Towards Loss Minimization in Power Distribution Systems using AI: the WatDist Algorithm

by

Derrick Earl Bouchard

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical Engineering

Waterloo, Ontario, Canada, 1996

0-612-21333-1

The University of Waterloo requires the signature of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

As electric power distribution systems continue to grow in size and complexity, Distribution Automation schemes become more attractive. One of the features that is desirable in an automated system is feeder reconfiguration for loss reduction. Reducing losses can result in substantial savings for a utility. Other benefits from loss reduction include released system capacity, and possible deferral or elimination of capital expenditures for system improvements and expansion. There is also improved voltage regulation as a result of reduced feeder voltage drop.

System reconfiguration is accomplished using existing switches in the network. For a given system, there will be a switching pattern that minimizes system losses. However, if there are N switches in a network, there are 2^N possible switching combinations, and the challenge of finding the optimum switching pattern to minimize losses becomes formidable as the number of switches increases.

In this thesis, a novel algorithm, WatDist, is introduced to solve the network reconfiguration for loss minimization problem. The proposed technique is based on artificial intelligence techniques applied to constraint satisfaction optimization problems. A critical review of earlier methods is presented to highlight their shortcomings. Computer simulations using WatDist demonstrate its advantages, including a high success rate in finding the global optimum, the final solution being independent of the initial configuration, and assurance that any solution offered will have a radial configuration with all loads connected and no constraint violations. A cost/benefit analysis demonstrates the significant contribution of the algorithm to distribution system analysis and operation.

Acknowledgements

I would like to thank Dr. Magdy Salama, one of my two thesis supervisors, for his invaluable help in seeing this thesis through to its successful completion. Dr. Salama has always shown professionalism, dedication and enthusiasm, and I am grateful for having had the opportunity to study and to work under his supervision.

I would also like to thank Dr. Aziz Chikhani, my other thesis supervisor, as well. I am extremely grateful for his confidence in my abilities, and his steady encouragement over the past few years. His firm (but always kind) nudging helped to keep this thesis on schedule. I feel very fortunate to have worked with him.

Scott Knight proved to be an excellent teacher and gifted software engineer who helped me with some of the more difficult concepts of C++. His willingness to take the time to thoroughly answer my numerous questions was greatly appreciated.

To my children, Gabrielle, Mallory and Craig, thank you for your patience on the many evenings and week-ends when we should have been together, but I was busy working on my thesis. Thank you, too, for pulling me away from my studies now and again to go for a bike ride, or to go to the park, or to play road hockey. Those breaks were surely needed, and I wish there had been more of them.

Finally, to my wife, Fioretta, my biggest thanks, although thanks is perhaps not strong enough. Thank you for having endured my hectic schedule of the past three years. Your support and encouragement have meant a great deal to me. I could not have done this thesis without you.

To my family, Fioretta, Gabrielle, Mallory and Craig,

with love

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Over the past few decades, distribution systems have received considerably less attention than have transmission and generating systems [1]. This is due mainly to the fact that transmission and generating systems are usually very capital intensive, and inadequacies in either often lead to widespread catastrophic consequences. Consequently, more effort has gone into ensuring the adequacy of this part of the power system. Distribution systems are relatively cheap, and outages have a very localized effect. However, while relatively inexpensive, large sums of money are spent collectively on such systems.

A radial distribution system consists of a set of series components including lines, cables, disconnects, busbars and transformers between a utility and its customers. A customer connected to any load point in a distribution system requires all of the components between the point of connection and the supply point to be operating.

Many distribution systems are designed and constructed as groups of single radial feeders. Some systems are constructed as meshed systems, but operated as single radial feeder systems by using normally-open switches in the mesh. These normally-open points reduce the amount of equipment exposed to a fault on any single feeder circuit. They also ensure that, in the event of a fault or during scheduled maintenance periods, the normally-open point can be closed and another opened using switches to minimize the total load disconnected from the system.

These switches can also be used to transfer loads among feeders to meet new load requirements, to make better use of system capacity, and to minimize I^2R losses in the distribution lines. For a given system, there will be a switching pattern that minimizes system losses. If there are N switches in a system, there are 2^N possible switching combinations. For modern distribution systems with thousands of load buses and hundreds or even thousands of switches, the challenge of finding the optimum switching pattern to minimize losses is formidable.

1.2 Thesis Scope and Objectives

The objectives for this thesis were the following:

a. to develop suitable data structures and algorithms for system reconfiguration;

b. to employ intelligent search techniques to develop a new algorithm for finding the optimum configuration to minimize distribution system losses of a test system; and,

c. to apply the algorithm to an actual distribution system, and to measure its performance in reducing energy losses over a period of one year.

The sequence of events that led to the production of this thesis was as follows:

a. a thorough investigation of previous work in the area was carried out, and the associated issues of system protection, load flow techniques and system planning were also examined;

b. a study of artificial intelligence techniques applied to constraint satisfaction optimization problems was made;

c. reconfiguration algorithms, as well as a new optimization algorithm for loss minimization through system reconfiguration, were developed; and,

d. two test systems were selected, and experimentation carried out to prove the validity of the new technique.

1.3 Thesis Organization

The material in this thesis is organized as follows:

- Chapter 2 provides the rationale for loss minimization, examines several techniques for reducing distribution system losses, and introduces loss minimization through system reconfiguration;

- Chapter 3 examines the difficulties in implementing loss minimization through reconfiguration;
- Chapter 4 reviews the techniques proposed by earlier researchers, beginning with the algorithm of Merlin and Back [2] in 1975;
- Chapter 5 explores constraint satisfaction optimization problems and defines several terms;
- Chapter 6 discusses the data structures and algorithms developed for this thesis for system reconfiguration;
- Chapter 7 explains the optimization algorithm, WatDist, developed for this thesis, and examines the results obtained when the algorithm is applied to a test system;
- Chapter 8 presents the results obtained when the algorithm is applied to an actual distribution system operating over the period of one year; and,
- Chapter 9 provides concluding remarks, and recommendations for future work.

Chapter 2

The Need for Distribution System Reconfiguration

2.1 Introduction

Power distribution systems provide the final link between a utility and its customers. These systems face demands for ever-increasing power requirements, high reliability, more automation, and greater control complexity. At the same time, utilities face a scarcity of available land in urban areas, ecological considerations, the undesirability of rate increases, and the necessity to minimize investments and operating expenses. Planners must consider all of these factors, and, simultaneously, attempt to minimize the cost of substations, feeders and laterals, as well as the cost of losses [3].

As the demand for electrical power continues to grow, so, too, does the public's awareness of environmental issues and energy conservation. Utilities must maximize their use of existing equipment and optimize existing system capabilities as a means of generating more capacity without construction of new facilities. It has been estimated that 5% to 13% of total system generation is wasted in the form of distribution system losses [4], and therefore the reduction of these losses is important. In [5], Grainger and Kendrew examined the distribution of losses in a distribution network. Their results are summarized in Table 2.1.

From Table 2.1, it can be seen that the biggest contributor to losses are the distribution transformers, accounting for 55.1% of all losses (with no-load losses three times that of losses under load), and representing 2.14% of the utility's revenue. The next largest contributor are the primary feeders, which account for 19.0% of all losses, and which represent 0.74% of the utility's revenues. Thus, reduction of losses represents an effective means of cutting the cost of power to a utility.

As well, there are other economic benefits resulting from loss minimization, including [3]:

- released generation capacity;
- released transmission capacity;
- released distribution substation capacity;
- reduced energy (copper) losses;
- reduced feeder voltage drop and consequently improved voltage regulation; and,
- deferral/elimination of capital expenditures for system improvements/expansion.

Segment	% of revenue		% of losses
Substation losses:		0.66	17.1
Primary feeders			
3ϕ	0.70		
1ϕ and 2ϕ	0.04		
Total feeder losses		0.74	19.0
Distribution transformers			
No-load loss	1.86		
Loaded loss	0.28		
Total transformer losses		2.14	55.1
Secondary feeder losses		0.13	3.4
Other losses		0.21	5.4
	TOTAL	3.88	100.0

Table 2.1. Summary of allocation of energy losses in a distribution system (from [5]).

2.2 Methods of reducing distribution system losses

Several techniques can be employed to reduce distribution system losses, and these will be examined in detail. These techniques are as follows [3]:

a. introduction of higher voltage levels;

b. reconductoring;

c. conservation voltage reduction;

d. installation of capacitors; and,

e. system reconfiguration.

Reference [6] provides benefit/cost ratios for various methods of loss reduction in distribution systems, and these are summarized in Table 2.2. It can be seen that the most expensive methods (in terms of benefit/cost ratio) are reconductoring and the introduction of higher voltage levels. System reconfiguration provides one of the more economical options.

Method of Loss Reduction	*Benefit/cost ratio*
Introduction of higher voltage levels	1.5 to 3
Reconductoring	0.6 to 7
Installation of capacitors	2 to 8
Reconfiguration	0 to 13

Table 2.2. Benefit/cost ratios for Various Methods of Loss Reduction (from [6]).

2.2.1 Introduction of higher voltage levels

The primary feeder voltage level is the most important factor affecting the system design, cost and operation. Operational and design aspects affected by the voltage level include feeder length and loading, the number and rating of distribution substations, system maintenance practices and type of pole-line design and construction [3]. In general, for a given percent voltage drop, the feeder length and loading are direct functions of the feeder voltage level, and may be expressed by a relationship known as the voltage-square rule. For example, if the feeder voltage is doubled, for the same voltage drop, the feeder can supply the same power four times the distance. The relationship is:

$$\textit{Voltage-square factor} = \left(\frac{V_{L\text{-}N,new}}{V_{L\text{-}N,old}}\right)^2 \tag{2.1}$$

Introduction of higher voltage levels involves extensive modification to existing networks, as well as to associated switchgear, transformers and substation equipment, and hence entails considerable cost to a utility that may or may not be economically feasible. For example, Toronto Hydro recently upgraded its distribution system to 13.8 kV, with projected savings of $620 million over 25 years [7]. On the other hand, as a counter-example, with shrinking margins as a result of its regulatory agency's refusal to allow rate increases, Ottawa Hydro recently decided to suspend its upgrade of its 4.16 kV system to 13.8 kV as too costly [8].

2.2.2 Reconductoring

The resistance, R, of a conductor of length, l, resistivity, ρ and cross-sectional area, A, is:

$$R = \rho\frac{l}{A} \tag{2.2}$$

It is apparent that line resistance can be decreased by using a conductor with a lower resistivity or by increasing the cross-sectional area of the conductor. The costs associated with reconductoring may be prohibitive, and probably are only justified in networks that are operating near their design capacity.

2.2.3 Conservation Voltage Reduction

Conservation voltage reduction (CVR) is a method by which utilities lower substation transformer voltages by a few percent to reduce peak demand. Although there have been several studies of CVR, it is not clear that CVR is of benefit to all utilities, as there is an associated loss of revenue to a utility when the peak demand is reduced.

Several American studies provide conflicting conclusions regarding CVR. In [9], De Steese *et al* note that there is a potential of a 0.765% reduction in energy consumption for each 1% reduction in average voltage for residential customers. While Snohomish County PUD in the state of Washington found that energy savings were achieved from the implementation of CVR on distribution circuits, these savings were highly variable from circuit to circuit and were difficult to measure accurately [10]. However, reducing the distribution primary voltage did not result in lower real and reactive power demand on the distribution circuits tested. In a study by Detroit Edison, the opposite results were found, in that reducing the distribution primary voltage did result in lower real and reactive power demands [11]. However, this study concluded with the following comments:

> "Although energy is saved when voltages are lowered, voltage reduction does not appear to be a practical, cost effective, and viable method of conserving energy. The cost of energy which customers would save would be offset by additional rate increases, additional operation and maintenance expenses, and it is likely that the quality of service for some customers would become a problem."

The performance and the operating life of equipment may be affected when the voltage at the terminals of the equipment deviates from its nameplate value. The effect may be minor or serious, depending on the deviation from the nameplate voltage rating and the characteristics of the equipment. In Canada, standards for voltage levels have been established [12], and utilities are bound by these standards. Practicing CVR may lead to excessive voltage drop along long, heavily loaded feeders, and thus voltages have to be carefully monitored when CVR is used.

For induction motors, Table 8 of reference [13] provides an indication of the general effects of voltage variations on induction motors. For a 10% voltage decrease, the starting and maximum running torque decrease 19%, while the slip increases 20-30%. There is a corresponding increase of 5-10% in the full-load current, corresponding to a temperature rise of 10-15%. Thus, in the case of predominantly induction motor loads, there will be an increase in line current, and hence an increase in line losses.

For resistance heating devices, the heat output varies approximately as the square of the impressed voltage. Thus, a 10% drop in voltage will cause a drop of approximately 19% in heat output. To produce the same amount of heating would then require the resistance heater to operate for longer periods, and thus energy conservation would not be achieved.

In summary, CVR is not beneficial when the loads are predominantly induction motors or resistance heating devices. As well, the reduction in voltage may lead to excessive voltage drops to some customers. The success of CVR is very system-dependent, and determined by such factors as predominant customer types, feeder lengths and loading.

2.2.4 Installation of Capacitors

The fundamental purpose of capacitors is to regulate the voltage and reactive power flows at the point where they are installed [3]. Shunt capacitors do not affect the current or power factor beyond their point of application, and generation of reactive power at a power plant and its supply to a load located at a far distance is not economically feasible. In Ontario, customers whose power factor is less than 90% pay for the excessive reactive power demanded, and it is in their best interests to carry out power factor correction [14]. The result is that many municipal utilities operate at near unity power factor, and hence installation of capacitors is often not warranted.

2.2.5 System Reconfiguration

Loss minimization through system reconfiguration can provide substantial savings to a utility. By applying loss minimization techniques, a distribution automation project by the Pennsylvania Power and Light Company projected potential savings of over $100k by reducing losses by 14.6% in one year for a 230 MW distribution network [15].

A radial distribution system consists of a set of series components including lines, cables, disconnects, busbars and transformers between a utility and its customers. Most distribution systems are designed and constructed as single radial feeder systems. Some systems are constructed as meshed systems, but operated as single radial feeder systems by using normally-open switches in the mesh. These normally open points reduce the amount of equipment exposed to a fault on any single feeder circuit and ensure that, in the event of a fault or during scheduled maintenance periods, the normally open point can be closed and another opened in order to minimize the total load disconnected from the system.

A typical system one-line diagram is shown in Figure 2.1. Sectionalizing switches along a feeder and interfeeder tie switches are used to maintain a radial structure. For every switch

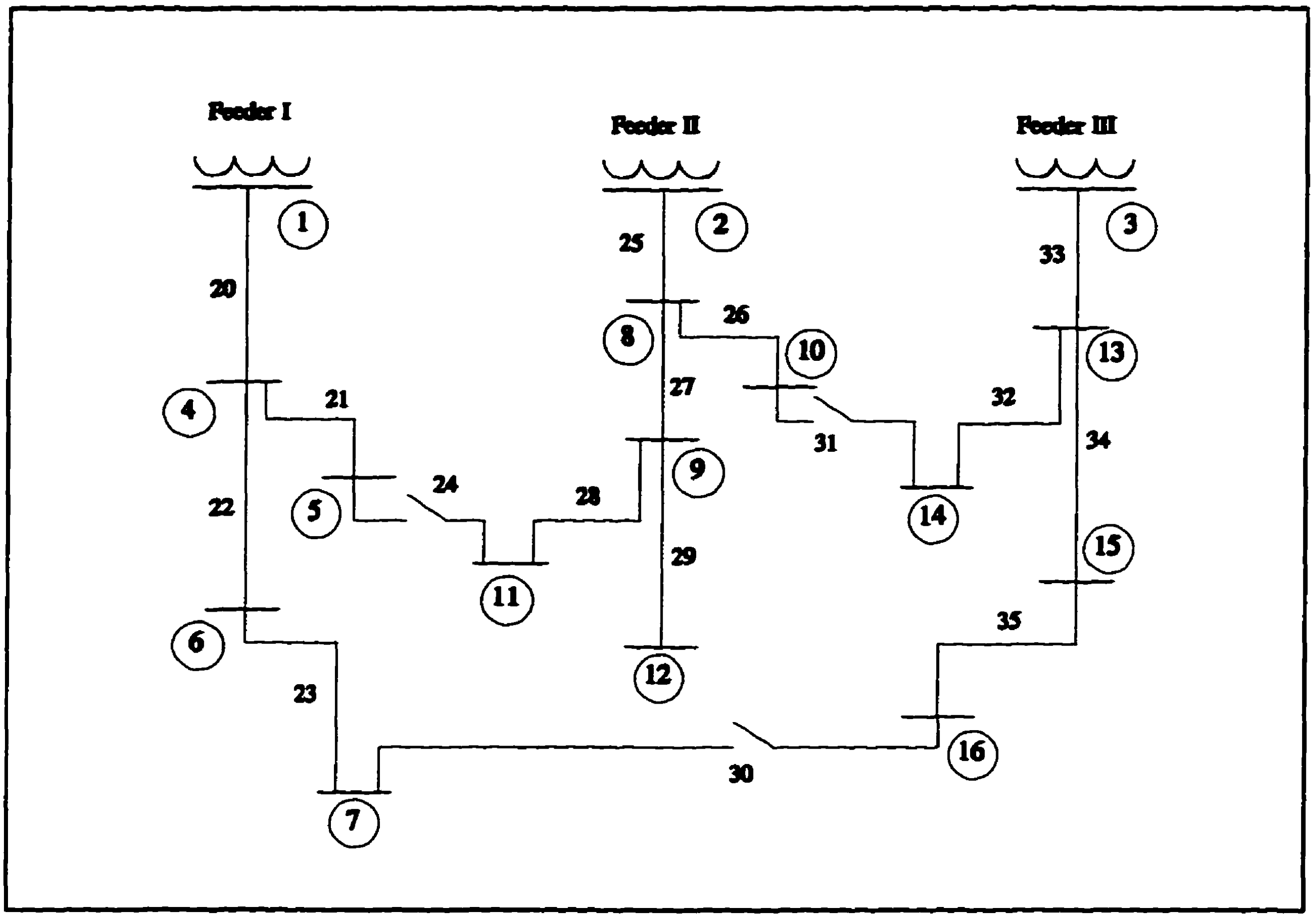


Figure 2.1. A distribution system with three feeders and sixteen sectionalizing switches, three of which are open.

closed, another is opened. The greater the number of sectionalizing (or tie) switches, the greater are the possibilities for reconfiguration. To have minimum losses, a network must be equipped with remotely-operated switches, preferably in every line section of the network to provide the maximum degree of flexibility. The primary benefit of a radial structure is that it simplifies fault-detection, allowing a utility to quickly dispatch repair crews where needed, and to isolate faulted sections so that service to other sections can be restored. Radial networks have lower short circuit currents and simpler switching and protective equipment than meshed networks [16]. The tradeoff is that these networks have lower overall reliability.

Power utilities are turning increasingly to computers and telecommunications to monitor and control power systems. Considering the size and complexity of a modern utility, a human operator cannot hope to control a power system without automated assistance. SCADA (supervisory control and data acquisition) systems generate large amounts of data that cannot be rapidly assessed by a human operator, and thus there is a desire to automate human decision-making tasks. SCADA systems allow the remote control of distribution system switches to improve system reliability through fault isolation and service restoration. These switches can also be used to transfer loads among feeders in a distribution system to meet new load requirements, and to make better use of system capacity.

Thus, loss minimization through system reconfiguration is an attractive option, as it uses existing equipment to reduce losses. Even those utilities that rely on manual switches can benefit from reconfiguration, although on a much-reduced basis, perhaps only carrying out reconfiguration once or twice per year.

Reference [17] describes the results of a distribution automation study conducted by Pacific Gas & Electric. Substation and feeder automation were identified as cost-effective areas benefiting from distribution automation. For feeders, automation included data acquisition from sectionalizers, line switches and fault indicators, as well as supervisory control of these devices for feeder reconfiguration and fault isolation. Economic benefits associated with feeder automation included reductions in capital expenditures due to deferment of additional feeders through more effective use of existing feeders, reductions in operations and maintenance costs, increased revenues as a result of loss reduction through feeder reconfiguration and faster service restoration following a fault.

The reconfiguration of an electrical distribution system to reduce losses also has a natural tendency to balance loading among circuits, putting the system in a better position to respond to emergency load transfers [18]. This is especially important when transformers and feeders are loaded close to their limits due to rapid load growth or delays in the construction of new substations and feeders [19].

Perhaps the most comprehensive study into system automation and reconfiguration has been that of the Athens Area Control Experiment (AACE), carried out by the Athens Utility Board (AUB), and thoroughly described in Reference [20]. The system automated during AACE was made up of 12 feeders, 35 load-break switches, 12 power reclosers, 5 voltage regulator banks, 29 capacitor banks and 21 fault detectors. A load-break switch was a three-phase, group-operated switch with an electric motor operator costing approximately $11k, whose purpose was to isolate faulted lines and transfer loads while a circuit was energized. These switches were also used as tie switches between feeders. Power reclosers are similar to load-break switches, except their purpose is to clear temporary faults.

AUB saw an improvement in conventional system reliability indices through automation. It was also discovered that there were significant intangible benefits and tangible cost benefits that were not measured through reliability indices. Automated fault detection and remote control of switches and breakers lead to (1) significant reduction in the time required to detect and locate faults; (2) faster isolation of faulted sections; and, (3) faster load restoration above and below a faulted zone. Additionally, outages were prevented, or the outage area was reduced (and hence the number of customers affected), costs were saved by automating tasks that had previously been performed manually, equipment problems were detected prior to catastrophic failure, and system safety was improved. It was also possible to detect such things as abnormal load conditions and insulation failure.

Reliability studies indicated that automation was fully justified. The time of customer interruption of power was highly sensitive to the switching time required to sectionalize a feeder and restore service after a fault, and thus remotely-operated switches in place of manual switches increased reliability. The switches installed by AUB were primarily to increase reliability rather than to optimize feeder loading. It was found that hourly load transfers were sufficient.

Part of the AACE studies included determining if capacity utilization could be improved as a result of automation. It was thought that remote load transfers between feeders would allow surplus equipment capacity at one location to made available to other locations quickly and easily. However, the results obtained were less than anticipated by AUB for a variety of reasons, including:

1. coincident peak loads on feeders;
2. "telescoped" feeder conductor diameters that decreased in discrete steps with distance from the substation; and,
3. reconfiguration increased the impedance between source and load. This served to increase losses, and at the same time increase the voltage drop along the feeder. The drop in voltage resulted in a drop in customer loads.

As a result of AACE, AUB recommended more automated switches to allow smaller load transfers. Most feeders had switches to sectionalize feeders into three or four zones, which allowed only relatively large load transfers.

In [21], Aoki *et al* review the principal functions of SCADA systems in distribution systems, and note that algorithms for "automatic load transfer for secure or economical operation require more research in order to put them into practical use as "no efficient algorithm for the sectionalizing-switch operation has been established, as it is a combinatorial optimization problem." At the same time, however, these algorithms are indispensable in distribution automation, and must be developed.

Thus, distribution feeder reconfiguration for loss minimization is important for utilities. However, few utilities have implemented reconfiguration [22], and their rationale for not doing is examined in Chapter 3.

Chapter 3

Why *is* Reconfiguration a Difficult Problem?

3.1 Introduction

As noted in the previous chapter, few utilities have implemented reconfiguration to minimize system losses. The main objection to reconfiguration is that it computationally expensive, i.e., as system size grows, so does computation time. If there are N switches in a distribution network, there are 2^N possible configurations. For modern urban distribution systems, the number of distribution transformers may reach two to three thousand, and each transformer may be supplied by four or five different feeders and substations [23]. Such systems are very complex, very difficult to monitor, and difficult to control optimally in real-time. Losses associated with each configuration must be calculated, and this requires a load flow. The problem is compounded by the desire to maintain the radial configuration of the distribution system and by operational constraints, i.e., ensuring feeders and transformers are not overloaded and ensuring voltage drop limitations are not exceeded. As well, there is a need for efficient data structures and algorithms that will permit reconfiguration in real-time.

3.2 The Reconfiguration Problem

3.2.1 Mathematical Representation of the Reconfiguration Problem

The reconfiguration problem can be expressed as follows:

Minimize

$$\sum_{i=1}^{n} I_i^2 R_i x_i \tag{3.1}$$

subject to

$$\sum_{i=1}^{n} S_{ij} = D_j + losses \tag{3.2}$$

$$S_{ij} \leq S_{ij,max} \tag{3.3}$$

$$\Delta V_{ij} \leq \Delta V_{ij,max} \tag{3.4}$$

$$\sum_{for\,all\,f_t} S_{f_t} = S_{f_t max} \tag{3.5}$$

where the variables are defined as follows:

R_i the resistance of line section i

I_i the current in line section i

x_i the state value of switch i, where

$$x_i = \begin{cases} 1, & \textit{if the switch is closed} \\ 0, & \textit{if the switch is open} \end{cases} \tag{3.6}$$

n number of buses

S_{ij} power flow along line section ij

D_j demand at bus j

ΔV_{ij} voltage drop across line section ij

$\Delta V_{ij,max}$ maximum allowable voltage drop across line section ij

S_{ft} power flow for feeder f_t

$S_{ft,max}$ maximum rated power flow for feeder f_t

f_t subset of feeders supplied by transformer t

In the above formulation of the reconfiguration problem, equation (3.1) represents the total losses of the distribution system. Equation (3.2) ensures that supply equals demand at every bus. Equation (3.3) ensures that a feeder is not overloaded (current, or thermal, limitation). Feeder voltage drop constraints are modeled by equation (3.4). Equation (3.5) ensures that transformer buses are not overloaded. As noted earlier, the system must also remain in a radial configuration.

Distribution losses are I^2R losses, and thus the problem is a nonlinear integer optimization problem, with a quadratic objective function, 0-1 type state variables, and linear constraint equations with state-dependent constraint formulae. The value of the objective function is determined from the power flow solution given settings of the control variables. At each iteration, a new power flow is required to determine a new system operating point. This issue will be addressed in detail shortly. The problem presents a heavy computational burden for even a moderately-sized distribution system.

3.2.2 Other Representations of the Problem

Other representations of the problem are possible. Roytelman *et al* [24] indicated that the problem could be formulated in various fashions, including:

1. minimizing active power loss:
2. minimizing power demand (losses + customer demands):
3. keeping system within constraints using minimum number of control actions.

Minimizing power demand is equivalent to determining the power injected at the substation buses. This representation is used in this thesis.

3.3 Power Flow Analysis

Power flow analysis is used to determine the steady-state powers and voltages at each bus in a distribution system, and is well described in many textbooks (for example, [25, 26]). Two of the more popular methods of power flow analysis are the Gauss-Seidel and Newton-Raphson techniques.

Although both the Gauss-Seidel and Newton-Raphson methods can be carried out very quickly on modern computers, the Gauss-Seidel method is the preferred method for distribution systems [26]. The method has the advantages of relative insensitivity to initial voltage estimates, small memory requirements and programming simplicity. For the Newton-Raphson method, the low X/R ratios of distribution systems often leads to ill-conditioned Jacobian matrices, with the result that the method fails to converge, or even diverges.

Reconfiguration for loss reduction typically involves evaluating many combinations of switching options to determine which option offers the lowest losses. Obviously, in a large system, even with a very fast computer, the time needed to complete a load flow to evaluate every option would be prohibitive, and is the main reason for not carrying out an exhaustive search of all switching combinations.

3.4 Dynamic Nature of Distribution Systems

Calculation of the losses for a configuration provides values for only one instant in time, based on current bus loads. However, distribution systems are very dynamic, and customers include industries, commercial centres and residential homes, all of which have changing load demands throughout the day, week and season of the year. A typical load profile over a twenty-four hour period for a residential load is shown in Figure 3.1. Thus, reconfiguration must be carried out on a regular basis (i.e., on-line and in real-time) as demand changes, further increasing the computational load.

3.5 Summary

Distribution system reconfiguration for loss minimization is a nonlinear optimization problem that presents an enormous computational burden for even systems of moderate size. Power flows must be carried out at each iteration to evaluate possible configurations, further adding to the computing time. Finally, the solution must be available in real-time if it is to be useful, due to the dynamic, time-varying nature of feeder loads.

In the next chapter, the work of previous researchers in solving the optimization problem is reviewed.

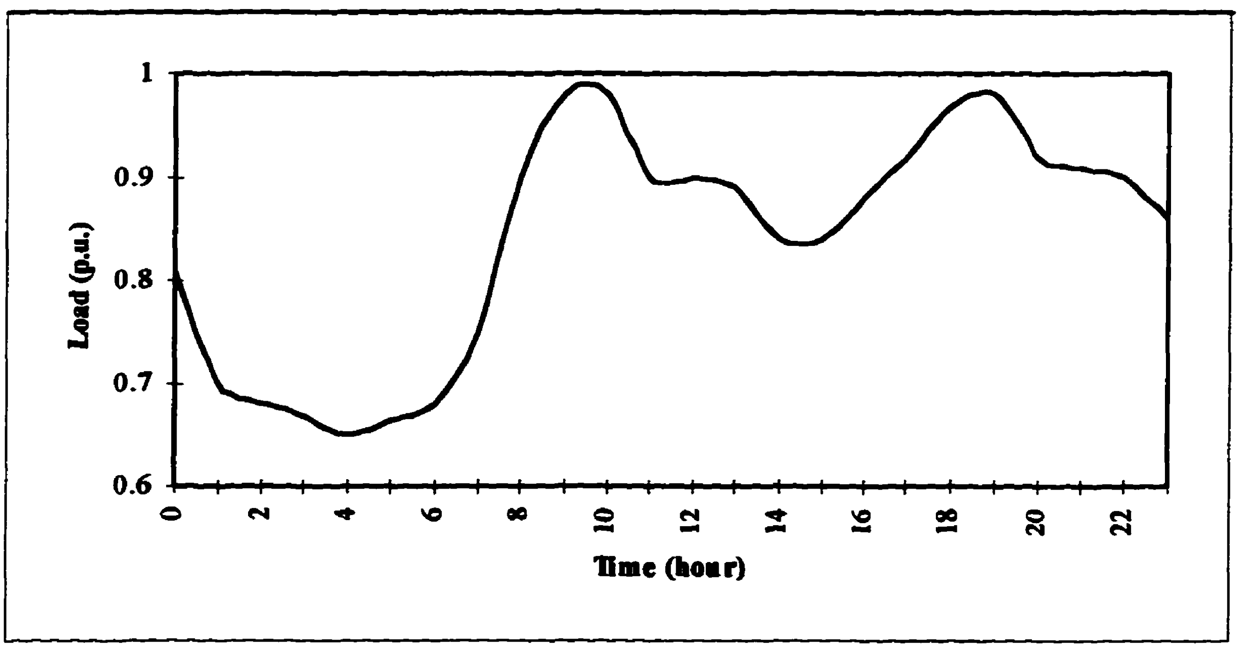


Figure 3.1. Load curve for a residential load for a winter week-day (based on data taken from Reference [27]).

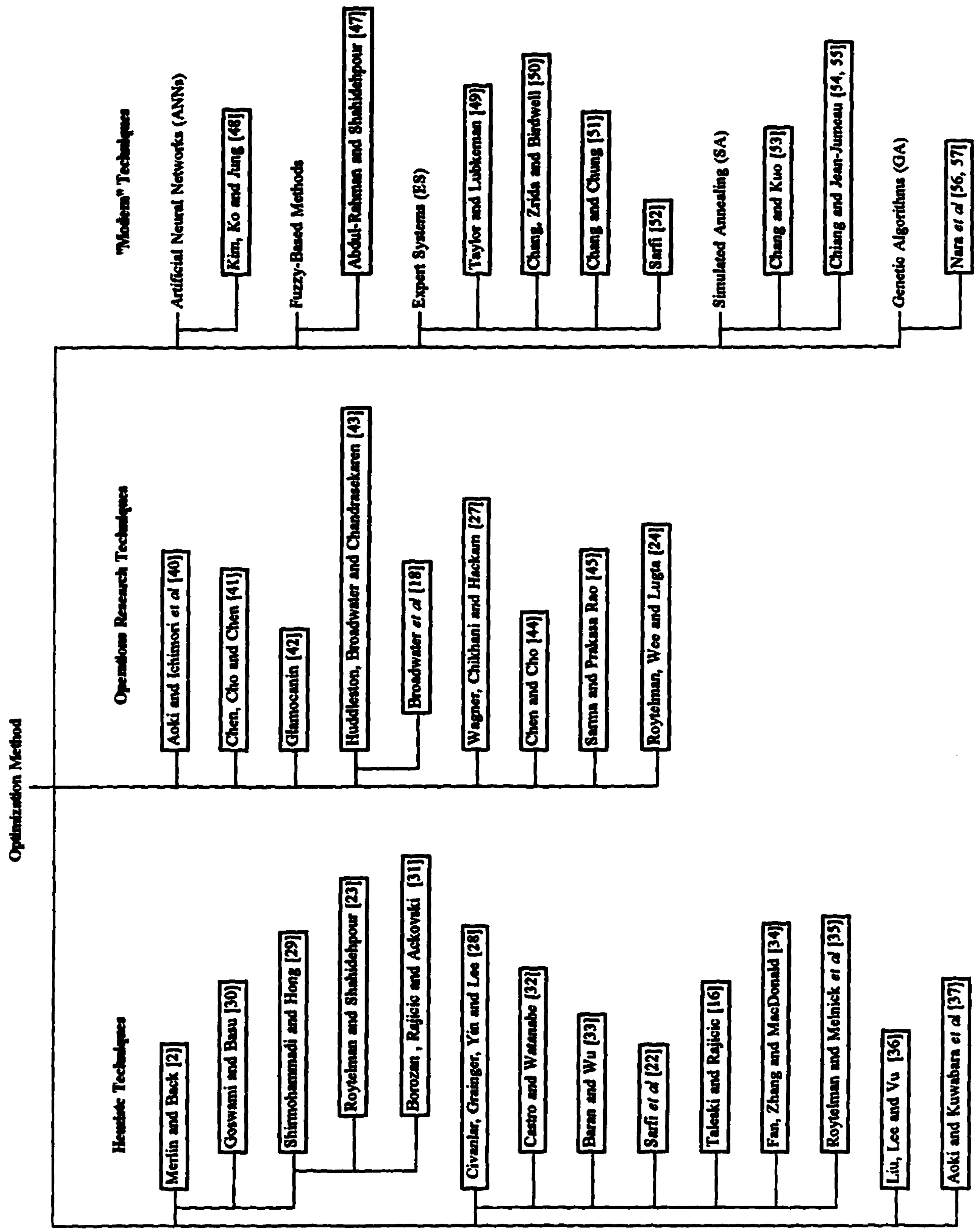
Chapter 4

Review of Recent Research in Reconfiguration

4.1 Introduction

In this chapter, recent research in reconfiguration is reviewed. Loss minimization by system reconfiguration continues to be a very active field of research. Beginning in 1975 with the work of Merlin and Back [2], a variety of techniques have since been proposed, including several algorithms that employ heuristics, methods based on classical operations research techniques, and algorithms that use "modern" techniques such as neural networks, expert systems and genetic algorithms. Figure 4.1 provides a list of the works reviewed in this chapter, divided broadly into the categories of (1) heuristic techniques; and, (2) operations

Figure 4.1. **Earlier Works into the Reconfiguration Problem**



research techniques; and, (3) "modern" techniques. The work of Civanlar, Grainger, Yin and Lee [28] is perhaps the most often cited reconfiguration algorithm, and will be examined in detail.

4.2 Algorithm Assessment Criteria

Optimization algorithms must select an alternative from among a very large set of possible solutions by using some form of numerical or nonnumerical computation to find a good (hopefully, the best) solution. To be successful, an algorithm must generate and examine all the alternative solutions, and not just a portion, i.e., it must be complete. Because of the large number of possible switching combinations in a distribution system, it is usually not feasible to examine each one. To date, no algorithm has been complete, and thus, no researchers can claim that their method finds the global optimal.

Most of the proposed algorithms ignore system voltage drop and thermal constraints that limit switching options and restrict loading. These operational restrictions are a crucial part of everyday operation for a utility, and an algorithm that ignores constraints will not be well-received.

There are three factors to consider in the design of an optimization algorithm:

a. *linear or nonlinear analysis:* Distribution system losses are nonlinear. For many algorithms, a nonlinear model leads to longer solution times, smaller problem capacity, or a greater likelihood of computational problems such as divergence. As will be seen, some researchers get around this problem by linearizing losses over a small operating region.

b. *problem size:* Distribution systems may have several thousand buses. For some optimization algorithms, this large number of buses will be too many, and an algorithm will be unsuccessful in finding a solution.

c. *radiality:* A radial configuration is difficult to enforce on many optimization techniques, as quite often the optimization algorithm senses (correctly) that splitting a load between two feeders will minimize losses. Regardless of which of two possible feeders to a load has the lowest I^2R losses, leaving a load connected to both feeders will lower the losses further.

To enforce radiality, a number of methods are used. First, the number of options considered by the algorithm can be limited to only radial configurations in a preoptimization of the "pruning" space (although this is difficult if the space is large). Second, loops found in the final configuration can be "radialized" in a postoptimization process. Third, the optimization algorithm can be modified to ensure it only seeks radial configurations.

4.3 Heuristic Techniques

To overcome the size limitation posed by modern distribution systems, or to reduce or eliminate the need to carry out power flows, many researchers turn to heuristics, or rules-of-thumb. The tradeoff becomes a question of solution quality versus computation time, i.e., finding the optimal solution in possibly an infinite amount of time, or finding a feasible suboptimal solution in a finite amount of time. There are several drawbacks to heuristic algorithms[29]:

1. the final network configuration often depends upon the initial configuration;
2. while losses may be reduced by employing heuristics, there is no guarantee that the final solution is optimal, or even near optimal; and,
3. even by employing heuristics, the computation time can still be quite large in a network of realistic size, which may contain thousands of branches and thousands of switches.

As well, most of the algorithms presented in this section ignore operational constraints, and thus they are of little interest to utilities. Most present-day distribution systems contain major components that operate close to their maximum load/capacity ratio, and thus it is crucial for algorithms to work around these limitations.

Even though heuristics are employed, at some point a power flow must be done to ensure constraints have not been violated. Having identified invalid solutions due to constraint violations, most methods are not capable of incorporating this knowledge in finding an alternative solution.

In this section, references [2, 16, 22, 23, and 28 - 38] are reviewed.

4.3.1 The Work of Merlin and Back, and Related Works

The first work to examine the problem of minimizing losses through distribution system reconfiguration was published in 1975 by Merlin and Back [2], who modeled the distribution system as a spanning tree structure, with line sections represented by the arcs of a graph, and the buses by the nodes. The final configuration that minimized losses was determined from the values found for binary variables associated with switch status. System constraints were neglected.

Merlin and Back approximated the behaviour of the distribution system by performing a DC power flow as a meshed network, accounting only for the real component of the current in loss calculations, and assuming differences in bus voltage angles were negligible. The strength of the algorithm of Merlin and Back was that a solution was obtained which was independent of the initial switch status. The algorithm of Merlin and Back required an iterative process of removing the branch with the lowest power flow and then performing a minimal loss power flow until a radial network was obtained.

This technique is similar to a technique examined by Willis *et al* [39] for the distribution system planning problem, which includes determining network layout, equipment size and capacities, and a radial switching pattern. Although the primary goal in this problem is not to reduce system losses, but to minimize costs, their comparison results are useful.

Willis *et al* compared several optimization techniques, and one of the techniques examined by Willis *et al* is null-point load flow. Null-point load flow is not an optimization technique *per se*. The method applies a network load flow to optimize a radial feeder system by closing all switches in the network, carrying out a load flow, and interpreting the results. The computed network flow will have null points in the system where the power flow is zero (or very small), and it is at these points that switches are opened to restore the radiality of the network. Hence, the method is very similar to the method of Merlin and Back, and to other techniques discussed in this section. The method has intuitive appeal, in that the optimal open points should be where the power flow is at a minimum (although this may or may not be true). Network load flow programs are widely available, proven and easy to apply, and computationally fast compared to many optimization algorithms.

The primary disadvantage of the null-point load flow method is that it does not respond directly to capacity constraints. Willis *et al* [39] found that results decreased dramatically with the null-point load flow method for the distribution system planning problem as the load/capacity ratio increased. Indeed, in comparing various optimization methods, they concluded that "the null-point load flow method appears to be of little practical value (in fact it could be termed a placebo, lulling the user into the belief that the system has been optimized when in fact it is far from it)." For example, Willis *et al* found that using numerical optimization techniques and genetic algorithms on a large distribution system (more than 10,000 nodes) resulted in savings of 9%, and a solution was found in approximately 60 seconds. For the same system using null-point load flow, savings of only 3% were found (although this solution was found in 20 seconds). If these results are extended to the method of Merlin and Back, and to other techniques discussed in this section, it is clear that the technique of performing load flows and opening switches in sections with the smallest power flow can lead to suboptimal solutions.

The method of Shirmohammadi and Hong [29] differed from that proposed by Merlin and Back only in the inclusion of feeder current constraints, and in the use of a compensation-based power flow technique to ensure that the behaviour of the weakly meshed distribution network is more accurately modeled. Both this method and that of Merlin and Back only minimize losses. However, they can not guarantee that an optimum solution will be found.

In reference [30], Goswami and Basu introduced an algorithm similar to that of Merlin and Back [2], the primary difference being that the distribution network is never represented as a meshed system. Goswami and Basu argued that it is invalid to model distribution systems as meshed networks, as the optimum flow pattern for a meshed network will be different than that of a radial configuration. Thus, Goswami and Basu close only one switch at a time, carry out a power flow, and then open the switch carrying the smallest current to open the loop and return the system to its radial configuration. The algorithm terminates when the switch that is opened is the same switch that was initially closed. Three methods were presented to select which switch to close: (1) the switch having the greatest voltage across it; (2) the switch having the smallest voltage across it; and, (3) at random. Goswami and Basu note that, for the 37-bus system studied, the method of switch selection did not affect the results.

Borozan *et al* [31] offered three algorithms to improve the method of Shirmohammadi and Hong. These algorithms were able to carry out the following operations faster than the original method: loop impedance matrix construction, partial re-ordering of network, and loop impedance matrix re-evaluation. Test results showed that the algorithms increased the speed of execution of Shirmohammadi and Hong's method, but the optimal solution was not always found, and voltage violations occurred.

Roytelman and Shahidehpour [23] use a method similar to that of Shirmohammadi and Hong. Their algorithm closes all open switches, carries out a load flow with branch reactance set to zero, and then opens the branch with the smallest current. The process is repeated until the network is restored to a radial configuration.

4.3.2 The Work of Civanlar, Grainger, Yin and Lee, and Related Works

The algorithm of Civanlar, Grainger, Yin and Lee [28] is perhaps the most often cited reconfiguration algorithm, and is often used as a bench mark to measure the performance of new algorithms. Civanlar, Grainger, Yin and Lee made use of heuristics to determine a configuration which would reduce losses. The following expression was developed to determine losses resulting from a load transfer between feeders:

$$\Delta P = Re\{2\left(\sum_{i \in D} I_i\right)(E_m - E_n)^*\} + R_{loop}\left|\sum_{i \in D} I_i\right|^2 \tag{4.1}$$

where:

D = the set of buses disconnected from feeder (I) and connected to another (II),

m = tie bus of feeder I to which loads of feeder II are to be connected,

n = tie bus of feeder II that will be connected to bus m via a tie switch,

I_i = complex bus current at bus i,

R_{loop} = series resistance of the path connecting two substations buses of feeder I and feeder II via closure of a specified tie switch,

E_m = component of $E = R_{BUS} I_{BUS}$ corresponding to bus m,

R_{BUS} = bus resistance matrix of feeder I before the load transfer,

I_{BUS} = vector of bus currents for feeder I, and

E_n = similar to E_m , but defined for bus n of feeder II.

The right-hand term in Equation (4.1) is always positive, and, hence, to have a drop in system losses (ΔP negative), it follows that a load transfer will only reduce system losses if there is a significant voltage difference across an open switch, and only if the load is being transferred from the higher voltage side to the lower voltage. The authors note that this may not be true if the phase angles of the two voltages are not similar, or if the complex bus currents are not in phase with phasor voltages (i.e., power factor not close to unity). They also note that the expression becomes less and less accurate as the amount of load transferred increases, or if the switch is close to a substation.

Civanlar *et al* proposed the following two heuristic rules:

Rule 1: *Loss reduction can only be attained if there is a significant voltage difference across an open tie switch.*

Rule 2: *Loss reduction will be achieved if loads on the higher voltage drop side of the tie switch are transferred to the other side.*

The high/low voltage rule is used to eliminate switching options for reconfiguration, and Equation (4.1) is then used to determine the change in system losses for the remaining switching possibilities. The option with the largest negative ΔP is selected and a power flow carried out. This process is repeated until there are no candidate switching options.

The advantages of the algorithm of Civanlar *et al* are that it allows rapid determination of a switching configuration which reduces losses. The disadvantages, however, must also be considered:

1. the method assures only a reduction in losses, and not a minimization of losses;
2. the proposed network configuration depends on the initial switch status;
3. the equation for estimating changes in system losses as a result of a load transfer is inaccurate if the load transferred is large, if the transfer occurs

close to a substation or if the power factor is not close to unity; and,

4. system constraints are ignored.

Exceptions to Rule 2 have been reported by other researchers. Baran and Wu [33] offer a counter-example to the high/low voltage rule, showing that it is possible to perform a branch exchange where a switching on the higher voltage side results in a positive loss reduction. As noted in the previous section, Goswami and Basu [30] found that the method of switch selection (switch having the greatest voltage across it, switch having the smallest voltage across it, or at random) did not affect their results. Hence, the two heuristic rules are only very approximate.

Castro and Watanabe [32] extended the work of Civanlar *et al* by making use of a more efficient search algorithm requiring less computational effort. Civanlar *et al* considered branching on only the most promising switching option, which reduced solution time, but increased the likelihood of finding a local minimum. Castro and Watanabe proposed selecting the maximum number of feasible switching operations at each stage of the algorithm, which offered the advantage of finding a better solution in a shorter time. However, a global optimum was not assured, and they continued to use the high/low voltage rule. System constraints were not considered.

Baran and Wu [33] followed the approach taken by Civanlar *et al*, extending the work by introducing two different methods to approximate the load flow in a system after a load transfer between two substations, feeders or laterals, and making use of a set of power flow equations developed specifically for radial distribution feeders. Power flow was described by a set of recursive equations that used the real power, reactive power and voltage magnitude at the sending end of a branch to express the same quantities at the receiving end of the branch. Knowing (or estimating) these quantities at the first node in a network, the same quantities were determined for downstream nodes on a feeder using the equations developed in a *forward update*. A similar set of equations was developed for a *backward update*, where the update started from the last node of a feeder and proceeded towards the substation. By successively applying the forward and backward updates, a power flow solution was obtained. The two power flow solutions offered are (1) a simplified version; and, (2) a full version of the power flow just discussed.

For a two feeder system with 32 buses and 5 tie lines, the optimal solution was found using the simplified method. Interestingly enough, the global optimum was found "by accident," as it estimated a branch exchange as positive when it was, in fact, negative, allowing it to perform more iterations to find the global solution. Using the second proposed method, and a full power flow solution, the algorithm was unable to find the global optimum.

Sarfi *et al* [22] partitioned the distribution system into groups of load buses, and then applied Civanlar's technique [28] to minimize losses within each group of buses. When tested on the same system used by Civanlar *et al* [28], Sarfi *et al* achieved similar loss reduction results for one set of bus partitions, but, for a different set of partitions, no loss reduction was achieved. The results presented indicate that the solution quality was very dependent on the assignment of buses to groups, and how the network was partitioned.

Taleski and Rajicic [16] extend Civanlar's method to minimizing energy losses instead of power losses by incorporating data from daily load curves.

Fan *et al* [34] use a single loop optimization technique whereby a normally-open switch is selected to be closed, and then the problem is to find a normally-closed switch to open in the loop such that line losses are minimized, similar to the method of Goswami and Basu [30]. The normally-open switch is selected by examining voltage differences across open switches to determine which switch experiences the largest voltage difference, similar to the method of Civanlar *et al*.

In [35], Roytelman *et al* sought to incorporate five objectives in a single objective function, including (1) minimization of feeder losses; (2) load balancing among supply transformers; (3) minimization of the worst voltage drop; (4) minimization of service interruption frequency; and, (5) balanced service to important customers (by ensuring all important customers are not served from the same transformer). Objectives were weighted as deemed necessary. A two-stage approach was used. In the first stage, an initial solution was found using a technique similar to that of Merlin and Back [2] to determine a radial network configuration. Then, the solution was improved by closing a switch and opening an adjacent one to see the change in objective function. Civanlar's formula was used to determine changes in losses resulting from a branch exchange.

In [38], Peponis *et al* combined reactive power control (through capacitor installation) and network reconfiguration. Peponis *et al* compared the Civanlar and Shirmohammadi algorithms, and found that the Civanlar technique was approximately four times faster, but that the final solution was very dependent upon the initial configuration.

4.3.3 Other heuristic algorithms

Liu, Lee and Vu offered two algorithms that they asserted would ensure a globally optimal solution [36]. One algorithm was based on a uniformly distributed load model (UDLM) and the other a concentrated load model (CLM). Liu, Lee and Vu demonstrated that by considering loads as current sinks, the current flowing through an arc could be represented by a sum of a basic current (y_k) and a constant (a_k).

The first algorithm identified which sectionalizing points had to be open for a minimal loss reconfiguration. Global optimality could not be assured when practical constraints such as line voltage drop were considered, and the authors noted that solutions could be found that violated the radial topology requirement. In this algorithm, if the system was assessed to be "non-optimal" (failure to meet set criteria), "non-optimal" feeder pairs would be selected and minimum loss positions determined until a tolerance was satisfied. Because the first algorithm relied on a piece-wise parabolic form of the loss function, sectionalizing points determined by the algorithm did not always correspond to actual switch positions, and hence the second algorithm was used to determine the actual switch positions for the optimal system configuration. This second algorithm differed from the first in that all "non-optimal" pairs were assessed using loss estimation formulae and only the pair with the greatest loss reduction selected.

In [37], Aoki *et al* note that reconfiguration is used in Japan to balance loads among feeders and transformers for fear of fault occurrence, as well as to reduce system losses, but that the main emphasis is on load balancing. The authors assume all section loads are known, that all feeders are of equal capacity, and that the system is initially in a feasible (but not necessarily optimal) state. Loads are transferred between feeders by determining which feeder has the largest load, and which has the smallest, while ensuring the radial structure of the distribution system is maintained. This process is carried out until feeders and transformers are loaded as equally as possible. There is no guarantee that system losses are reduced.

4.4 Previous Research Based on Operations Research Methods

Numerical optimization techniques apply computed numerical formulae and procedures to search (usually iteratively) for the best configuration. Previous works based on numerical optimization techniques include [18, 24, and 40 - 45]. The advantages of these techniques include convergence to the mathematically optimal configuration, and that proven algorithms are widely available and understood. However, the disadvantages include mathematical complexity which may make programming and diagnosis difficult, and convergence that takes so long to be of no practical value.

Linear programming (LP) methods require all relations to be linear or approximated by linear functions and were popular in the early 1980s for solving such power system problems as the capacitor placement problem [46]. Only smaller systems were considered, as the computation times for larger systems made LP methods impractical. As well, solutions obtained were not always optimal, due to approximations introduced by the linearized models [47].

Few researchers have based solution to the reconfiguration problem solely on linear programming methods. Aoki *et al* [40] divide distribution lines into segments according to the differences of the load distributions and line constants. The status of each switch in a system (open or closed) should be solved as a discrete optimization problem, but, since there are many switches, as well as line and voltage constraints to consider, finding a feasible solution would lead to large computation times. Aoki *et al* approximate the variables identifying the locations of normally open switches as continuous variables, and, after solving the continuous problem, the location of the open switches is determined by rounding the solution to the nearest actual switch. The authors note that their solution is not necessarily optimal, but that on a 59-bus test system they were able to reduce losses by 5%.

In [41], Chen *et al* develop equations to determine total feeder loss using regression analysis. The method develops an equation based on a specific feeder and the hourly load pattern at each bus. The method would not be suitable for the reconfiguration problem, as it would be necessary to recalculate the coefficients for the regression equation for each change of configuration, which would be difficult to accomplish in real-time.

Glamocanin formulated the reconfiguration problem as a transshipment problem with quadratic costs [42]. Using Glamocanin's method, it was first necessary to obtain a feasible solution as the starting basic solution. The quadratic simplex method was then used to improve the solution. System constraints were not included.

Huddleston, Broadwater and Chandrasekaren [43] offer a reconfiguration algorithm based on modelling the distribution system by a quadratic loss function as a function of switching currents and a set of feeder current constraints. Their algorithm assumes that the distribution system has a unity power factor (typical for many urban distribution systems), and thus the loss function can be constructed as a DC model. The algorithm of Huddleston *et al* looks for feeder sections having negligible currents to indicate open switches. The quadratic optimization problem was solved by using the IMSL subroutine QPROG.

The problem that is solved is a continuous problem, yet the currents are switched discretely, and thus solution results have to be interpreted before they can be implemented. In addition, Huddleston, Broadwater and Chandrasekaren note that a distribution system with n circuits and m open circuits requires a solution with $n + 2m$ unknowns. For the relatively small sample system of Civanlar *et al*, with 16 sections and 3 open circuits, this would lead to 22 unknowns. As the size of the distribution system increased, so too would the number of unknowns, as would the computation time.

The work of Broadwater *et al* [18] builds upon the previous work of Civanlar *et al* [28], as well as that of Huddleston *et al* [43]. The work of Broadwater *et al* uses Civanlar's switching rule, a single switch pair operation per iteration and a direct search method incorporating Huddleston's loss function, including voltage and current constraints. The reconfiguration algorithm proposed calculates losses for each possible switch pair operation. A load flow is required at each iteration, after a new base case is developed as a result of a switching operation. There is no discussion of how long the algorithm takes to find a solution for a system. However, for a large distribution system, evaluating the losses for every possible switching combination would not be feasible.

Wagner *et al* presented a reconfiguration algorithm based on a solution of a linear transportation problem [27]. Feeder line section losses were approximated by a piece-wise-linear function, and the problem solved using the stepping stone algorithm. Feeder voltage and thermal constraints were included in the simplifications introduced by the linearizations. Using a working 44 kV distribution network, a comparison of the method offered by Wagner *et al* was made with the methods proposed by Shirmohammadi and Hong [29], as well as that of Civanlar, Grainer *et al* [28]. For a small system such as the one presented with the work of Wagner *et al*, convergence was rapidly obtained. It is not clear that analysis of a more realistic distribution system would be possible in a real-time implementation.

In [44], Chen *et al* solve the problem as a binary integer programming problem using branch and bound techniques to minimize a cost function that includes I^2R losses, as well as labour and line switch costs. In this case, voltage constraints are included. To reduce computation time, Chen *et al* estimate the losses for all possible switching combinations using a DC load flow where the imaginary parts of the voltage, current and line impedance are ignored. The switching configuration that offers the largest reduction of line losses is then selected. The method is essentially an exhaustive search of all possible switching combinations using an approximation for the load flow, and it is doubtful the method could be applied to larger systems.

Sarma and Prakasa Rao [45] use a 0-1 integer programming approach. However, the proposed method only considers opening switches on either side of currently-open switches (although multiple switches are considered at each iteration), and voltage and current constraints are ignored. For the 16-bus test system proposed by Civanlar *et al*, the algorithm finds that the best solution has switches 15, 17 and 26 open (which is not the global minimum found by Civanlar *et al*).

Roytelman *et al* [24] used a gradient search technique, varying the search direction according to the largest derivative of the objective function with respect to the control variables. The partial derivatives were computed as the differences in the objective function divided by corresponding increments in the discrete variables, i.e.,

$$\frac{\partial F^k}{\partial x_i^k} = \frac{F^{k+1} - F^k}{x_i^{k+1} - x_i^k} \tag{4.2}$$

where k, $k + 1$ are the current and next positions for control variable, x_i. Constraints were included in the objective function as penalties normalized to power. Being a gradient descent method, the final solution depends on the initial configuration, and there is a risk of being trapped in a local minimum.

4.5 Previous Work Based on "Modern" Techniques

"Modern" techniques include techniques based on artificial neural networks, fuzzy systems, expert systems, simulated annealing and genetic algorithms. These techniques have been applied to the distribution system reconfiguration problem, and include references [47 - 57].

4.5.1 Work Based on Artificial Neural Networks

Artificial neural networks (ANNs) have been proposed for many power system applications [58]. Their use appears well-suited to reconfiguration, as they can be used to map the relationship between the highly non-linear nature of a load pattern to a network topology which offers minimal line losses. Perhaps the most widely-used ANN is the backpropagation network [59]. A typical backpropagation network has an input layer, one or more hidden layers, and an output layer. Each layer is fully connected to the succeeding layer, and each layer consists of a number of Processing Elements (PEs). A processing element is shown in Figure 4.2.

In this figure, it can be seen that there are a number of inputs. Each input is multiplied by a weight associated with that input, and the sum of the weighted inputs is determined. The output of the PE is determined by a transfer function, which can be the sigmoid function, hyperbolic tangent or sine functions.

If the network has some global error function associated with it, it is assumed that all processing elements and connections are to blame for an error (or for the actual output not being the same as the desired output), and responsibility for the error is affixed by propagating the output error backward through the connections to the previous layer. This process is repeated until the input layer is reached. Hence, the name "backpropagation." The input is forward propagated through the layers to the output layer, the error at the output determined, and then the error is back propagated through the network from the output to the input layer. During training, the global error function is minimized by adjusting the weights.

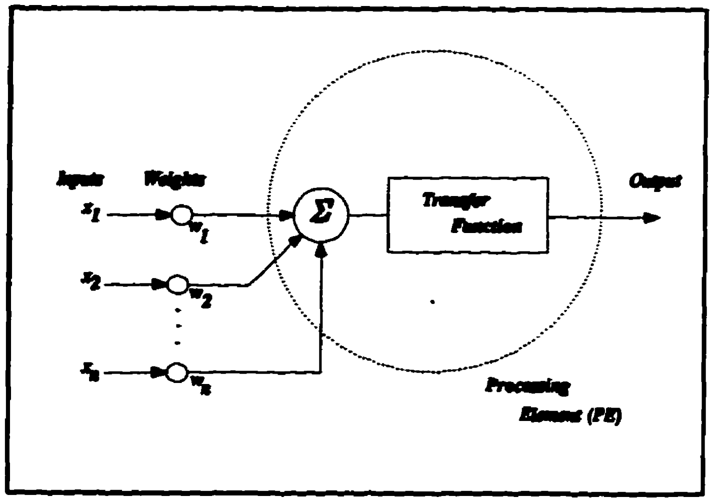


Figure 4.2. A processing element (PE).

ANNs prove themselves most useful in applications such as load forecasting where existing models do not have enough accuracy, and where vast amounts of historical data are available. Although the use of ANNs can offer reduced solution times for even large problems, three factors appear to limit their usefulness to a utility for the loss minimization problem [60]:

1. a considerable amount of time is required for collecting data and for training the neural network, as loads vary with the time of day and season of the year, as well as by customer type, resulting in enormous amounts of data;
2. training must be performed for each utility's network and subsequent changes in the system must be accounted for; and,
3. accurate training data must be acquired to ensure that the ANN offers meaningful results.

Kim, Ko and Jung proposed a two-stage algorithm based upon ANNs for distribution system reconfiguration for loss minimization [48]. They proposed dividing the distribution network into load zones, with each load zone having a distinct set of two ANNs trained to classify the loading level and to reconfigure the zone based upon the assigned loads. The use of ANNs offered a fast solution, as no load flow operations were required within the solution algorithm. A multi-layered feedforward network topology was selected for the ANN in view of the adaptive learning capability of this topology. Training data was obtained by a solution of a quadratic programming problem, whose constraints included line voltage drop and current limits. Although good results were obtained by this algorithm, the training data used was simulated data for a small system. The massive amount of data needed to accurately model a system of realistic size. as well as the network training time, would most likely preclude this approach.

4.5.2 Work Based on Fuzzy Systems

Fuzzy set theory and fuzzy logic was introduced in the 1960s by Zadeh [61] as a formal tool for dealing with uncertainty, where vague descriptions for variables may be more or less precise (less or more fuzzy, respectively), depending upon the certainty with which a variable can be described. For example, it may be said that the load on a feeder is heavy. How heavy is "heavy?" Fuzzy set theory is employed to deal with this uncertainty. In the fuzzy domain, each variable is associated with a membership function that indicates the degree of satisfaction of the variable from zero to unity, and expressed by a set of ordered pairs, i.e.,

$$X = \{(x, \mu(x) \mid x \text{ is a possible value of variable } X\},$$

where $\mu(x)$ is the membership function which denotes the possibility that variable X has the value x. A fuzzy set A of X is defined to be the set of ordered pairs, $\{(x, \mu_A(x)\}$, where $x \in X$ and $\mu_A(x) \in [0,1]$ is the degree of x in A.

The concepts of fuzzy logic can be illustrated with the example shown in Figure 4.3. Consider a variable, L, (representing the load) having the set of values *{light, medium, heavy}*. The values of the load are the labels of the fuzzy sets, $A_{low}(L)$, $A_{medium}(L)$, and $A_{heavy}(L)$ on the domain of numeric loads, L. In this case, a load of 0.38 p.u. is interpreted to be *low* with degree 0.21 and *medium* with degree 0.64. *If-then* rules are then used to relate these imprecise relationships.

Although the title of Reference [47] is "A Fuzzy-Based Optimal Reactive Power Control," this paper addresses, in fact, the reconfiguration problem, noting in its abstract that the authors present a mathematical formulation of the optimal reactive power control problem, where the objectives are "to minimize real power losses and improve the voltage profile of a given system."

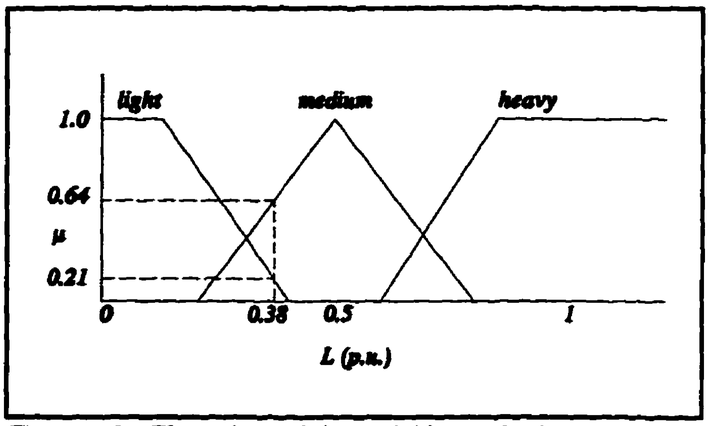


Figure 4.3. The values of the variable *load - light, medium* and *heavy* - are the labels of the fuzzy sets.

In [47], Abdul-Rahman and Shahidehpour use a fuzzy-based linear programming approach. Voltage and reactive power constraints were "fuzzified," i.e., made less rigid (or crisp), and hence allowed small voltage and power flow violations (the level of which would be determined by system operators). The method was tested on a 6-bus and 30-bus system, and voltage violations of 2% and 3%, respectively, were allowed, although not all buses were subject to the same violation. For the 6-bus case, it was found that the proposed method found a better solution than traditional LP methods with fewer iterations, but that the time per iteration was longer for the fuzzy method. For the 30-bus method, traditional LP methods found a better solution, but in a longer time. A large system of realistic size was not examined. The method can not guarantee an optimum solution. As well, an operator must specify the acceptable violation limits, and for which feeders and buses, and this may not be obvious.

4.5.3 Work Based on Expert Systems

An expert system is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice [62]. The system may act as an assistant to a human decision maker, or completely fulfil a function that normally requires human expertise. Unlike traditional sequential computer programs, expert systems simulate human reasoning about a problem, rather than simulating the problem itself. Heuristics are often employed.

Taylor and Lubkeman [49] proposed an expert system for distribution system reconfiguration based upon extensions of the rules of Civanlar, Grainger *et al* [28]. Taylor and Lubkeman described the primary objective of their work as being to avoid transformer overloads, feeder thermal overloads, and abnormal voltages. By satisfying these criteria, they asserted that they would simultaneously find a configuration for loss minimization. Taylor and Lubkeman used a best-first strategy to reduce the solution time. Five different rule sets were developed to drive the inference of the expert system developed. Following each decision, a load flow calculation was necessary to update the network's operating status. The use of the heuristic rules was demonstrated to reduce the search space considerably. However, the use of a best-first search strategy precludes the guarantee of finding a globally-optimal solution.

Chang, Zrida and Birdwell introduced the requirements for a knowledge-based software package for analysis and control of distribution systems [50]. The knowledge base would employ tools specific to distribution analysis to ensure precise, representative modelling. Reconfiguration for loss reduction figures prominently in their proposed package and would be driven by an expert system.

Chang and Chung [51] describe the development of an expert system for on-line use of power system operators in a SCADA environment. The proposed system uses the method of Aoki *et al* [40] to determine which loads to transfer, as well as several heuristic rules proposed by system operators. Chang and Chung note that the method of [40] was implemented in the computer language, Prolog, which, while being a useful language for expert systems, is not well-suited to the "number-crunching" required by the reconfiguration problem.

Recently, Sarfi [52] proposed an expert system combined with fuzzy logic. The method of Civanlar *et al* [28] was used to obtain an initial, suboptimum configuration. Then, several rules were proposed to further optimize the network, taken into account network constraints. A large part of this work was based on conservation voltage reduction, which, as discussed earlier, may or may not be viable. As well, Sarfi only examined adjacent switches when considering which switch to close, leading to the possibility of suboptimum solutions.

4.5.4 Work Based on Simulated Annealing

Simulated annealing (SA) is an algorithmic approach to the solution of optimization problems. The name of the algorithm comes from the analogy between solving optimization problems and the simulation of annealing of solids. When a material is annealed, it is first heated to a high temperature, and then slowly cooled according to a cooling schedule to reach a desired state. At the highest temperature, the particles in the material are arranged haphazardly. As they cool, they form a lattice structure, and reach a minimum energy state. First proposed by Metropolis et *al* [63] in 1953, interest in the algorithm grew following a paper by Kirkpatrick et *al* [64] in 1983.

SA is an iterative improvement technique. The cost of an initial configuration is determined, and then the configuration is given a small perturbation. The cost of the perturbed configuration is calculated, and, if the cost is less, the new configuration becomes the current configuration. If the cost is greater, the new configuration is accepted probabilistically, i.e., with probability exp $-(\Delta/T)$, where Δ is the change in the cost function and T the current temperature. This provides a means for the system to escape from local minima, although the probability of this occurring decreases as annealing progresses.

To use this algorithm, a cost function must be derived. Several parameters must be specified empirically, including an initial temperature and a cooling schedule. The cooling schedule includes the final value of the temperature, how long the system is held at each temperature, and how the temperature is reduced. If the initial temperature is not high enough, the system will not be completely at random at the start. If the cooling schedule is too fast, the system hardens into a state that may not be globally optimal. If the cooling schedule is too slow, computation time suffers. An initial configuration must also be selected, usually by random assignment.

Chang and Kuo [53] use simulated annealing to solve the reconfiguration problem after introducing an approximated power flow. The algorithm presented in [53] selects an open tie-line switch to close at random, and then opens a switch immediately adjacent to it to preserve the network's radial topology. A criticism of this algorithm is that any sectionalizing switch between the now-connected substations could be opened, not just the ones next to the just-closed tie-line switch. There is a strong possibility that the optimum solution will be missed if such a narrow range for switch selection is used.

Chang and Kuo use the following cooling schedule:

$$T_{i+1} = \alpha T_i \tag{4.3}$$

where T_i is the temperature for the ith iteration, and α the cooling rate. The initial temperature must be specified. The cooling rate is a number between 0 and 1, and the authors use 0.9 for two test cases, and 0.95 for a third test case, but it would appear that trial and error is needed to find its optimal value. There is no discussion of how to choose the cooling rate. As well, the authors note that if the rate is set too low, the solution quality is poor. Test cases ranging from 13 to 69 buses revealed it was possible to reduce system losses by 5% to 55%, with the amount of loss reduction increasing as the number of buses increased.

The work of Chang and Kuo [53] is very similar to that of Chiang and Jean-Jumeau, who four years earlier developed a two-stage algorithm based on simulated annealing [54, 55]. The algorithm of Chiang and Jean-Jumeau proposed to include both optimal loss reduction and load balancing in a multi-objective, non-differentiable objective optimization problem with both equality and inequality constraints. The primary difference between the two works is how the power flow is carried out at each step in the algorithm - Chiang and Jean-Jumeau use a fast decoupled load flow, while Chang and Kuo use a method similar to that proposed in [33].

4.5.5 Work based on Genetic Algorithms

Simulated evolution is intrinsically a robust search and optimization technique whose process can be applied to engineering problems where heuristic solutions are not available or provide unsatisfactory results [65]. The physical processes involved include reproduction, competition and selection. During reproduction, an individual's genetic program is transferred to its offspring. Competition is the result of expanding populations and finite resources, and selection is the result of competitive replication.

In the past few years, interest has grown in solving problems using algorithms based on the principles of biological evolution [66]. These algorithms maintain a population of potential solutions, have some selection process based on the fitness of individuals within the population, and have some recombination operators. Perhaps the best-known of these methods is Holland's Genetic Algorithm [67]. Evolution programs are essentially probabilistic algorithms that maintain a population of n individuals, $P(t) = \{ x_1^t, \ldots, x_n^t \}$ at iteration t. Each individual represents a possible solution for the problem at hand implemented as a data structure, S. Each solution x_i^t is evaluated to determine its "fitness." The better individuals are selected to be parents for the next generation, and a new population, $P(t+1)$, for the next iteration is generated. Some of the offspring will undergo transformation as a result of application of genetic operators. *Mutation* is the operation whereby new individuals are created by making small changes to single individuals, typically on a single bit ($m_i : S \rightarrow S$), while high order transformations c_j create new individuals by combining parts from several individuals in an operation known as *crossover* ($c_j : S \times \ldots \times S \rightarrow S$). After several generations, the program converges, and the best individual hopefully represents the optimum solution [68].

Genetic algorithms manipulate a population of potential solutions to an optimization problem by operating on an encoded representation of the solutions equivalent to the genetic material of individuals in nature. Solutions are encoded as strings of binary bits. Each solution has associated with it a fitness value that allows the solution to be compared to all other solutions in the gene pool. The higher the fitness value, the higher the chances that an individual will survive and reproduce, and the larger its representation in the population. During reproduction, crossover is used to exchange portions of genetic material between strings. Mutation also occurs to cause sporadic and random alteration of bits. This plays the role of regenerating lost genetic material.

Crossover is the crucial operation of genetic algorithms, and is illustrated in Figure 4.4. Pairs of strings are picked at random from the population for crossover. In single-point crossover, a crossover point is picked at random, and portions of the strings beyond that point are exchanged to form new strings. Crossover only proceeds if a randomly generated number in the range [0,1] is greater than the crossover rate (specified at the start of the algorithm).

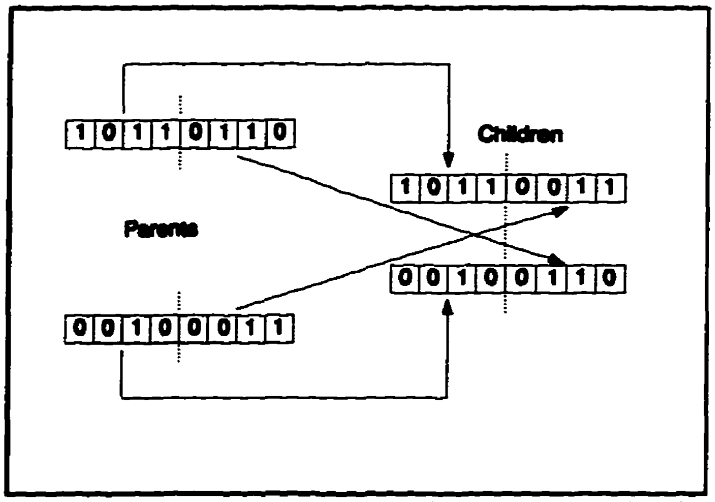


Figure 4.4 Crossover of two parents having a crossover point of 4.

After crossover, strings are mutated by flipping a bit. The bit to be flipped is selected at random, and mutation proceeds only if a randomly generated number in the range [0,1] is greater than the mutation rate specified at the start of the algorithm. Mutation allows strings to recover lost genetic material. For example, if all of the bits in all of the strings in a population have converged to 1, crossover can not generate a 0. However, mutation would allow this to occur.

There can be problems when using GAs. When the algorithm is first run, the string having the highest fitness function can generate a disproportionate number of offspring that may lead to premature convergence. Conversely, after many iterations, the population tends to become homogeneous as the variance in fitness values is small. As a result, equal number of offsprings are allocated to all strings, thus eliminating the driving force that promotes better strings [69].

A second issue involves the crossover mechanism. Users must decide how many crossover points will be used, and typically use one or two. Similarly, the crossover rate must be specified. This must also be done for the mutation rate [69].

Users must also specify the population size. Large populations increase a population's diversity, and help to prevent premature convergence, but, at the same, increase the time needed for the population to converge to an optimal solution. It appears that population size, crossover rate and mutation rate are inter-related, and that small populations need relatively large crossover and mutation rates compared to those of large populations [69].

Another problem with most implementations of genetic algorithms to date is their domain independence, and the difficulty in dealing with constraints. Since most GAs use binary strings - 1's and 0's - how constraints are implemented is very important. As noted in [68]:

> "Constraints that cannot be violated can be implemented by imposing great penalties on individuals that violate them, by imposing moderate penalties, or by creating decoders of the representation that avoid creating individuals violating the constraint. Each of these solutions has advantages and disadvantages. If one incorporates a high penalty into the evaluation routine and the domain is one in which production of an individual violating the constraint is likely, one runs the risk of creating a genetic algorithm that spends most of its time evaluating illegal

individuals. Further, it can happen that when a legal individual is found, it drives the others out and the population converges on it without finding better individuals, since the likely paths to other legal individuals require the production of illegal individuals as intermediate structures, and the penalties for violating the constraint make it unlikely that such intermediate structures will reproduce. If one imposes moderate penalties, the system may evolve individuals that violate the constraint but are rated better than those that do not because the rest of the evaluation function can be satisfied better by accepting the moderate constraint penalty than by avoiding it. If one builds a 'decoder' into the evaluation procedure that intelligently avoids building an illegal individual from the chromosome, the result is frequently too computation-intensive to run. Further, not all constraints can be easily implemented in this way."

A decoder looks at solutions to determine if they violate any constraints, and discards any found. Alternatively, a decoder can be used to "correct" infeasible solutions. Problems with decoders may include excessive computation time, the possibility that constraints may not be easily checked, and that they are very specific to the problem at hand.

4.5.5.1 Applications of Genetic Algorithms to the Problem of Distribution System Reconfiguration

Nara *et al* [56] applied a genetic algorithm to minimize distribution system losses. Noting that the problem of distribution system reconfiguration for loss minimization is a problem of determining the position of open sectionalizing switches, they formulated the problem as a 0-1 integer programming problem with the following assumptions:

1. section loads were uniformly distributed, balanced constant current loads;
2. the power factor of section loads was 1.0;

3. the current phase shift due to line impedance was negligible; and,

4. the maximum voltage drop occurs at the end of a feeder, as capacitors are usually not installed in urban distribution systems.

Sections and switches were represented as binary numbers and subjected to mutation and crossover. A fitness function was developed to minimize losses, and included penalty terms for line and transformer capacity constraint violations, and excessive voltage drop. To avoid the need to carry out a power flow at each iteration, Nara *et al* developed an expression similar to that of Civanlar *et al* [28] that allowed the estimation of the change in losses as a result of a branch exchange (assuming an initial power flow was available).

The results for two test systems are shown below in Table 4.1, along with a comparison of the results using simulated annealing (SA).

CASE	Initial losses (kW)	Crossover rate	Mutation rate	Losses after 1000 iterations of GA (kW)	Computation time of GA (min)	Losses after SA (kW)	Computation time of SA (min)
A	341.13	0.6	0.008	288.54	61.5	285.81	227.4
B	980.13	0.6	0.0003	790.39	1181.1	653.08	3109.9

Table 4.1 Summary of the results presented in reference [56].

Case A is a test system containing 106 sectionalizing switches, with 10 normally open. Case B is an actual urban distribution system with 1692 switches. It can be seen that application of the GA and SA algorithms lead to lower system losses (as compared to the initial losses). SA outperformed GA in both cases, although the computation time is considerably longer for SA.

However, even the computation time needed for GA is excessive, taking over an hour for the smaller system, and nearly 20 hours for the larger one. This can hardly be considered useful for real-time operation. Similar results were seen in [57], where Nara and Kitagawa repeated the work of [56], but added distribution transformer losses as part of the minimization process. As expected, the additional parameter increased computation time. For the system described as Case A above (106 sectionalizing switches with 10 normally open), the cpu time required on an Apollo workstation was 102 minutes, and the optimal configuration was not found. In a system with 271 switches with 56 open switches, the cpu time was 567 minutes (nearly 10 hours).

Interestingly enough, solutions were obtained for the same systems using simulated annealing, with solution times of 615 minutes for the first case and 2556 minutes (i.e., nearly 43 hours!) for the second, although the results obtained surpassed those of the genetic algorithm.

The excessive computation time is not surprising. As noted earlier, how constraints are handled strongly affects the performance of a GA. If the likelihood of producing illegal individuals is high, the GA wastes much of its time evaluating them. In both [56] and [57], Nara *et al* assigned penalties to strings that violated voltage, line capacity and transformer capacity constraints, with the result that many solutions that violated one or more of the constraints were in all likelihood produced. Computation time was needed to evaluate these illegal solutions. As well, Nara *et al* indicate that some solutions left the system in a loop configuration, or left some sections de-energized. These problems were not handled in the constraints, and thus part of the computation time was needed to check for those conditions.

In [56], it was noted that there were 1000 iterations of GA for each case. However, figures presented showed that much of the improvement came in the first 200 iterations, but there was no way to know that the best solution had been found. No stopping mechanism was incorporated (save the 1000 iteration limit), nor was any mechanism proposed whereby the algorithm could "zoom in" on the best solution to try to find a better solution in a smaller search region.

4.6 Summary

Distribution system loss minimization through system reconfiguration is a difficult problem that has been investigated by many researchers. Most algorithms proposed to date suffer from one or more of the following shortcomings:

a. losses are reduced, but not necessarily optimized, in that locally optimum solutions are found instead of global optimums;

b. excessive computation time allows application to only small distribution systems of unrealistic size, or restricts their use to off-line applications;

c. the final solution depends upon the initial system configuration; or,

d. numeric parameter values (such as the cooling rate and initial temperature in simulated annealing) must be heuristically specified and/or fine tuned.

Perhaps the biggest shortcoming of most of the algorithms in this chapter is that the system voltage drop and thermal constraints have been ignored. Bound by standards for maximum allowable voltage drop, and by equipment current-carrying capacity, utilities would be reluctant to embrace any method that did not consider operational constraints.

In the next chapter, modelling the reconfiguration problem as a constraint satisfaction optimization problem - which is the basis of this thesis - is introduced.

Chapter 5

Constraint Satisfaction Optimization Problems

5.1 Introduction

Having examined the need for network reconfiguration to minimize losses, and the efforts of earlier researchers to develop efficient algorithms for the problem, attention is now turned to Constraint Satisfaction Problems (CSP) and Constraint Satisfaction Optimization Problems (CSOP). In this chapter, the CSP and CSOP are first reveiwed, followed by an examination of some of the general-purpose algorithms used to solve them, as well as the issues involved in finding a solution. In the following chapters, the reconfiguration problem is modeled as a CSOP, and a method to solve it is introduced.

5.2 A Review of Constraint Satisfaction Problems

In optimization problems, each facet of the problem can be considered a state variable. The state space holds all possible combinations of values of state variables, and can be thought of as an N-dimensional space, where N is the number of state variables. The size of each dimension is equal to the length of the list of the possible choices for that state variable. The size of the state space is equal to the product of the size of all dimensions.

The solution space is a subset of the state space, and includes all acceptable solutions. The solution space is spread throughout the state space, with no means to identify its members other than by evaluation by a set of tests. For a small state space, it may be possible to perform an exhaustive search, in which every possible solution is examined. A subsequent evaluation of the resulting solution space will reveal the optimum solution.

A Constraint Satisfaction Problem (CSP) is a problem with a finite set of variables, each associated with a finite domain, and a set of constraints which restrict the values that the variables can simultaneously take [70].

The domain of a variable is the set of all possible values that can be assigned to the variable. For example, if x is a variable, then D_x denotes its domain.

A label (assignment) is a variable-value pair that represents the assignment of the value to a variable. For example, $<x,v>$ denotes the label of assigning the value v to the variable x. $<x,v>$ is only meaningful if v is in the domain of x (i.e., $x \in D_x$).

A compound label is the simultaneous assignment of values to a set of variables. For example, $(<x_1, v_1> <x_2, v_2> \ldots <x_n, v_n>)$ denotes the compound label of assigning $v_1, v_2, \ldots, v_n$ to $x_1, x_2, \ldots, x_n$, respectively. A k-compound label assigns k values to k variables simultaneously. The variables of a compound label is the set of all variables which appear in the compound label.

A constraint on a set of variables is a restriction on the values that they can take on simultaneously. C_S denotes the constraint on the set of variables S.

A CSP can be more formally described in terms of the above definitions. A constraint satisfaction problem is a triple:

$$(Z,D,C)$$

where Z is a finite set of variables $\{x_1, x_2, \ldots, x_n\}$;

D is a function which maps every variable in Z to a set of objects of arbitrary type:

$$D: Z \rightarrow \text{finite set of objects (of any type)}$$

D_x is the set of objects mapped from x by D. These objects are possible values of x and the set D_x is the domain of x. C is a finite (possibly empty) set of constraints.

The task in a CSP is to assign a value to each variable such that all constraints are satisfied simultaneously.

An algorithm is said to be sound if every result that is returned by the algorithm is a solution. For a CSP, this means that any compound label which is returned by the algorithm contains labels for every variable, and this compound label satisfies all the constraints in the problem.

An algorithm is said to be complete if it can find every solution. Normally, an exhaustive search is not possible. Thus, an efficient method for navigating the state space is needed to determine the solution space, and the optimal solution within the solution space.

5.3 Existing Algorithms and Previous Work

Many CSP algorithms are constructive in nature, and can be viewed as a search through the space of partial variable assignments. One variable is instantiated at a time until all variables have been assigned and no constraint violations occur. When no consistent instantiation for the next variable can be found, backtracking is employed, i.e., one of the previously instantiated variables must be incorrect, and is unassigned. Deciding which values to assign to variables, and which variable to backtrack to, is an on-going topic of research [70].

The simple backtracking (BT) algorithm is a general search strategy which has been widely used in problem solving [71]. The basic operation is to pick one variable at a time, and consider one value for it at a time, making sure that the newly-picked label is compatible with all of the labels picked thus far. If labelling the current variable with the picked value violates any constraints, then an alternative value must be picked. When all of the variables have been labelled, then the problem is solved. If no value can be assigned to a variable without violating any constraints, the label which was last picked is revised, and an alternative value is assigned to that variable - hence the name backtracking.

A similar technique is the best-first search technique for evaluating solutions in a large state space. An example of the best-first search technique is the A* algorithm proposed by Hart *et al* in 1968 [72]. A solution and its evaluation are combined into a node, which in turn is placed on a tree. The best available node on the tree is chosen for improvement (called the Parent node). Any improvement produces a Child node. When no further improvements are possible from a parent node, it is closed. The optimal solution is the node with the best available solution.

In general, there are several ways to search for a possible solution, the most common being [73]:

1. depth-first searches;
2. breadth-first searches;
3. least-cost searches; and,
4. hill-climbing searches.

The depth-first search explores each possible path to its conclusion before another path is tried. If the search is implemented as a tree, the search becomes an inorder tree traversal. In the worst case, it degenerates into an exhaustive search. The performance of depth-first searches can be poor when a particularly long branch with no solution at the end is explored, as the search wastes considerable time in exploring the branch, as well as backtracking to the goal.

In breadth-first search, each node on the same level is checked before the search proceeds to the next deeper level. As with the depth-first search, a solution is guaranteed, because eventually the search degenerates into an exhaustive search. The performance of the breadth-first search is poor when the solution is several layers deep.

In a hill-climbing search, the algorithm chooses as its next step the node that appears to place it closest to the goal. This method can become trapped in local minima. As well, situations arise in which all next steps look equally good (or bad), in which case hill climbing is no better than depth-first searching.

A least-cost search takes the path of least resistance. It is similar to the hill-climbing search, except it looks for valleys instead of mountains. In general, hill climbing produces a solution with the least nodes visited, while least-cost finds a path that requires the least effort.

If multiple solutions exist for a problem, it is possible to generate those solutions without redundancy by removing solutions already found. One method of doing this is by the path-removal method, which removes nodes that form a current solution from the knowledge base and then attempts to find another. In essence, limbs are pruned from the tree. A second method, node removal, simply removes the last node in the current solution path and tries again.

The term optimal often means the best solution that can be found using one of the various search techniques. It may not actually be the best solution, which can only be truly found by an exhaustive search.

Many of these methods can be improved by employing heuristics to guide the search, or to reduce the search space. Heuristics are criteria, methods or principals for deciding which among several alternative courses of action promises to be the most effective in order to achieve some goal [71]. Heuristics are basically rules of thumb used to guide the search.

The heuristic repair method is a hill-climbing algorithm that attempts to minimize the number of conflicts. When a label which violates some constraints is picked for revision, the value which violates the least number of constraints is picked. Ties are resolved randomly. A CSP solution technique that can be classified as a randomized heuristic repair method was recently proposed by Minton *et al* in [74]. This technique keeps all variables instantiated at all times, and moves towards a better solution by changing values one variable at a time. Since all variables are always instantiated, *anytime* processing is possible, i.e, the algorithm can be stopped at any point in its processing to provide the best solution it has found so far. Anytime processing is not usually possible with constructive CSP techniques, since all of the variables are instantiated only when a complete solution has been found.

Minton *et al* called their algorithm the MIN-CONFLICTS heuristic, and applied it to some scheduling problems, most notably the scheduling of observations for the Hubble Space Telescope. Their algorithm is simple: find a CSP variable V_c in conflict with another variable and assign to V_c the value that minimizes the number of conflicts. Minton *et al* used this heuristic to solve the *n*-queens problem, i.e., the problem of placing *n* queens on an *n* x *n* board such that no two queens share the same column, row or diagonal. Prior to the MIN-CONFLICTS heuristic, CSP algorithms were unable to solve the problem with even 100 queens. Minton *et al* were able to solve the 1 million queens problem efficiently.

Iterative improvement such as the MIN-CONFLICTS heuristic do have drawbacks. First, they are not complete. However, for the size of the many problems being examined, it would not be possible to do a complete search in most instances. A more serious drawback is that the techniques can easily get caught in local minima. This is especially true for highly-constrained problems where the number of solutions is relatively small.

5.4 Constraint Satisfaction Optimization Problem (CSOP)

Relatively little work has been done in the CSOP by the research community [70]. A CSOP is defined as a CSP together with an optimization function f which maps every solution tuple to a numerical value:

$$(Z, D, C, f)$$

where (Z, D, C) is a CSP, and if S is the set of solution tuples of (Z, D, C), then

$$f\colon\ S \rightarrow \text{numerical value.}$$

The task in a CSOP is to find the solution tuple with the optimal (minimum or maximum) f-value, as determined by the application-dependent optimization function, f. Given a minimization problem with objective function f and feasible region S, a typical search algorithm requires that for each point $n_i \in S$, there is associated a neighbourhood $N(n_i) \subset S$. Given a current point $n_i \in S$, the set $N(n_i)$ is searched for a point n_{i+1} with $f(n_{i+1}) < f(n_i)$. If such a point exists, it becomes the new current solution, and the process is repeated. Otherwise, n_i is retained as a local optimum, and a set of feasible search states is generated, and each of them is locally improved within its neighbourhood.

How the search is carried out is important. Stochastic search methods can be useful for CSOPs, if the user is willing to sacrifice completeness for speed. These methods move from one point to the next in the search space nondeterministically, often guided by heuristics. The next move is partly determined by the outcome of the previous move.

Stochastic, or probabilistic, methods, do not offer an absolute guarantee of finding the global minimum with success. Rather, they tend to minimize the expected error in the approximation of the global minimum or to maximize the probability that the error is less than a pre-fixed bound. If X^* represents the global optimum, the probability that a feasible solution is within a distance, ε, of X^* approaches 1 as the sample size increases. If the sampling distribution is uniform over S and $f(X)$ is continuous, then an even stronger result holds: the sample point with the lowest function value converges to the global minimum with probability 1 (or almost surely). This can be stated more formally as follows [75]:

> If $f(X)$ is continuous, and y_N is the smallest function value found in a sample of size N, then y_N converges to the global minimum value (X^*) with probability 1

(or almost surely) with increasing N, i.e.,

$$p(\lim_{N \to \infty} y_N) = f(X^*)) = 1 \tag{5.1}$$

Because the global minimum is guaranteed in a probabilistic sense, stopping rules play a critical role in determining the trade-off between reliability and convergence rate (computation time). Stopping rules are basically heuristic.

Pure random search is perhaps the oldest and most primitive approach to solving an optimization problem. The objective function is evaluated at N points drawn at random from the feasible region, S, and the point where the function value is the lowest is a candidate for being the global minimum. While simple to implement, the technique suffers from poor reliability, particularly as the problem dimension increases.

A technique known as controlled random search (CRS) uses a clustering approach to locate the global minimum in unconstrained optimization problems. It is built around the premise that a predetermined number of values, which have been retained because of their low objective function values, will tend to cluster around local minima after a number of trials. The clusters are then graded, and the best cluster is identified. The best cluster is then taken as the global minimum.

In CRS, N uniformly selected points are selected over the feasible set S, and their respective function values determined and stored. At each iteration, a point P is obtained, and, if $f(P) \leq f(M)$, then M is replaced by P. Conversely, if $f(P) \geq f(M)$, the trial is discarded and a new point obtained. As the algorithm proceeds, the N stored points will tend to cluster around local minima. Eventually, the cluster associated with the global minimum will be the only one left.

In [75], Petit-Pas compared nine stochastic and deterministic optimization techniques. CRS proved superior for the thirty-seven test problems examined. Although scoring second in terms of reliability and robustness, CRS had the highest efficiency (level of effort required for parameter management as well as ease of changing test problems) and the fastest convergence rate (almost seven times faster than simulated annealing).

5.5 Modelling the Loss Minimization Problem as a CSOP

The loss minimization problem of a distribution system can be modeled as a CSOP. In this case, the variables are the position (open or closed) of switches in the distribution network. The positions of these switches are constrained by voltage drop, thermal and capacity limits. Finally, the optimization function, f, is simply to minimize I^2R losses in all sections, i.e.,

Minimize

$$\sum_{i=1}^{n} I_i^2 R_i x_i \tag{5.2}$$

where the variables are defined as follows:

R_i the resistance of line section i

I_i the current in line section i

x_i the state value of switch i, where

$$x_i = \begin{cases} 1, & \textit{if the switch is closed} \\ 0, & \textit{if the switch is open} \end{cases} \tag{5.3}$$

As noted earlier, minimizing losses is equivalent to minimizing the power injected at substation buses.

A stochastic technique can be employed to solve the resulting problem formulation. In essence, genetic algorithms and simulated annealing can be considered to be stochastic methods for solving CSOPs. However, both of these techniques require the setting of parameters (the selection of which may not be obvious) that have a major impact on the performance of the algorithms.

5.6 Summary

This chapter has introduced the definitions associated with CSPs and CSOPs, some of the algorithms available for their solution, and how the loss minimization through system reconfiguration problem can be modeled as a CSOP. Stochastic search methods have proven successful in solving CSOPs. In the next chapter, algorithms and data structures for network reconfiguration developed as part of this thesis are presented. Following that, a new algorithm for loss minimization called WATDIST is introduced. This is a stochastic search technique that has proven to be much more successful than earlier stochastic methods.

Chapter 6

Algorithms and Data Structures for Network Reconfiguraton

6.1 Introduction

Although most of the research effort into the reconfiguration problem has focused on the combinatorial optimization problem of finding the position of switches to minimize losses, a second critically-important issue is the automated representation of system data. Information passes continuously between the data structures and the optimization techniques used to minimize losses through reconfiguration. Efficient algorithms and data structures are needed for the optimization techniques to run in real-time.

In this chapter, the following items, and the data structures and algorithms developed in this thesis to represent them, are discussed:

a. a representation of the physical aspects of the system, including both static and dynamic information;

b. a method of building the bus admittance matrix from that representation;

c. a method of determining possible switching options for reconfiguration;

d. a method of ensuring that a switching option retains the radial network configuration; and,

e. a method of updating system data following reconfiguration.

Distribution system buses and sections are represented as adjacency and incidence matrices. Three combined algorithms are presented for reconfiguration. The first determines the admittance matrix for a feeder. The second algorithm performs a branch exchange, ensuring no loads are left disconnected from the system, and that the system's radial configuration is maintained. A third algorithm updates system data for the new configuration. When combined in a single program with an appropriate optimization technique , these algorithms will allow a utility to implement a robust and efficient loss minimization program through network reconfiguration.

6.2 Automated Representation of Network Data

There are two important issues associated with the reconfiguration problem, and these are represented conceptually in Figure 6.1. The first issue is where much of the research effort has been directed in the loss minimization through system reconfiguration problem: the combinatorial optimization problem of finding the position of switches to minimize losses in a timely manner given the large size of modern distribution systems. This is represented by the lower bubble in Figure 6.1.

A second, equally-important issue is that of the automated representation of system data. Information passes continuously between the data structures and the optimization techniques used to minimize losses through reconfiguration. How well these data structures represent the distribution system, and how well they can be employed for reconfiguration, is a determining factor in the speed of any optimization technique. Efficient algorithms and data structures are needed for the optimization techniques to run in real-time.

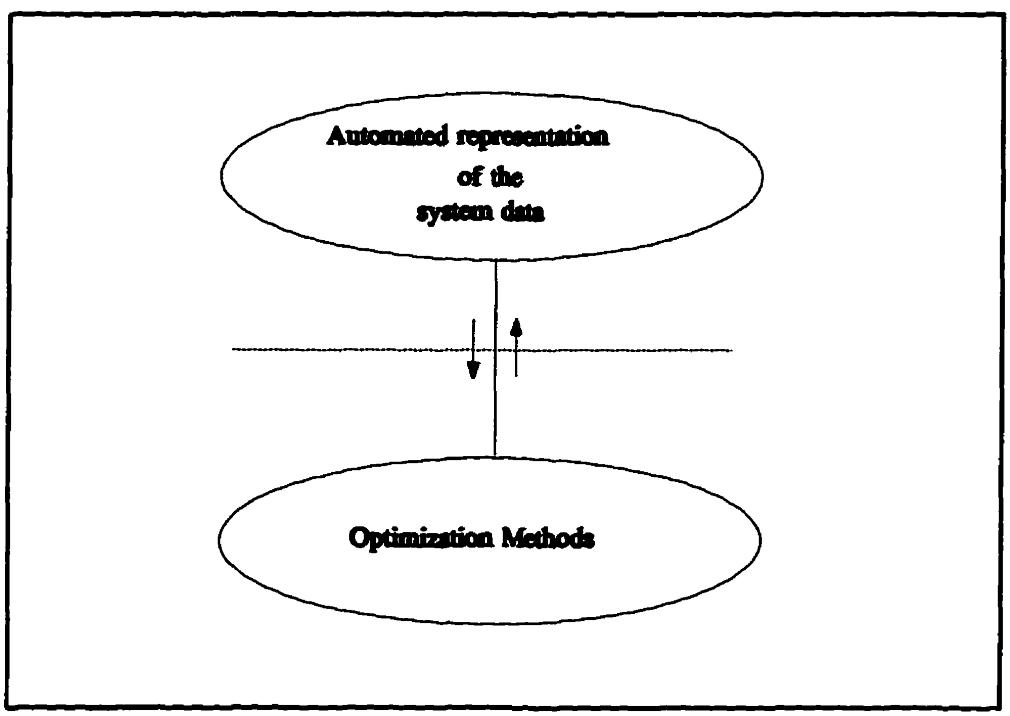


Figure 6.1. Conceptualization of issues involved in reconfiguration.

Castro *et al* [32] were perhaps the first to propose data structures for distribution feeder reconfiguration. They proposed a fixed data structure referred to as a switch table, which represented the actual distribution system configuration in tabular form. Data in the switch table was static, except for one column which indicated switch status (open or closed). The dynamic structure of the distribution network was represented as a multipath tree, and was generated from the switch table; however, how the structure was generated was not presented.

Cespedes [76] proposed a branch and node nomenclature that gave the branches a number that coincided with one of the end nodes of the same branch, allowing the representation of the network by a single vector. However, the representation presented only considered static networks, and would not be useful for the load re-allocation or reconfiguration problem. Borozan *et al* [31] proposed algorithms which improved the efficiency of the methods of Shirmohammadi and Hong [29]; however, they are not general enough to be used in other algorithms.

Most recently, Roytelman *et al* [24] proposed using branch and node tables to represent system topology. Although the method was outlined with an example, detailed algorithms were not presented. Thus, it is clear that the issue of the automated representation of network data has been largely ignored, and the area needs to be explored and developed.

The automated representation of system data (the upper half of Figure 6.1) is crucial to the success of any reconfiguration operation, and includes:

a. a representation of the physical aspects of the system;

b. a method of building the bus admittance matrix from that representation;

c. a method of determining possible switching options for reconfiguration;

d. a method of ensuring that a switching option retains the radial network configuration; and,

e. a method of updating system data following reconfiguration.

These issues are shown conceptually in Figure 6.2. Physical aspects of the system include both static and dynamic information. Static information includes section impedances, while dynamic information includes switch information (open or closed).

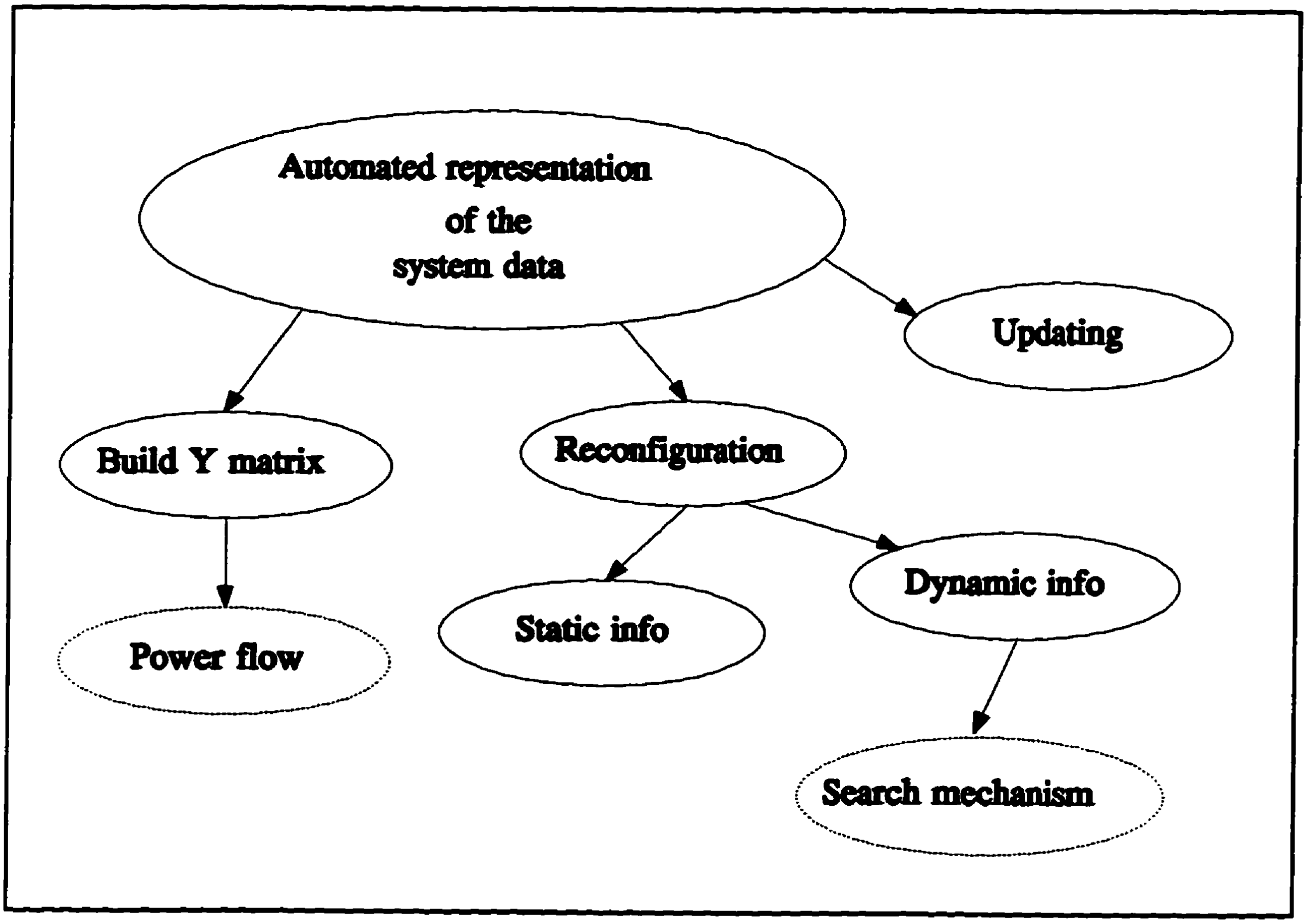


Figure 6.2. Elements of an automated representation of network data (items enclosed within dashes are not considered part of network data).

6.3 Distribution System Mathematical Model

The formation of a suitable mathematical model is the first step in the analysis of a distribution system. The model must describe the characteristics of each component in the system, as well as define the relations that govern the interconnection of these components. A convenient mathematical model is a network matrix equation.

Electrical networks are governed by Kirchoff's Laws, and hence the elements of a network matrix depend on the selection of currents and voltages, which are the independent variables in a network. Thus, the elements of a network matrix will be impedances or admittances. While this network matrix conveniently represents the electrical characteristics of the individual network components, it does not provide any information pertaining to network connections. Thus, the matrix network must be transformed into a matrix that describes the performance of the interconnected network.

To describe the geometrical structure of a network, network components can be replaced by single line segments, irrespective of the characteristics of the components. These line segments are called elements, and their terminals are called nodes. A node and an element are incident if the node is a terminal of the element. Nodes can be incident to one or more elements. Two nodes are adjacent if they share a common element. Nodes can be adjacent to one or more nodes.

A distribution system is made up of an interconnected set of elements whose performance is described by $n - 1$ independent nodal equations, where n is the number of nodes. In matrix notation, the performance equation is:

$$I_{BUS} = Y_{BUS} E_{BUS} \tag{6.1}$$

where E_{BUS} is the vector of bus voltages measured with respect to a reference bus

I_{BUS} is the vector of impressed bus currents

Y_{BUS} is the bus admittance matrix

The bus admittance matrix is formed by a simple and straight-forward procedure. A diagonal element, Y_{pp}, of this matrix is equal to the sum of the admittances connected to bus p. An off-diagonal element, Y_{pq}, is equal to the negative of the admittance of the network element connected bus p to bus q. Since the bus matrix is sparse, relatively few elements have to be calculated. As well, computer memory can be saved because it is not necessary to store the zero elements. In contrast, the bus impedance matrix is a full matrix with no zero elements. Consequently, it requires considerably more computer memory, and more time to compute each element.

The bus admittance matrix can be obtained from the inverse of the bus impedance matrix, if it is available. The majority of load flow programs use the bus admittance matrix, as the use of this approach is the most economical from the point of view of computer time and memory requirements [77].

In performing reconfiguration, it is necessary to revise system data before proceeding from case to case, and thus the network matrix must be modified. When the bus admittance matrix is used, it is necessary to recompute only those elements of the matrix that are associated with the terminals of the sections being changed. Relatively few matrix elements are associated with any one bus, and thus network changes can be effected simply and quickly. In contrast, in order to modify the bus impedance network, all network elements have to be modified.

6.4 Some graph theory

Distribution systems can be modelled as graphs. Consider the two-feeder network of Figure 6.3. Circled numbers represent buses, while uncircled letters represent feeder sections. The dashed section (section *d*) between buses 5 and 7 represents a currently-open sectionalizing switch. If this switch were to be closed, switch *a*, *b*, *g*, *f* or *e* would have to be opened to re-establish the network's radial configuration.

It is very easy to see how the network is connected from this figure; however, this representation is of no value for computer processing. However, the network can be represented conveniently and naturally by two matrices: an adjacency and an incidence matrix. It will be seen that the adjacency matrix provides information about bus connections, and the incidence matrix contains branch (or section) data.

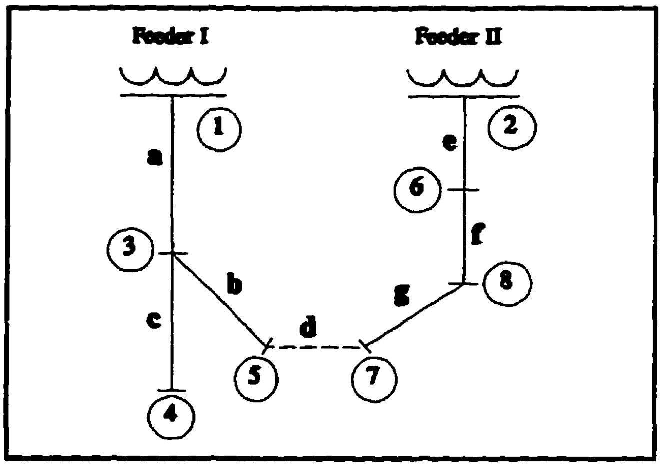


Figure 6.3. A small, two-feeder distribution system. The circled numbers are buses, and the letters represent sections.

If G is a graph with n vertices and e edges, the *adjacency matrix*, $\mathbf{X} = [x_{ij}]$, can be defined as an n x n matrix with elements as follows [78]:

$$x_{ij} = 1, \quad \text{if there is an edge between the } i\text{th and } j\text{th vertices; and,}$$
$$= 0, \quad \text{otherwise.}$$

The adjacency matrix, **X**, for the network of Figure 6.3 is shown in Equation (6.2). This is a binary symmetric matrix. A count of the number of 1's in a row (or column) indicates how many edges are incident on a vertex. Although no such rows or columns are present, a row or column of all zeroes would indicate an isolated vertex.

Because it is the reconfiguration problem that is of interest, it is useful to have a method to indicate *possible* connections. This can be done by substituting a -1 for the 0 where applicable in Equation (6.2) (for buses 5 and 7 in this case), and this is shown in Equation (6.3).

It can be seen that the matrix of Equation (6.3) is a sparse matrix, and hence a more compact representation is required, and thus the adjacency matrix is rewritten as a series of vectors, as shown in Equation (6.4). The zero at the end of each vector is simply a marker to indicate there are no further elements in the vector.

$$\begin{array}{c c} & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} & \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Equation (6.2) Adjacency matrix for the network of Figure 1.

$$\begin{array}{c c} & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} & \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Equation (6.3) Adjacency matrix, with *-1* to indicate *possible* connections.

$$\begin{aligned} X[1] &= [3,0] \\ X[2] &= [6,0] \\ X[3] &= [1,4,5,0] \\ X[4] &= [3,0] \\ X[5] &= [3,-7,0] \\ X[6] &= [2,8,0] \\ X[7] &= [-5,8,0] \\ X[8] &= [6,7,0] \end{aligned}$$

Equation (6.4) Adjacency matrix written as a series of vectors.

It is also possible to define an *incident matrix*, *A(G)*, for graph *G* with *n* vertices and *e* edges. *A(G)* is an *n* x *n* matrix with elements [78]:

a_{ij} = 1, if the jth edge e_j is incident on the ith vertex v_i; and,

= 0, otherwise.

The incidence matrix for a network is not unique and depends upon the orientation of the graph and the selection of branches and nodes. The incident matrix for the network of Figure 6.3 can be written as a series of vectors, as shown in Equation (6.5) for the system of Figure 6.3. Both $A(G)$ and $X(G)$ contain all of the information about a radial network.

$$
\begin{aligned}
A[1] &= [a,0] \\
A[2] &= [e,0] \\
A[3] &= [a,b,c,0] \\
A[4] &= [c,0] \\
A[5] &= [b,-d,0] \\
A[6] &= [e,f,0] \\
A[7] &= [-d,g,0] \\
A[8] &= [f,g,0]
\end{aligned}
$$

Equation (6.5) Incident matrix written as a set of vectors.

6.5 Algorithms

Before discussing the algorithms and looking at an example, it is first necessary to examine the data structures used within the algorithms. There are primarily two data structures used: one containing section data, and the other bus data. Section data includes which bus is to the "left" or "right" of a section (as viewed on the on-line diagram, and could also be considered as the upstream and downstream buses), the status of sectionalizing switches in the section (closed or open) and section impedance and admittance. Bus data includes the complex load (real and reactive power), which substation bus is supplying the load bus, and the adjacent and incident vector for each bus. Two C++ data structures were created to represent section and bus data, and they are illustrated in Figure 6.4.

```
// section data

typedef struct
        {int left;
         int right;
         int status;
         complex impedance;
         complex admittance;
        } section_table;

// bus data

typedef struct
        {complex load;
         int parent_bus_root;
         int adjacent[4];
         int incident[4];
        }  bus_table;
```

Figure 6.4. Data structures used to represent section and bus data.

Three arrays are dynamically allocated: an array *section* of type *section_table* of size equal to the number of sections, an array *bus* of type *bus_table* of size equal to the number of buses, and a three-dimensional array *y* of type complex, where the first dimension is the number of feeders, and the remaining two dimensions are equal to the number of buses.

To illustrate the use of three algorithms proposed in this paper, it is useful to present an example to show how the algorithms can be used as they are discussed. We will consider the case where feeder 2 is overloaded at its substation in the two-feeder system of Figure 6.3, and we wish to transfer part of its load to feeder 1. The steps to follow in such a situation, as well as the algorithms to use, are shown in pseudo-code in Figure 6.5.

1. use Algorithm 1 to build a graph of each feeder, and determine the associated Y-admittance matrix;
2. carry out a load flow;
3. perform a single branch exchange;
    (1) select two feeders to connect together by closing an open switch;
    (2) using Algorithm 2, build a path between the two now-connected substation buses;
    (3) to restore the radial configuration select a switch on the path between the two substation buses to open;
4. use Algorithm 1 to build a graph of each feeder, and determine the associated Y-admittance matrix;
5. carry out a load flow;
6. update system data using Algorithm 3;

Figure 6.5. Illustration example to relieve overload at substation bus 2.

6.5.1 Algorithm 1 - Graph Builder

After reading the system data, including section data and switch status, as well as bus data and the system adjacency and incidence matrices, *Algorithm 1 - Graph Builder* (shown as Figure 6.8 at the end of the chapter) forms the graph that represents each feeder, and builds its bus admittance matrix. The algorithm builds the graph by treating each feeder as an *n*-ary tree, carrying out a preorder, or depth-first, traversal, and taking the substation bus as the root node of the tree. As each bus is visited, it is assigned a local bus number. For the two-feeder network of Figure 6.3, two graphs would be created, represented internally as (1) a 4-bus system with node 1 corresponding to Bus 1, node 2 corresponding to Bus 3, node 3 to Bus 4, and node 4 to Bus 5; and, (2) a 4-bus system with node 1 corresponding to Bus 2, node 2 to Bus 6, node 3 to Bus 8, and node 4 to Bus 7.

The admittance matrix is used for load-flow studies to determine bus voltages and section currents given the scheduled real and reactive power at each bus. In general, the bus admittance matrix is formed using the admittance of all elements, as follows [81]:

$$Y_{kk} = \sum_{\substack{j=0 \\ j \neq k}}^{N} y_{kj} \qquad k = 1, 2, \ldots, N \text{ diagonals}$$
$$Y_{kj} = Y_{jk} = -y_{jk} \qquad \text{nondiagonals} \tag{6.6}$$

Given the bus loads and the bus admittance matrix, a load-flow study can then be carried out using any load-flow technique specified by the user (in this thesis, the Gauss-Seidel method is used - it will be discussed in detail in the next chapter).

6.5.2 Algorithm 2 - Path Builder

Algorithm 2 - Path Builder connects two feeders together momentarily, until the decision is made as to which switch to open to re-establish the radial configuration of the network. The algorithm is shown at the end of the chapter as Figure 6.9. Algorithm 2 builds a path by treating each feeder as an n-ary tree, carrying out a preorder, or depth-first, traversal, and taking the substation bus as the root node of the tree. Each time a leaf node is reached (i.e., no left or right subtree, and hence a dead-end), the algorithm backtracks and tries a new path. For the two-feeder network of Figure 6.3, the algorithm would visit the buses in the following order: 1, 3, 4, 3, 5, 7, 8, 6, 2.

Using this approach, Algorithm 2 ensures that opening candidate switches does not create any loops within the network, and that all buses remain connected to a feeder. It should be noted that the algorithms presented in this chapter will *not* determine which switch to close or open - another algorithm or operator intervention is required. Algorithm 2 closes the switch it is directed to close, and determines the path between the two substation buses by providing a list of *possible* switches that can be opened to restore a network to its radial configuration.

Consider the situation where it has been determined that feeder 2 is overloaded. Sectionalizing switch *d* can be closed, and then Algorithm 2 used to determine the path between substation buses 1 and 2, i.e., *a, b, d, g, f, e* (see Figure 6.6). Opening any one of these switches will return the system to its radial configuration. It is left to the user's discretion to select an optimization technique to determine which switch to open, as the algorithms presented in this chapter do not perform that function.

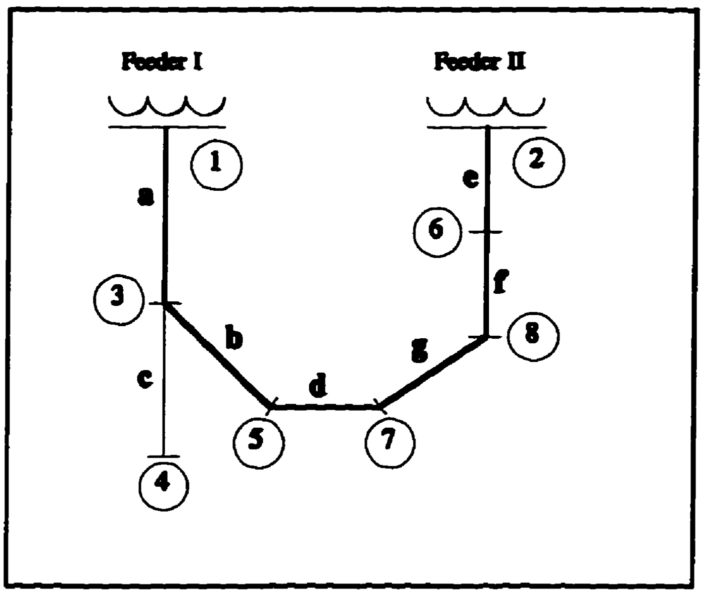


Figure 6.6. Path determined by Algorithm 2 (shown by heavy lines).

6.5.3 Algorithm 3 - System Update

Algorithm 3 - System Update is used to update system data following reconfiguration. The algorithm is shown at the end of the chapter as Figure 6.10. Using the example outlined in the previous paragraph, assume that an optimization technique (or operator) has determined that sectionalizing switch *g* should be opened. The resulting network would be as it appears in Figure 6.7. Algorithm 3 would update the adjacency and incident matrices to the values shown in Equation (7) and Equation (8), respectively.

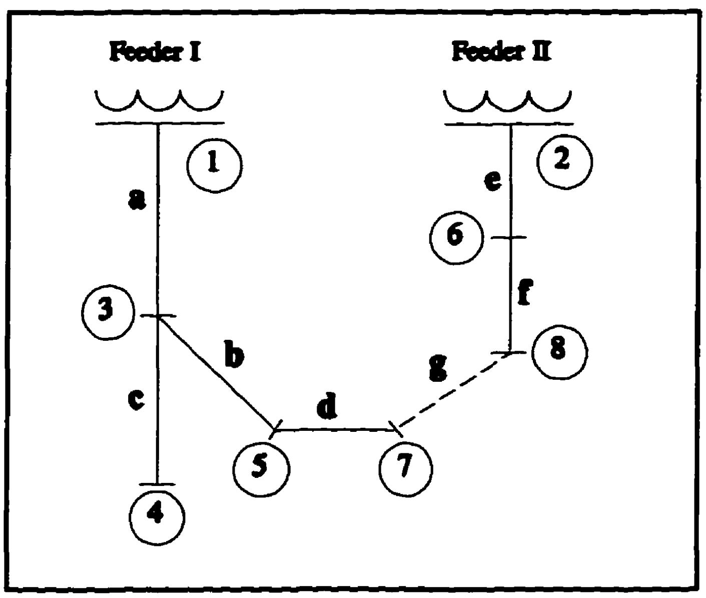


Figure 6.7. Final system configuration.

$$
\begin{aligned}
X[1] &= [3,0] \\
X[2] &= [6,0] \\
X[3] &= [1,4,5,0] \\
X[4] &= [3,0] \\
X[5] &= [3,7,0] \\
X[6] &= [2,8,0] \\
X[7] &= [5,-8,0] \\
X[8] &= [6,-7,0]
\end{aligned}
$$

Equation (7) Updated adjacency matrix.

$$
\begin{aligned}
A[1] &= [a,0] \\
A[2] &= [e,0] \\
A[3] &= [a,b,c,0] \\
A[4] &= [c,0] \\
A[5] &= [b,d,0] \\
A[6] &= [e,f,0] \\
A[7] &= [d,-g,0] \\
A[8] &= [f,-g,0]
\end{aligned}
$$

Equation (8) Updated incident matrix.

6.6 Summary

To date, most of the research effort directed at reducing system distribution system losses through reconfiguration has focused on the problem of which sectionalizing switches to open and close. The issues of building the bus admittance matrix, creating links between feeders and updating system data following configuration have been largely ignored.

This chapter has presented three algorithms that can be readily used for system configuration, and introduced data structures to represent system data. The algorithms ensure that the radial network topology commonly used in distribution networks is retained, while at the same time ensuring that no loads are left disconnected from the remainder of the system.

Although no optimization techniques *per se* have been introduced, the data structures and algorithms presented will be used as the framework for the optimization technique to be used in succeeding chapters. These algorithms should also provide researchers with a framework to explore new methods for reducing losses, and, for utilities, a framework to undertake system reconfiguration studies.

Attention is now turned to the optimization technique developed in this thesis, and presented as an algorithm called WatDist. This algorithm was developed specifically for this thesis research, and, it will be seen, is a very flexible and efficient method for reducing losses in power distribution systems.

```
Step 1:  set:    ▪ v = x (x is the root-node where the search begins);
                 ▪ i = 0;
                 ▪ stack = x;

Step 2:  set:    ▸ i = i+1
                 ▸ num(v) = i
                 ▸ ynum(i) = v (not required, but used later on)

Step 3:  from v, look for a node that is unvisited, by examining adjacency list values, adjacent[v,0..4]

                 if a node is found, then
                     ▸ push node value to stack;
                     ▸ set v = node value;
                     ▸ go to Step 2;
                 else,
                     pop v from the stack
                     if v = x,
                          the search is complete;
                               go to Step 4
                          else,
                               go to Step 3.

Step 4:
    for (i=1 to the number of buses connected to the feeder)
         {ival=ynum[feeder_number][i];
          k=1;
          neighbour=bus[ival].adjacent[1];

          while (neighbour != 0)
              {if (neighbour > 0)
                   {y[feeder_number][i][i] =
                        y[feeder_number][i][i] + section[(bus[ival].incident[k])].admittance;
                    entry=num[feeder_number][neighbour];
                    y[feeder_number][i][entry] = -section[(bus[ival].incident[k])].admittance;
                   }
                   k=k+1;
                   neighbour=bus[ival].adjacent[k];
              } // end while
    } // end for
```

Figure 6.8. *Algorithm 1: Graph Builder.* Forms the feeder graph and the associated Y-admittance matrix.

```
Set:    endbus1 = x (where x is the root node);
        v=x;
        count=-1;
        not_done=TRUE;
Push 1 to stack;
Push v to stack;

while (not_done)
    {i=v;
    j=1;
     while (bus[i].adjacent[j]==stack OR bus[i].adjacent[j] <= 0)
        {if (bus[i].adjacent[j]==stack) j=j+1;
         if (bus[i].adjacent[j]<0) j=j+1;
         if (bus[i].adjacent[j] == 0)
            {while (bus[i].adjacent[j] == 0)
                {pop i from stack;
                 v=i;
                 pop j from stack;
                 count=count-1;
                }
             j=j+1;
            }
        }

    push j to stack;
    push v to stack;
    count=count+1;
    link[count]=bus[i].incident[j];
    v=bus[i].adjacent[j];

    for (k=1;k<=NUMBER_OF_FEEDERS;k++)
        {if (bus[i].adjacent[j]==feeder_root_node[k] AND bus[i].adjacent[i]!=endbus1)
            {endbus2=feeder_root_node[k];
             not_done=FALSE;
            }
        }
    }
```

Figure 6.9. *Algorithm 2: Path Builder.* Creation of link between two feeders.

```
Set v=root_bus;
Set not_done=TRUE;
Push 1 to stack;
Push v to stack;

while (not_done)
    {i=v;
     j=1;

     while (bus[i].adjacent[j]==stack[stack_ptr] OR  bus[i].adjacent[j] <= 0 AND not_done)
        {if (bus[i].adjacent[j]==stack) j=j+1;
         if (bus[i].adjacent[j] <= 0)
            {while (bus[i].adjacent[j] <= 0)
                {pop i off stack;
                 v=i;
                 pop j off stack;
                 if  ((stack==root_bus) AND (bus[i].adjacent[j+1]==0)) not_done=FALSE;
                }
             j=j+1;
            }
        }

    if (not_done)
        {push j to stack;
         push v to stack;
         x=bus[i].adjacent[j];
         bus_root[x]=root_bus;
         v=bus[i].adjacent[j];
        }
}
```

Figure 6.10. *Algorithm 3: System Update.*

Chapter 7

WatDist: An Algorithm for Network Reconfiguration

7.1 Introduction

In this chapter, WatDist is introduced. WatDist is an algorithm developed as part of this thesis to solve the problem of loss minimization through network reconfiguration. WatDist models the loss minimization problem as a CSOP, and solves it stochastically. The steps in the algorithm are explained, and simulation results are presented.

7.2 WatDist

The loss minimization problem through system reconfiguration is a constraint satisfaction optimization problem (CSOP). Historically, utilities have only focused on finding a feasible solution configuration with as little search effort as possible. This task is known as *satisficing* [71]. The difference between the optimization problem and its satisficing counterpart is substantial: finding a feasible configuration is usually trivial, but finding a configuration that minimizes losses and satisfies all constraints is NP-hard. Traditionally, the property of NP-hardness is used as a measure of what separates tasks that can be solved computationally with realistic resources from those than cannot [79]. As noted earlier, the optimality requirement is often relaxed to reduce the search effort, and thus a "semi-optimization" problem is the result, i.e., a balance is made between the quality of the solution found and the cost of searching for a solution.

To solve the loss minimization problem, the following items are required to allow its solution on a computer:

a. a symbolic structure or code to represent each candidate solution in the solution space (a database);

b. tools to transform the encoding of one solution to that of another in order to scan the search space systematically (operators or production rules); and,

c. a method of scheduling these transformations so as to produce the desired solution as quickly as possible (a control strategy).

The data structures and algorithms presented in Chapter 5 fulfil requirement (a) above, and also satisfy requirement (b), in that they are able to reconfigure a system, and hence transform the system from one solution to another. The final requirement, i.e., that of a control strategy, will now be examined.

It is desirable for a control strategy to have two features:

a. to not leave any stone unturned (unless it is certain there is nothing under it).

b. to not turn any stone more than once.

The first requirement guarantees that the desired solution is not missed if one exists. The second prevents inefficiency through repetitious computations.

The algorithm developed for this thesis is similar to the randomized heuristic repair method proposed by Minton *et al* in [74] (and discussed in Chapter 5). It is an iterative technique based on the branch exchange method, i.e., the transfer of loads (or branches) between feeders. Similar techniques have been used to design gas pipelines and computer networks, most notably the U.S. Department of Defence's ARPA network of the early 1970s [80].

The WatDist algorithm is presented in Figure 7.1. Given an initial system configuration, WatDist uses Algorithm 1 (of Chapter 6) to build a graph for each feeder, and determines the associated bus admittance matrices. The power injected at substation buses is then determined, providing an initial base for comparison of solutions.

The algorithm then changes the assignment of two of the switch positions, and determines the power injected at substation buses. If the power of the new configuration is less than that of the current solution, the new configuration becomes the current solution. Such changes are repeated until a preset maximum number of changes (MAX-FLIPS) is reached. This process is repeated as needed up to a maximum of MAX-TRIES times.

```
Step 1:  assume an initial switch configuration;
Step 2:  use Algorithm 1 to build feeder graphs, and determine the associated Y-admittance matrices;
Step 3:  determine the power injected at substation buses;
Step 4:  initialize two counters:  (1) generations; and, (2) no_change_count.

Step 5:
         while (iterations < MAX-TRIES)
                 increment iterations counter;
                 if (no_change_count < MAX-FLIPS)
                         then
                                 perform a single branch exchange (see below);
                                 increment no_change_count counter;
                         else
                                 perform 2 or 3 branch exchanges (see below);
                                 reset no_change_count counter;
                 if (switching combination has already been examined) go to Step 5
Step 5a:
                 use Algorithm 1 to build feeder graphs and associated Y-admittance matrices;
                 determine power injected at substation buses;
                 if there are constraint violations
                                 then
                                         transfer some of the load on the feeder with constraint
                                                 violations to the other feeder
                                         go to Step 5a
                 if total power injected for new configuration is less than that of current solution
                                 then
                                         accept this new configuration as the current solution;
                                         update system data using Algorithm 3;
                                         reset no_change_counter to zero;

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Branch Exchange

1.       from the open_switch_list, select at random a switch to close;
2.       using Algorithm 2, build a path between the two now-connected substation buses;
3.       at random, select a switch on the path between the two substation buses to open;
```

Figure 7.1. The WatDist algorithm.

The WatDist procedure requires the settings of the two parameters - MAX-FLIPS and MAX-TRIES - which determine, respectively, how many branch exchanges the procedure will attempt before giving up and starting in a new part of the search space, and how many times this search can be restarted before quitting. As a rough guideline, setting MAX-FLIPS to the number of switches is sufficient. The purpose of MAX-FLIPS is to ensure the algorithm does not get trapped in a local minimum - if there has been no improvement in the solution after MAX-FLIPS iterations, the algorithm moves to a new region of the search space. The setting of MAX-TRIES will generally be determined by the amount of time that one wants to spend looking for an assignment, which in turn is determined by when reconfiguration is to take place.

A currently-open switch is selected at random to be closed. Algorithm 2 (from Chapter 6) is then used to build a path between the two substations that are now connected. Next, at random, a switch in the path is opened, and thus a branch exchange takes place.

After carrying out a branch exchange, the algorithm determines if the current network configuration has been examined previously. Previous configurations are stored in a pattern matrix. Although this step could be eliminated, it does speed up the process of searching for the optimum configuration, as much of the processing time involved in the algorithm is taken up by load flows. By eliminating the need to re-perform load flows on configurations already examined, WatDist is able to find an optimum solution much faster. This step is also required to prevent the algorithm from oscillating between two configurations following a load transfer between two feeders (described below).

If the configuration has not been previously examined, graphs for the two feeders are constructed (using Algorithm 1), and their bus-admittance matrices determined. A load flow is carried out for the two affected feeders (because the remaining feeders are not involved in the branch exchange, there is no need to carry out load flows for them). Although WatDist carries out a number of load flows during its execution, these load flows are on a very small subset of the entire distribution network, and thus can be carried out very quickly.

With the load flow complete, and if there are no constraint violations, it is possible to determine the total power injected at substation buses. If the total power injected is less than that of the current solution, the new configuration becomes the current solution. Algorithm 3 (from Chapter 6) then updates the system data.

If there have been constraint violations, an attempt is made to transfer some of the load on the feeder with the constraint violations to the other feeder involved in the branch exchange. This is accomplished by closing the switch that was opened at random following the building of the path between the two substations, and opening the switch to the left or right of that switch, as needed, to relieve the load.

WatDist explores potential solutions that are "close" to the one currently being considered. Specifically, we explore the set of assignments that differ from the current set on only two variables. Another feature of WatDist is that the variable whose assignment is to be changed is chosen at random from those that could give an equally good improvement. Such non-determinism makes it very unlikely that the algorithm makes the same sequence of changes over and over again.

The algorithm uses a generate-and-test process, creating new solutions rather than eliminating objects from pre-existing sets. The latter approach might be thought of as an operations research approach that begins with a large set of potential solutions that eventually are trimmed down to a single solution.

7.3 Load flow

Because load flow calculations consume much of the processing time for WatDist, it is important to examine the issue. Load flow calculations provide power flows and voltages of a network for specified terminal or bus conditions. There are four quantities associated with each bus: real and reactive power $(P_p - jQ_p)$, and voltage magnitude and phase angle ($E_p = E \angle \phi$). For each bus, two of the four quantities are specified. Distribution systems typically do not have any generation of power, and hence most buses will be load buses where the real and reactive power are specified.

The solution of the load flow problem is initiated by assuming voltages for all buses except the substation bus (where the voltage is specified and remains fixed). Then, currents can be calculated for all buses from the bus loading equation [77]:

$$I_p = \frac{P_p - jQ_p}{E_p^*}, \qquad p = 1,2,\ldots,n \tag{7.1}$$

where n is the number of buses in the network. The performance of the network is given by

$$I_{BUS} = Y_{BUS} E_{BUS} \tag{7.2}$$

Thus, a set of n - 1 simultaneous equations can be written in the form

$$E_p = \frac{1}{Y_{pp}}\left(I_p - \sum_{\substack{q=1 \\ q \neq p}}^{n} Y_{pq}E_q\right) \tag{7.3}$$

Substituting Equation (7.1) into Equation (7.3) gives

$$E_p = \frac{1}{Y_{pp}}\left(\frac{P_p - jQ_p}{E_p^*} - \sum_{\substack{q=1 \\ q \neq p}}^{n} Y_{pq}E_q\right) \tag{7.4}$$

which involves only bus voltages as variables. By formulating the load flow problem in this manner, the resulting set of nonlinear equations can be solved iteratively until the change in bus voltages is smaller than some specified tolerance, ε.

To reduce the computing time, several arithmetic operations can be performed in advance of initiating the iterative calculations. These operations involve only constant quantities (admittance matrix values and bus loads). By letting

$$\frac{1}{Y_{pp}} = A_p \tag{7.5}$$

equation (7.5) can be written

$$E_p = \frac{(P_p - jQ_p)A_p}{E_p^*} - \sum_{\substack{q=1 \\ q \neq p}}^{n} Y_{pq}A_pE_q \tag{7.6}$$

Letting

$$(P_p - jQ_p)L_p = B_p \tag{7.7}$$

and

$$Y_{pq}L_p = C_{pq} \tag{7.8}$$

equation (7.6) can be written as

$$E_p = \frac{B_p}{E_p^*} - \sum_{\substack{q=1 \\ q \neq p}}^{n} C_{pq}E_q \tag{7.9}$$

where B and C are simply constants defined in equations (7.7) and (7.8), respectively.

Thus, bus voltages are solved iteratively, based on the most recent estimate of the voltage. This method of load flow solution is known as the Gauss-Seidel technique, and is the method used in this thesis.

The correctness of the load flow algorithm, and its implementation in software, were verified by comparing the results obtained by the algorithm with results obtained by other researchers. Table 7.1 (on page 103) provides the bus voltages obtained using the WatDist algorithm for a distribution system having 3 feeders and 16 buses. These voltages are identical to the results presented by Civanlar *et al* in reference [28] (although there are minor differences of less than 1° in the phase angle).

The Gauss-Seidel technique is an iterative technique, and may take a considerable number of iterations to reach convergence (or may fail to converge). To overcome this problem in the past, some researchers have turned to D.C. load flows (for example, [2]). However, it was felt that the introduction of such approximations was both unnecessary, in that for the two systems examined in this thesis, convergence and speed of convergence were not significant issues. It was also felt that approximations were undesirable, as they would reduce the accuracy of the results. Finally, load flows at each step in the algorithm are carried out only for the feeders that have been affected by a switching operation, and not the entire system. Thus, a load flow is only needed for a small subset of the network.

7.3.1 Load sensitivity to voltage

Prior to the start of the Athens Area Control Experiment (AACE), it was thought that transferring loads between feeders would see an increase in the power injected at the substations as a result of increased losses (due to the use of telescoping feeder sections), but the reverse occurred [21]. Further analysis showed that initial predictions were based on a constant power model. However, simulations revealed that a constant current model most accurately models most real-life loads, due to load sensitivity to voltage. AACE proposed a model that represented each load as a parallel connection of constant-power, constant-current and constant-impedance components. The model is shown in Figure 7.2.

If $P_L(V)$ is the real power consumed by some load as a function of the applied voltage, V, a Taylor series expansion of $P_L(V)$ can be written as

$$P_L(V) = P_L(V_0) + P_L'(V_0)(V - V_0) + \frac{1}{2}P_L(V_0)''(V - V_0)^2 + HOT \qquad (7.10)$$

where V_0 is the nominal voltage and HOT stands for "higher-order terms." Since distribution voltages are typically within 5% of their nominal values, the HOT can be ignored, allowing Equation (7.10) to be written as

$$P_L(V) = \left[P_L(V_0) - V_0P_L'(V_0) + \frac{1}{2}V_0^2P_L''(V_0)\right] + \left[P_L'(V_0) - V_0P_L''(V_0)\right]V + \left[\frac{1}{2}P_L''(V_0)\right]V^2 \tag{7.11}$$

Defining

$$\begin{aligned} P &= P_L(V_0) - V_0P_L'(V_0) + \frac{1}{2}V_0^2P_L''(V_0) \\ I_r &= P_L'(V_0) - V_0P_L''(V_0) \\ R^{-1} &= \frac{1}{2}P_L''(V_0) \end{aligned} \tag{7.12}$$

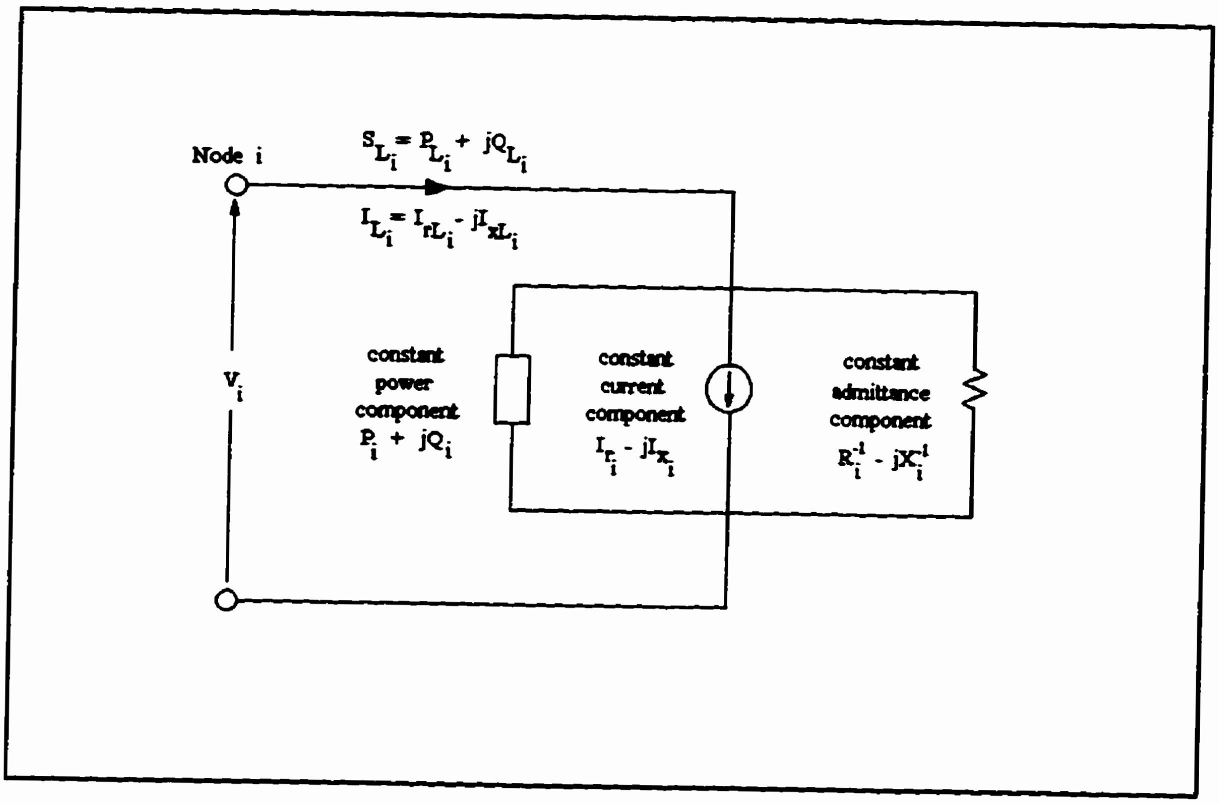


Figure 7.2. Load model at node *i*.

Then,

$$P_L(V) = P + I_r V + R^{-1} V^2 \tag{7.13}$$

which can be interpreted as the power consumed by a parallel connection of a constant-power sink, P, a constant-current sink, I_r, and a constant resistance, R. Similar expressions can be written to model reactive power.

It is difficult to determine the model parameters without knowing the sensitivities of each load type. For the AACE, it was assumed that each load component (constant power, constant current and constance impedance) contributed a given fraction of the total load at nominal voltage, i.e.,

$$\begin{aligned} \epsilon_P &= \frac{P}{P_L(V_0)} \\ \epsilon_{I_r} &= \frac{I_r V_0}{P_L(V_0)} \\ \epsilon_R &= \frac{R^{-1} V_0^2}{P_L(V_0)} \end{aligned} \tag{7.14}$$

where

$$\epsilon_P + \epsilon_{I_r} + \epsilon_R = 1 \tag{7.15}$$

The advantage of this method is that a load can be made to assume any voltage dependence by varying its component contribution parameters (ε_P, ε_{I_r}, ε_R). The disadvantage is that model parameters have to be specified by distribution system engineers, and this process may not be evident.

In the WatDist algorithm, a constant power load model is used. This is perhaps the most common load model in use today by researchers and utilities alike. However, as discussed earlier, it may not be the most accurate for a specific distribution system, which may require the more accurate model presented above. The constant power load model does, however, produce the most conservative results when calculating line losses, and thus serves as a lower bound on the losses a utility might see in practice.

The inclusion of the above load model in WatDist could easily be carried out by modifying the load flow portion of the program. However, without specific operational experience to determine the component contribution parameters, it is difficult to offer a general-purpose load model, and hence the constant power load model is used.

7.4 Simulation Results

To demonstrate the effectiveness of the WatDist algorithm in solving the network reconfiguration for loss minimization problem, the system shown in Figure 7.2 was used. This system was originally used by Civanlar *et al* [28] to illustrate the performance of their algorithm. In this system, 3 feeders supply 16 load buses. There are also 16 sections, each with a sectionalizing switch. Three of these switches must remain open to ensure the system maintains its radial configuration. System data is shown in Table 7.1.

For this system, there are 2^{16} (65,536) possible switching combinations, or configurations. However, of those, only 189 are valid configurations, as the remainder leave loads disconnected. For example, if switch 20, 25 and 33 are open, none of the buses are supplied. The number of valid configurations was determined using Algorithm 1: Graph Builder, from Chapter 6.

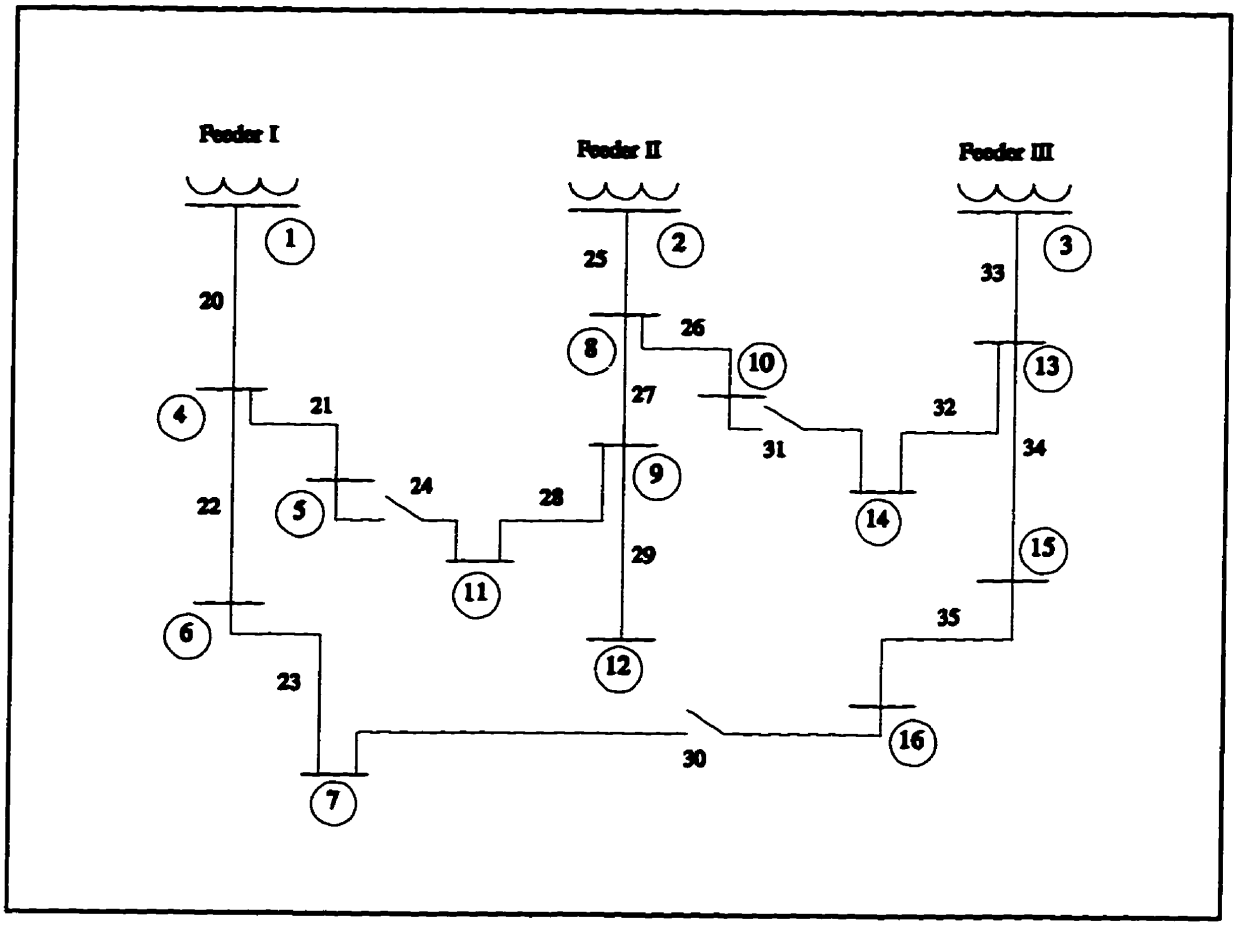


Figure 7.3. Test distribution system with 3 feeders and 16 sectionalizing switches, 3 of which are open (circled numbers represent buses; uncircled numbers are switches).

Start/End Bus	Section Data (p.u.)		End Bus Load		End Bus Capacitance	End Bus Voltage
	Resistance	Reactance	MW	MVAR	MVAR	p.u.
1-4	0.08	0.10	2.0	1.6		0.991/-1.564
4-5	0.08	0.11	3.0	1.5	1.1	0.988/-1.561
4-6	0.09	0.18	2.0	0.8	1.2	0.986/-1.559
6-7	0.04	0.04	1.5	1.2		0.985/-1.558
2-8	0.11	0.11	4.0	2.7		0.979/-1.557
8-9	0.08	0.11	5.0	3.0	1.2	0.971/-1.545
8-10	0.11	0.11	1.0	0.9		0.971/-1.544
9-11	0.11	0.11	0.6	0.1	0.6	0.969/-1.539
9-12	0.08	0.11	4.5	2.0	3.7	0.977/-1.557
3-13	0.11	0.11	1.0	0.9		0.994/-1.565
13-14	0.09	0.12	1.0	0.7	1.8	0.995/-1.563
13-15	0.08	0.11	1.0	0.9		0.992/-1.562
15-16	0.04	0.04	2.1	1.0	1.8	0.991/-1.560
5-11	0.04	0.04				
10-14	0.04	0.04				
7-16	0.09	0.12				

Table 7.1 Sample system data (from [28]). System base is 100 MVA, 230 kV.

The number of configurations can be further reduced to 59 if the requirement is made that each of the substation nodes (buses 1, 2 and 3) must be connected. This is a realistic requirement that would probably occur in an operational context, as most utilities would use all of their substations to provide power.

A power flow was carried out for each of the valid 189 configurations, and it was found that, for this system, the optimal configuration to minimize losses has switches 26, 28 and 30 open. This is also the optimal configuration found by Civanlar *et al* [28], and validates the algorithms of Chapter 6 in constructing the Y admittance matrix, as well as the load flow. It should be noted that finding this result is very sensitive to the specification of the bus voltage convergence used in the power flow. The stopping criteria used for the power flow was the difference in bus voltages between iterations, and this had to be set to quite a small value (i.e., $\varepsilon = 10^{-6}$) to obtain these results. Larger values of ε allowed the load flow to converge to incorrect bus voltages.

Early in the research, it was apparent that three approaches were possible for performing a branch exchange: (1) open a closed-switch at random and close another at random; (2) examine only switches adjacent to a currently-open switch; or, (3) close a switch that is currently open, and then open a switch between the two substations that are now connected as a result of the closed switch.

When the problem was first examined, the first approach was used, i.e., a branch exchange was carried out by opening a switch at random, and then closing another at random. This approach is simple to implement in computer code, as it is only necessary to generate two random numbers, confirm that the system is still radial, and carry out a new power flow to determine if losses have been reduced.

However, this approach was not successful. As noted above, there are only 189 possible valid configurations, leaving 65,347 invalid ones (sections disconnected). Thus, the probability of randomly generating an invalid solution is much higher than that of generating a valid one, and a great deal of effort is wasted in looking at invalid solutions. This is similar to what was reported in [56] by Nara *et al*, and is probably the main reason for the excessive computation time required for their technique.

Sarfi [52] noted a similar problem. Sarfi's method started with opening a switch, and then identified a switch as a candidate for closure. However, the algorithm then had to determine if radiality was maintained by the switching operation. Experimentally, Sarfi discovered that "most of the switching operations could not be employed to preserve a radial network." The method was able to locate "an optimal solution, but possibly infeasible solution."

The second approach - that of only examining switches adjacent to a currently-open switch - was felt to be too limiting, and ran the risk of not finding the global optimum.

The third approach - close a switch that is currently open, and then open a switch - was much more successful than the first approach, although much more difficult to implement. At random, an open switch was closed, resulting in two substations now being joined together. To restore the radial configuration of the two feeders, a section was opened at random. This approach is more effective, since the search space now consists solely of valid configurations (although there is no guarantee that constraints will not be violated). A power flow was then carried out, but only on the two affected feeders - no power flow was needed for the third feeder, as conditions had not changed for it.

Voltage constraints were considered within the algorithm. It should be noted for this configuration, of the 189 valid configurations, 101 of those become invalid once voltage constraints are imposed. Only configurations having voltages within the favourable range, i.e., voltage magnitude greater than 0.95 p.u., were accepted (Reference [3] provides a description of operating voltage ranges).

For the purposes of this simulation, it is assumed that each section has a sectionalizing switch, and three of these remain open to ensure the radial topology of the network. There are 16 sections, each with a sectionalizing switch, allowing a possibility of 2^{16} (65,536) configurations. The initial power injected at the three substation buses is 29.21 MW, with a connected bus load of 28.7 MW, and thus the losses represent 0.51 MW, or 1.75% of the total load.

Upon convergence to the global minimum, the power injected at the substation buses is 29.17 MW, with the connected bus load unchanged at 28.7 MW. Losses now represent 0.47 MW, or 1.61 % of the total load. Thus, through system reconfiguration, losses have been reduced by 8.51%.

Since WatDist uses a probabilistic approach, 1000 trials were carried out. Table 7.2 provides a statistical description of the trials. For 1000 trials, the global optimum was found in every case, with a mean of 11.114 iterations (and standard deviation of 7.942). Running on an Intel-based Pentium 90 MHz computer, and with the WatDist algorithm implemented in C++, the mean time to find the global optimum for those trials was 0.174 seconds, with a standard deviation of 0.079 seconds. Thus, it is apparent that the algorithm is very fast. For all of the trials, the algorithm found the optimum configuration within 60 iterations.

Figure 7.4 indicates the rate of optimization of the best solution at each iteration averaged over 100 trials. It is evident that optimization rapidly approaches the optimum. An upper and lower bound are also shown. These bounds enclose the region in which all of the trial solutions lay. It can be seen that all solutions converge to the mean as the number of iterations increases. By the sixtieth iteration, all trials have converged to the optimum solution.

Mean number of iterations	11.114
Standard deviation	7.942
Mean time (seconds)	0.174
Standard Deviation (seconds)	0.079

Table 7.2. Statistical description of the number of iterations needed to achieve the optimum configuration to minimize losses (average of 1000 trials).

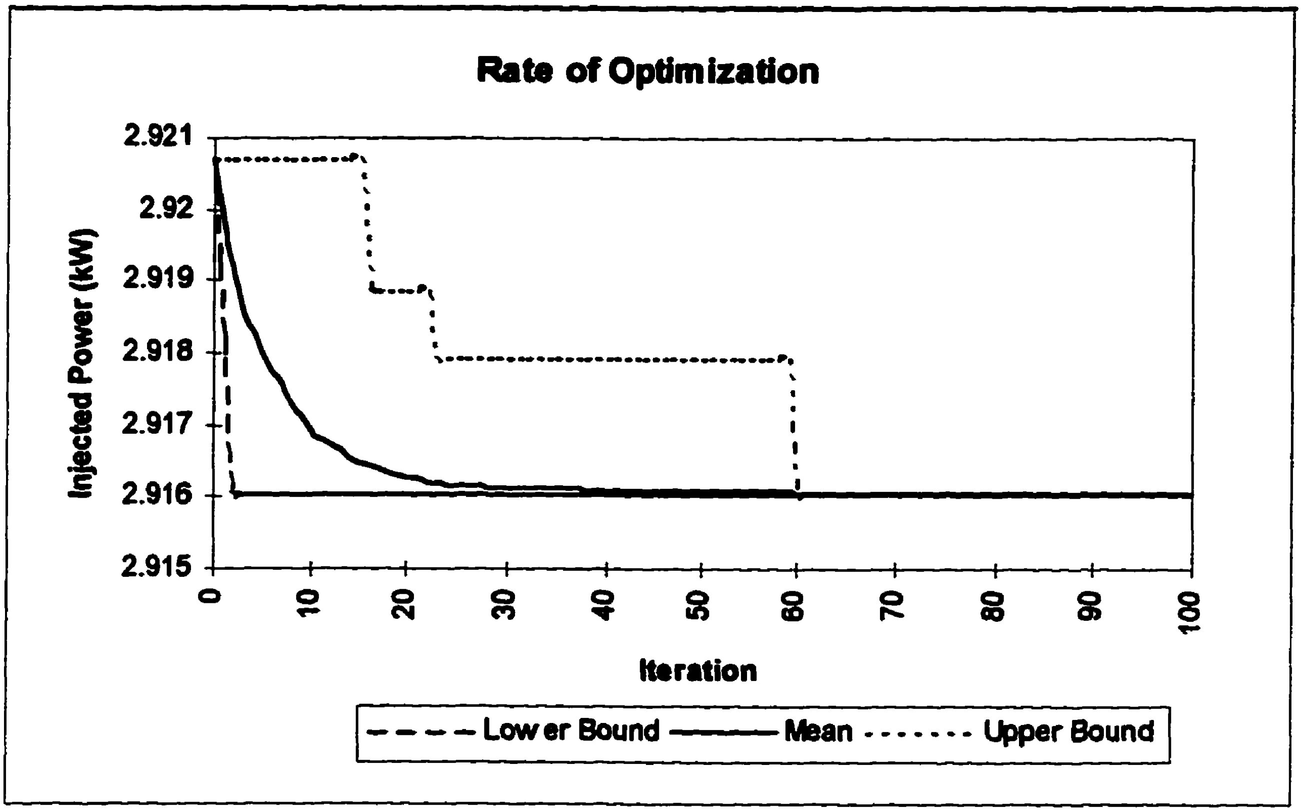


Figure 7.4. Rate of optimization of the mean fitness. The upper and lower bounds enclose the best and poorest solutions at each iteration over the sample.

An additional benefit of reconfiguration to reduce losses is an improvement in the load bus voltage profile, as can be seen in Figure 7.5. The average bus voltage initially was 0.984 per-unit volts. Following reconfiguration to the optimum configuration to minimize losses, the average bus voltage rose to 0.986 per-unit volts.

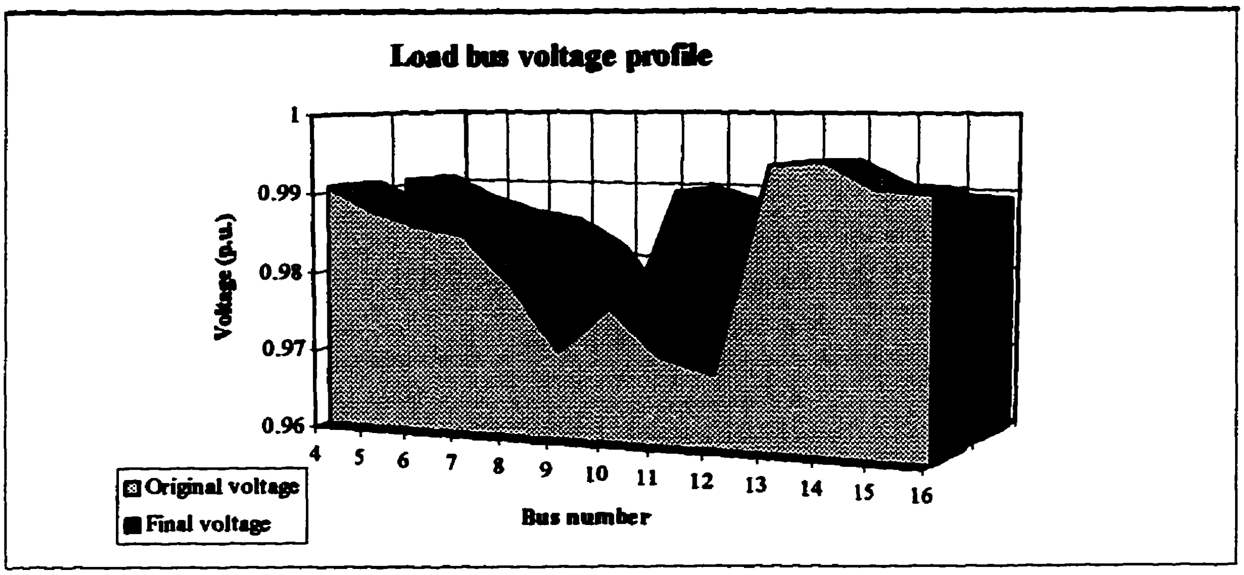


Figure 7.5. Load bus voltages, initially and following reconfiguration to the global optimum.

7.5 Summary

For the system under study, WatDist had no trouble in finding the optimal assignment all of the time. System losses were reduced by 8.51% when the optimum configuration was found. Although this may appear inconsequential, for a large distribution system supplying many MW of power, any reduction in losses is important.

To date, none of the methods available in the literature can claim to guarantee that a global minimum will be found, and the WatDist algorithm is no different. However, the algorithm offers many clear advantages over earlier methods, including:

a. a high success rate in finding the global minimum;

b. the final solution is independent of the initial configuration;

c. the algorithm is fast;

d. assurance that any solution offered will have a radial configuration with all loads connected and with no constraint violations;

e. the algorithm runs in "anytime," i.e., at any time during processing, the current solution is the best solution; and,

f. the relative ease in changing the objective function to optimize system performance for other factors.

This chapter has introduced the WatDist algorithm, and has proven its strength and utility on a bench mark system that has been used by many other researchers. Having verified that the WatDist algorithm works, in the next chapter the algorithm is used to minimize energy losses over an entire year.

Chapter 8

Practical Application of the WatDist Algorithm

8.1 Introduction

A system operator has four primary concerns when performing a simple feeder-to-feeder load transfer [21]:

1. what are the switching alternatives?
2. is the conductor capacity adequate?
3. will the voltage/loss profile be acceptable for both feeders?
4. will there be drastic changes in short-circuit duty that may disrupt protection?

In this chapter, practical implementation issues are examined, including load sensitivity to voltage, protection coordination and switching operations, is carried out. The performance of the WatDist algorithm on an actual distribution system in reducing losses over the period of one year is examined in detail.

8.2 Protection Coordination

The reliability of electric power supply is extremely important. Distribution system reliability is increased by various devices that isolate faulted sections, and also prevent excessive damage to circuits and related equipment in the event of a fault.

Distribution system protection is considered both a science and an art [82]. The determination of such things as fault currents, equipment ratings and whether or not protective devices will be properly coordinated is based on engineering principles. However, other aspects are not well-defined, including such things as zones of protection, and device location and type. As well, local conditions may influence the design of protection systems, and thus it is difficult to generalize to all circuits.

In a well-designed distribution system, protective devices will be coordinated. Protective devices are selected with characteristics that complement one another to ensure a minimum number of customers are affected by faults. For example, for a feeder with two reclosers, the downstream recloser should trip before the upstream recloser for all downstream faults, and should lockout before the upstream recloser for all permanent faults.

To date, no researchers in the area of loss minimization through system reconfiguration have considered the issue of protection coordination of automatic reclosers, fuses and sectionalizing switches in feeder circuits. Nearly all of the effort has been directed towards finding an efficient algorithm for minimizing losses.

It could be argued that power distribution system designers consider reconfiguration when planning protection of distribution system circuits. Automatic switches used to redistribute loading among a group of interconnected distribution circuits to minimize losses should be designed such that protection is adequate for all possible configurations of a system. While manual and computerized tools exist to design a protection system for a fixed configuration of a radial distribution system, there are few computerized tools for designing protection schemes for all possible configurations [83]. Such tools could work with smart protective devices, whose settings are controlled by a computer as the circuit configurations change, or with protective devices whose settings remain fixed.

Some work has been done in attempting to coordinate system protection with reconfiguration, but the work is directed towards system planning, and not system operation. Broadwater *et al* [83] proposed a protection system design algorithm which encompassed all possible circuit configurations of a radial distribution circuit. Based on eleven database tables and a predetermined set of coordination rules, the algorithm performed computerized placement and selection of coordinating devices.

Hsu and Jwo-Hwu [84] proposed an algorithm for the planning of distribution feeder reconfiguration which accounted for protective device coordination. A set of switchable regions within which switch operations are allowed is identified. Once the distribution system has been planned using the proposed algorithm, the network can be reconfigured, with all protective devices coordinated, by changing the open/close position of switches in the switchable regions.

More recently, Peponis *et al* [85] proposed an algorithm for distribution system planning that takes into account the protection scheme applied, network reliability and voltage drop, as well as minimizing energy losses. Loss minimization is achieved by reconfiguration and by the installation of capacitors. The proposed method is used for short-term planning of distribution networks for approximately one year. The method assumes autoreclosers are used in overhead networks to clear temporary faults, and that fuses or sectionalizers are installed at the beginning of each lateral circuit.

Witte *et al* [86] describe reclosers used with microprocessor controls to modify the basic time current characteristics or the operating sequence of automatic circuit reclosers. The paper discusses a PC-based coordination package which includes an expert system module to determine device coordination on radial distribution feeders.

In [87], Horowitz *et al* examine an adaptive protection scheme for transmission systems. An adaptive protection scheme seeks to adjust various protective devices to prevailing power system conditions, and is based on a microprocessor-based system of protection. The implementation of an adaptive scheme is practical and straightforward with the advent of digital technology in a microprocessor-based system.

The issue of protection coordination is not considered as part of the WATDIST algorithm. Protection coordination would have to be considered before implementation in an actual distribution system, but, because of the variety of protection schemes in use, it is impossible to provide a general-purpose algorithm for all situations. However, WATDIST is easily modified to prevent certain switching patterns, and it would not be difficult for a utility to tailor WATDIST to the utility's needs.

8.3 Switching Operations

To ensure a minimal disruption of customer service, and to reduce the number of switching operations carried out, once a new configuration has been identified, a switching sequence must be determined. The sequence must seek to minimize transient effects, and ensure that there are no temporary constraint violations, such as exceeding protective device capacity. This issue is not explored in this thesis, as algorithms are already available to carry out switching operations. For example, Aoki *et al* [88] provide such an algorithm.

The frequency of system optimization and switching must be considered when reconfiguration for loss minimization is part of a utility's day-to-day operation. A trade-off must be made between realizing maximum savings through loss reduction and increased maintenance and repair costs to switchgear as a result of more frequent switching operations. As well, the effect of transients caused by switching operations must be considered. Reference [89] suggests allowing 10 minutes as the amount of time needed to adequately dampen system transients.

In an operational context, hourly switching is adequate, and this has been suggested by several researchers [18, 44].

8.4 Application of WatDist to an Actual Distribution System

WATDIST was applied to a second test system consisting of a three-feeder, 3200 kVA network with 35 load points. The system is presented in Figure 8.1, and system data in Table 8.1.

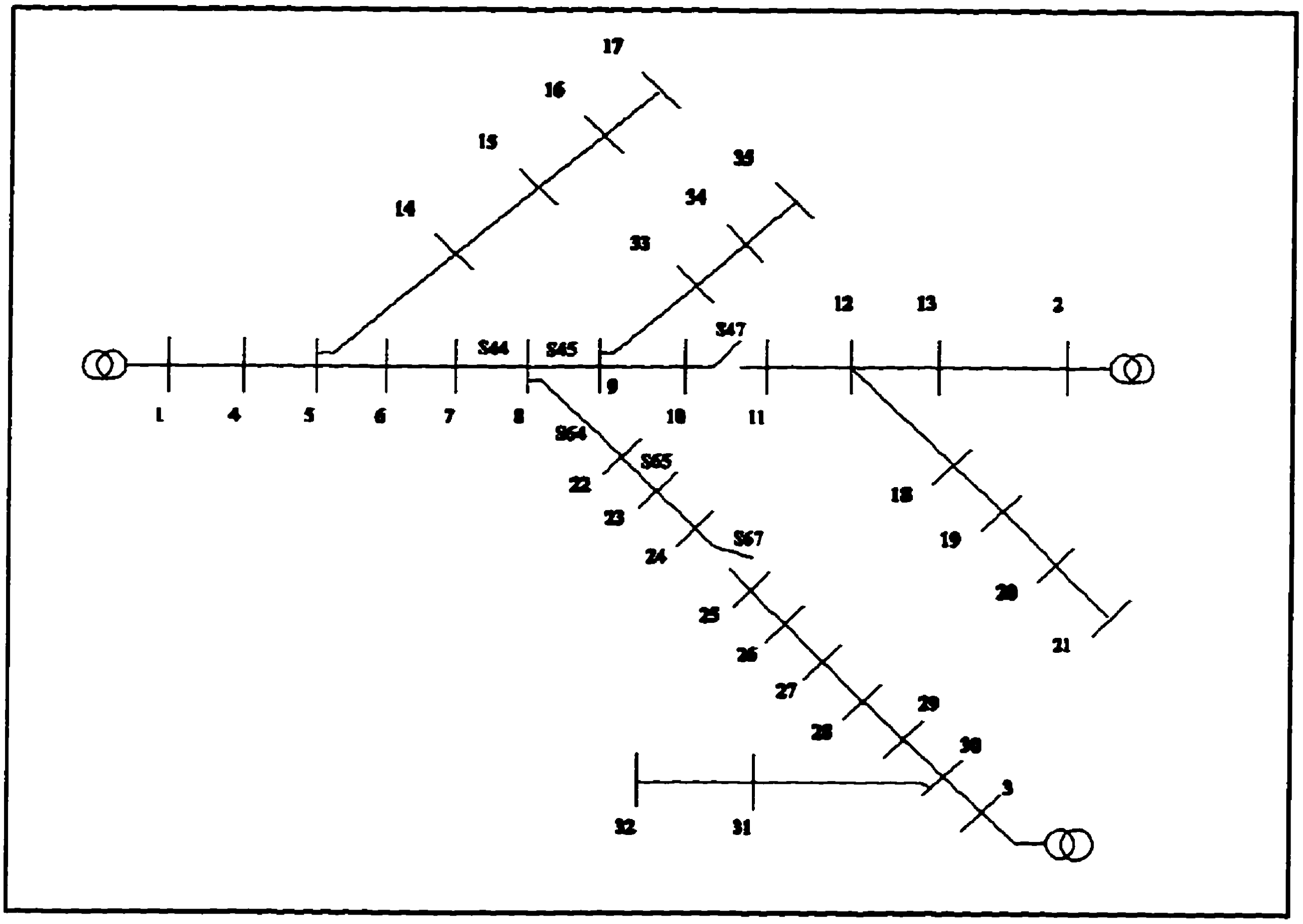


Figure 8.1. Second test system network topology (initial configuration).

Start/End Bus	Load type	Section Data (p.u.)		End Bus Load	
		Resistance	Reactance	kW	kVAR
1-4	Ind	0.05	0.01	230.0	69.0
4-5	Ind	0.015	0.002	0.0	0.0
5-14	Res	0.01	0.002	72.0	21.0
14-15	Res	0.05	0.01	72.0	21.0
15-16	Res	0.01	0.002	72.0	21.0
16-17	Res	0.01	0.002	72.0	21.0
5-6	Ind	0.01	0.002	230.0	69.0
6-7	Ind	0.05	0.01	230.0	69.0
7-8	Ind	0.01	0.002	0.0	0.0
8-9	Ind	0.01	0.002	0.0	0.0
8-22	Res	0.04	0.02	57.0	7.0
22-23	Res	0.01	0.002	57.0	7.0
23-24	Res	0.01	0.002	57.0	7.0
9-10	Ind	0.01	0.002	230.0	69.0
9-33	Com	0.05	0.01	72.0	21.0
33-34	Com	0.01	0.002	72.0	21.0
34-35	Com	0.01	0.002	72.0	21.0
2-13	Ind	0.01	0.002	230.0	69.0
13-12	Ind	0.05	0.01	0.0	0.0
12-11	Ind	0.01	0.002	230.0	69.0
12-18	Com	0.05	0.01	57.0	7.0
18-19	Com	0.01	0.002	57.0	7.0
19-20	Com	0.01	0.002	57.0	7.0
20-21	Com	0.05	0.01	57.0	7.0
3-30	Res	0.05	0.01	57.0	7.0
30-31	Res	0.01	0.002	57.0	7.0
31-32	Res	0.01	0.002	57.0	7.0
30-29	Res	0.01	0.002	57.0	7.0
29-28	Res	0.01	0.002	57.0	7.0
28-27	Res	0.05	0.01	57.0	7.0
27-26	Res	0.01	0.002	57.0	7.0
26-25	Res	0.02	0.002	57.0	7.0

Table 8.1. System data for the second test network.

The loads in this system are a mix of residential and commercial, with some light industrial, and are in the form of transformer stations, each with its own capacity. In-line switches have been included in most line sections for the simulation carried out here. This would not necessarily be done in practice, but was done to test the performance of the WatDist algorithm.

Although the system is by no means large compared to many distribution systems, its use as a test system allows presentation of the details of simulation results in this thesis. Because of the simulation of one year of operation of the system, simulation times were lengthy, even for this system, as a full year of 24-hour-a-day operation was examined. The system was used to test the reconfiguration algorithm on a realistic system where loads vary over time. In this way, the amount of loss reduction over a one-year period was examined. The size of the system is comparable to the size of systems used by other researchers, including [28] (16 buses), [30] (37 buses), [33] (32 buses) and [47] (30 buses).

Load curves were necessary for each of the three load types: residential, commercial and industrial. The method used to specify loads is the method proposed by Wagner in reference [27]. In consultation with Ontario Hydro, Wagner concluded that loads could be divided into three time periods, as follows:

SUMMER - June, July, August

WINTER - December, January, February

SPRING/FALL - March, April, May, September, October, November

Loads tended to follow the same patterns for each of these three time periods.

Wagner also concluded that within those time periods, typical daily load patterns emerged that depended on whether the day was a week-day or week-end/holiday. Daily load curves also depended upon the three load types.

System load changes for an entire year can be simulated by using only 18 load profiles, consisting of 3 for each of the year's time periods, either week-day or week-end, and for each of the 3 load types (3 times 2 times 3).

In further consultations with Ontario Hydro, Wagner also found that industrial load curves do not change significantly with the seasons, as industries' requirements for heating and/or air conditioning represent only a very small portion of their demand.

For residential customers, the summer peak demand tends to occur in the afternoon, when air conditioning loads predominated. In the winter, peak demand shifts to early morning, when heating loads were predominant. It was found that spring and fall demand are approximately 75% of the winter demand.

Using this information, load profiles were produced for the test system, and are shown in Figure 8.2 to Figure 8.6. For the sake of the simulation, it was assumed that the industrial load curve did not change throughout the year. As well, residential and commercial loads used the same load curves. This method of simulating load changes is by no means accurate, but it does provide a reasonable method for changing load profiles to examine the performance of the WATDIST algorithm in reducing losses.

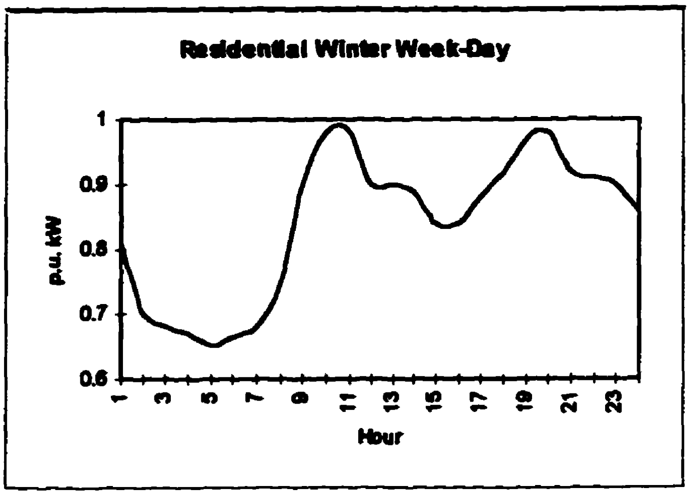


Figure 8.2. Load profile for residential loads on a winter week-day.

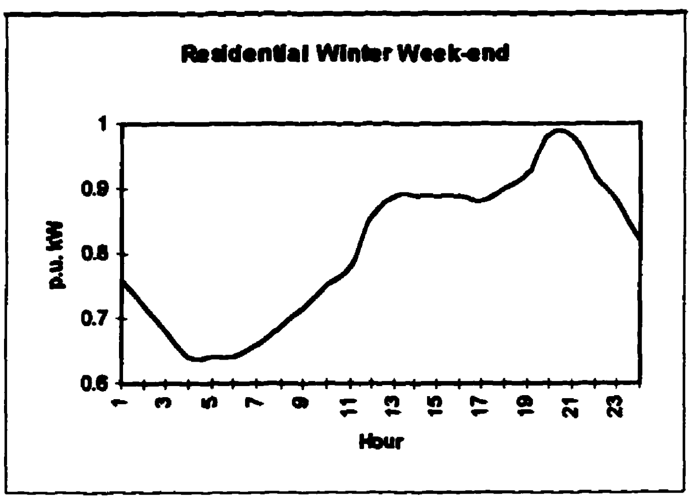


Figure 8.3. Load profile for residential loads on a winter week-end day.

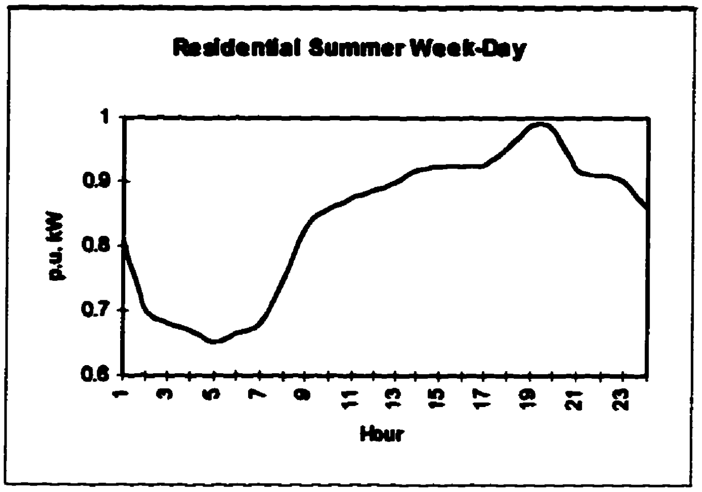


Figure 8.4. Load curve for residential loads on a summer week-day.

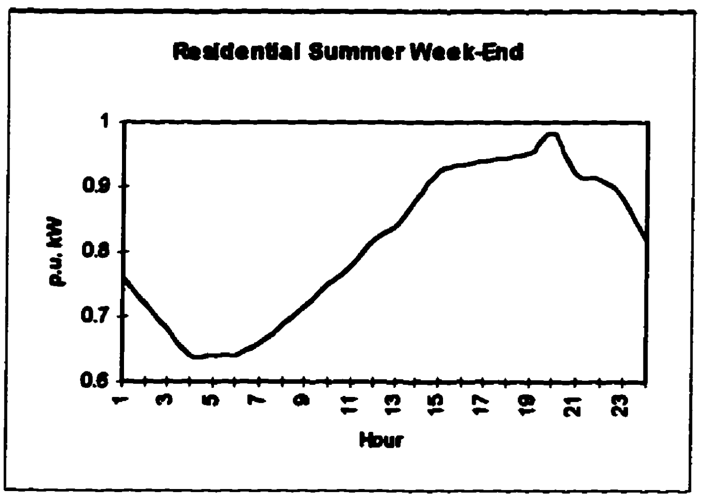


Figure 8.5. Load curve for residential loads on a summer week-end.

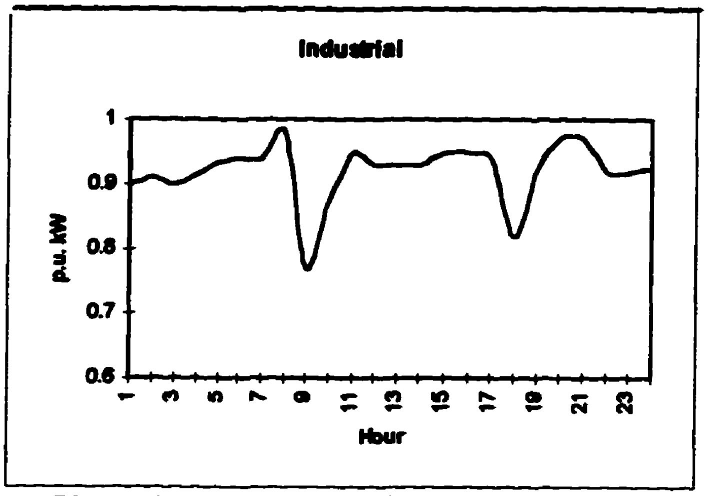


Figure 8.6. Load curve for industrial loads.

Having derived load curves for each load type, it was necessary to assign those load profiles to the actual system loads. Loads on the system are in the form of transformer stations, each with its own capacity. The total system capacity is 3200 kVA, with a peak demand of 2739 kW. Therefore, the load at each load point, *i*, at a particular instant in time, *t*, is assigned as:

$$P_{i_t} = \frac{P_{station}}{P_{total}} \times P_L(t) \tag{8.1}$$

where P_{it} is the load at the load point, *i*, at time, *t*;

P_{total} is the total system capacity from Table 8.1;

$P_{station}$ is the substation capacity, and shown in Table 8.1 as the end-bus load in kW; and,

$P_L(t)$ provides the load profile information, which is a function of time, and which is expressed in per-unit values.

In this fashion, the peak system load will be 2739 kW, and the total load will be distributed among the various load points in proportion to their substation capacities.

8.5 Simulation Results

The performance of the algorithm was examined in a simulation of the operation of the test network over a period of one year, with loads varying according to the load type, season of the year, and time of day. To more realistically match the unpredictability of loads over the course of the day, load curves were multiplied by a random number between 0.85 and 1.15. This number was generated by generating a random number having a uniform distribution between 0 and 1, multiplying it by 0.3, and adding it to 0.85.

In the first part of the simulation, operation of the distribution was carried out for the period of one year with the switches between buses 10 and 11 (identified as Switch S47 in Figure 8.1 for simulation purposes), and between buses 24 and 25 (identified as Switch S67), remaining open for the entire year, with no reconfiguration. In the second part of the simulation, the same loads were assigned to buses as had been assigned in the first part of the simulation. However, in the second part, reconfiguration was allowed to occur, and the WatDist algorithm was employed to determine which switches should be open at each hour of each day.

As noted earlier, the total power supplied by all substation buses is the quantity optimized in the WatDist algorithm, and thus this quantity was recorded during the simulation. As well, the total loads connected to load buses were recorded, as were the losses. These recordings were made at 6 p.m. daily, as it is at this time that load peaks occur in most of the load curves. Daily energy supplied was also recorded. To determine which switches were involved in the reconfiguration, switch status was also recorded.

8.5.1 Results on a Yearly Basis

Figure 8.7 shows the total energy supplied to the distribution system by month. It is evident that using the WatDist algorithm ensures a reduction in the energy supplied, and hence in the losses, as each month shows a reduction in the energy supplied when the WatDist algorithm is used. The reduction in energy supplied is quantified in Figure 8.8 and in Figure 8.9. From Figure 8.8, it is clear that energy savings, on average, are approximately 1.5%. It is not clear why there is a drop in those savings during the month of July, and it is most likely a statistical anomaly as a result of the simulation and the load curves. Figure 8.9 provides the cumulative system energy by month, and also shows the energy saved by using the WatDist algorithm. Over the course of one year, the energy savings are 1.67%.

The financial benefit resulting from these savings would be of interest to a utility. These savings are difficult to assess, as energy costs are usually negotiated, and depend on such factors as peak load demand, power factor, and time of use. However, it is felt that a lower bound of \$0.03/kWh is reasonable, as is a peak charge of \$0.07/kWh. Based on these limits, cost savings for this test system would be between \$10,586 and \$24,700 annually.

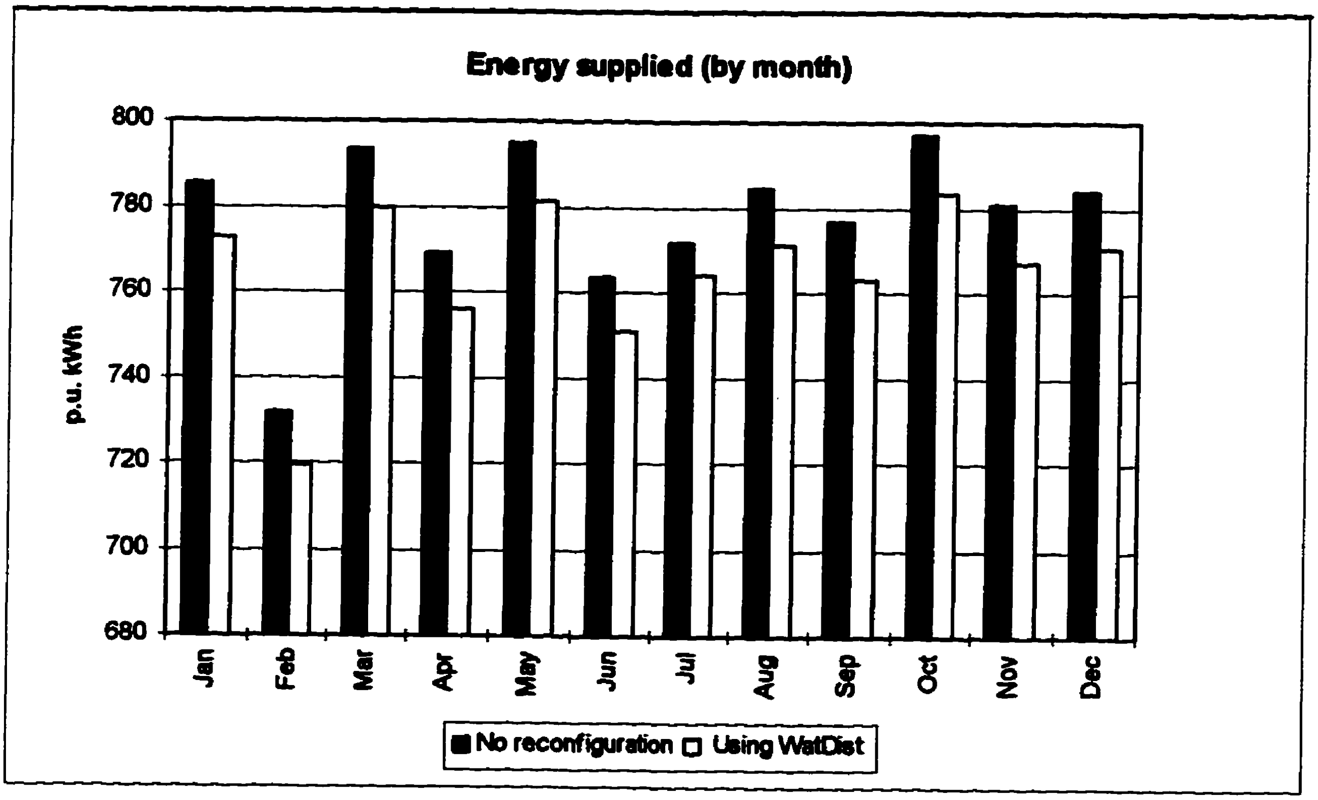


Figure 8.7. Total system energy supplied each month.

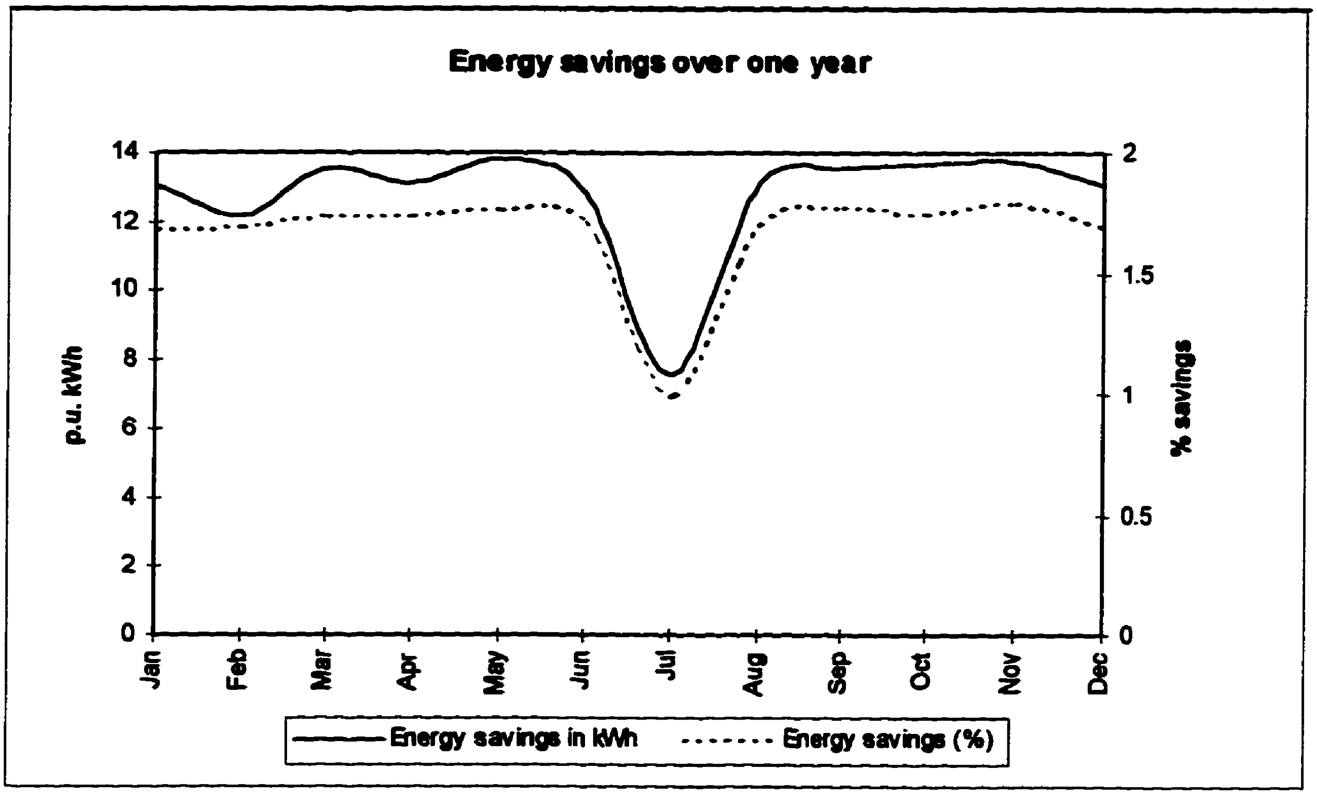


Figure 8.8. Energy savings over one year for each month, in kWh and as a percentage, as a result of using the WatDist algorithm.

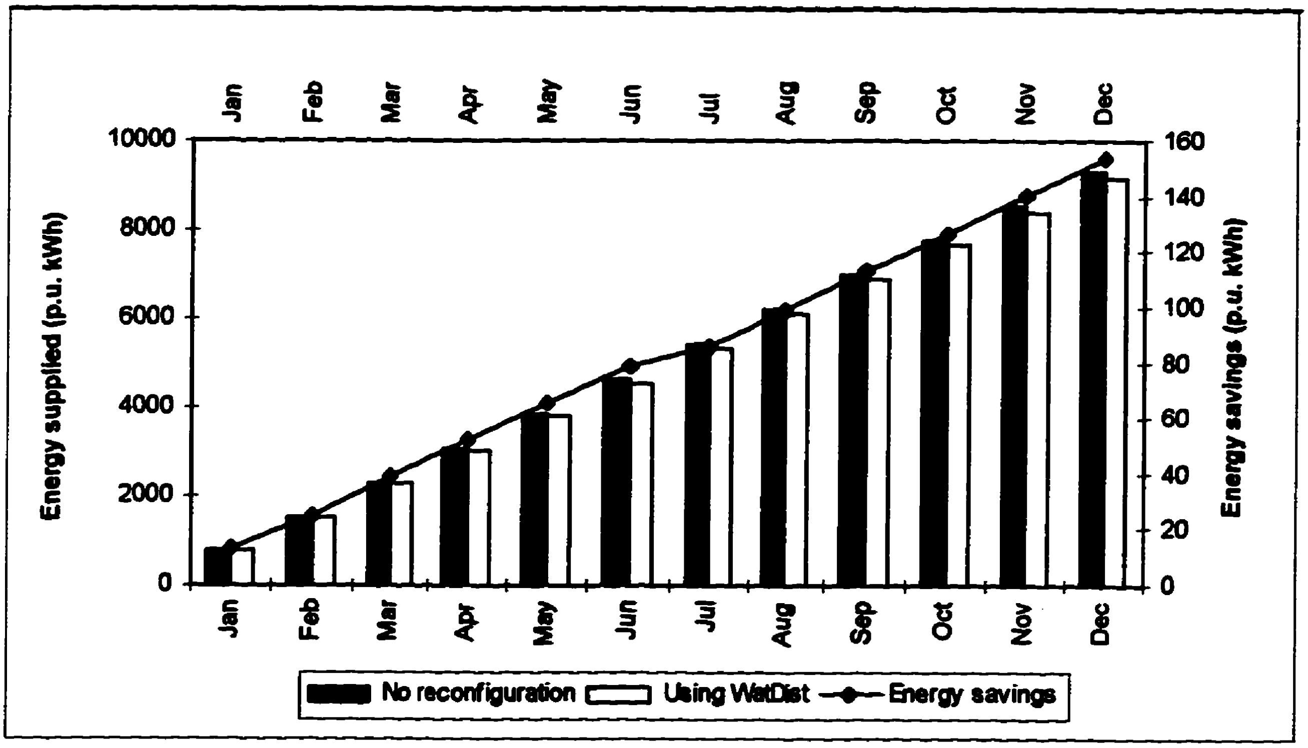


Figure 8.9. Total cumulative system energy, by month, and showing energy saved by using the WatDist algorithm.

Although there are potentially 33 switches available for reconfiguration, it turns out that very few of them are required for reconfiguration. Indeed, simulation results revealed that only 4 switches need to be automated:

1. the switch between buses 7 and 8 (Switch S44);
2. the switch between buses 8 and 9 (Switch S45);
3. the switch between buses 8 and 22 (Switch S64); and,
4. the switch between buses 22 and 23 (Switch S65).

All other switches remain closed for the entire simulation, including the two initially open (Switches 47 and 67). Table 8.2 shows the percentage of time each of the four switches involved in the reconfiguration is open, and it is apparent that Switch S44 and Switch S45 should be automated, and Switch S64 left open all of the time. If a cost of $11k is assumed for an automated switch, automating these two switches would cost $22k. Based on the energy savings as a result of using the WatDist algorithm, this cost would be recovered in a little over 2 years, using the lower bound for cost savings. Thus, it is clear that automating these switches, and carrying out reconfiguration, is attractive from a financial point-of-view.

The percentage of time open for Switches S44 and S45 is shown graphically in Figure 8.10. A typical duty cycle for a single day for these two switches is shown in Figure 8.11 and Figure 8.12. It can be seen from these two figures that when Switch S44 is open, Switch S45 is closed, and vice versa.

	Switch S44	Switch S45	Switch S64	Switch S65
Jan	65.68	34.32	100.00	0.00
Feb	68.62	31.38	100.00	0.00
Mar	63.40	36.60	100.00	0.00
Apr	64.13	35.87	99.86	0.14
May	64.61	65.39	99.87	0.13
Jun	65.10	34.90	100.00	0.00
Jul	64.21	35.79	99.87	0.13
Aug	61.39	38.74	99.87	0.13
Sep	63.85	36.15	100.00	0.00
Oct	64.34	35.66	99.87	0.13
Nov	61.08	38.92	100.00	0.00
Dec	60.59	39.41	99.87	0.13
Average	63.92	36.09	99.92	0.08

Table 8.2. Percentage of time open for the 4 switches involved in reconfiguration.

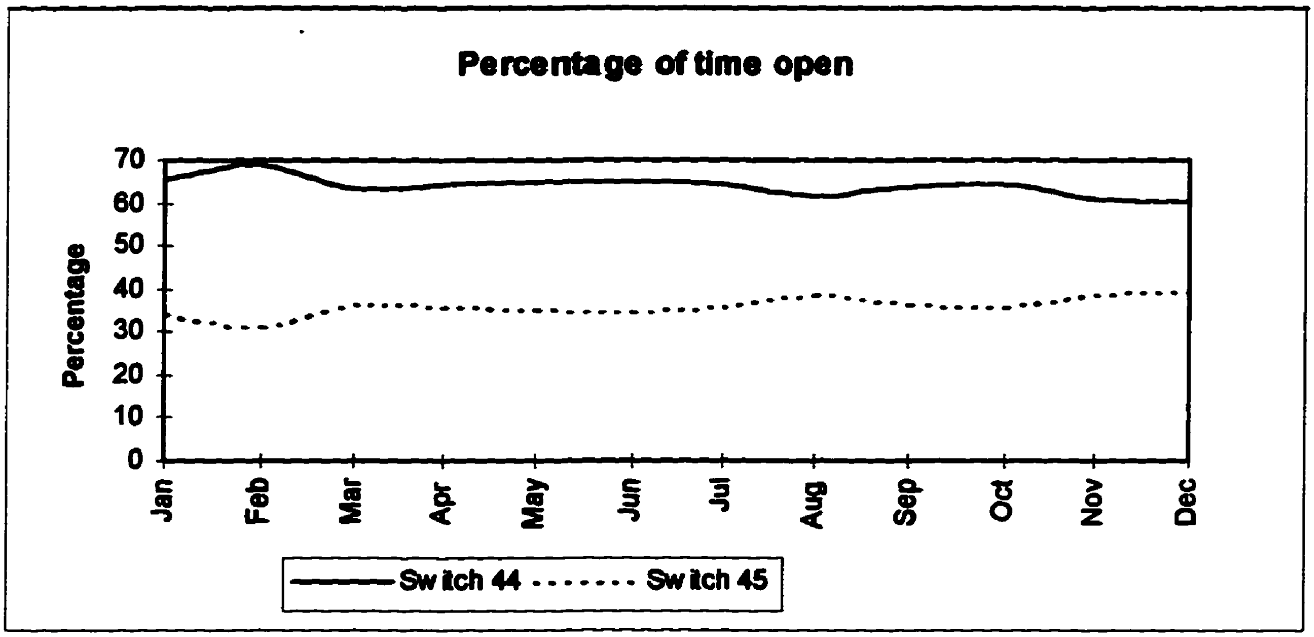


Figure 8.10. Percentage of time open for Switches S44 and S45 over one year.

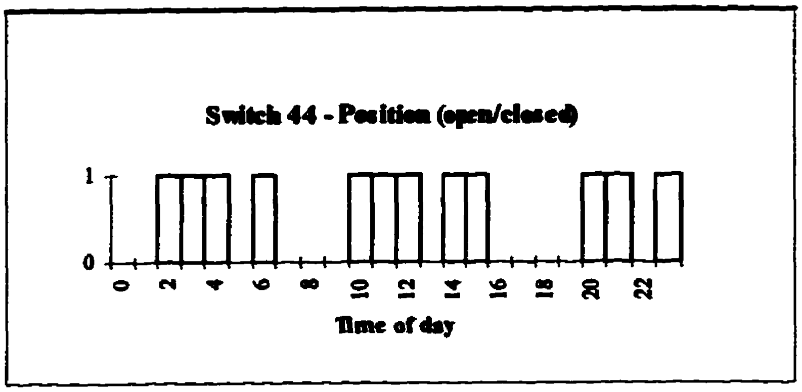


Figure 8.11. Hourly position of Switch S44 for 1 January.

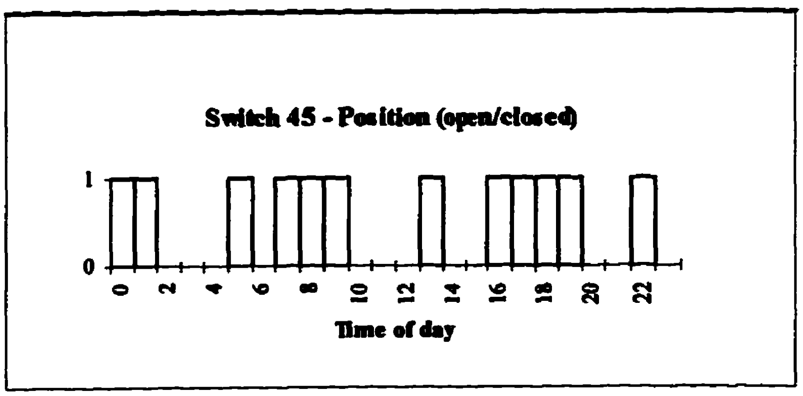


Figure 8.12. Hourly position of Switch S45 for 1 January.

It should be noted that losses have been reduced in every instance with no constraint violations, with no loads left disconnected, and with the radiality of the network assured.

8.5.2 Results on a Monthly Basis

This section shows the results of applying the WatDist algorithm, and the energy savings that are possible, on a monthly basis. One month is presented for each season, i.e., winter, spring, summer and fall.

The total connected loads at load buses at 6 p.m. daily are shown in Figure 8.14 through Figure 8.17. Finally, Figure 8.18 through Figure 8.21 show the line losses at 6 p.m. daily, with no reconfiguration and when the WatDist algorithms used. Figure 8.22 to Figure 8.25 show the energy supplied each day for a month of each season, and the cumulative energy savings. In every case, WatDist reduces line losses.

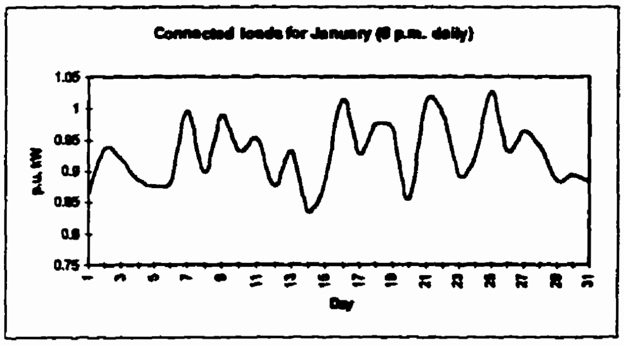


Figure 8.14. Connected loads for winter month (January), taken at 6 p.m. daily.

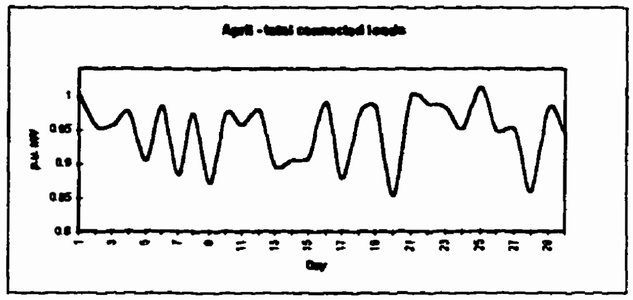


Figure 8.15. Connected loads for a spring month (April), taken at 6 p.m. daily.

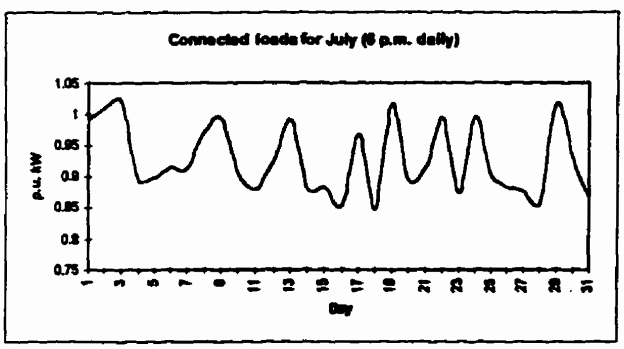


Figure 8.16. Connected loads for summer month (July), taken at 6 p.m. daily.

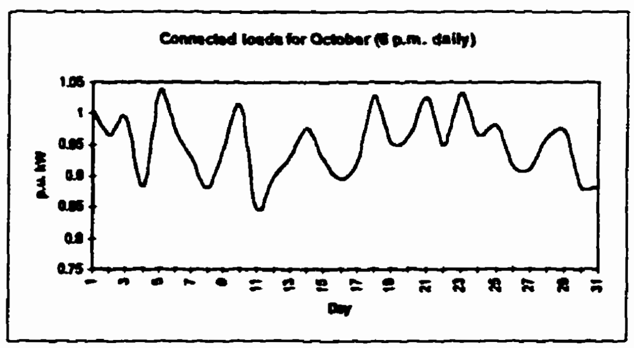


Figure 8.17. Connected loads for fall month (October), taken at 6 p.m. daily.

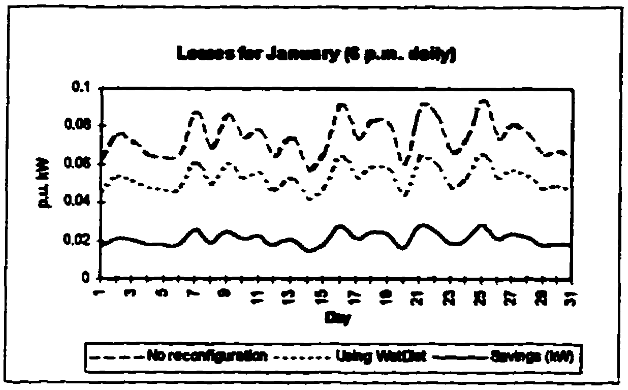


Figure 8.18. Line losses at 6 p.m. daily for a winter month (January), in kWh.

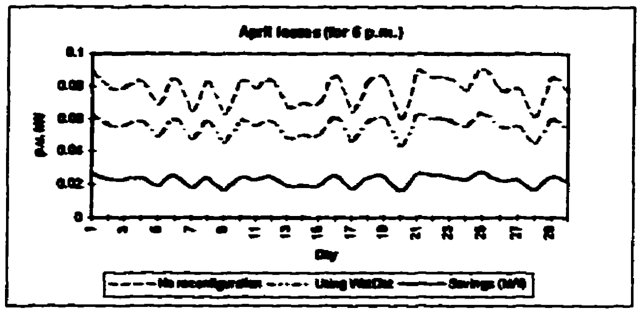


Figure 8.19. Line losses at 6 p.m. daily for a spring month (April), in kWh.

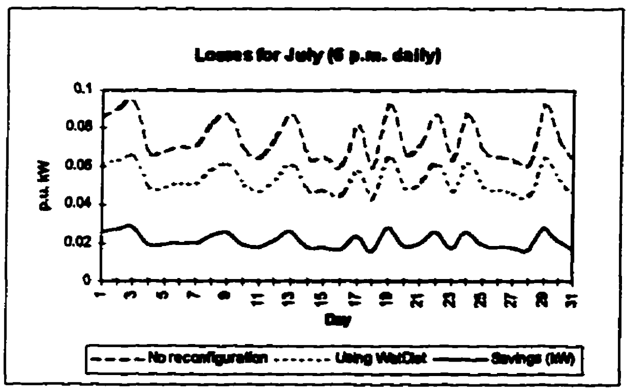


Figure 8.20. Line losses at 6 p.m. daily for a summer month (July), in kWh.

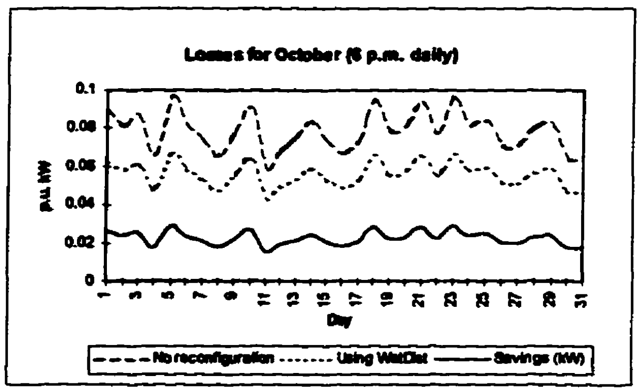


Figure 8.21. Line losses at 6 p.m. daily for a fall month (October), in kWh.

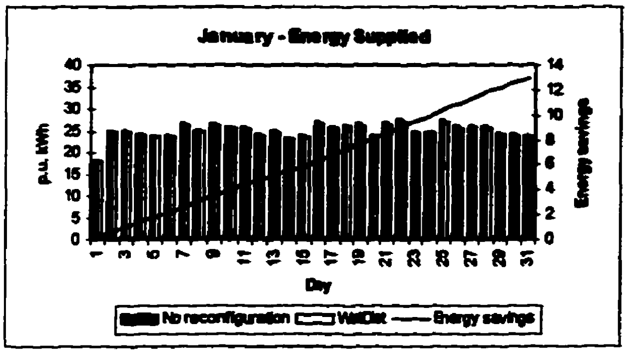


Figure 8.22. Energy supplied for a winter month (January), with energy savings using WatDist.

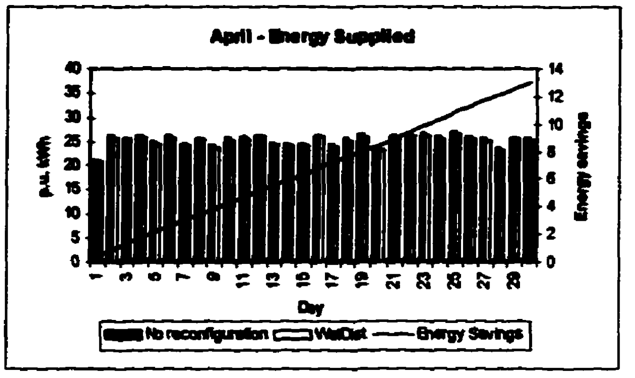


Figure 8.23. Energy supplied for a spring month (April), and energy saved using WatDist.

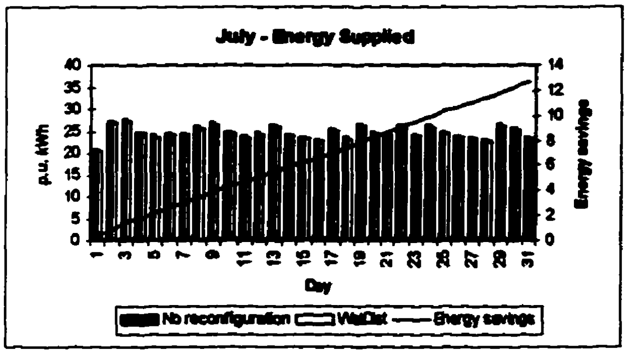


Figure 8.24. Energy supplied for a summer month (July), with energy savings using WatDist.

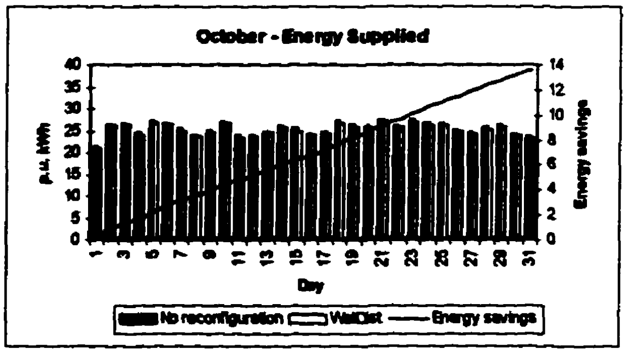


Figure 8.25 Energy supplied for a fall month (October), and energy savings using WatDist.

8.5.3 Results on a Daily Basis

It is clear that WatDist reduces line losses, and hence the energy supplied to the network. In this section, the performance of WatDist over a one-day period (New Year's Day) is examined.

Figure 8.26 provides the system load curve for 1 January. The total losses without reconfiguration, and after employing the WatDist algorithm, are presented in Figure 8.27, and it can be seen that reconfiguration on an hourly basis reduces line losses.

8.6 Summary

The usefulness of the WatDist algorithm has been clearly demonstrated in this chapter. For every hour of every day, losses have been reduced. In addition, in every instance, there were no constraint violations, there were no disconnected loads, and the radiality of the network was preserved.

Over a one-year period, the energy savings are 1.67% of the total energy supplied, or 353 MWh for the system examined. The financial benefit resulting from these energy savings range between a lower bound of $10,586 and an upper bound of $24,700, based on current Ontario Hydro rates.

This simulation has also served to identify that not all switches need to be automated in a distribution system to realize energy savings. In fact, very few of the switches need to be automated, and, for the system examined, only two sections were identified as requiring automated switches to allow reconfiguration on an hourly basis. It is clear that the WatDist algorithm could be used during planning, using projected loads to identify which sections would most benefit from automation.

The results of this simulation also indicate that a utility that did not have automated switches (and whose position had to be changed manually) could benefit by using WatDist to identify switches that should remain open most of the time. While this would not minimize losses, it would certainly reduce them.

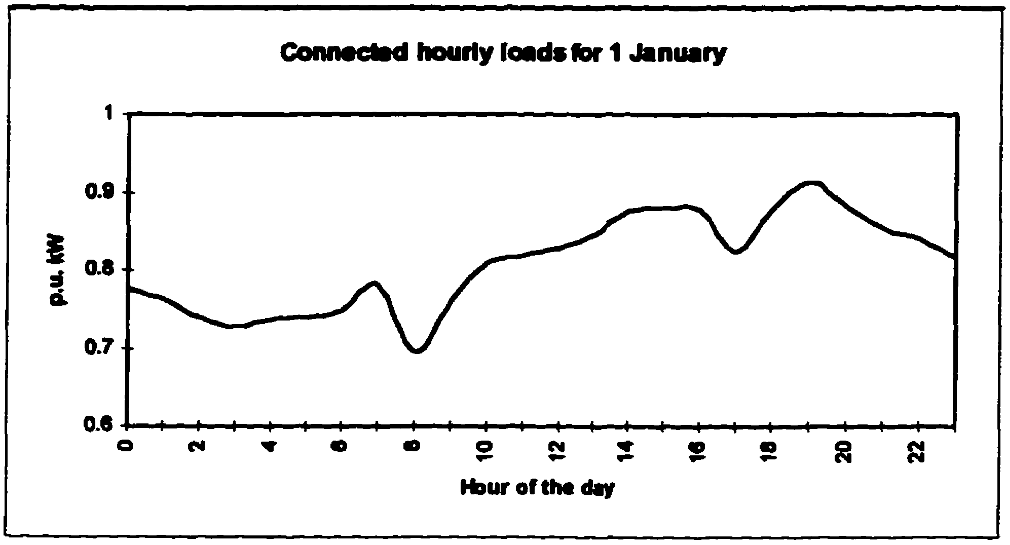


Figure 8.26. System load curve for New Year's Day.

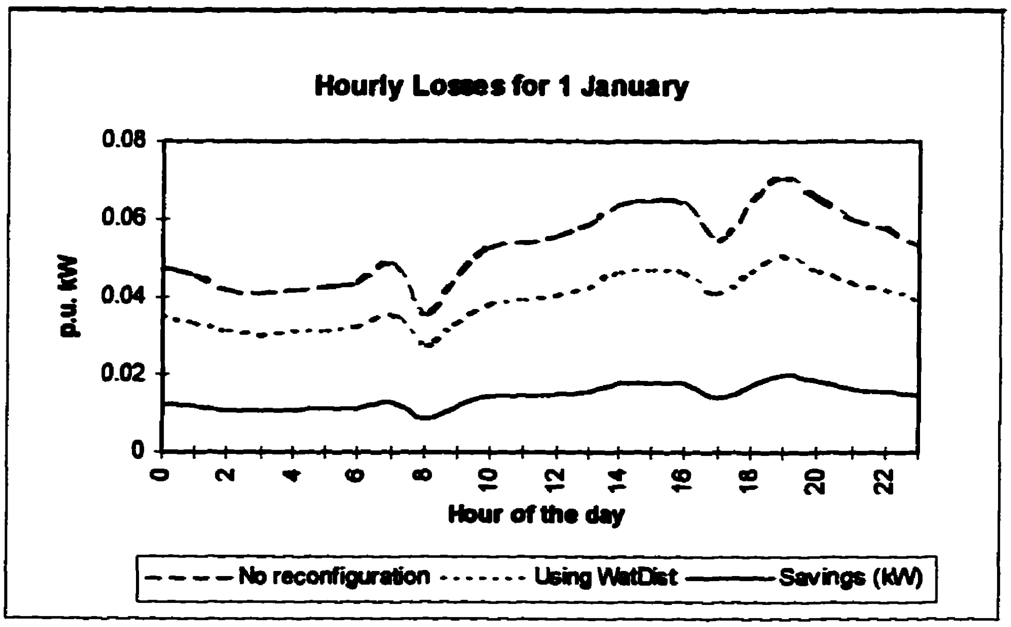


Figure 8.27. Reduction in losses through use of the WatDist for New Year's Day.

Chapter 9

Conclusions and Recommendations for Future Work

9.1 Conclusions

Loss minimization through system reconfiguration offers utilities the opportunity to reduce energy costs using existing equipment, and, at the same time, release system capacity. As outlined in the earlier chapters of this thesis, many algorithms have been proposed for loss minimization, but few have been adopted for use. Reasons for this most likely include inadequate consideration of system constraints, excessive computation time and difficulty in implementing the algorithms in software.

It became clear in the early stages of this research that there were two issues involved in the loss minimization problem. The first issue was the optimization problem itself, and finding a method that provided optimal solutions in real-time while satisfying system constraints. A second and equally-important issue was how system data was represented. This was an important issue, because it affected the performance of whatever optimization technique was employed for loss minimization. Data structures were needed to store system information, including static information such as section impedances and substation capacities, as well as for dynamic information, including switch status and bus loads. Algorithms were also then needed to efficiently manipulate those data structures.

Three algorithms (Graph Builder, Path Builder and System Update) have been proposed in this thesis for network reconfiguration studies. The Graph Builder algorithm can be used to build a graph of a distribution feeder, and to build its admittance matrix. Path Builder determines a path between two adjacent feeders, providing a list of sections between feeder root nodes. An optimization technique can then be employed to determine which section should be opened to restore a system to its radial configuration, or the section to be opened can be specified by an operator. Finally, the System Update algorithm is used to update all system dynamic information following reconfiguration.

The use of Graph Builder allows the construction of admittance matrices for each feeder in a distribution system. Having this feature allows the WatDist algorithm to perform load flows on a small subset of the entire distribution system, and hence considerably reduces computation time. Traditionally, load flows are carried out for entire systems.

The use of Path Builder allows the selection of all switches between two connected substations for consideration in the optimization process, rather than the narrow selection of switches adjacent to an open switch, as has been used by other researchers (for example, as reported in [52] and [53]). The building of the path, and the opening of a switch in the path, ensures that any solution offered by the WatDist algorithm will be a feasible solution that guarantees all loads are connected and the network's radial configuration is retained.

These three algorithms are a significant contribution to distribution system operation, and provide a strong framework for other researchers in the fields of loss minimization, distribution system planning, and load restoration (to be discussed shortly).

Having developed the necessary data structures, and algorithms for manipulating them, the next step was to select an optimization technique to perform the loss minimization function. After a thorough review of earlier methods and their shortcomings, it became clear that the branch exchange method was the most promising, and artificial intelligence techniques were adopted. The result was the WatDist algorithm.

The WatDist algorithm captures the essential features of distribution systems. It allows users to monitor the state of the optimization, and a computation time can be specified. This is in contrast to some optimization techniques which are "all-or-nothing," in that interim solutions are not available. The algorithm can be run until a solution is needed, and can be run for several thousand iterations to determine if a better solution is possible.

As noted previously, different researchers have had different perceptions of the loss minimization problem, in that some have attempted to minimize losses, others have attempted to balance feeder loading, while others have been concerned with security of supply to important customers. The WatDist algorithm is perhaps unique in that it is a simple matter to modify it to account for other objectives during optimization (for example, to consider load balancing among supply transformers, minimization of the worst voltage drop, minimization of service interruption frequency, balanced service to important customers, or a combination of these objectives, such as considered in [35] by Roytelman *et al*).

The algorithm does not require continuous optimization functions, and nonlinear functions are treated in the same manner as linear ones. Functions containing many local minima and several variables are easily handled.

The WatDist algorithm provides a significant contribution to distribution system operation. It identifies which switch should be opened for loss minimization through system configuration, and can be useful in determining which switches should be automated in a system, or which switches should be left open most of the time to reduce line losses.

As mentioned earlier, the algorithm assures continuity of supply to all load buses while retaining a radial configuration, as well as providing sound solutions that do not violate system constraints. The algorithm runs in "anytime," i.e., the current solution is the best solution. In the case where the global minimum is not found, losses will be reduced. WatDist is fast.

From an algorithmic point of view, implementation of the algorithm in software is straightforward. There are no parameters to set and/or "fine tune." The final solution is independent of the original configuration, and the algorithm has a very high success rate in finding the global minimum. It is relatively simple to change the objective function to optimize system performance for other factors.

9.2 Recommendations for Future Work

Distribution system planning and system restoration are two applications that are similar to the reconfiguration for loss minimization problem. In distribution system planning, planners attempt to determine network layouts, the position of switches and protective devices, and equipment capacities based on expected loads. In system restoration, operators attempt to restore power to as many customers as possible following system outages as a result of faults caused by weather, animals and accidents. In both cases, many configurations must be compared before deciding upon a final configuration. Thus, application of the WatDist algorithm to these problems should be examined.

A SCADA system interface should be developed for the WatDist algorithm. This would allow the algorithm to access on-line SCADA data, and allow WatDist to access historical system data, such as load curves for various customers. Instead of reacting to bus loads and determining a configuration to minimize losses based on information that is almost out-of-date as soon as it is recorded, it would be beneficial to have a method of predicting loads ahead of time. Access to SCADA data and a suitable forecasting technique would allow a system to be ready to respond to expected loads.

As faster computers become available, it can probably be said with some certainty that the execution speed of WatDist will be reduced. However, for very large distribution systems, it would probably be beneficial to use more than one computer to determine the optimal configuration. Several computers could be slaves to a master computer that would direct them to determine system demands for a specific configuration. In this case, communication and coordination problems become issues. However, such a parallel implementation should be feasible, and could be examined.

Some fine tuning of the WatDist algorithm may be possible. Since much of the processing time is spent building bus admittance matrices and carrying out load flows, it makes sense to remember which switching combinations have already been examined, to ensure that they are not re-examined. However, an efficient method is needed to record those combinations that have already been examined. In this thesis, a simple pattern matrix recorded switching combinations already examined. However, as the number of switches increases, this method would require a large sparse matrix that would waste computer memory. Hence, a more efficient method is needed.

Because the WatDist algorithm was originally implemented in FORTRAN, many of its structures do not take advantage of the use of pointers. It is not clear that using pointers would reduce computation times, but an examination of this issue is warranted.

References

[1] Billinton, R. and Allan, R.N., Reliability Evaluation of Power Systems, Pitman Advanced Publishing Program, London, 1984.

[2] A. Merlin and H. Back, "Search for a minimal-loss operating spanning tree configuration in an urban power distribution system," *Proceedings of the 5th Power System Computation Conference*, Cambridge, U.K., 1975, pp. 1-18.

[3] Turan Gönen, *Electric Power Distribution System Engineering*, McGraw-Hill, Toronto, 1986.

[4] J.B. Bunch, R.D. Miller and J.E. Wheeler, "Distribution System Integrated Voltage and Reactive Power Control," *IEEE Transactions on Power Apparatus and Systems*, vol PAS-101, no. 2, February, 1982, pp. 284-289.

[5] J.J. Grainger and T.J. Kendrew, "Evaluation of Technical Losses on Electric Distribution Systems," *Proceedings of the 10th International Conference on Electricity Distribution*, Brighton, United Kingdom, 8-12 May 1989, pp. 488-493.

[6] W.M. Ritchie, P.W. Beard, A. Barker and D.G.T. Lewis, "Loss Reduction - an overview of the problems and solutions," *Power Technology International*, 1988, pp. 191-194.

[7] R.B. Mason, "Switching over to savings: Toronto Hydro's switch to 13.8 kV saves money and trees," *Electrical Business*, Vol. 31, No. 3, March 1995, pp. 12-13.

[8] Personal correspondence with Colin Clark, Superintendent of Stations and Metering, Ottawa Hydro, March 26, 1996.

[9] J.G. De Steese, S. B. Merrick and B.W. Kennedy, "Estimating Methodology for a Large Regional Application of Conservation Voltage Reduction," *IEEE Transactions on Power Systems*, Vol. 5, No. 3, August 1990, pp. 862-870.

[10] Barry Kennedy and R.H. Fletcher, "Conservation Voltage Reduction (CVR) at Snohomish County PUD," *IEEE Transactions on Power Systems*, Vol. 6, No. 3, August 1991, pp. 986-997.

[11] Theodore Drupa and Haukur Asgeirsson, "The Effects of Reduced Voltage on Distribution Circuit Loads", *IEEE Transactions on Power Systems*, Vol. PWRS-2, No. 4, November 1987, pp. 1013-1018.

[12] CAN3-C235-83, Preferred Voltage Levels for AC Systems, 0 to 50,000 V (Canadian Standards Association).

[13] Power Systems Engineering Committee of the IEEE Industry Applications Society, *IEEE Recommended Practice for Electric Power Distribution for Industrial Plants*, The Institute of Electrical and Electronics Engineers, New York, NY, 1986.

[14] *Ontario Hydro Energy Management Series - Rates and Billing*, Ontario Hydro Publication 11-200200 REV:11/91 5k.

[15] R.E. Lee and C.L. Brooks, "A Method and its Application to Evaluate Automated Distribution Control," *IEEE Transactions on Power Delivery*, vol. 3, no. 3, July 1988, pp. 1232-1240.

[16] R. Taleski and D. Rajicic, "Distribution Network Reconfiguration for Energy Loss Reduction," *Proceedings of the 1996 IEEE/PES Winter Meeting*, January 21-26, 1996, Baltimore, Md..

[17] D.L. Brown, J.W. Skeen, P. Daryani and F.A. Rahimi, "Prospects for Distribution Automation at Pacific Gas & Electric Company," *IEEE Transactions on Power Delivery*, vol. 6, no. 4, 1991, pp. 1946-1954.

[18] R.P. Broadwater, A.H. Khan, H.E. Shaalan and R.E. Lee, "Time Varying Load Analysis to Reduce Distribution Losses through Reconfiguration," *IEEE Transactions on Power Delivery*, vol. 8, no. 1, Jan 1993, pp. 294-300.

[19] Y.Y. Hsu, Y. Jwo-Hwu, S.S. Liu, Y.W. Chen, H.C. Feng and Y.M. Lee, "Transformer and Feeder Load Balancing using a Heuristic Search Approach," *IEEE Transactions on Power Systems*, vol. 8, no. 1, Feb 1993, pp. 184-190.

[20] K. Aoki, K. Nara, T. Satoh, M. Itoh, H. Kuwabara, M. Kitagawa and K. Yamanaka, "Totally Automated Switching in Distribution Systems," *IEEE Transactions on Power Delivery*, vol. 5, no. 1, 1990, pp. 514-526.

[21] P.A. Gnadt and J.S. Lawler, *Automating Electric Utility Distribution Systems: the Athens Automation and Control Experiment*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[22] R.J. Sarfi, M.M.A. Salama and A.Y. Chikhani, "Distribution System Reconfiguration for Loss Reduction: An Algorithm Based on Network Partitioning Theory," *Proceedings of PICA 1995*, May 7-12, Salt Lake City, Utah, pp. 503-509.

[23] I. Roytelman and S.M. Shahidehpour, "Practical Aspects of Distribution Automation in Normal and Emergency Conditions," *IEEE Transactions on Power Delivery*, Vol. 8. No. 4, October 1993, pp. 2002-2008.

[24] I. Roytelman, B.K. Wee and R.L. Lugta, "Volt/Var Control Algorithms for Modern Distribution Management System," *IEEE Transactions on Power Apparatus and Systems*, Vol. 10, No. 3, August, 1995, pp. 1454-1460.

[25] Arthur R. Bergen, *Power Systems Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

[26] W.D. Stevenson, *Elements of Power System Analysis*, McGraw-Hill, Toronto, 1975.

[27] T.P. Wagner, A.Y. Chikhani and R. Hackam, "Feeder Reconfiguration for Loss Reduction: An Application of Distribution Automation," *IEEE Transactions on Power Delivery*, vol. 6, no. 4, 1991, pp. 1992-1931.

[28] S. Civanlar, J.J. Grainger, H. Yin and S.S.H. Lee, "Distribution Reconfiguration for Loss Reduction," *IEEE Transactions on Power Delivery*, vol. 3, no. 3, 1988, pp. 1217-1223.

[29] D. Shirmohammadi and H.W. Hong, "Reconfiguration of Electric Distribution Networks for Resistive Line Losses," *IEEE Transactions on Power Delivery*, vol. 4, no. 2, 1989, pp. 1492-1498.

[30] 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, no. 3, 1992, pp. 1484-1491.

[31] V. Borozan, D. Rajicic and R. Ackovski, "Improved Method for Loss Minimization in Distribution Networks," *IEEE Transactions on Power Systems*, Vol. 10, No. 3, August, 1995, pp. 1420-1425.

[32] C.A. Castro and A.A. Watanabe, "An Efficient Reconfiguration Algorithm for Loss Reduction of Distribution Systems," *Electric Power Systems Research*, vol. 18, 1990, pp. 137-144.

[33] M.E. Baran and F.F. Wu, "Network Reconfiguration in Distribution Systems for Loss Reduction and Load Balancing," *IEEE Transactions on Power Delivery*, vol. 4, no. 2, 1989, pp. 1401-1407.

[34] J.-Y. Fan, L. Zhang and J.D. MacDonald, "Distribution Network Reconfiguration: Single Loop Optimization," *Proceedings of the 1996 IEEE/PES Winter Meeting*, January 21-26, 1996, Baltimore, Md.

[35] I. Roytelman, V. Melnick, S.S.H. Lee and R.L. Lugta, " Multi-Objective Feeder Reconfiguration by Distribution Management System," *Proceedings of the 1995 IEEE Power Industry Computer Applications Conference*, Salt Lake City, Utah, May 7-12, 1995.

[36] C.C. Liu, S.J. Lee and K. Vu, "Loss Minimization of Distribution Feeders: Optimality and Algorithms," *IEEE Transactions on Power Delivery*, vol. 4, no. 2, 1989, pp. 1281-1289.

[37] K. Aoki, H. Kuwabara, T. Satoh, M. Kanezashi, "An Efficient Algorithm for Load Balancing of Transformers and Feeders by Switch Operation in Large Scale Distribution Systems," *IEEE Transactions on Power Delivery*, vol. 3, no. 4, 1987, pp. 1865-1872.

[38] G.J. Peponis, M.P. Papadopoulos and N.D. Hatziargyriou, "Distribution Network Reconfiguration to Minimize Resistive Line Losses," *IEEE Transactions on Power Delivery*, Vol. 10, No. 3, July, 1995, pp. 1338-1342.

[39] H. Lee Willis, Hahn Tram, Michael V. Engel and Linda Finley, "Selecting and Applying Distribution Optimization Methods," *IEEE Computer Applications in Power*, Vol. 9, No. 1, January 1996, pp. 12-17.

[40] K. Aoki, T. Ichimori, M. Kanezashi, "Normal State Optimal Load Allocation in Distributions Systems," *IEEE Transactions on Power Delivery*, vol. PWRD-2, no. 1, 1987, pp. 147-155.

[41] C.S. Chen, M.Y. Cho and Y.W. Chen, "Development of Simplified Loss Models for Distribution System Analysis," *IEEE Transactions on Power Delivery*, vol. 9, no. 3, 1994, pp. 1545-1551.

[42] V. Glamocanin, "Optimal Loss Reduction of Distribution Networks," *IEEE Transactions on Power Systems*, vol. 5., no. 3, 1991, pp. 774-782.

[43] C.T. Huddleston, R.P. Broadwater and A. Chandrasekaran, "Reconfiguration Algorithm for Minimizing Losses in Radial Electric Distribution Systems," *Electric Power Systems Research*, vol. 18, 1990, pp. 57-60.

[44] C.S. Chen and M.Y. Cho, "Energy Loss Reduction by Critical Switches," *IEEE Transactions on Power Delivery*, vol. 8, no. 3, 1993, pp. 1246-1253.

[45] N.D.R. Sarma and K.S. Prakasa Rao, "A new 0-1 integer programming method of feeder reconfiguration for loss minimization in distribution systems," *Electric Power Systems Research*, Vol. 33, 1995, pp. 125-131.

[46] J.K. Delson and S.M Shahidehpour, "Linear Programming Applications to Power System Economics, Planning and Operations," *IEEE Transactions on Power Delivery*, vol. 7, no. 4, 1992, pp. 1155-1163.

[47] K.H. Abdul-Rahman and S.M. Shadidehpour, "A Fuzzy-Based Optimal Reactive Power Control," *IEEE Transactions on Power Systems*, vol. 8, no. 2, May 1993, pp. 662-670.

[48] K. Kim, Y. Ko and K.H. Hung, "Artificial Neural Network Based Feeder Reconfiguration for Loss Reduction in Distribution Systems," *IEEE Transactions on Power Delivery*, vol. 8, no. 3, July 1993, pp. 1356-1366.

[49] T. Taylor and Lubkeman, "Implementation of Heuristic Search Strategies for Distribution Feeder Reconfiguration," *IEEE Transactions on Power Delivery*, vol. 5, no. 1, 1990, pp. 239-246.

[50] G. Chang, J. Zrida and J.D. Birdwell, "Knowledge-Based Distribution System Analysis and Reconfiguration," *IEEE Transactions on Power Systems*, vol. 5, no. 3, August, 1990, pp. 744-749.

[51] C.S. Chang and T.S. Chung, "An Expert System for On-Line Security - Economic Load Allocation on Distribution Systems, " *IEEE Transactions on Power Delivery*, vol. 5, no. 1, 1990, pp. 467-473.

[52] Robert J. Sarfi, *Reduction of Distribution System Losses and Improvement of Service Quality: A Novel Application Based upon a Coupling of Knowledge-Based and Numerical Methods*, PhD Thesis, University of Waterloo, Waterloo, Canada, 1996.

[53] Hong-Chan Chang and Cheng-Chien Kuo, "Network reconfiguration in distribution systems using simulated annealing," *Electric Power Systems Research*, vol. 29, 1994, pp. 227-238.

[54] H.D. Chiang and R. Jean-Jumeau, "Optimal Network Reconfiguration in Distribution Systems, Part I: A New Formulation and a Solution Methodology, " *IEEE Transactions on Power Delivery*, vol. 5, no. 4, 1990, pp. 1902-1909.

[55] H.D. Chiang and R. Jean-Jumeau, "Optimal Network Reconfiguration in Distribution Systems, Part II: A Solution Algorithm and Numerical Results, " *IEEE Transactions on Power Delivery*, vol. 5, no. 3, 1990, pp. 1568-1575.

[56] K. Nara, A. Shiose, M. Kitagawa and T. Ishihara, "Implementation of Genetic Algorithm for Distribution Systems Loss Minimum Re-Configuration," *IEEE Transactions on Power* Systems, vol. 7, no. 3, August 1992, pp. 1044-1051.

[57] K. Nara and M. Kitagawa, "Distribution System Copper and Iron Loss Minimization by Genetic Algorithm," *Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems*, Yokohama, Japan, April 19-22, 1993, pp. 151-156.

[58] V.S.S. Vankayala and N.D. Rao, "Artificial neural networks and their applications to power systems - a bibliographical survey, " *Electric Power Systems Research*, vol. 28, no. 1, 1993, pp. 67-79.

[59] James A. Freeman and David M. Skapura, *Neural Networks - Algorithms, Applications and Programming Techniques*, Addison-Wesley Publishing Company, Toronto, 1991.

[60] R.J. Sarfi, M.M.A. Salama and A.Y. Chikhani, "A survey of the state of the art in distribution system reconfiguration for system loss reduction," *Electric Power Systems Research*, vol. 31, no.1, 1994, pp. 61-70.

[61] Zadeh, L.A., "Fuzzy Sets," *Information and Control*, vol. 8, 1965, pp. 338-353.

[62] P. Jackson, *Introduction to Expert Systems*, Addison-Wesley, Don Mills, Ontario, 1990.

[63] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, "Equation for state calculations by fast computing machines," *Journal of Chemical Physics*, vol 21, 1953, pp. 1087-1092.

[64] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol 220, 1983, pp. 671-680.

[65] D.B. Fogel, "An Introduction to Simulated Evolutionary Computing," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, January 1994, pp. 3-14.

[66] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company, Don Mills, 1989.

[67] J. Holland, "Genetic Algorithms," *Scientific American*, July 1992, pp. 66-72.

[68] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, New York, 1992.

[69] M. Srinivas and L.M. Patnaik, "Genetic Algorithms: A Survey," *IEEE Computer*, vol. 27 , no. 6, 1994, pp. 17-26.

[70] Edward Tsang, *Foundations of Constraint Satisfaction*, Academic Press, Toronto, 1993.

[71] Judea Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley Publishing Company, Don Mills, Ontario, 1985.

[72] P.E. Hart, N.J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on SSC*, Vol. 4, 1968, pp. 273-281.

[73] Herbert Schildt, *C: The Complete Reference,*, McGraw-Hill, Berkeley, California, 1988.

[74] Steven Minton, Mark Johnston, Andrew Phillips and Philip Laird, "Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems," *Artificial Intelligence*, Vol. 58, 1992, pp. 161-205.

[75] J.G.M. Petit-Pas, *Unconstrained Global Optimization: Theory, Algorithms and Applications*, MSc Thesis, Royal Military College of Canada, Kingston, Ontario, 1989.

[76] R. Cespedes, "New Method for the Analysis of Distribution Networks," *IEEE Transactions on Power Delivery*, vol. 5, no. 1, January 1990, pp. 391-396.

[77] Glenn W. Stagg and Ahmed H. El-Abiad, *Computer Methods in Power System Analysis*, McGraw-Hill Book Company, Toronto, 1968.

[78] N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, Englewood Cliffs, N.J., 1974.

[79] Bert Selman, Hector Levesque and David Mitchell, "A New Method for Solving Hard Satisfiability Problems," *Proceedings of the Tenth National Conference on Artificial Intelligence*, July 12-16, 1992, San Jose, California, pp. 440 - 446.

[80] H. Frank and I.T. Frisch, "Network Analysis," *Scientific American*, Vol. 223, July 1970, pp. 94-103.

[81] B.R. Gungor, *Power Systems*, Harcourt Brace Jovanovich, Toronto, 1988.

[82] IEEE Tutorial Course, *Application and Coordination of Reclosers, Sectionalizers, and Fuses*, IEEE Publication 80 EH157-8-PWR, 1980.

[83] Robert P. Broadwater, Jeffrey C. Thompson, Robert E. Lee and Hamidreza Maghdan-D, "Computer-Aided Protection System Design with Reconfiguration," *IEEE Transactions on Power Delivery*, Vol. 6, No. 1, January 1991, pp. 260-266.

[84] Yuan-Yih Hsu and Yi Jwo-Hwu, "Planning of Distribution Feeder Reconfiguration with Protective Device Coordination," *IEEE Transactions on Power Delivery*, Vol. 8, No. 3, July 1992, pp. 1340-1347.

[85] G.J. Peponis, M.P. Papadopoulos and N.D. Hatziagyriou, "Optimal Operation of Distribution Networks," *IEEE Transactions on Power Systems*, Vol. 11, No. 1, February 1996, pp. 59-67.

[86] J.F. Witte, S.R. Mendis, M.T. Bishop and J.A. Kischefsky, "Computer-Aided Recloser Applications for Distribution Systems," *IEEE Computer Applications in Power*, Vol. 5, No. 3, July 1992, pp. 27-32.

[87] S.H. Horowitz, A.G. Phadke and J.S. Thorp, "Adaptive Transmission System Relaying," *IEEE Transactions on Power Delivery*, Vol. 3, No. 4, October 1988, pp. 1436-1445.

[88] K. Aoki, K. Nara, T. Satoh, M. Itoh, H. Kuwabara, M. Kitagawa and K. Yamanaka, "Totally Automated Switching in Distribution Systems," *IEEE Transactions on Power Delivery*, Vol. 5, No. 1, January 1990, pp. 314-320.

[89] T.J.E. Miller, *Reactive Power Control in Electric Systems*, John Wiley & Sons, Toronto, 1982.