1981.

[10] J. B. A. London, L. F. C. Alberto and N. G. Bretas, "Network observability: a fast topological approach to identify critical measurements," in *Proceedings of International Conference on Power System Technology*, vol. 2, pp. 583-588, Dec. 2000.

[11] H. Mori and S. Tzuzuki, "A fast method for topological observability analysis using a minimum spanning tree technique," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 491-501, 1991.

[12] Y. Wu, “Improved measurement placement and topology processing in power system state estimation,” Ph.D. dissertation, Texas A&M University, 2007.

[13] J. Lei, “Using graph theory to resolve state estimator issues faced by deregulated power systems,” Ph.D. dissertation, Texas A&M University, 2007.

[14] R. Merlin and H. Back, “Search for a minimal-loss operating spanning tree configuration for an urban power distribution system,” in *Proceedings of 5th Power System Computation Conference (PSCC),* Cambridge, Sep. 1975.

[15] P. M. S. Carvalho, L. A. F. M. Ferreira, and L. M. F. Barruncho, “On spanning-tree recombination in evolutionary large-scale network problems—Application to electrical distribution planning,” *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 6, pp. 623-630, Dec. 2001.

[16] Y. Li and X. Chang, “A MST–based and new GA supported distribution network planning,” in *Proceedings of 2011 International Conference on Mechatronic Science*, Jilin, China, Aug. 2011.

[17] K. Singh, “Electricity network reliability optimization,” in *Proceedings of Operational Research Society of New Zealand (ORSNZ) Conference 2001*, University of Canterbury, Christchurch, New Zealand, Nov. 30-Dec. 1, 2001. [Online]. Available: http://www.orsnz.org.nz/conf36/papers/Singh.pdf

[18] J. Li, “Reconfiguration of power networks based on graph-theoretic algorithms,” Ph.D. dissertation, Iowa State University, 2010.

[19] K. Huang, D. A. Cartes, and S. K. Srivastava, “A multi-agent based algorithm for ring-structured shipboard power system reconfiguration,” in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 530-535, 2005.

[20] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*. Amsterdam, Netherlands: North-Holland, 1992.

[21] M. Hazewinkel, “Steiner tree problem,” in *Encyclopaedia of Mathematics*, Springer, 2001. ISBN 978-1556080104. [Online]. Available: http://eom.springer.de/s/s110270.htm

[22] C. Coulston and R. Weisshach “Routing transmission lines via steiner trees,” *IEEE Power Engineering Society General Meeting*, Jul. 2003.

[23] E. Míguez, J. Cidrás, E. Díaz-Dorado, and J. L. García-Dornelas, “An improved branch-exchange algorithm for large-scale distribution network planning,” *IEEE Transactions on Power Systems*, vol. 17, no. 4, pp. 931-936, Nov. 2002.

[24] Y. Song, H. B. Gooi, and C. K. Chan, “Algorithms for automatic generation of one-line diagrams,” in *IEE Proceedings on Generation, Transmission, and Distribution*, vol. 147, no. 5, Sep. 2000.

[25] A. D. A. Mota and L. T. M. Mota, “Drawing meshed one-line diagrams of electric power systems using a modified controlled spring embedder algorithm enhanced with geospatial data,” *Journal of Computer Science*, vol. 7, no. 2, pp. 234-241, 2011.

[26] P. Eades, “A heuristic for graph drawing,” *Congressus Nutnerantiunt,* vol. 42, pp. 149–160, 1984.

[27] N. Quinn and M. Breur, “A force directed component placement procedure for printed circuit boards,” *IEEE Transactions on Circuits and Systems, CAS-26,* vol. 6, pp. 377–388, 1979.

[28] T. Kamada and S. Kawai, “An algorithm for drawing general undirected graphs,” *Information Processing Letters,* vol. 31, pp. 7–15, 1989.

[29] T. M. J. Fruchterman and E. M. Reingold, “Graph drawing by force-directed placement,” *Software – Practice and Experience*, vol. 21, pp. 1129-1164, Nov. 1991.

[30] U. Fayyad, G. P.-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *AI Magazine*, pp. 37-53, 1996.

[31] A. O. M. Saleh and M. A. Laughton “Cluster analysis of power-system networks for array processing solutions,” *IEE Proceedings*, vol. 132, no. 4, Jul. 1985.

[32] S. Madan, W.-K. Son, and K. E. Bollinger, “Applications of data mining for power systems,” in *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1997.

[33] J. A. Steel, J. R. McDonald, and C. D’Arcy, “Knowledge discovery in databases: Applications in the electrical power engineering domain,” in *Proceedings of IEE Colloquium IT Strategies Information Overload*, Dec. 1997.

[34] A. Asheibi, D. Stirling, and D. Robinson, “Identification of load power quality characteristics using data mining,” in *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Ottawa, pp. 157-162, May 2006.

[35] K. Rogers and T. Overbye, “Clustering of power system data and its use in load pocket identification,” in *Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS),* Jan. 2011.

[36] H. Mori, “State-of-the-art overview on data mining in power systems,” in *Proceedings of IEEE Power Systems Conference and Exposition (PSCE),* 2006.

[37] M. McGranaghan, “Making connections: Asset management and the smart grid,” *IEEE Power and Energy Magazine*, vol. 8, pp. 16-22, Nov. - Dec. 2010.

[38] P. Myrda, “Optimizing assets: Smart grid-enabled asset management,” *IEEE Power and Energy Magazine*, vol. 8, pp. 109-112, Nov. - Dec. 2010.

[39] T. Gibson, A. Kulkarni, K. Kleese-Van Dam, and T. Critchlow, “The feasibility of moving PMU data in the future power grid,” in *Proceedings of Cigré Conference on Power Systems*, Halifax, Sep. 6-8, 2011, Canada.

[40] K. Rogers, “Data mining of power system data,” Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2011.

[41] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” *ACM Computing Surveys (CSUR)*, vol. 31, no. 3, pp. 264-323, Sep. 1999.

[42] F. C. Schweppe and J. Wildes, “Power system static-state estimation, part I: Exact model,” *IEEE Transactions on Power Apparatus and Systems*, vol. 89, no. 1, pp. 120-125, 1970.

[43] A. Monticelli, *State Estimation in Electric Power Systems: A Generalized Approach,* Boston: Kluwer Academic Publishers, 1999.

[44] A. Debs, “Parameter estimation for power systems in the steady-state,” *IEEE Transactions on Power Systems*, vol.19, no. 6, Dec. 1974.

[45] G. L. Kusic and D. L. Garrison, “Measurement of transmission line parameters from SCADA data,” in *Proceedings of the IEEE PES Power Systems Conference and Exposition (PSCE)*, 2004, pp. 440-445.

[46] J. Zhu and A. Abur, “Identification of network parameter errors,” *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 586-592, May 2006.

[47] P. Zarco and A.G. Exposito, “Power system parameter estimation: A survey”, *IEEE Transactions on Power Systems*, vol. 15, no.1, Feb. 2000.

[48] K. Rogers, S. Dutta, T. J. Overbye, and J. Gronquist, “Estimation of transmission line parameters from historical data,” in *Proceedings of Hawaii International Conference on System Sciences (HICSS),* Jan. 2013.

[49] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, J. Wiley, 1996, ISBN 0-471-58699-4.

[50] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation.* New York: Marcel Dekker, 2004.

[51] H. K. Alfares and M. Nazeeruddin, “Electric load forecasting: Literature survey and classification of methods,” *International Journal of Systems Science*, vol. 33, no. 1, pp. 23-34, 2002.

[52] G. A. N. Mbamalu and M. E. El-Hawary, “Load forecasting via suboptimal seasonal autoregressive models and iteratively reweighted least squares estimation,” *IEEE Transactions on Power Systems*, vol. 8, pp. 343-348, 1992.

[53] A. K. Jain, R. P. W. Duin, and M. Jianchang, “Statistical pattern recognition: A review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 4-37, Jan. 2000.

[54] R. Podmore, “Identification of coherent generators for dynamic equivalents,” *IEEE Transactions on Power Apparatus and Systems*, vol. 97, no. 4, pp. 1344-1354, 1978.

[55] J. R. Winkelman, J. H. Chow, B. C. Bowler, B. Avramovic, and P. V. Kokotovic, “An analysis of interarea dynamics of multi-machine systems,” *IEEE Transactions on Power Apparatus and Systems*, vol. 100, no.2, pp. 754-763, Feb. 1981.

[56] B. Yang, V. Vittal, and G. T. Heydt, “Slow coherency based controlled islanding – A demonstration of the approach on the August 14, 2003 blackout scenario,” *IEEE Transactions on Power Systems*, vol. 21, no. 4, pp. 1840-1847, Nov. 2006.

[57] L. Wang, M. Klein, S. Yirga, and P. Kundur, “Dynamic reduction of large power systems for stability studies,” *IEEE Transactions on Power Systems*, vol. 12, no. 2, pp. 889-895, May 1997.

[58] H. A. Alsafih and R. Dunn, “Determination of coherent clusters in a multi-machine power system based on wide-area signal measurements,” in *Proceedings of IEEE Power and Energy Society General Meeting,* 2010.

[59] X. Tao, H. Renmu, W. Peng, and X. Dongjie, “Applications of data mining technique for power system transient stability prediction,” in *Proceedings of 2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies*, Hong Kong, Apr. 2004.

[60] Z. Yu, X. Zhou, and Z. Wu, “Transient stability boundary visualization for power system,” in *Proceedings of IEEE International Conference on Power System Technology*, Oct. 2006.

[61] V. Kurbatsky and N. Tomin, “Monitoring of expected operating conditions of electric power systems on the basis of the modern cluster methods,” in *Proceedings of IEEE 10th International Conference on Environment and Electrical Engineering (EEEIC)*, Rome, Italy, May 2011.

[62] Z. Yu, X. Zhou, and Z. Wu, “Fast transient stability assessment based on data mining for large-scale power system,” in *Proceedings of 2005 IEEE PES Transmission and Distribution Conference & Exhibition: Asia and Pacific*, Dalian, China, 2005.

[63] A. Chakrabortty, “Handling the data explosion in tomorrow’s power systems,” *IEEE Smart Grid Newsletter*, Sep. 2011.

[64] P. J. Labuschagne, “Automatic clustering with application to time dependent fault detection in chemical processes,” M. Eng. Thesis, University of Pretoria, Pretoria, 2008.

[65] G. Heydt and E. Gunther, “Post-measurement processing of electric power quality data,” *IEEE Transactions on Power Delivery*, vol. 11, no. 4, pp. 1853-1859, Oct. 1996.

[66] E. Y. Hamid and Z.-I. Kawasaki, “Wavelet-based data compression of power system disturbances using the minimum description length criterion,” *IEEE Transactions on Power Delivery*, vol. 17, no. 2, pp. 460-466, Apr. 2002.

[67] A. Moore, “Efficient memory based learning for robot control,” Ph.D. dissertation, Carnegie-Mellon University, 1991.

[68] Y. Hu, C. Li, J. Li, Y.-H. Han, X.-W. Li, W. Wang, H.-W. Li, L.-T. Wang, and X.-Q. Wen, “Test data compression based on clustered random access scan,” in *Proceedings of Asian Test Symposium,* 2006.

[69] V. Arya, J. Hazra, P. Kodeswaran, D. Seetharam, N. Banerjee, and S. Kalyanaraman, “CPS-Net: In-network aggregation for synchrophasor applications,” in *Proceedings of Third International Conference on Communication Systems and Networks (COMSNETS)*, 2011.

[70] D. Y. Wong, G. J. Rogers, B. Porretta, and P. Kundur, “Eigenvalue analysis of very large power systems,” *IEEE Transactions on Power Systems*, vol. 3, pp. 472-480, May 1988.

[71] J. F. Hauer, C. J. Demeure, and L. L Scharf, “Initial results in prony analysis of power system response signals,” *IEEE Transactions on Power Systems*, vol. 5, pp. 80-89, Feb. 1990.

[72] J. N. Wrubel and R. Hoffman, “The new energy management system at PSE&G,” *IEEE Computer Applications in Power*, Jul. 1988, pp. 12-15.

[73] "Technology Assessment of Power System Visualization," EPRI Technical report 1017795, Palo Alto, CA, 2009.

[74] C. Ware, *Information Visualization: Perception for Design*, 2nd ed. Boston, MA: Morgan Kaufmann, 2004.

[75] R. Klump, W. Wu, and G. Dooley, "Displaying aggregate data, interrelated quantities, and data trends in electric power systems," in *Proceedings of 36th Hawaii International Conference on System Sciences (HICSS)*, Waikaloa, HI, Jan. 2003.

[76] E. Tufte, *The Visual Display of Quantitative Information*, Cheshire, CT: Graphics Press, 1983.

[77] C. Ware, "Quantitative texton sequences for legible bivariate maps," *IEEE Transactions on Visualization and Computer Graphics,* vol. 15, pp. 1523-1529, Dec. 2009.

[78] J. D. Weber and T. J. Overbye, "Voltage contours for power system visualization," *IEEE Transactions on Power Systems*, vol. 15, no. 1, pp. 404-409, Feb. 2000.

[79] E. Tufte, *Beautiful Evidence*. Cheshire, CT: Graphics Press, 2006.

[80] T. J. Overbye, E. Rantanen, and S. Judd, "Electric power control center visualization using geographic data views," in *Proceedings of 2007 iREP Symposium- Bulk Power System Dynamics and Control - VII, Revitalizing Operational Reliability*, Charleston, SC, USA, Aug. 2007.

[81] G. J. Cokkinides, A. P. Sakis Meliopoulos, G. Stefopoulos, R. Alaileh, and A. Mohan, "Visualization and characterization of stability swings via GPS-synchronized data," in *Proceedings of 40th Hawaii International Conference on System Sciences (HICSS)*, Jan. 2007.

[82] Y. Zhang, L. Chen, Y. Ye, P. Markham, J. Bank, J. Dong, Z. Yuan, Z. Lin, and Y. Liu, "Visualization of wide area measurement information from the FNET system," *IEEE Power and Energy Society General Meeting*, Jul. 2011.

[83] Z. Zhong, C. Xu, B. Billian, L. Zhang, S.-J. S. Tsai, R.W. Conners, V. A. Centeno, A. G. Phadke, and Y. Liu, "Power system frequency monitoring network (FNET) implementation," *IEEE Transactions on Power Systems*, vol. 20, no. 4, Nov. 2005.

[84] J. E. Tate and T. J. Overbye, "Contouring for power systems using graphical processing unit," in *Proceedings of 41st Hawaii International Conference on System Sciences (HICSS)*, Waikoloa, HI, Jan. 2008.

[85] J. B. MacQueen, "Some methods for classification and analysis of multivariate

observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, pp. 281-297, 1967.

[86] A tutorial on clustering algorithms – K-means, [Online]. Available: http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html

[87] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, pp. 129-137, 1982.

[88] T. W. Liao, "Clustering of time series data – A survey," *Pattern Recognition*, vol. 38, pp. 1857-1874, 2005.

[89] L. J. Heyer, S. Kruglyak, and S. Yooseph, "Exploring expression data: Identification and analysis of coexpressed genes," *Genome Research*, vol. 9, pp. 1106-1115, Nov. 1999.

[90] QT (Quality Threshold) Clustering, Agilent Technologies Inc., 2005. [Online]. Available: http://www.chem.agilent.com/cag/bsp/products/gsgx/Downloads/pdf/qt_clustering.pdf

[91] A. Danalis, C. McCurdy, and J. S. Vetter, "Efficient quality threshold clustering for parallel architectures," in *Proceedings of IEEE 26th International Parallel and Distributed Processing Symposium*, 2012.

[92] O. Dan and H. Mocian, "Scalable web mining with newistic," in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, 2009.

[93] Algorithms for nearest-neighbor search, [Online]. Available: http://simsearch.yury.name/tutorial.html

[94] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509-517, Sep. 1975.

[95] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software*, vol. 3, no. 3, pp. 209-226, Sep. 1977.

[96] R. Weber, H.-J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity search methods in high dimensional spaces," in *Proceedings of 24th VLDB conference*, New York, 1998.

[97] R. Klump, R. E. Wilson, and K. E. Martin, "Visualizing real-time security threats using hybrid SCADA/PMU measurement displays," in *Proceedings of 38th Hawaii International Conference on System Sciences (HICSS)*, Poipu, HI, Jan. 2005.

[98] Sparkline theory and practice. [Online]. Available: http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0001OR

[99] IEEE PES Wind Plant Collector System Design WG, “Wind power plant collector system design considerations,” in *Proceedings of IEEE Power and Energy Society General Meeting*, Calgary, Canada, Jul. 2009.

[100] H. L. Willis, H. Tram, M. V. Enge, and L. Finley, “Optimization applications to power distribution,” *IEEE Computer Applications in Power*, Oct. 1995.

[101] T. Short, *Electric Power Distribution Handbook.* CRC Press, 2003.

[102] S. K. Goswami and S. K. Basu, “A new algorithm for the reconfiguration of distribution feeders for loss minimization,” *IEEE Transactions on Power Delivery*, vol. 7, pp. 1484-1490, 1992.

[103] Roy Maclean, “Electrical system design for the proposed one Gigawatt Beatrice Offshore Wind Farm,” M.S. Thesis, University of Strathclyde, Scotland, Sep. 2004.

[104] IEEE PES Wind Plant Collector System Design WG, “Wind power plant grounding, overvoltage protection, and insulation coordination,” in *Proceedings of IEEE Power and Energy Society General Meeting*, Calgary, Canada, Jul. 2009.

[105] IEEE PES Wind Plant Collector System Design WG, “Wind power plant substation and collector system redundancy, reliability, and economics,” in *Proceedings of IEEE Power and Energy Society General Meeting*, Calgary, Canada, Jul. 2009.

[106] IEEE PES Wind Plant Collector System Design WG, “Design and application of cables and overhead lines in wind power plants,” in *Proceedings of IEEE Transmission and Distribution Conference*, New Orleans, 2010.

[107] A. Hertz, O. Marcotte, A. Mdimagh, M. Carreau, and F. Welt, “Optimizing the design of a wind farm collection network,” *Information Systems and Operational Research* (*INFOR)*, vol. 50, no. 2, pp. 95-104, Apr. 2012. [Online]. Available: http://www.gerad.ca/~alainh/Eoliennes.pdf

[108] P. Fagerfjäll, “Optimizing wind farm layout–More bang for the buck using mixed integer linear programming,” M.S. Thesis, Chalmers University of Technology and Gothenburg University, 2010.

[109] C. Berzan, K. Veeramachaneni, J. McDermott, and U.-M. O’Reilly, “Algorithms for cable network design on large-scale wind farms,” Technical Report, MIT, 2011. [Online]. Available:
http://thirld.com/files/msrp_techreport.pdf

[110] S. Dutta and T. J. Overbye, "A quality-threshold based collector system cable layout design," in *Proceedings of IEEE Power and Energy Conference at Illinois (PECI)*, Urbana, IL, Feb. 25-26, 2011.

[111] E. Muljadi, C. P. Butterfield, A. Ellis, J. Mechenbier, J. Hochheimer, R. Young, N. Miller, R. Delmerico, R. Zavadil, and J. C. Smith, "Equivalencing the collector system of a large wind power plant," in *Proceedings of 2006 IEEE Power and Energy Society General Meeting*, Montreal, Canada, Jun. 2006.

[112] S. Dutta and T. J. Overbye, "Optimal wind farm collector system topology design considering total trenching length," in *IEEE Transactions on Sustainable Energy*, vol. 3, no. 3, pp. 339-348, 2012.

[113] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, pp. 1389-1401, 1957.

[114] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," in *Proceedings of the American Mathematical Society*, vol. 7, 1956.

[115] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, "The algorithms of Kruskal and Prim," in *Introduction to Algorithms*, 3rd ed. MIT Press, Section 23.2: pp. 631-638, 2009.

[116] D. Cheriton and R. E. Tarjan, "Finding minimum spanning trees," *SIAM Journal on Computing*, vol. 5, pp. 724-741, Dec. 1976.

[117] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269-271, 1959.

[118] A. Ivanov and A. Tuzhilin, *Minimal Networks: The Steiner Problem and Its Generalizations,* CRC Press, 1994.

[119] D. Hochbaum, *Approximation Algorithms for NP-Hard Problems*, PWS Publishing, 1997.

[120] A. Steger, The Steiner tree problem, [Online]. Available: http://www.ti.inf.ethz.ch/ew/courses/ApproxAlgs0304/problem_sets/approx-steiner.pdf, Apr. 2004.

[121] GeoSteiner – Software for computing Steiner trees. [Online]. Available: http://www.diku.dk/hjemmesider/ansatte/martinz/geosteiner/

[122] D. M. Warme, P. Winter, and M. Zachariasen, "Exact algorithms for plane Steiner tree problems: A computational study," Technical report, Department of Computer Science, University of Copenhagen, Denmark, 1998.

[123] D. M. Warme, “Spanning trees in hypergraphs with applications to Steiner trees,” Ph.D. Thesis, Computer Science Dept., The University of Virginia, 1998.

[124] P. Winter and M. Zachariasen, “Euclidean Steiner minimum trees: An improved exact algorithm,” *Networks*, vol. 30, pp. 149-166, 1997.

[125] M. Zachariasen, “Rectilinear full Steiner tree generation,” *Networks*, vol. 33, pp. 125-143, 1999.

[126] M. Zachariasen, “Algorithms for plane Steiner tree problems,” Ph.D. dissertation, Department of Computer Science, University of Copenhagen, 1998.

[127] E. N. Gilbert and H. O. Pollak, “Steiner minimal trees,” *SIAM Journal on Applied Mathematics*, vol. 16, 1968.

[128] D. Z. Du and F. K. Hwang, “The Steiner ratio conjecture of Gilbert and Pollak is true,” in *Proceedings of National Academy of Sciences*, vol. 87, pp. 9464-9466, Dec. 1990. [Online]. Available:
http://www.pnas.org/content/87/23/9464.full.pdf

[129] D. Dreyer and M. Overton, “Two heuristics for the Euclidean Steiner tree problem,” *Journal of Global Optimization*, vol. 13, pp. 95-106, 1998.

[130] M. Clayton, “How enormous batteries could safeguard the power grid,” The Christian Science Monitor, Mar. 22, 2009. [Online]. Available:
http://www.csmonitor.com/Innovation/Responsible-Tech/2009/0322/how-enormous-batteries-could-safeguard-the-power-grid

[131] T. Bannow, “Minnesota tests nation’s first wind-to-battery storage,” Mar. 31, 2009. [Online]. Available:
http://www.mndaily.com/2009/03/31/minnesota-tests-nation%E2%80%99s-first-wind-battery-storage

[132] M. LaMonica, “Saving wind power for later,” Feb. 10, 2006. [Online]. Available:
http://news.cnet.com/Saving-wind-power-for-later/2100-11392_3-6170659.html

[133] Wind power and electricity markets, Utility Wind Integration Group, 2011. [Online]. Available:
http://www.uwig.org/windinmarketstableOct2011.pdf.

[134] R. Gold, “Natural gas tilts at windmills in power feud,” *The Wall Street Journal*, Mar. 2, 2010. [Online]. Available:
http://online.wsj.com.

[135] J. G.-González, R. M. Ruiz de la Muela, L. M. Santos, and A. M. González, “Stochastic joint optimization of wind generation and pumped-storage units in an electricity market,” *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 460-468, May 2008.

[136] M. Korpaas, A. T. Holen, and R. Hildrum, “Operation and sizing of energy storage for wind power plants in a market system,” *Electrical Power and Energy Systems*, vol. 25, pp. 599-606, 2003.

[137] M. T. Vespucci, F. Maggioni, M. Bertocchi, and M. Innorta, “A stochastic model for the daily coordination of pumped storage hydro plants and wind power plants,” *Annual Operations Research*, Jul. 2010.

[138] S. Dutta and T. J. Overbye, “Optimal storage coordination for minimal wind generation schedule deviation,” in *Proceedings of IEEE North American Power Symposium (NAPS)*, Arlington, TX, Sep. 26-28, 2010.

[139] S. Dutta and T. Overbye, “Optimal storage scheduling for minimizing schedule deviations considering variability of generated wind power,” in *Proceedings of IEEE Power Systems Conference and Exposition*, Phoenix, AZ, Mar. 2011.

[140] S. M. Schoenung, J. M. Eyer, J. J. Iannucci, and S.A. Horgan, “Energy storage for a competitive power market,” Annual Reviews, *Energy Environment*, 1996.

[141] W. B. Powell, *Approximate Dynamic Programming.* John Wiley & Sons Inc., 2010.

[142] A. F. Veinott, “Optimal policy for a multi-product, dynamic, non-stationary inventory problem,” *Management Science*, vol. 12, pp. 206-222, 1965.

[143] E. Ignall and A. F. Venott, “Optimality of myopic inventory policies for several substitute products,” *Managment Science*, vol. 15, pp. 284-304, 1969.

[144] Y. Makarov, P. Du, M. C. W. Kintner-Meyer, C. Jin, and H. F. Illian, “Optimal size of energy storage to accommodate high penetration of renewable resources in WECC system,” *IEEE Explore*, 2010. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05434768

[145] P. Pinson, G. Papaefthymiou, B. Klockl, and J. Verboomen, “Dynamic sizing of energy storage for hedging wind power forecast uncertainty,” in *Proceedings of IEEE Power and Energy Society General Meeting*, Calgary, Canada, 2009.

[146] Q. Li, S. S. Choi, Y. Yuan, and D. L. Yao, “On the determination of battery energy storage capacity and short-term power dispatch of a wind farm,” *IEEE Transactions on Sustainable Energy*, vol. 2, no. 2, pp. 148-158, Apr. 2011.

[147] Y. Makarov, Z. Huang, P. V. Etingov, J. Ma, R. T. Guttromson, K. Subbarao, and B. B. Chakrabarti, “Integration of wind generation and load forecast uncertainties into power grid operations,” in *Proceedings of IEEE Transmission and Distribution Conference and Exposition*, Apr. 2010.

[148] S. Dutta and R. Sharma, “Optimal storage sizing for integrating wind and load forecast uncertainties,” in *Proceedings of 2012 IEEE Innovative Smart Grid Technologies (ISGT),* Washington, D. C., Jan. 16-20, 2012.

[149] R. Sharma and S. Dutta, “Optimal storage sizing for integrating wind and load forecast uncertainties,” Patent (pending), 2012.

[150] S. Sen and J. L. Higle, “An introductory tutorial on stochastic linear programming models,” *Institute for* Operations *Research and the Management Sciences*, *Interfaces*, vol. 29, no. 2, pp. 33-61, Mar.-Apr. 1999. [Online]. Available: http://www.sie.arizona.edu/faculty/higle/images/pdf/Sen%26Higle99.pdf

[151] C. Singh, X. Luo and H. Kim, “Power system adequacy and security calculations using Monte Carlo simulation incorporating intelligent system methodology,” in *Proceedings of 9th International Conference on Probabilistic Methods applied to Power Systems*, KTH, Stockholm, Sweden, Jun. 2006.

[152] Y. Makarov, Z. Huang, P. V. Etingov, J. Ma, R. T. Guttromson, K. Subbarao, and B. B. Chakrabarti, “Integration of wind generation and load forecast uncertainties into power grid operations,” in *Proceedings of IEEE Transmission and Distribution Conference and Exposition*, Apr. 2010.

[153] Y. Makarov, Z. Huang, P. V. Etingov, J. Ma, R. T. Guttromson, K. Subbarao, and B. B. Chakrabarti, “Incorporating wind generation and load forecast uncertainties into power grid operations,” PNNL Wind Energy Management System EMS Integration Project, Jan. 2010. [Online]. Available:
http://www.ntis.gov/ordering.htm

[154] Load Forecast Uncertainty – Historical analysis and recommendation, MISO, Jun. 2011. [Online]. Available:
https://www.midwestiso.org/_layouts/MISO/ECM/Redirect.aspx?ID=97031 load forecast uncertainty.

[155] *Electric Energy Storage Technology Options: A White Paper Primer on Applications, Costs, and Benefits*, EPRI, Palo Alto, CA, pp. 4-27, 2010.

[156] Independent Electricity System Operator (IESO), Ontario. [Online]. Available: http://www.ieso.ca/imoweb/transInfo/demand.asp

[157] Building integration tutorial – Commercial buildings. [Online]. Available: http://www.apep.uci.edu/der/buildingintegration/2/BuildingTemplates/Office.aspx

[158] PJM – Operational analysis. [Online]. Available: http://www.pjm.com/markets-and-operations/ops-analysis.aspx

[159] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, "A review on the forecasting of wind speed and generated power," *Renewable and Sustainable Energy Reviews*, vol. 13, pp. 915-920, 2009.

[160] E. Ela and B. Kirby, "ERCOT event on February 26, 2008: Lessons learned," Technical report, NREL, 2008.

[161] I. Damousis, M. Alexiadis, J. Theocharis, and P. Dokopoulos, "A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation," *IEEE Transactions on Energy Conversion*, vol.19, no. 2, pp. 352-61, 2004.

[162] M. Alexiadis, P. Dokopoulos, H. Sahsamanoglou, and I. Manousaridis, "Short-term forecasting of wind speed and related electrical power," *Solar Energy*, vol. 63, no. 1, pp. 61-68, 1998.

[163] T. Barbounis and J. Theocharis, "A locally recurrent fuzzy neural network with application to the wind speed prediction using spatial correlation," *Neurocomputing*, vol. 70, pp. 1525-1542, 2007.

[164] M. Bilgili, B. Sahin, and A. Yasar, "Application of artificial neural networks for the wind speed prediction of target station using reference stations data," *Renewable Energy*, vol. 32, pp. 2350-2356, 2007.

[165] G. Riahy and M. Abedi, "Short term wind speed forecasting for wind turbine applications using linear prediction method," *Renewable Energy*, vol. 33, pp. 35-41, 2008.

[166] S. Dutta and T. J. Overbye, "Prediction of short term power output of wind farms based on least squares method," in *Proceedings of IEEE Power and Energy Society General Meeting*, Minneapolis, MN, Jul. 26-29, 2010.

[167] I. Damousis, M. Alexiadis, J. Theocharis, and P. Dokopoulos, "A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation," *IEEE Transactions on Energy Conversion*, vol.19, no. 2, pp. 352-61, 2004.

[168] D. Bechrakis and P. Sparis, "Correlation of wind speed between neighboring measuring stations," *IEEE Transactions on Energy Conversion,* vol. 19, no. 2, pp. 400-406, 2004.

[169] M. Alexiadis, P. Dokopoulos, H. Sahsamanoglou, and I. Manousaridis, “Short-term forecasting of wind speed and related electrical power,” *Solar Energy*, vol. 63, no. 1, pp. 61-68, 1998.