# NETWORK RECONFIGURATION FOR LOSS REDUCTION IN THREE-PHASE POWER DISTRIBUTION SYSTEMS

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Ray Daniel Zimmerman

May 1992

# ABSTRACT

Power distribution systems typically have tie and sectionalizing switches whose states determine the topological configuration of the network. The system configuration affects the efficiency with which the power supplied by the substation is transferred to the load. Power companies are interested in finding the most efficient configuration, the one which minimizes the real power loss of their three-phase distribution systems.

In this thesis the network reconfiguration problem is formulated as single objective optimization problem with equality and inequality constraints. The proposed solution to this problem is based on a general combinatorial optimization algorithm known as simulated annealing. To ensure that a solution is feasible it must satisfy Kirchhoff's voltage and current laws, which in a three-phase distribution system can be expressed as the three-phase power flow equations. The derivation of these equations is presented along with a summary of related three-phase system modeling.

The simulated annealing algorithm is described in a general context and then applied specifically to the network reconfiguration problem. Also presented here is a description of the implementation of this solution algorithm in a C language program.

This program was tested on a Sun workstation, given an example system with 147 buses and 12 switches. The algorithm converged to the optimal solution in a matter of minutes demonstrating the feasibility of using sim-

ulated annealing to solve the problem of network reconfiguration for loss reduction in a three-phase power distribution system. These results provide the basis for the extension of existing methods for single-phase or balanced systems to the more complex and increasingly more necessary three-phase unbalanced case.

# Biographical Sketch

Ray Daniel Zimmerman was born in Ephrata, PA on December 17, 1965. Four years later he moved with his family to a chicken farm in rural Lancaster County, PA where he lived until he began studying Electrical Engineering in September of 1984. As an undergraduate at Drexel University in Philadelphia, PA he participated in a cooperative education program which involved working for six month periods at each of the following companies: IBM Corporation, Research Triangle Park, NC, Evaluation Associates, Bala Cynwyd, PA, and UNISYS Corporation, Tredyffrin, PA. In each of these positions he did various computer hardware and software related tasks. In 1986 he was awarded the Eastman-Kodak Company Scholarship. He received a Bachelor of Science degree in Electrical Engineering from Drexel University in June 1989. In August of the same year he began graduate studies in Electrical Engineering at Cornell University in Ithaca, NY.

to Esther

# Acknowledgments

*O Lord my God, I will give you thanks forever.* Psalm 30:12

I want to give my appreciation to my advisor, Dr. Hsiao-Dong Chiang, for his invaluable help in getting me started on this project, his constant generosity in providing the necessary tools to do the work, and the many group dinners (which served as motivation). I thank Dr. James Thorp for his role on my Special Committee as well. I would like to acknowledge Dr. Jianzhong Tong and the Paralogix Corporation for their respective contributions to the load flow program. My appreciation also goes to René Jean-Jumeau and Pauline Bennett for their encouragement and experience.

My deepest appreciation go to all of my family and friends who prayed for me so faithfully during the time of the writing of this thesis. My parents have been a constant and invaluable support to me throughout every stage of my education and certainly no less during the writing of this thesis. Thanks, Mom and Dad. Thanks also to Ernie for proofreading the final draft and offering helpful stylistic comments. A very special thanks to Esther for her love, prayers, and many letters without which I'm sure I would have despaired. Muchísimas gracias. I'd also like to acknowledge my housemates Karl Johnson, Tom Krauss, and Mark Lattery for their constant support and their great patience with me during the month that I didn't exist around the house. Special thanks go to Tom for his helpful comments on the first draft. Thanks also to my entire family at Covenant Church, especially Ray and Gretchen Crognale, for their love and prayers.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

In the past decade or so, with the advances in communication and data processing technology, electric utility companies have become very interested in *distribution automation*. It is apparent that with the increasing complexity of power distribution systems, it is becoming essential to automate some tasks that have always been done manually. It has also been estimated that utilities could save as much as 10% of their annual maintenance and operating expenses by taking advantage of this technology [14].

One important area in which distribution automation is being applied is the area of *network reconfiguration*. Network reconfiguration refers to the closing and opening of switches in a power distribution system in order to alter the network topology, and thus the flow of power from the substation to the customers. There are two primary reasons to reconfigure a distribution network during normal operation. Depending on the current loading conditions, reconfiguration may become necessary in order to eliminate overloads on specific system components such as transformers or line sections. In this case it is known as *load balancing*. As the loading conditions on the system change it may also become profitable to reconfigure in order to reduce the real power losses in the network. This is usually referred to as network reconfiguration for *loss reduction* and is the topic of this thesis.

Network reconfiguration in both of these cases can be classified as a minimal spanning tree problem, which is known to be an NP–complete combinatorial optimization problem. A method is needed to quickly find the network configuration which minimizes the total real power loss of the network while satisfying all of the system constraints. Several approaches have been applied to the solution of this problem with varying degrees of success. Heuristic methods [3; 13; 20] have been used successfully to find sub-optimal solutions rapidly. The genetic algorithm [19] and simulated annealing [11; 12], which require much more computation time, have been used to find optimal solutions. It seems that these methods have only been applied to relatively small, balanced, or single-phase distribution systems. Power utility companies currently need an algorithm which can be applied to their large three-phase *unbalanced* distribution systems.

This thesis presents a method based on simulated annealing. In Chapter 2 the problem is formulated as single objective optimization problem with equality and inequality constraints. The three-phase power flow problem, one of the important elements in any network reconfiguration method, is presented in a general formulation in Chapter 3 along with a brief discussion of solution techniques. Chapter 4 describes the simulated annealing algorithm in a general context as a tool for solving combinatorial optimization problems. Chapter 5 gives a description of how simulated annealing can be applied to the problem of network reconfiguration for loss reduction in a three-phase distribution system. Based on the solution methodology presented in Chapter 5, a program was written in C language implementing the algorithm for solving for the optimal network configuration. This program was run on a 147 bus example system and the results of this simulation are presented in Chapter 6. The last chapter discusses some of the conclusions drawn from this study and presents some ideas for extending the work covered in this thesis.

# Chapter 2

# Problem Formulation

In order to remain competitive, it is becoming more and more important for power distribution companies to be able to meet efficiently the demands of their customers. This means that one of their goals is to be able to find an operating state for a large, three-phase, unbalanced distribution network which minimizes the cost for the power company supplying the power, while satisfying the requirements of the customer. This chapter introduces some useful notation and presents a formulation of the network reconfiguration problem for loss reduction.

Let $u$ denote the current configuration of a large, three-phase, unbalanced distribution system whose operating state is specified by $x$. Let $f(x, u)$ be the *cost function* (also known as the *objective function*) which gives a relative measure of the cost of operating the system in configuration $u$ and corresponding state $x$. In order for a configuration $u$ to be a valid solution to the problem it must satisfy certain *topological constraints*. The corresponding state $x$ must be consistent with Kirchhoff's current and voltage laws to satisfy the *electrical constraints* and must satisfy the *operational con-*

*straints* of the system by not exceeding the physical limitations of any of the system components. It must also satisfy the customers' real power demands while maintaining bus voltages within appropriate bounds. The customer requirements are also called *load constraints*. The objective is to find a network configuration $u$ which minimizes $f(x, u)$ while satisfying all of the above constraints.

This problem can be given in a very general formulation as a single objective function with equality and inequality constraints

$$\underset{u \in S}{\text{minimize}} \ f(x, u) \tag{2.1}$$

$$\text{such that:} \quad \begin{cases} F(x, u) = \mathbf{0} \\ G(x, u) \leq \mathbf{0} \end{cases} \tag{2.2}$$

where $S$ is the set of all possible network configurations, and $F$ and $G$ are non-linear functions used to express the constraints mentioned above. Any solution $u$ satisfying the constraints of Equation (2.2) is called a *feasible* configuration.

## 2.1 Search Space

The *search space* for this problem is the set of all possible network configurations. Once the general layout[1] of the distribution network is specified, the specific topology is determined by the status of each of the switches in the system. Switches which are normally open are called *tie switches* and normally closed switches are known as *sectionalizing switches*. Specifying

[1] General layout here refers to the connectivity of the network, i.e. which bus is connected to which via which line, transformer, or switch, etc.

the open/closed status of each switch completely characterizes the topology of the network. So if the total number of tie and sectionalizing switches in the system is $n_s$, the current configuration can be represented as a vector $u = [u_1, u_2, \ldots, u_{n_s}]^T$ of individual switch states $u_i \in \{0, 1\}$, $1 \le i \le n_s$, where $u_i = 1$ indicates that switch $i$ is closed, and $u_i = 0$ indicates that it's open. Let the search space, the space of all possible configurations $u$, be denoted by $S = \{0, 1\}^{n_s}$.

## 2.2 State Space

In order to calculate the cost function and check the constraints it is necessary to have complete information on the voltage magnitudes and angles at each bus. This information is included in the state variable $x$.

Let $|V_i| = [|V_i^a|, |V_i^b|, |V_i^c|]^T$ and $\theta_i = [\theta_i^a, \theta_i^b, \theta_i^c]^T$ be the voltage magnitudes and angles respectively for phases $a$, $b$, and $c$ at bus $i$. Given a three-phase distribution network with a total of $n$ buses, where bus 1 is the substation and buses 2, 3, ..., $n$ are the load buses, the state variable can be denoted by $x = [\theta_2, \ldots, \theta_n, |V_2|, \ldots, |V_n|]^T$, and the *state space* is $\mathbb{R}^{6(n-1)}$.

## 2.3 Cost Function

The *cost function* or *objective function*, $f : S \times \mathbb{R}^{6(n-1)} \to \mathbb{R}$, maps the current configuration $u$ and corresponding state $x$ into a real number which gives a relative measure of the cost of that configuration. It is the criterion used for determining whether $u$ is better than any other configuration. If, for instance, there exists a feasible configuration $u \in S$ which yields an operating state $x$ such that $f(x\ ,u\ ) \le f(x, u)$ for all feasible $u \quad S$ and corresponding $x \quad \mathbb{R}^{6(n-1)}$, then the solution $u$ is the *global optimum*. There are many factors which could be considered in evaluating the relative quality of one network configuration over another. In this application the goal is to reduce real power losses in the system, thereby reducing the cost of supplying the necessary power for a given system loading condition.

Let $f(x, u)$ be the sum of the real power losses in each line, transformer, and voltage regulator in the system

$$f(x, u) = \left( \sum_{i=1}^{n_l} P_i^{line} \right) + \left( \sum_{j=1}^{n_t} P_j^{transformer} \right) + \left( \sum_{k=1}^{n_r} P_k^{regulator} \right) \tag{2.3}$$

where $P_i^{line}$, $P_j^{transformer}$, and $P_k^{regulator}$ represent the real power lost in line $i$, transformer $j$, and voltage regulator $k$, respectively, and $n_l$, $n_t$, and $n_r$ are the number of lines, transformers, and voltage regulators, respectively, in the system. Stated another way, the total power loss in the system is the total power input to the system minus the total power delivered to the loads

$$f(x, u) = P_{input} - P_{delivered} \tag{2.4}$$

Given the proper scaling this cost function would give the number of dollars lost due to real power losses in the system. A more complete formulation might also include the cost of switching to configuration $u$ from the current operating configuration.

## 2.4 Constraints

Not every configuration $u$ $S$ is a reasonable solution to the network reconfiguration problem. For example, if all of the switches were put in the *open* state and all bus voltages set to zero, the real power losses in the system would also be zero, but a distribution system operated in this state would obviously cause the utility company to lose customers. So it is necessary to specify which states are *feasible* and which ones are not. As was mentioned earlier, this involves four types of constraints:

1. topological constraints
2. electrical constraints
3. operational constraints
4. load constraints

These constraints can be expressed as equality and inequality constraints as in Equation (2.2).

### 2.4.1 Topological Constraints

The topology or layout of the system is constrained to be the radial configuration which is typical in power distribution networks. This means that no loops are allowed in the network.

The network configuration is also constrained to be a connected topology such that each bus is connected via at least one path to the substation. The combination of these two requirements classifies the feasible topology as a *spanning tree*. Figure 2.1 shows a typical feasible radial configuration of a distribution system with a main feeder and 7 laterals.

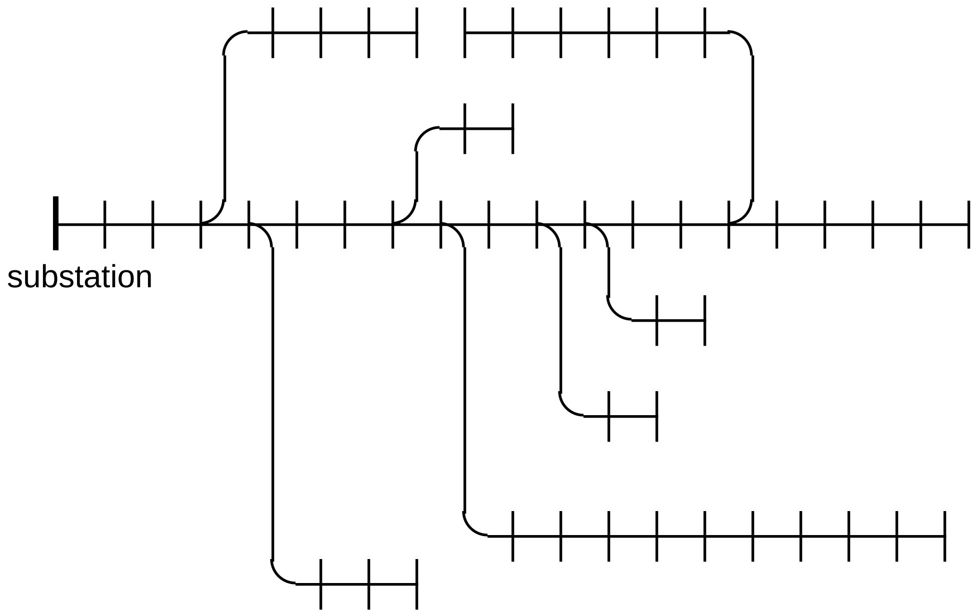


**Figure 2.1** Typical Radial Distribution System

### 2.4.2 Electrical Constraints

Being an electrical circuit, the state of a power system network must also satisfy Kirchhoff's voltage and current laws. Since a distribution system can be quite large, involving thousands of buses, the formulation of these constraints can be rather involved. This topic is treated in much more detail in Chapter 3 which presents a derivation of the three-phase power flow equations. Equation (3.15) shows that these electrical requirements can be expressed in a compact form as the equality constraint, $F(x, u) = \mathbf{0}$, given in Equation (2.2).

### 2.4.3 Operational Constraints

It is possible that the network configuration which theoretically minimizes the real power losses in the system might require one or several of the components in the system to be operated at a level beyond its physical limitations. This obviously must be disallowed. Each line, transformer, and switch in the system has a certain thermal limitation which restricts the maximum allowable current through that component. In general, these physical limitations can be accounted for by constraining line currents, line flows, and bus voltages to lie within appropriate bounds. These operational constraints are inequality constraints which can be included in $G(x, u) \le \mathbf{0}$ of Equation (2.2).

### 2.4.4 Load Constraints

The power company's customers have certain requirements for the electrical power they receive. For example, one expects to get approximately 110 Volts at 60 Hz from a wall outlet. The power company must be able to maintain a certain voltage level at each bus in the system while supplying the power demanded by each customer. This inequality constraint, which requires the voltage magnitude of each phase $p$ at each bus $i$ to lie in the appropriate range,

$$\left|V_i^p\right|_{min} \le \left|V_i^p\right| \le \left|V_i^p\right|_{max} \tag{2.5}$$

can also be included in the inequality constraint in Equation (2.2), namely, $G(x, u) \le \mathbf{0}$.

# Chapter 3

# Three-Phase Distribution Power Flow

One of the most important tools for the power engineer is the *power flow*, or *load flow* study. The power flow study is the basic calculation used to determine the state of a given power system operating at steady-state under the specified conditions of power input, power demand, and network configuration.

In a distribution system there is typically one voltage specified bus, the substation bus, which is connected radially to the load buses. The solution to the power flow problem provides information on the voltage magnitudes and angles at each bus, the real and reactive power supplied or absorbed at the substation, the real and reactive power flows in each line section, and the system losses.

The results of a load flow analysis can be used for operational purposes to evaluate various operating states of an existing system. They can also be used in the planning stages to evaluate possible future systems. In the network reconfiguration problem the load flow study is used to calculate the overall real power loss for a given system configuration in order to rank it against other configurations. The results of the load flow are also used in the evaluation of the electrical, load, and operational constraints.

## 3.1 Three-Phase vs. Single-Phase Power Flow

For certain applications it is not necessary to take into account potential system imbalance, therefore it is sufficient to model the system as a balanced three-phase system. When this is the case, per phase analysis can be used to formulate a single-phase power flow problem.

However, it is not always possible to completely balance the system loads, and transmission line impedances can be unbalanced due to untransposed lines sharing the same right of way. As distribution systems become larger and more complex, it becomes more important to take into account the system imbalance. Some of the effects of system imbalance, according to [2] and [7], are negative sequence currents causing problems with motors, zero sequence currents causing protective relays to malfunction, increased system loss, decreased system capacity, and an increase in inductive coupling between parallel lines and feeders.

In any case, no power system is completely balanced and sometimes the additional complexity of a three-phase load flow study is necessary to model the system closely enough to accurately acquire the information of interest.

## 3.2 Component Models

Realistic mathematical representations for each of the system components are needed in order to achieve accurate and meaningful results from the power flow study. Detailed models for distribution system components such as line sections, shunt elements, cogenerators, transformers, and loads can be found in [8]. A brief summary of some of these typical three-phase models is given below.

### 3.2.1 Conductor Model

The conductors for each of the line sections in the network can be represented by the standard compound $\pi$-equivalent model. Figure 3.1 shows a schematic representation of a line section between bus $i$ and bus $k$. The

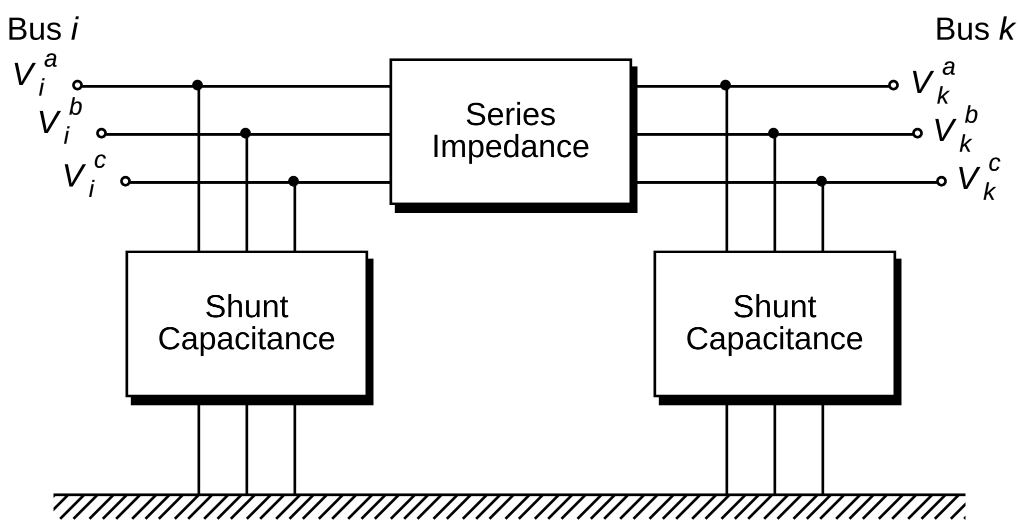


**Figure 3.1** Three-Phase Conductor Model

series impedance of the line is included in the series arm of the $\pi$-equivalent circuit and the line charging effects are accounted for by dividing the total capacitance to ground between the two shunt arms as shown in Figure 3.2. The series impedance and the shunt capacitance for a three-phase line are 3 x 3 complex matrices which take into account the mutual inductive coupling between the phases.

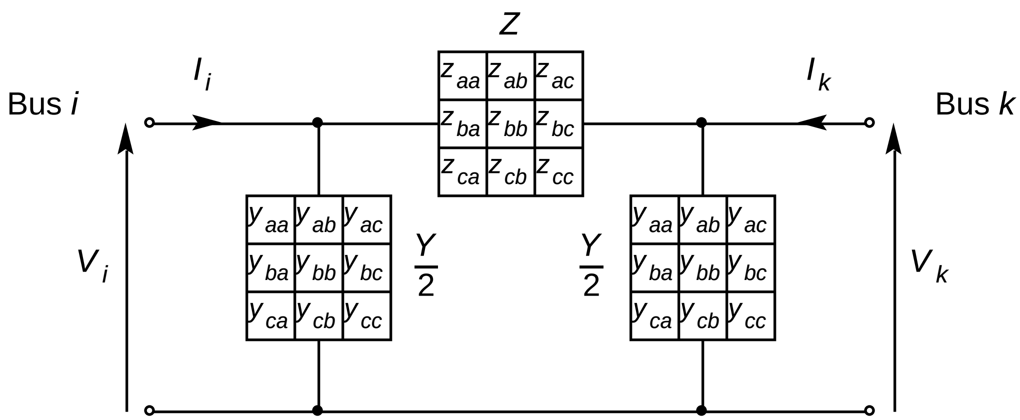


**Figure 3.2** Compound π-equivalent Model for Three-Phase Conductor

If $Z$ and $Y$ are the 3 x 3 matrices representing the series impedance and shunt admittance, respectively, then the admittance matrix $Y_{ik}$ for a three-phase conductor between buses $i$ and $k$ is the 6 x 6 matrix

$$Y_{ik} = \begin{bmatrix} Z^{-1} + \frac{1}{2}Y & -Z^{-1} \\ -Z^{-1} & Z^{-1} + \frac{1}{2}Y \end{bmatrix} \tag{3.1}$$

In other words, the voltages and currents labeled by the 3 x 1 vectors $V_i$, $V_k$, $I_i$, and $I_k$ in Figure 3.2, can be related by

$$\begin{bmatrix} I_i \\ I_k \end{bmatrix} = Y_{ik} \begin{bmatrix} V_i \\ V_k \end{bmatrix} \tag{3.2}$$

In the case of secondary distribution networks, which typically have relatively low voltage levels, the line charging effects may be negligible and the model is often simplified by neglecting the shunt capacitance.

### 3.2.2 Shunt Capacitor Model

Shunt capacitors, which act as sources of reactive power, are often placed at strategic locations throughout the network where they might be useful for power factor or voltage profile improvement, or VAR compensation. A

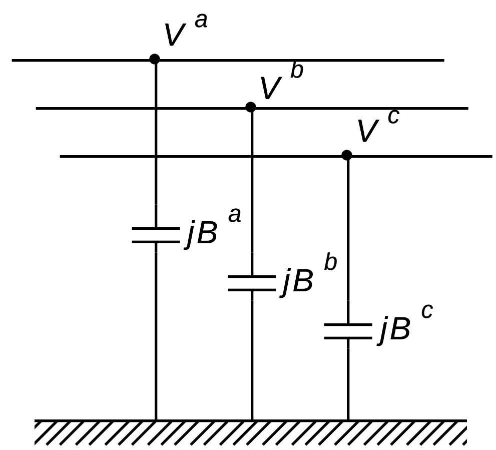


**Figure 3.3** Three-Phase Shunt Capacitor Model

three-phase shunt capacitor, as shown in Figure 3.3, can be modeled by treating each of the three phases as a constant admittance.

### 3.2.3 Cogenerator Model

Cogenerators are becoming much more common in distribution systems as many industries attempt to save money by converting heat generated by other processes into usable electrical energy. Typically these cogenerators are designed to maintain a constant real power output at a constant power factor. In other words, they are modeled as constant complex power devices.

### 3.2.4 Transformer Model

It is important to have a realistic three-phase representation of the transformers found in distribution systems in order to analyze their effects on system loss. This model should take into account transformer core losses since these losses can account for a significant percentage of the power losses in a distribution system. The basic model for the three-phase transformer can be represented by the diagram in Figure 3.4. The exact form of

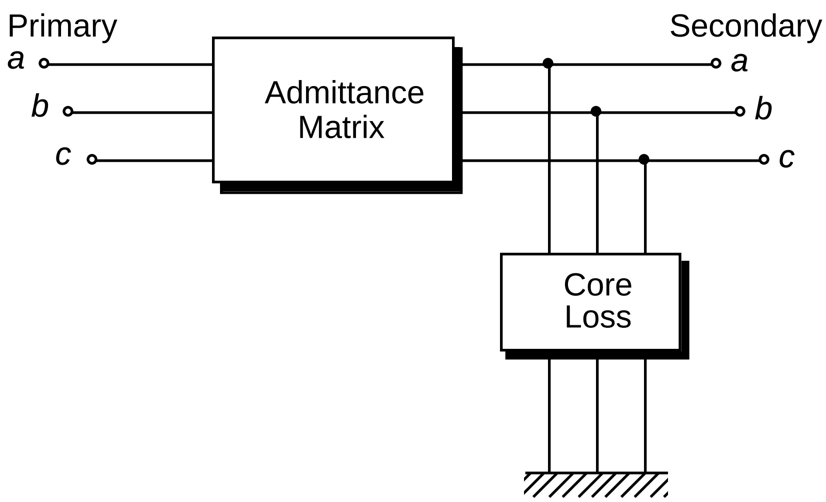


**Figure 3.4** Three-Phase Transformer Model

the admittance matrix depends on the type of connection and is detailed, along with the core losses, in [8].

### 3.2.5 Load Model

For the purpose of analyzing the steady-state behavior of a distribution system the loads or demands are assumed to be constant complex power elements. In other words, load buses are modeled as PQ specified buses.

## 3.3 Three-Phase Power Flow Equations

Given a system with a total of $n$ buses, define a bus voltage vector, $V_{bus}$, and a bus injection current vector, $I_{bus}$, as

$$\begin{aligned} V_{bus} &= [V_1^a, V_1^b, V_1^c, V_2^a, V_2^b, V_2^c, \quad , V_n^a, V_n^b, V_n^c]^T \text{, and} \\ I_{bus} &= [I_1^a, I_1^b, I_1^c, I_2^a, I_2^b, I_2^c, \quad , I_n^a, I_n^b, I_n^c]^T \end{aligned} \tag{3.3}$$

where $V_i^p$ and $I_i^p$ are complex values representing the voltage and injected current, respectively, of phase $p$ at bus $i$. With the appropriate models for each of the system components, it is now possible to construct $Y_{bus}$, the system admittance matrix[1] which relates the bus voltages and currents according to Kirchhoff's voltage and current laws

$$I_{bus} = Y_{bus} V_{bus} \tag{3.4}$$

$Y_{bus} = [Y_{ik}^{pm}]$ is a $3n$ x $3n$ complex matrix whose element $Y_{ik}^{pm}$ relates the voltage $V_k^m$ to the current $I_i^p$.

The goal of the load flow study is to determine the values of the $V_{bus}$ vector based on the specified network configuration and loading conditions. The

[1] In the network reconfiguration problem, as formulated in Chapter 2, the system admittance matrix $Y_{bus}$ is a function of the current configuration $u$.

representation of the power system network given by Equation (3.4) provides a framework for formulating the power flow equations and developing algorithms for solving them.

Rewriting Equation (3.4) as a summation of the individual matrix and vector components gives the injected current of phase $p$ at bus $i$ as

$$I_i^p = \sum_{k=1}^{n} \sum_{m=a}^{c} Y_{ik}^{pm} V_k^m \tag{3.5}$$

An expression for the corresponding injected power then can be derived as follows:

$$\begin{aligned} S_i^p &= V_i^p I_i^p \\ &= V_i^p \left( \sum_{k=1}^{n} \sum_{m=a}^{c} Y_{ik}^{pm} V_k^m \right) \\ &= \sum_{k=1}^{n} \sum_{m=a}^{c} V_i^p (V_k^m) \quad (Y_{ik}^{pm}) \\ &= \sum_{k=1}^{n} \sum_{m=a}^{c} \left( \left|V_i^p\right| e^{j\theta_i^p} \right) \left( \left|V_k^m\right| e^{-j\theta_k^m} \right) (Y_{ik}^{pm}) \end{aligned} \tag{3.6}$$

Here the bus voltages have been expressed in polar form as $V_i^p = \left|V_i^p\right| e^{j\theta_i^p}$. Applying Euler's formula, and rearranging terms in Equation (3.6) yields

$$S_i^p = \sum_{k=1}^{n} \sum_{m=a}^{c} \left|V_i^p\right| \left|V_k^m\right| (\cos\theta_{ik}^{pm} + j\sin\theta_{ik}^{pm}) \, (G_{ik}^{pm} - jB_{ik}^{pm}) \tag{3.7}$$

where $\theta_{ik}^{pm} = \theta_i^p - \theta_k^m$, and $Y_{ik}^{pm} = G_{ik}^{pm} + jB_{ik}^{pm}$. Separating this expression into real and imaginary parts then gives the following expression of the injected real and reactive powers for phase $p$ at bus $i$ in terms of the phase voltage magnitudes and angles

$$S_i^p = P_i^p + jQ_i^p \tag{3.8}$$

where

$$P_i^p = \left|V_i^p\right| \sum_{k=1}^{n} \sum_{m=a}^{c} \left|V_k^m\right| [G_{ik}^{pm} \cos\theta_{ik}^{pm} + B_{ik}^{pm} \sin\theta_{ik}^{pm}] \tag{3.9}$$

$$Q_i^p = \left|V_i^p\right| \sum_{k=1}^{n} \sum_{m=a}^{c} \left|V_k^m\right| [G_{ik}^{pm} \sin\theta_{ik}^{pm} - B_{ik}^{pm} \cos\theta_{ik}^{pm}] \tag{3.10}$$

In a power system operating at steady-state, Equations (3.9) and (3.10), known as the *three-phase power flow equations*, must hold for each of the three phases ($p = a, b, c$) at each bus ($i = 1, 2, \ldots, n$).

## 3.4 Problem Formulation

In general, each bus in the system can be classified into one of three categories:

1. *PQ bus*: injected real and reactive powers are known, voltage magnitude and angle are unknown.
2. *PV bus*: injected real power and voltage magnitude are known, injected reactive power and voltage angle are unknown.[2]
3. *Slack* or *swing bus*: voltage magnitude and angle are known, injected real and reactive powers are unknown.[3]

[2] PV buses are used for generator buses in transmission systems, and are not typically used in distribution systems.

In a typical distribution system all buses are PQ buses except the one voltage specified bus, the substation, which is taken as the swing bus.

Given a three-phase distribution network which has a total of $n$ buses, where bus 1 is the substation[4] and buses 2, 3, …, $n$ are the load buses, let $x = [\theta_2, \ldots, \theta_n, |V_2|, \ldots, |V_n|]^T$ be the vector of unknowns, where $\theta_i = [\theta_i^a, \theta_i^b, \theta_i^c]^T$ and $|V_i| = [|V_i^a|, |V_i^b|, |V_i^c|]^T$. Then the three-phase power flow problem can be stated as follows:

- Find $x$ such that the real and reactive power mismatches, $\Delta P(x)$ and $\Delta Q(x)$, are zero at each of the PQ buses.

In other words, find $x$ such that

$$\Delta P_i^p(x) = 0 \tag{3.11}$$

$$\Delta Q_i^p(x) = 0 \tag{3.12}$$

for $i = 2, 3, \ldots, n$ and $p = a, b, c$, where the power mismatches are given by

$$\Delta P_i^p(x) = P_i^p - |V_i^p| \sum_{k=1}^{n} \sum_{m=a}^{c} |V_k^m| [G_{ik}^{pm} \cos\theta_{ik}^{pm} + B_{ik}^{pm} \sin\theta_{ik}^{pm}] \tag{3.13}$$

$$\Delta Q_j^p(x) = Q_j^p - |V_j^p| \sum_{k=1}^{n} \sum_{m=a}^{c} |V_k^m| [G_{jk}^{pm} \sin\theta_{jk}^{pm} - B_{jk}^{pm} \cos\theta_{jk}^{pm}] \tag{3.14}$$

Keeping in mind that $Y_{bus}$, the system admittance matrix, is a function of the current network configuration $u$, Equations (3.11) - (3.14) can be

[3] By conservation of complex power, the real power losses of the system must be equal to the real power supplied minus the real power delivered to the loads. Therefore the injected real power cannot be arbitrarily specified at every bus. Generally it is specified at all buses *except* the one slack bus.

[4] That is, the swing bus.

expressed in compact form, as in Equation (2.2), as an equality constraint on $x$ and $u$

$$F(x, u) = \mathbf{0} \tag{3.15}$$

After solving this system of $(6n - 6)$ non-linear algebraic equations and $(6n - 6)$ unknowns for $x$, the injected real and reactive power at the slack bus can be found using Equations (3.9) and (3.10). Since at this point all of the bus voltages are known, the line flows and line losses can be calculated using the model for the three-phase line conductor given in Section 3.2.1 “Conductor Model” on page 13.

## 3.5 Comments on Formulation and Solution Algorithms

Since the load flow problem is a system of non-linear algebraic equations, there is no closed form solution. Therefore the algorithms used to solve this problem are iterative in nature. The real and reactive power mismatches are calculated using an initial value for $x$, such as the current operating point or a *flat start* configuration.[5] The mismatch values are then used to update the value of $x$. If the initial guess is close enough to the solution and a good method is used to update $x$ the algorithm will converge to a valid solution. This type of iterative algorithm can be summarized by the steps given in Table 3.1.

At least two methods, the Gauss method and the Newton-Raphson method, are commonly used to solve the power flow equations. The Gauss method

[5] A *flat start* configuration is with $x$ such that $\left|V_i^p\right| = 1$ and $\theta_i^p = 0$ for $i = 2, 3, \quad, n$ and $p = a, b, c$.

**Table 3.1** Iterative Power Flow Solution

| Basic Algorithm | |
|---|---|
| 1 | Assume initial values for $x$. |
| 2 | Calculate $\Delta P$ and $\Delta Q$. |
| 3* | Use $\Delta P$ and $\Delta Q$ to update $x$. |
| 4 | Repeat 2 and 3 until $\Delta P$ and $\Delta Q$ are smaller than some tolerance. |

*The primary differences between algorithms are typically found in this step.

requires fewer calculations per step since Newton-Raphson requires the calculation of a Jacobian matrix, however Newton-Raphson has fast quadratic convergence properties.

Due to the radial nature of the distribution systems being considered, the power flow problem could be reformulated with a reduced set of equations similar to the formulation presented in [9] and [10]. In this case, the unknowns are taken to be the real and reactive power leaving the substation and each of the branching nodes in the system. So for a network with a main feeder and $m$ laterals the number of unknowns is $6(m + 1)$. All other voltages, currents, and power flows can be calculated directly from these quantities. As was demonstrated for the single-phase or balanced case in [9], a Newton or decoupled quasi-Newton method can be used for very fast solution of the radial distribution load flow problem using this type of formulation.

# Chapter 4

# Simulated Annealing

The concept of *simulated annealing* was first introduced in the field of optimization in the early 1980's by Kirkpatrick *et. al.* [15; 16] and independently by Černy [6]. Simulated annealing is a robust, general-purpose combinatorial optimization algorithm based on probabilistic methods which has been applied successfully to many areas such as VLSI circuit design, neural-networks, image processing, code design, and capacitor placement in power systems.

## 4.1 Combinatorial Optimization

A *combinatorial optimization* problem is a minimization or maximization problem which involves finding the *optimal* or "best" solution out of a set of possible alternatives. It can be completely characterized by the *search space* and the *cost function* or *objective function*.

The search space $S$ is a finite or countably infinite set of possible solutions, and the objective function $f : S \to \mathbb{R}$ maps each point in the search space into the real line, to give a measure of how "good" a solution is relative to the others. In the minimization case[1] the desired optimal solution $x_{opt}$ is one for which

$$f(x_{opt}) \le f(x), \text{ for all } x \quad S \tag{4.1}$$

The problem then can be stated simply as

$$\underset{x \quad S}{\text{minimize}} \; f(x) \tag{4.2}$$

The solution $x_{opt}$ is called a *global optimum* and its objective value, the optimal cost, is denoted $f_{opt} = f(x_{opt})$. Since there can be more than one solution satisfying the conditions of Equation (4.1), the set of globally optimal solutions will be denoted by $S_{opt}$.

## 4.2 Analogy to Physical Annealing

The name *simulated annealing* comes from an analogy between combinatorial optimization and the physical process of annealing. In physical annealing a solid is cooled very slowly, starting from a high temperature, in order to achieve a state of minimum internal energy. It is cooled slowly so that *thermal equilibrium* is achieved at each temperature. Thermal equilibrium can be characterized by the Boltzmann distribution

$$P_T\{\mathbf{X} = x\} = \frac{e^{-E_x / k_B T}}{\sum_{\text{all states } i} e^{-E_i / k_B T}} \tag{4.3}$$

where $\mathbf{X}$ is a random variable indicating the current state, $E_x$ is the energy of state $x$, $k_B$ is Boltzmann's constant, and $T$ is temperature.

[1] The maximization problem is analogous and will not be discussed here.

The evolution of the state of a solid in a heat bath toward thermal equilibrium can be efficiently simulated by a simple algorithm based on Monte Carlo techniques which was proposed by Metropolis *et. al.* [18] in 1953. The *Metropolis algorithm* takes the current state $x$, and generates a new state $y$ by applying some small perturbation. The transition from state $x$ to state $y$ is then accepted with probability

$$P_{accept}(x, y) = \begin{cases} 1, & \text{if } E_x - E_y \le 0 \\ e^{-(E_x - E_y)/k_B T}, & \text{if } E_x - E_y > 0 \end{cases} \tag{4.4}$$

If accepted, $y$ becomes the current state and the procedure is repeated. This acceptance rule is known as the *Metropolis criterion*.

Given a particular combinatorial optimization problem let the solution $x$ correspond to the current state of the solid, the cost function $f(x)$ correspond to the energy of the current state, and the control parameter $T$ correspond to the temperature of the solid. The simulated annealing algorithm consists simply of iterating the Metropolis algorithm for decreasing values of the artificial temperature parameter $T$.

**Table 4.1** Simulated vs. Physical Annealing

| **Optimization Problem** | **Physical System** |
|---|---|
| solution $x$ | current state of the solid |
| cost or objective value $f(x)$ | energy of current state |
| control parameter $T$ | temperature |
| optimal solution $x_{opt}$ | ground state |
| simulated annealing | gradual cooling |

Some of the analogies between the thermal process of physical annealing and the artificial process of simulated annealing in a combinatorial optimization problem are summarized in Table 4.1.

## 4.3 The Simulated Annealing Algorithm

Simulated annealing falls into a category of optimization algorithms known as *probabilistic* methods, since there is some randomness involved in determining the path taken in search of the solution. A sequence of solutions are generated by randomly creating a new solution via a perturbation to the current solution and then accepting or rejecting the new point with a certain probability which is dependent on the temperature and the change in the objective function.

### 4.3.1 Acceptance Probability

The probability of accepting a proposed move, known as the *acceptance probability*, is given by the expression

$$P_{accept}(\Delta f, T) = \begin{cases} 1, & \Delta f \leq 0 \\ e^{-\Delta f / T}, & \Delta f > 0 \end{cases} \tag{4.5}$$

or alternatively

$$P_{accept}(\Delta f, T) = \min(1, e^{-\Delta f / T}) \tag{4.6}$$

and is the equivalent of Equation (4.4). Here $\Delta f$ is the change in the cost function corresponding to the proposed change in state, and $T$ is the artificial control parameter representing temperature. The fact that the acceptance probability is exactly 1 when $\Delta f$ is negative indicates that moves which improve the cost are always accepted. However, unlike "greedy" algorithms which allow only decreases in cost, simulated annealing may accept cost increases, enabling the algorithm to escape from local minima. As illustrated in Figure 4.1, the probability of accepting a cost increase is nearly 1 when the temperature parameter $T$ is much greater than the increase in cost. Generally the algorithm is started at a very high tempera-

$$\boxed{P_a(\Delta f, T) = \min(1, e^{-\Delta f / T})}$$

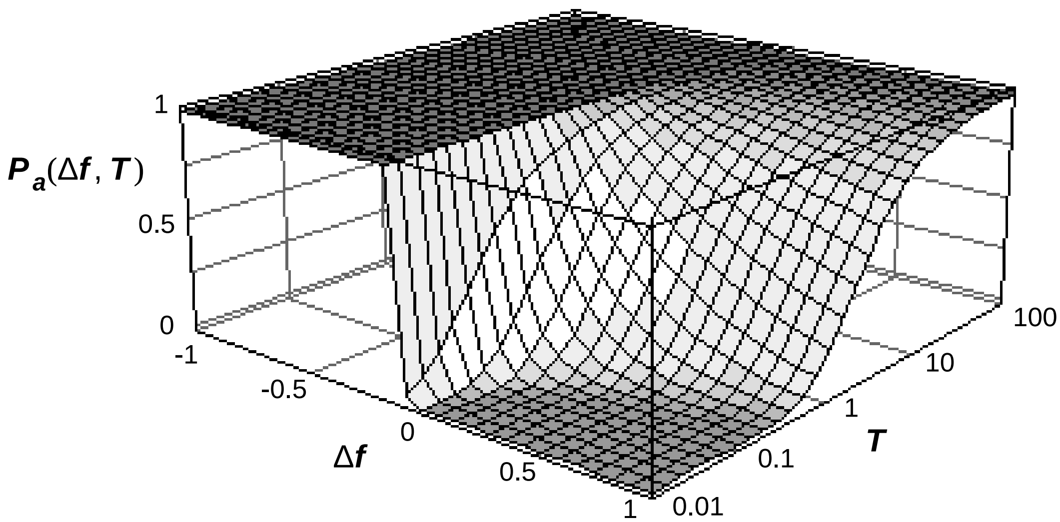


**Figure 4.1** Acceptance Probability vs. Temperature & Change in Objective Function

ture accepting nearly all proposed transitions. As the temperature decreases simulated annealing behaves more and more like a "greedy" algorithm, rarely accepting moves which entail an increase in the objective function.

Intuitively this means that it starts by searching globally, not being restricted by the local terrain of the objective function, but as the temperature is lowered it gravitates toward solutions with lower cost, until finally, at very low temperatures, it is simply doing "greedy" local[2] search.

### 4.3.2 Asymptotic Convergence Characteristics

Given a particular perturbation mechanism, define the *generation probability matrix* $G(T)$ to be the matrix whose $(i, j)^{\text{th}}$ entry is the probability of generating state $j$ from state $i$. Determining the entries of the *acceptance probability matrix* $A(T)$ according to Equation (4.5) or Equation (4.6) yields a matrix whose $(i, j)^{\text{th}}$ entry is the probability of accepting a proposed transition from state $i$ to state $j$. The complete probability of *moving* from state $i$ to state $j$ can be characterized by a *transition probability matrix* $P(T)$ whose $(i, j)^{\text{th}}$ entry can be expressed

$$P_{ij}(T) = \begin{cases} G_{ij}(T) A_{ij}(T), & \text{if } i \neq j \\ 1 - \sum_{k \in S,\, k \neq i} P_{ik}(T), & \text{if } i = j \end{cases} \tag{4.7}$$

With certain conditions on the generation and acceptance probability matrices, as described in [17], the simulated annealing algorithm operating at a fixed temperature $T$ generates a homogeneous Markov chain with transition matrix $P(T)$. Given an infinite number of transitions at a fixed temperature $T$, this Markov chain has a stationary distribution $\mathbf{q}(T)$ whose $i^{\text{th}}$ component, the probability of being in state $i$ after an infinite number of transitions, is given by

[2] *Local* in the same sense that the perturbations used to generate new states are *local* perturbations.

$$\mathbf{q}_i(T) = \frac{e^{-f(i)/T}}{\sum_{k\quad S} e^{-f(k)/T}} \tag{4.8}$$

This distribution is equivalent to the Boltzmann distribution given in Equation (4.3). It can be shown that this stationary distribution yields the following result:

$$\mathbf{q}_i = \lim_{T \to 0} \mathbf{q}_i(T) = \begin{cases} \dfrac{1}{|S_{opt}|}, & \text{if } i \quad S_{opt} \\ 0, & \text{if } i \quad S_{opt} \end{cases} \tag{4.9}$$

In other words, given a suitable perturbation mechanism and an infinite number of transitions, simulated annealing will find a globally optimal solution to an optimization problem with probability 1.

### 4.3.3 Finite Time Approximations

Since it is clearly not possible in practice to run the algorithm for an infinite number of iterations, finite time approximations are used. In the finite time algorithm the details of the approximation are specified by what is known as the *cooling schedule*.

The cooling schedule is very important with respect to the convergence and speed of convergence of the simulated annealing algorithm. It specifies starting value of the temperature parameter, how and when the temperature is updated throughout the annealing process, and when to stop the algorithm. Many different types of cooling schedules have been proposed [17], some very simple and others more complex.

**Table 4.2** Elements of a Cooling Schedule

| | Cooling Schedule Components |
|---|---|
| 1 | Initial Temperature |
| 2 | Temperature Update Procedure |
| 3 | Markov Chain Length |
| 4 | Termination Criterion |

One of the most basic cooling schedules involves setting the initial temperature to some large[3] constant $T_0$ and proceeding as follows:

1. Run the Metropolis algorithm for $n$ iterations, for some constant $n$, to generate a length $n$ Markov chain.
2. If the final state of the last $m$ Markov chains has remained unchanged, where $m$ is a small constant, then stop the algorithm, otherwise continue to step 3.
3. Update the temperature $T$ by multiplying by a constant $\alpha$, where $\alpha$ is between 0 and 1.
4. Go to step 1.

Finite time approximations to the simulated annealing algorithm cannot guarantee convergence to an optimal solution. However in many cases they are able to find near optimal solutions in a reasonable amount of time.

## 4.4 Implementation

One of the appealing features of the simulated annealing algorithm is its ease of implementation. The basic algorithm can be coded in only a few

[3] Large enough that the initial acceptance ratio is nearly 1.

```
x = Initial_State;
f = Cost(x);
T = Initial_Temperature();
do  {
     do  {
          new_x = Apply_Perturbation_To(x);
          Δf = Cost(new_x) - f;
          if ( (Δf<0) OR (random[0,1]<exp(-Δf/T)) )  {
               x = new_x;
               f = f + Δf;
          }
     }  while Not_At_Equilibrium();
     T = Update_Temperature(T);
}  while Exit_Condition_Not_Met();
```

**Figure 4.2** The Algorithm (pseudo C code)

lines, as illustrated by the pseudo C language example in Figure 4.2, and is independent of the particular optimization problem being solved.

There are four basic components which must be specified for each particular optimization problem:

1. The *search space*.
2. The *objective function* or *cost function* to be minimized on the given search space.
3. The *perturbation mechanism* used to generate a new solution from the current one.
4. The *cooling schedule* which includes the initial temperature, the procedure for updating the temperature, and the termination criterion used to determine the stopping point for the algorithm.

The search space and cost function are specified as part of the formulation of the problem, while the perturbation mechanism and cooling schedule deal more with the parameters which control the search.

# Chapter 5

## Implementation of the Solution Algorithm

Based on the problem formulation presented in Chapter 2, a C language program was developed using the three-phase load flow and simulated annealing tools discussed in Chapters 3 and 4, respectively. This chapter details the implementation of the simulated annealing algorithm; namely, it specifies the four components necessary to apply simulated annealing to a specific optimization problem, in this case the problem of loss reduction in a three-phase power distribution network. These four elements are listed again in Table 5.1.

**Table 5.1** Implementing Simulated Annealing

<table>
<tr><th colspan="3">4 Elements of Implementation</th></tr>
<tr><td>1</td><td>Search Space</td><td rowspan="2">specified as part of the problem formulation</td></tr>
<tr><td>2</td><td>Objective or Cost Function</td></tr>
<tr><td>3</td><td>Perturbation Mechanism</td><td rowspan="2">specified by as part of the solution algorithm</td></tr>
<tr><td>4</td><td>Cooling Schedule</td></tr>
</table>

## 5.1 Search Space

In the network reconfiguration problem the solution space being searched by the simulated annealing algorithm is the space of all possible network configurations as described in Section 2.1 “Search Space” on page 5. The solution is of the form $u = [u_1, u_2, \ldots, u_{n_s}]^T$, where $n_s$ is the number of switches in the system, and $u_i$ gives the open/closed status of switch $i$. The search space, the space of all possible configurations $u$, is $S = \{0, 1\}^{n_s}$.

## 5.2 Objective Function

The objective in this network reconfiguration problem is to minimize the total real power loss in the distribution system in order to reduce the cost of supplying the electrical power demanded by the loads. Real power loss includes the losses due to the resistance of the transmission lines, thermal losses in the transformers, and voltage regulator losses.

These individual losses can be calculated from the results of the power flow analysis. As previously given in Equation (2.3), the total objective function can be expressed as

$$f(x, u) = \left( \sum_{i=1}^{n_l} P_i^{line} \right) + \left( \sum_{j=1}^{n_t} P_j^{transformer} \right) + \left( \sum_{k=1}^{n_r} P_k^{regulator} \right) \tag{5.1}$$

where $P_i^{line}$, $P_j^{transformer}$, and $P_k^{regulator}$ represent the real power lost in line $i$, transformer $j$, and voltage regulator $k$, respectively, and $n_l$, $n_t$, and $n_r$ are the number of lines, transformers, and voltage regulators, respectively, in the system.

The cost of switching from one network configuration to another could easily be included in the cost function as well, as was mentioned in Section 2.3 “Cost Function” on page 6. However, it was not included in this implementation because of a lack of switching cost data.

## 5.3 Perturbation Mechanism

Although simulated annealing is generally thought of as a tool for unconstrained optimization, it can also be used to solve problems with constraints. The constraints can be taken into account in the perturbation mechanism used to generate new states. In this implementation the perturbation mechanism proposes only feasible solutions, ensuring that the solution can be reached via a sequence of feasible configurations. In actuality, certain types of non-feasible solutions may be generated but they are thrown out and not given as proposed solutions to the simulated annealing algorithm. In this way, the constraints are taken into account by the generation probability $G(T)$, mentioned in Section 4.3.2 “Asymptotic Convergence Characteristics” on page 28.

In this implementation, a new feasible point is generated according to the procedure given in Table 5.2.

### 5.3.1 Topological Constraints

The topological constraints are restrictions on the vector $u$ of switch states. Suppose there are a total of $n_s = n_{open} + n_{closed}$ switches in the system, where $n_{open}$ is the number of normally open switches and $n_{closed}$ is the number of normally closed switches.

**Table 5.2** Perturbation Mechanism

| | **Given a feasible system configuration ...** |
|---|---|
| 1 | Randomly choose a new network configuration which is consistent with the given *topological* constraints. |
| 2 | Run a power flow study to find bus voltages which are consistent with the given *electrical* constraints. |
| 3 | Check to see if this new solution satisfies the *load* and *operational* constraints ...<br>• *no* go to step 1<br>• *yes* propose new solution to algorithm |

In order to have a network which is a *spanning tree,* a radial network with each bus connected via some path to the substation bus, there must never be more than $n_{open}$ open switches. Opening another switch would separate the network into two unconnected parts. However, closing any switch in the network would create a loop, destroying the radial structure. Therefore, there can never be less than $n_{open}$ open switches either. In other words the number of open switches must always be exactly $n_{open}$, and the number of closed switches must always be exactly $n_{closed}$. With this restriction the total number of feasible configurations would be

$$\binom{n_s}{n_{open}} = \binom{n_s}{n_{closed}} = \frac{n_s!}{n_{open}!\,(n_s - n_{open})!} \tag{5.2}$$

This number, however, is actually an upper bound since not all of these configurations are spanning trees.

Given one radial configuration, a new radial configuration can be found by using the simple two step procedure implemented in this program.

1. Close exactly one tie switch, selected at random from the set of all tie switches. This creates a loop in the system since it was originally a radial network.

2. Open exactly one sectionalizing switch, selected at random from the set of sectionalizing switches located in the loop created in step 1.

In order to be able to efficiently traverse the network to find potential sectionalizing switches for breaking the loop, it is necessary to store some redundant connectivity information in the network data structures. The connectivity information is stored in two arrays, one for buses and one for lines. Each element in the bus array contains a field specifying the incoming line and a field containing a list of the outgoing lines. In the line array each element contains two fields specifying the buses at the sending and receiving ends of the line.

These data structures make it possible to find the loop created by closing a tie line switch as follows:

1. Starting with the bus at the sending end[1] of the tie line switch traverse the network toward the source by looking alternatively at the “incoming line” field of the current bus and the “sending bus” of the current line. Record all branching buses[2] until the source is reached.

2. Starting with the bus at the receiving end of the tie line switch traverse the network toward the source as described step 1, com-

[1] The sending end is chosen arbitrarily since at this point the “direction” of this line is not yet known. It will depend on which sectionalizing switch is opened.

[2] A branching bus is one with more than one outgoing line.

paring each branching node encountered with those recorded in step 1. Stop when a common node is found.

3. The loop consists of the two branches connecting this common node to either end of the tie line switch.

This very elementary change in configuration results in a new valid configuration which is in some sense “near” the previous configuration and can be thought of as being obtained by a small perturbation to the previous solution. This method of generating a new value for $u$ ensures that only feasible configurations will be proposed. Although other methods could be used to generate new feasible points, this very simple one ensures that any solution generated by the simulated annealing algorithm will be reachable through a sequence of simple switch changes, *all* of which result in configurations which satisfy the topological constraints.

### 5.3.2 Electrical Constraints

Any feasible solution must satisfy Kirchhoff's voltage and current laws while providing the specified amount of power to each load. In other words, a feasible solution must be consistent with Equations (3.9) and (3.10), the power flow equations derived in Section 3.3 “Three-Phase Power Flow Equations” on page 17.

In order to find feasible values for the bus voltage magnitudes and angles in $x$, a power flow analysis is run using the new network topology found by the method described in Section 5.3.1 “Topological Constraints” on page 34. In this implementation the system modeling and power flow solution are handled by a three-phase distribution load flow program from Paralogix

Corporation in Syracuse, NY. This program currently uses the Gauss method to solve the power flow equations.

In the cases where the power flow solution algorithm does not converge within 50 iterations of the Gauss method, that particular network configuration is considered to be *infeasible* and a new configuration is generated. In other words, when step 2 in Table 5.2 cannot yield a solution which satisfies the given electrical constraints, then return to step 1.

### 5.3.3 Load and Operational Constraints

Given the results of a convergent load flow solution for a feasible system configuration, it is very easy to check the load and operational constraints. The load flow solution yields information about bus voltage magnitudes, line flows, and line currents. Ensuring that each of these quantities lies within the specified bounds requires only a simple comparison.

If all of the bus voltage magnitudes, line flows, and line currents lie in the appropriate ranges, the configuration is taken to be *feasible* and is proposed as a new solution to the simulated annealing algorithm. If any of these constraints are violated the configuration is considered to be *infeasible* and a new configuration is generated starting at step 1 in Table 5.2.

## 5.4 Cooling Schedule

There are many variations in the simulated annealing algorithm when it comes to cooling schedules. Determining an effective cooling schedule is often a matter of trial and error. In this implementation a relatively simple cooling schedule was chosen and a few adjustments were made to the parameters based on the results of several trial simulations.

### 5.4.1 Initial Temperature

The starting value for the temperature parameter is found by running several iterations of the Metropolis algorithm[3] at a temperature of 1. If the *acceptance ratio*[4] for the first 10 iterations is less than 90%, the temperature is doubled and the algorithm is run for 10 more iterations. This procedure is repeated until the acceptance ratio becomes greater than or equal to 90%. At that point the value of the temperature is taken as the initial temperature for the simulated annealing algorithm.

This method ensures that the initial temperature is high enough that the majority of moves are accepted at the start of the annealing process.

### 5.4.2 Temperature Update

The temperature parameter is updated according to a very simple, commonly used rule. After the number of iterations determined by the procedure detailed in Section 5.4.3 “Markov Chain Length”, the temperature parameter is multiplied by a constant $\alpha$, where $0 < \alpha < 1$. In this implementation $\alpha = 0.85$ was used.

---

[3] The *Metropolis algorithm* is described in Section 4.2 “Analogy to Physical Annealing” on page 25.

[4] The *acceptance ratio* is the number of accepted state transitions divided by the number of proposed transitions.

### 5.4.3 Markov Chain Length

The goal of determining the optimal length of Markov chains is to allow the algorithm to reach equilibrium at a given temperature. After reaching equilibrium, continuing to generate new states at that temperature is not necessary.

In this implementation the Markov chain is terminated, in other words, the temperature is updated, after 10 moves have been accepted or 50 moves have been proposed, whichever comes first.

### 5.4.4 Termination Criterion

The algorithm should be stopped once the solution has been found and the temperature is low enough that the state is essentially “frozen”. If the last state in the Markov chain has remained unchanged for 4 iterations, the state is assumed to be “frozen” and the search is terminated.

## 5.5 Comments on Implementation

The calculation of the objective function and the evaluation of constraints both require the results of a power flow study. Because the power flow study itself requires an iterative procedure to solve a set of non-linear algebraic equations, the majority of computation time in this implementation goes into running power flow studies. Therefore the efficiency of the algorithm used to solve the power flow problem has a tremendous impact on the efficiency and speed of the loss reduction optimization problem.

# Chapter 6

# Simulation Results

This chapter is a discussion of the results obtained by applying the program described in Chapter 5 to an example test system. The implementation is exactly as described in Chapter 5 except that in this simulation the load and operational constraint checking was disabled since the corresponding data was not available. The program was written in C language and run on a Sun SPARCstation 2.

## 6.1 Example Test System

The system used to test the solution algorithm is the 147 bus three-phase distribution network shown in Figure 6.1. There are a total of 12 switches in the network, three tie switches and nine sectionalizing switches. According to the upper bound given by Equation (5.2) there are less than 220 feasible radial network configurations. A complete enumeration of these configurations shows that the actual number is less than half of this upper bound. There are exactly 104 configurations which satisfy the topological constraints described in Section 2.4.1 “Topological Constraints” on page 8.

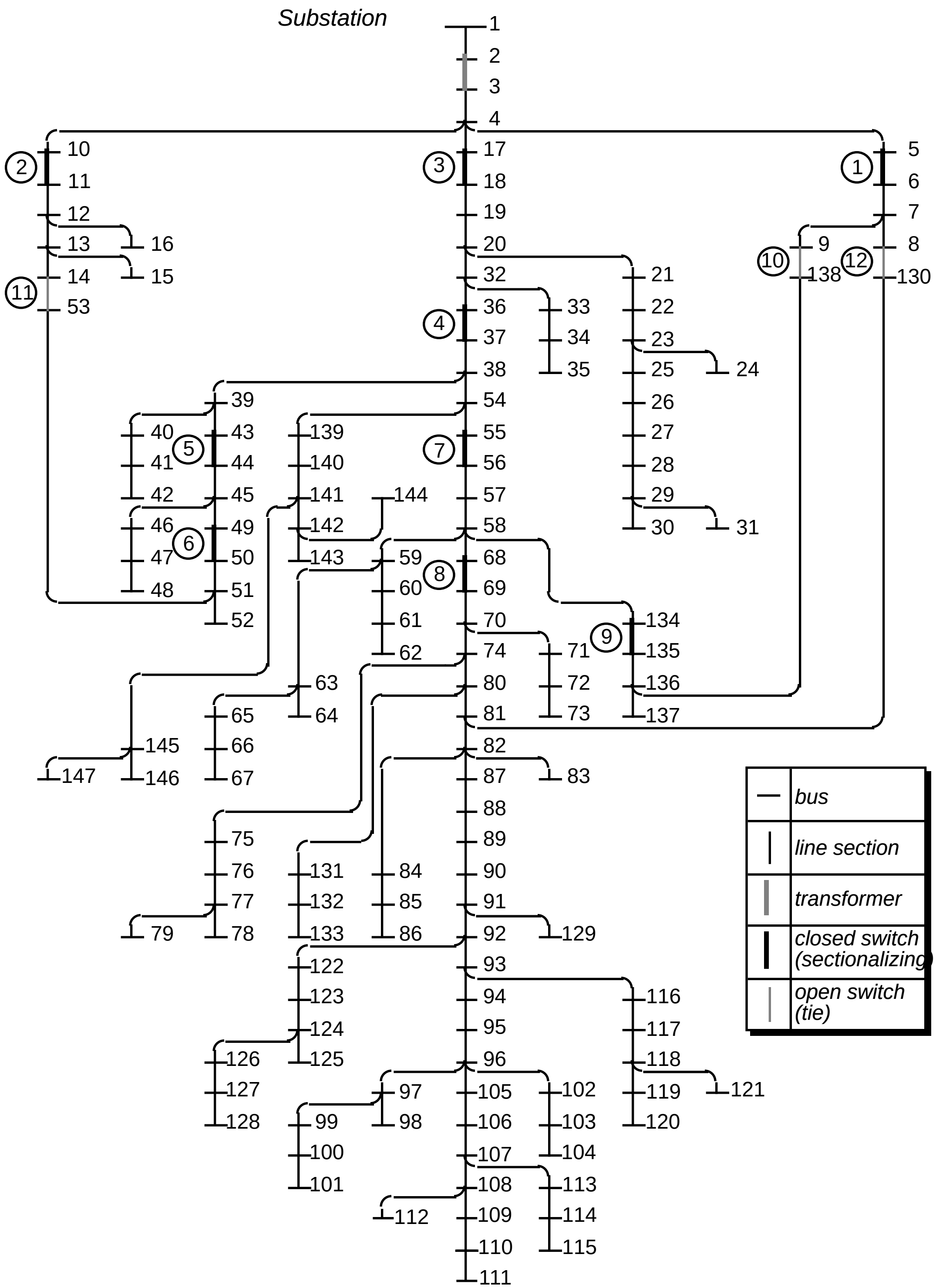


**Figure 6.1** Example Test System: Initial Configuration

This system contains 80 three-phase buses, 66 single-phase buses, and one two-phase bus, for a total of 147 buses. The total real power demanded by the load is 1.662 p.u. The nominal voltage at bus 1, the slack bus, is 1.025 p.u.

## 6.2 Initial Configuration

In the initial configuration, the one shown in Figure 6.1, switches 10, 11, and 12 are open, and switches 1 through 9 are closed. The results of the power flow study show that the total real power supplied to the network is 1.6865 p.u. Given a total real power demand of 1.6620 p.u., this yields a total real power loss in the system of 0.0245 p.u., or 1.455% of the power supplied.

## 6.3 Final Configuration

In the final, optimal configuration found in the simulation, switches 6, 9, and 12 are open, and the rest are closed. This configuration is shown in Figure 6.2. The results of the power flow study show that the total real power supplied to the network is 1.6812 p.u. Given a total real power demand of 1.6620 p.u., this yields a total real power loss in the system of 0.0192 p.u., or 1.143% of the power supplied.

The real power loss in this optimal network is 21.6% less than that of the initial system. This reduction in losses makes it possible to decrease the power supplied by the substation by 0.314%. A summary of these results is presented below in Table 6.1.

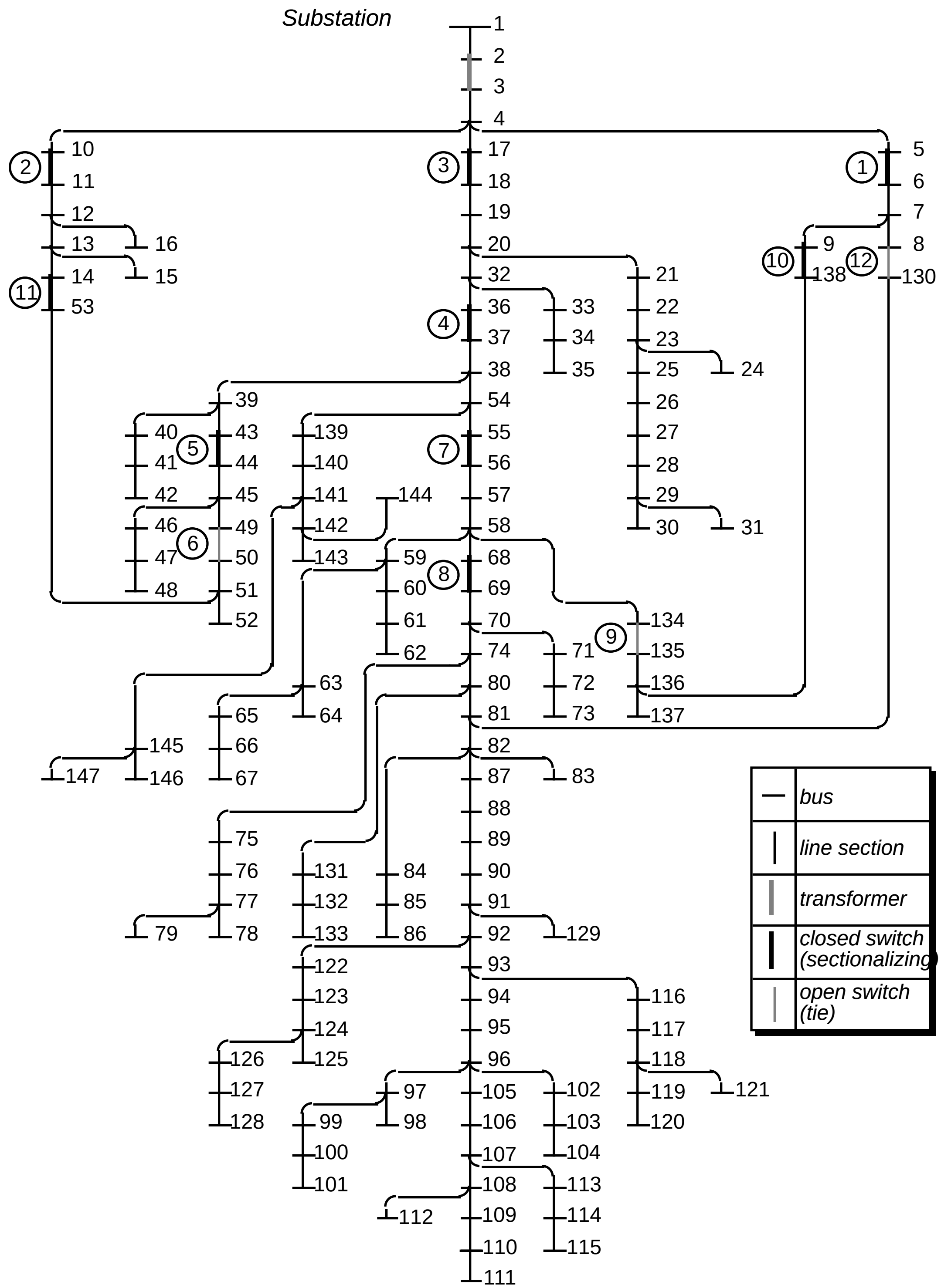


**Figure 6.2** Example Test System: Optimal Configuration

**Table 6.1** Summary of Results

| Total Real Power | Initial | Final | Change | % Change |
|---|---|---|---|---|
| input | 1.6865 p.u. | 1.6812 p.u. | -0.0053 p.u. | -0.314% |
| delivered | 1.6620 p.u. | 1.6620 p.u. | – | – |
| loss | 0.0245 p.u. | 0.0192 p.u. | -0.0053 p.u. | -21.6% |
| % loss | 1.455% | 1.143% | -0.313% | -21.5% |

Not only is the total real power loss for the system reduced, the system's overall voltage profile is improved as well. In Figure 6.3 the voltage magnitude[1] is plotted versus the bus number for the initial system configuration and for the optimal system configuration. The bus voltage is increased at all but a few buses and the minimum voltage is increased from 0.946 p.u. to 0.960 p.u. This voltage occurs at bus 101, a single-phase bus on phase b, and can be noted in Figure 6.3. Decreasing system losses often improves the quality of service to the customer, in addition to decreasing the cost of the service.

## 6.4 Discussion of Convergence Behavior

The simulated annealing program was run on a Sun SPARCstation 2 to find the optimal configuration of the example system shown in Figure 6.1. The search terminated after generating 48 Markov chains, making 970 proposed moves, and running 1000 power flow calculations.

The load flow program was not able to converge in less than 50 iterations for nine of the configurations. These configurations were attempted 30

[1] For two and three-phase buses this is the average magnitude of the phases.

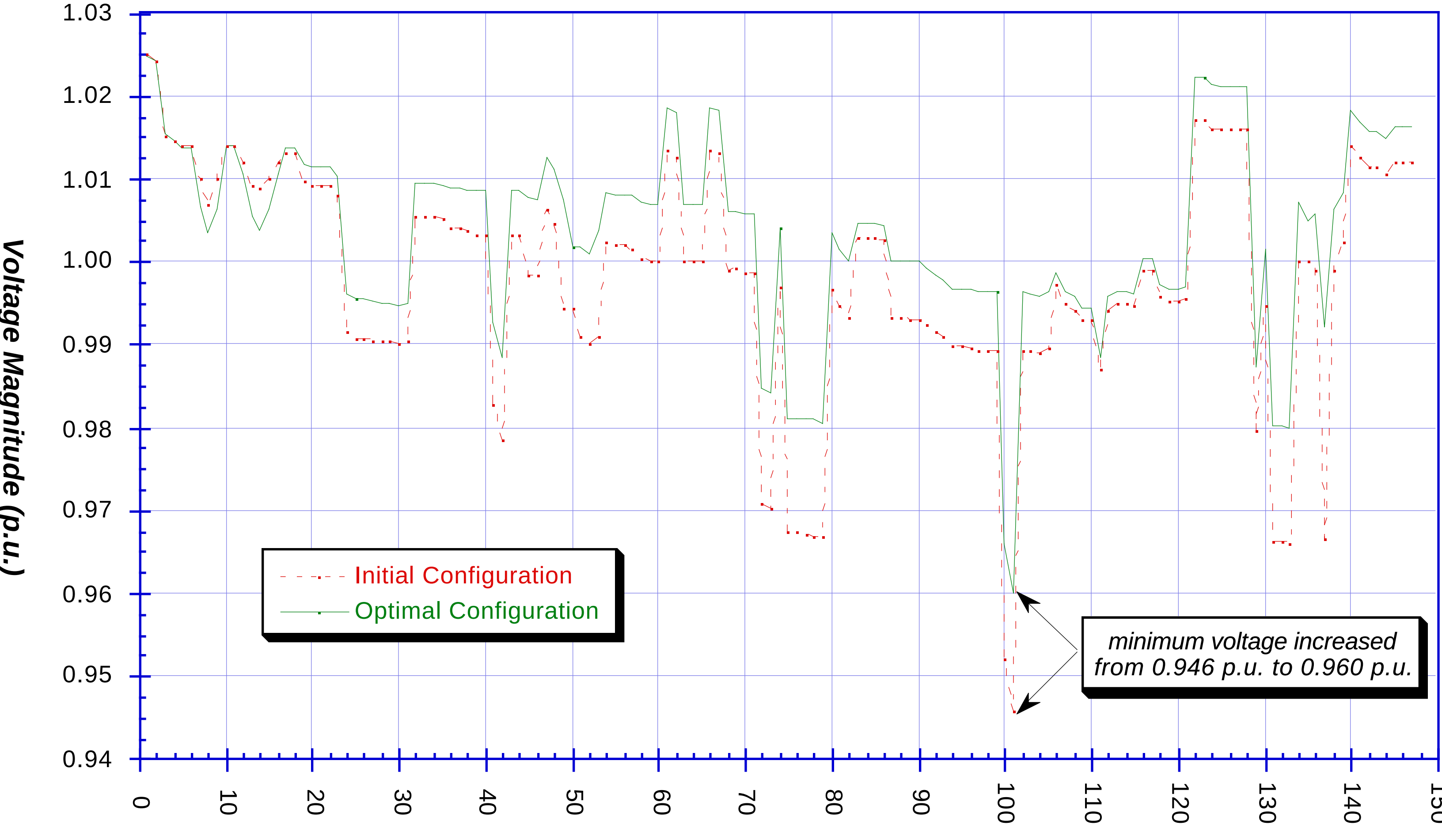


**Figure 6.3** System Voltage Profiles

times during the entire search. In other words, 0.3% of the configurations analyzed resulted in infeasibility. In the process of the search 98 of the 104 total configurations were generated, some of them many times. Later analysis showed that each of the six configurations not generated was either infeasible, yielding a non-convergent power flow, or had a high objective value relative to other solutions. In any case, it was verified that the solution found by the simulated annealing algorithm was indeed a *unique global optimum* for this problem.

The temperature parameter was started at 1 and updated by multiplying by a factor of 0.85 at the end of each Markov chain, yielding the exponential decrease shown in Figure 6.4. Since 48 Markov chains were generated,

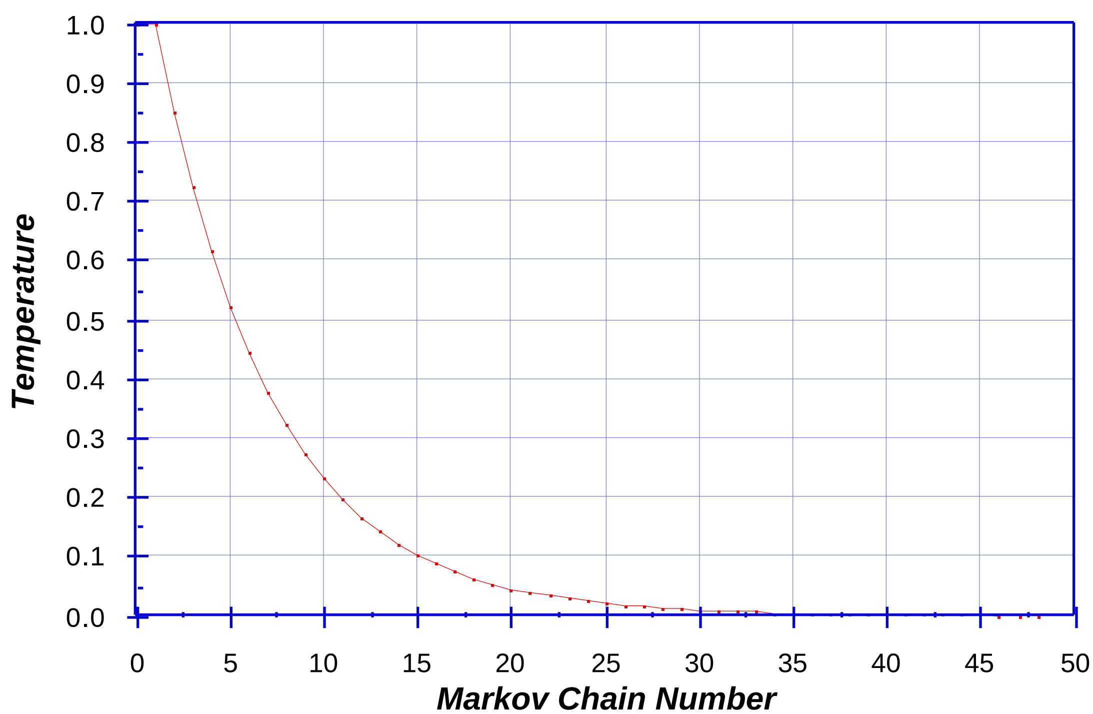


**Figure 6.4** Temperature vs. Markov Chain Number

the temperature was updated 47 times yielding a final temperature of $(0.85)^{47} = 4.82 \times 10^{-4}$.

The starting temperature of 1, found by the procedure described in Section 5.4.1 “Initial Temperature” on page 39, was high enough to ensure an initial acceptance ratio of 1. As the temperature decreases the accep-

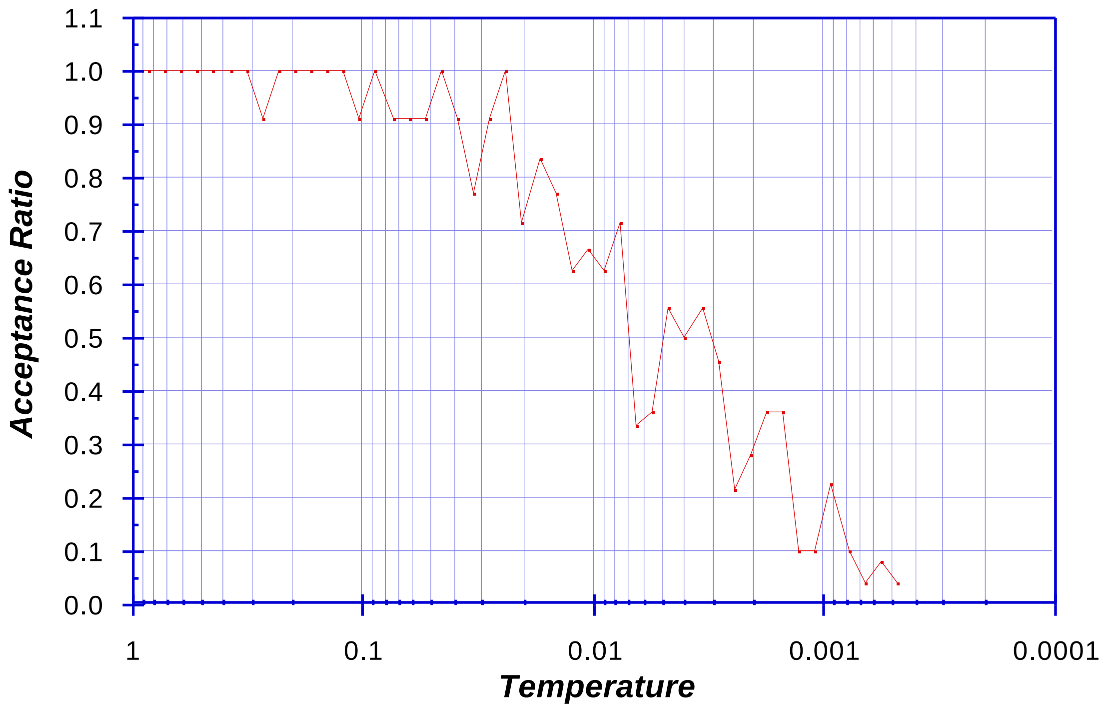


**Figure 6.5** Acceptance Ratio vs. Temperature

tance ratio also decreases as shown in Figure 6.5. By the time the algorithm terminates the acceptance ratio is down to 0.04, indicating 2 accepted transitions in a Markov chain of length 50. This can be considered a nearly “frozen” state.

Figure 6.6 shows the evolution of the objective function value at the end of each Markov chain as the temperature parameter is decreased. From the

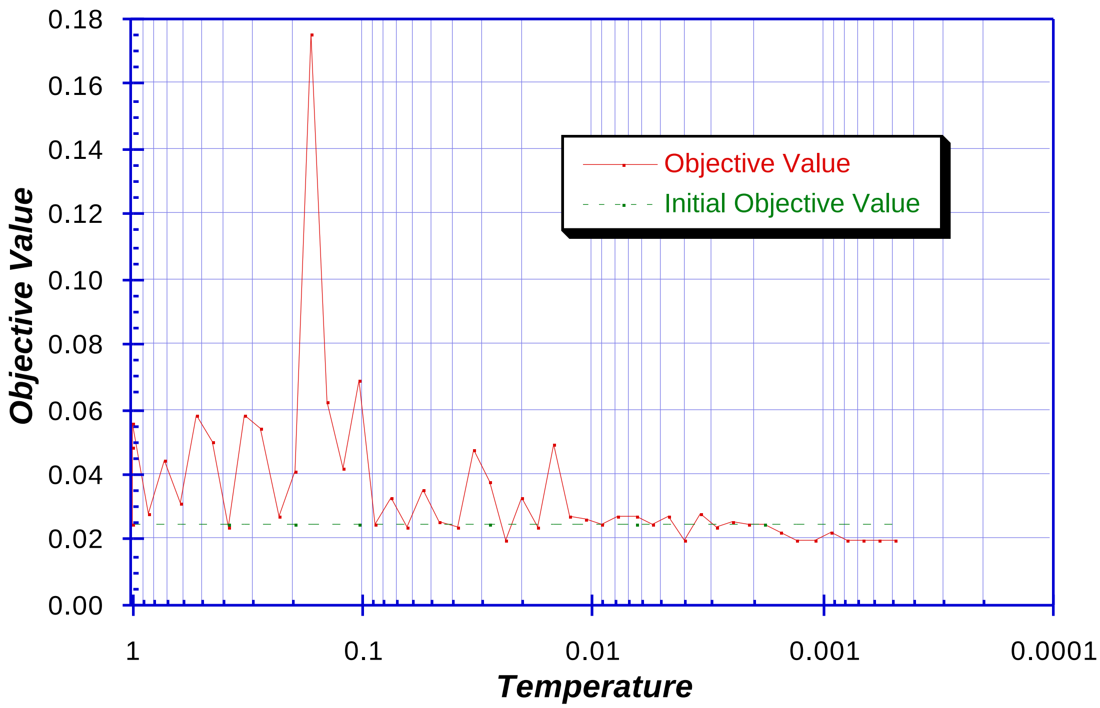


**Figure 6.6** Objective Function Value vs. Temperature

analysis of the solutions generated during the search it was found that 13 of the 104 configurations had real power losses less than the 0.0245 p.u. of the initial configuration. In fact 23.5% of the proposed moves were to configurations which improved on the initial system. In Figure 6.7 and Figure 6.8 the objective value is plotted for each accepted move, 33.9% of which are better than the initial configuration. Although the optimal solution was generated very near the beginning of the algorithm, at the 21$^{st}$ iteration, and again a total of 24 times, there is no way of knowing that it is the optimal solution without allowing the algorithm to terminate. Even then, because it is a finite time approximation to the simulated annealing algorithm, there is no guarantee that it is optimal. In this case, because of the small size of the feasible region of the solution space to this problem, it

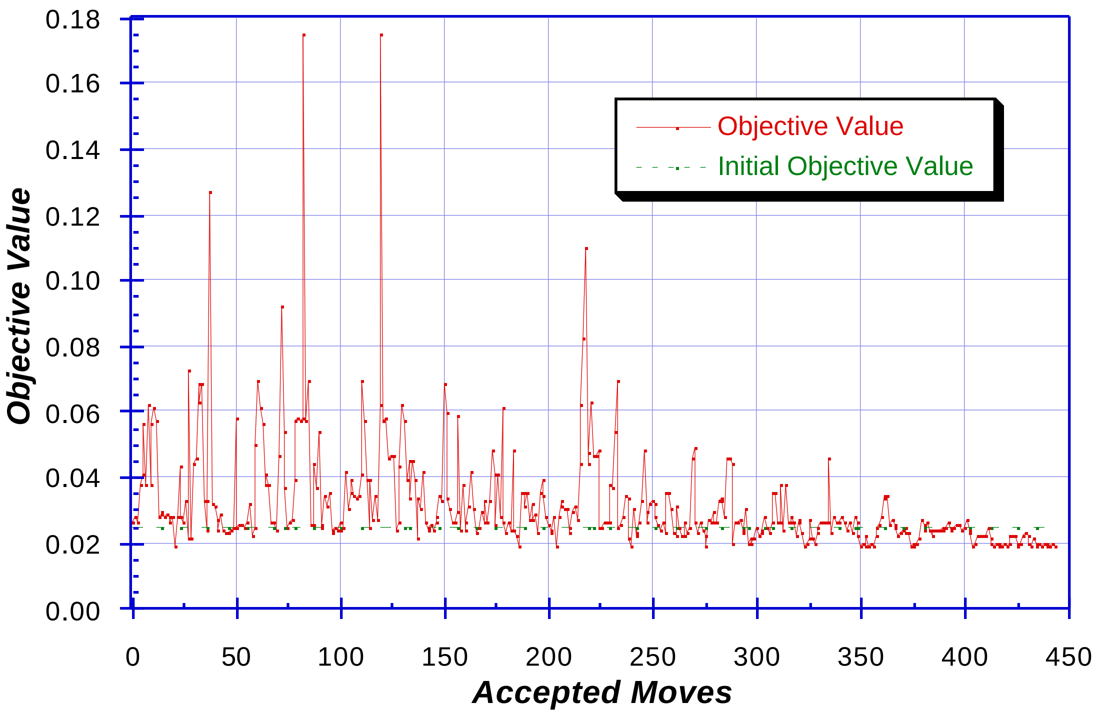


**Figure 6.7** Objective Function Value vs. Accepted Moves

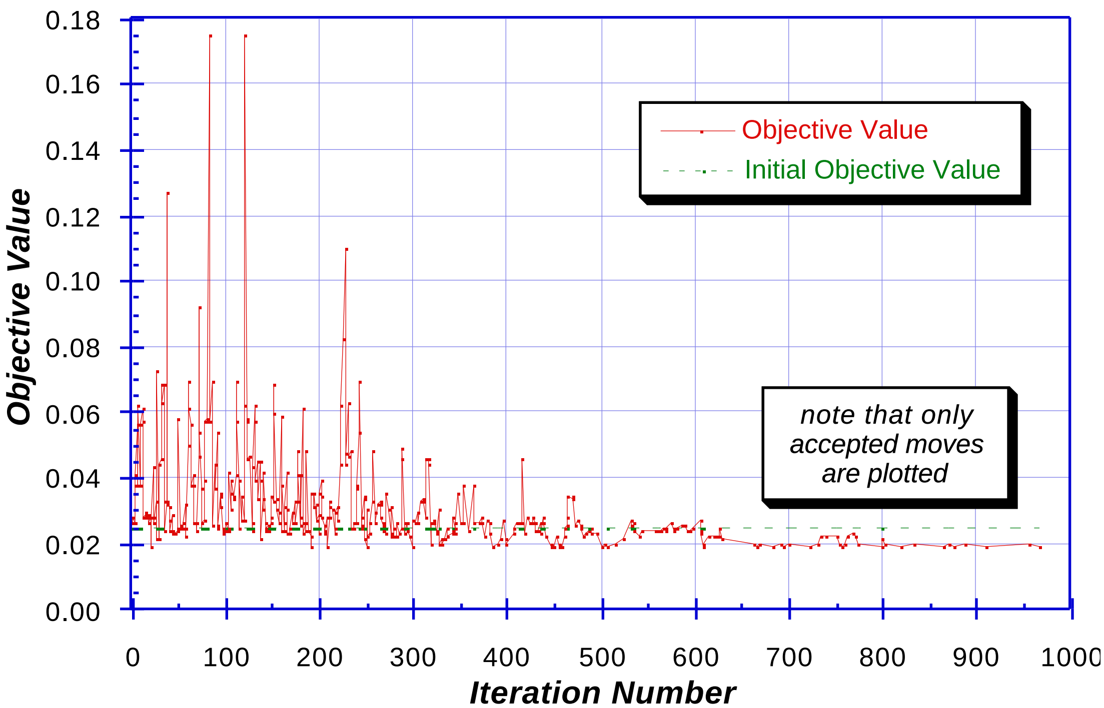


**Figure 6.8** Objective Function Value vs. Iteration Number

was possible to verify via an exhaustive search that the optimal solution was indeed found by the finite time algorithm.

# Chapter 7

# Conclusions

The objective of this thesis is to show that the simulated annealing algorithm can be used successfully to find the configuration of a *three-phase* power distribution network which minimizes the overall real power losses of the system. This work is to provide a basis for the development of algorithms and computer programs which could be included in the software used by power utility companies to make effective decisions regarding the configuration of their distribution networks.

## 7.1 What Was Accomplished

In this thesis a summary of the basic theory of three-phase distribution power flow was presented to provide the background for the understanding of one of the primary elements in the solution of the network reconfiguration problem. A general description of the simulated annealing algorithm was also given to provide a foundation for the implementation used in the simulation. A C language program was written, based on the theory presented here, in order to test the algorithm on an example system.

Simulated annealing has been used before to perform network reconfiguration for loss reduction [11; 12], but in this case only for the single-phase or three-phase *balanced* case. Because of the growing complexity of distribution systems it is important to be able to perform this type of analysis and optimization on larger, more complex, *unbalanced* three-phase systems, which was the focus of this study.

As discussed in Chapter 6, this program *was* able to find the optimal configuration of the 147 bus, 12 switch example system, demonstrating the feasibility of such an approach for the solution of this problem. However, refinements to the algorithm, especially the cooling schedule and perturbation mechanism, may be necessary in order to achieve satisfactory performance on large, real-world systems consisting of thousands of buses.

## 7.2 What Remains to be Done

The program which was written to do this simulation was a simple, limited simulation tool which would not be adequate for everyday use in a real-world distribution system. In addition to the need for more extensive testing, there is also much that could be done to improve and extend the capabilities of the current implementation.

### 7.2.1 Refining the Current Implementation

Fortunately, there are several obvious improvements that could be made to the program as it stands. Much of the code could be cleaned up to make it execute more efficiently. For example, in the current implementation the system admittance matrix, $Y_{bus}$, is reconstructed from scratch each time the configuration is changed, that is, each time a power flow calculation is needed. Since the change in the network topology is limited to changing two switches, most of the matrix will remain the same, implying that the matrix could be updated simply by taking into account the changes. Since $Y_{bus}$ is stored in a sparse data structure this could require some reordering. However, the savings gained from not having to completely rebuild the matrix should outweigh the cost of reordering the matrix, especially for large systems.

The current implementation also requires the solution of the load flow equations each time a new configuration is generated throughout the search algorithm. Even if the configuration had been previously generated and the real power losses had already been calculated, a complete power flow solution is done to determine these losses for the calculation of the objective function. Since simulated annealing may generate a given solution many times during the course of the search, much time could be saved by storing the solutions with their objective values as they are generated. In a relatively small system, the amount of storage necessary for this would probably worth the increase in speed. For an extremely large system where the probability of generating solutions many times is much smaller, the extra memory required to implement this may outweigh the advantages of the increased speed of the algorithm.

A third area which needs to be refined is the cooling schedule. In this simulation only one very simple cooling schedule was tried and the results given in Chapter 6 correspond to preliminary results achieved by the initial parameter values presented in Chapter 5. The parameters in this cooling schedule could be tuned in order to achieve maximum efficiency of the algorithm. Other more complex cooling schedules could also prove to be more effective.

### 7.2.2 Extending the Current Implementation

Besides the minor refinements to the current implementation discussed above, there are also ways in which it could be extended to include other relevant tools in the solution of the reconfiguration problem.

As discussed in Section 5.5 “Comments on Implementation” on page 40, the characteristics of the algorithm used to solve the power flow problem have a tremendous impact on the efficiency of the network reconfiguration algorithm. In this implementation the Gauss method was used to solve the full system of power flow equations as derived in Section 3.3 “Three-Phase Power Flow Equations” on page 17. It may prove beneficial to investigate other load flow solution algorithms. Specifically, the formulation of the distribution load flow problem presented in [9] could be extended to the three-phase case. This formulation, using a reduced set of equations, along with the fast decoupled solution methods could significantly reduce computation time. In fact, at the beginning of the simulated annealing algorithm, at a high temperature, it is not essential that the objective values be so precise, implying that even a rough approximation to the load flow solution may suffice during the first part of the search.

Other perturbation mechanisms for simulated annealing could also be investigated. The current perturbation mechanism involves changing only two switches at a time. This constrains the moves to be very local in nature and could cause the algorithm to converge to a sub-optimal solution if the global minimum is far from the starting point. This problem was not encountered in the test system, probably due to the small size of the system. Another perturbation mechanism which allows more global changes was presented by Chiang and Jean-Jumeau in [11; 12] and could be included in this three-phase implementation as well.

Much of the search time, at least in the example system used in this simulation, was spent searching in the neighborhood of the optimal solution. Certain heuristic methods have been developed for single-phase or balanced systems [13] which yield relatively quick convergence but without any guarantee that the solution is a *global* minimum. These methods use rules based on local information and might be able to effectively solve a problem which was started by a more global algorithm like simulated annealing. It seems that a hybrid approach, combining the speed of a heuristic method with the global search of simulated annealing, could possibly lead to fast, global search algorithm.

The formulation of the network reconfiguration problem could also be extended to include other constraints. In a practical setting it may be advantageous to be able to constrain the maximum number of allowable switch changes. Load balancing could also be included as a constraint in the loss reduction problem.

In the current formulation, as presented in Chapter 2 and Chapter 3, loads are considered to be constant PQ devices. Although this may be an accurate model for finding the instantaneous power loss, the actual loads in a distribution system vary throughout a given day. Therefore finding the configuration which minimizes the real power loss in the system at a given instant may not actually yield the configuration which would be most efficient over a longer period of time.

To take this into account, a time-varying load model could be used. In this type of formulation, a typical day is divided into a number of time windows within which the loads are considered to be constant PQ. A load flow solution is calculated for each window, giving an estimate of the real power losses at any instant in that time window. The real power loss for a given window multiplied by the length of the window yields the total energy loss in the system during that period of time. The objective function then is taken as the total daily energy loss in the system found by summing the energy loss values for each of the time windows. The problem could even be extended, as in [5], to solve for optimal switching patterns to accommodate seasonal and daily variations in the load profile.

Even reducing the total energy loss of the system may not result in a configuration which is optimal in some broader sense. It may be necessary to consider other objective functions such as load balancing. A multi-objective formulation such as the one applied to a single-phase system in [11; 12], could easily be extended to the three-phase unbalanced case.

Lastly, one of the most important things remaining to be done involves the power utility companies themselves. Before distribution automation tools, such as a network reconfiguration program, can be used to their potential many of the existing power distribution networks must be upgraded to provide the necessary data acquisition and control devices. This may involve the installation of new equipment such as remote sensors and controllers, and data transmission equipment. Few existing systems are currently equipped to take full advantage of these new tools.

# Bibliography


[1] E. H. L. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons, 1989.

[2] J. Arrillaga, C. P. Arnold, and B. J. Harker, *Computer Modelling of Electrical Power Systems*, John Wiley & Sons, 1983.

[3] M. E. Baran and F. F. Wu, "Network Reconfiguration in Distribution Systems for Loss Reduction and Load Balancing", *IEEE Transaction on Power Delivery*, Vol. 4, No. 2, April 1989, pp. 1401-1407.

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

[5] R. P. Broadwater, A. H. Khan, H. E. Shaalan, and R. E. Lee, "Time Varying Load Analysis To Reduce Distribution Losses Through Reconfiguration", *presented at IEEE/PES 1992 Winter Meeting*, New York, NY, January 1992, 92 WM 269-1.

[6] V. Černy, "Thermodynamic Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm", *Journal of Optimization Theory and Applications*, Vol. 45, 1985, pp. 41-51.

[7] T. H. Chen, M. S. Chen, K. J. Hwang, P. Kotas, and E. Chebli, "Distribution System Power Flow Analysis - A Rigid Approach", *IEEE Transactions on Power Delivery*, Vol. 6, No. 3, July 1991, pp. 1146-1152.

[8] T. H. Chen, "Generalized Distribution Analysis System", Ph.D. Dissertation, The University of Texas at Arlington, May 1990.

[9] H. D. Chiang, "A Decoupled Load Flow Method for Distribution Power Networks: Algorithms, Analysis and Convergence Study", *Electrical Power & Energy Systems*, Vol. 13, No. 3, June 1991, pp. 130-138.

[10] H. D. Chiang and M. E. Baran, "On the Existence and Uniqueness of Load Flow Solution for Radial Distribution Power Networks", *IEEE Transactions on Circuits and Systems*, Vol. 37, No. 3, March 1990, pp. 410-416.

[11] H. D. Chiang and R. M. Jean-Jumeau, “Optimal Network Reconfigurations in Distribution Systems: Part 1: A New Formulation and A Solution Methodology”, *IEEE Transactions on Power Delivery,* Vol. 5, No. 4, 1990.

[12] H. D. Chiang and R. M. Jean-Jumeau, “Optimal Network Reconfigurations in Distribution Systems: Part 2: A Solution Algorithm and Numerical Results”, *IEEE Transactions on Power Delivery,* Vol. 5, No. 3, 1990.

[13] S. Civanlar, J. J. Grainger, H. Yin, and S. S. H. Lee, “Distribution Feeder Reconfiguration for Loss Reduction”, *IEEE Transactions on Power Delivery,* Vol. 3, No. 3, July 1988, pp. 1217-1223.

[14] T. J. Kendrew, J. A. Marks, “Automated Distribution Comes of Age”, *IEEE Computer Applications in Power,* Jan. 1989, pp. 7-10.

[15] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, “Optimization by Simulated Annealing”, *IBM Research Report RC 9355*, 1982.

[16] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, “Optimization by Simulated Annealing”, *Science*, Vol. 220, 1983, pp. 671-680.

[17] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Reidel, Dordrecht, 1987.

[18] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of State Calculations by Fast Computing Machines”, *Journal of Chemical Physics*, Vol. 21, 1953, pp. 1087-1092.

[19] K. Nara, A. Shiose, M. Kitagawa, and T. Ishihara, “Implementation of Genetic Algorithm for Distribution Systems Loss Minimum Reconfiguration”, *presented at IEEE/PES 1991 Summer Meeting,* San Diego, CA, July 1991, 91 SM 467-1.

[20] T. Taylor and D. Lubkeman, “Implementation of Heuristic Search Strategies for Distribution Feeder Reconfiguration”, *IEEE Transactions on Power Delivery,* Vol. 5, No. 1, January 1990, pp. 239-246.