Smart Distribution Power Systems Reconfiguration using a Novel Multi-agent Approach

by

Michael Mansour

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2013

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The few past years have witnessed a huge leap in the field of the smart grid communication networks in which many theories are being developed, and many applications are being evolved to accommodate the implementation of the smart grid concepts. Distribution power systems are considered to be one of the first leading fields having the strong desire of applying the smart grid concepts; resulting in the emersion of the smart distribution power systems, which are the future visualization of the distribution systems having both the ability of smart acting, and the capabilities of automation, self-healing, and decentralized control. For the sake of the real implementation of the smart distribution power systems, the main functions performed by the traditional systems have to be performed by the new smart systems as well, taking into account the new features and properties of those smart systems. One of those main functions is the ability of power networks optimal reconfiguration to minimize the system's power loss while preserving the system radial topology.

The proposed reconfiguration methodology targets the utilization of a hybrid genetic algorithm with two fuzzy controllers that could converge to the global optimal network configuration with the fastest convergence rate consuming the least computational time. The first fuzzy controller is designed to reject any infeasible system configurations that might show up in the population of the genetic algorithm and violate the system radial topology, while the second fuzzy controller is designed to adapt the mutation rate of the genetic algorithm. Consequently, a novel multi-agent system is proposed and designed to perform the reconfiguration application in smart distribution power systems employing the concepts of distributed processing and decentralized control demanded by those systems. A multi-agent system employs a group of intelligent agents that have the capabilities of autonomy, reactivity, pro-activity, and sociality. Those agents cooperate with each other in order to perform a certain function through their powerful abilities to communicate, socialize, and make a common decision in a decentralized fashion based on the information retrieved from the surrounding environment and compiles with their ultimate objective.

Acknowledgements

“For I am the LORD, your God, who takes hold of your right hand and says to you, Do not fear, I will help you” (Isaiah 41:13)

Thank you my LORD for taking care of me, protecting me, and putting me on the right track

I would like to express my deep appreciation to my supervisors Prof. Magdy Salama and Dr. Tarek El-Fouly for their great efforts and contributions towards the work done in this thesis, and their guidance, leadership, and creation of the appropriate climate for research.

Thanks are not enough to express my gratitude towards everyone who has helped, guided, and supported me during my pursuit of distinction:

> Dr. Hany Milio and Dr. Gehan Ghally; Amgad and Amira Waniss; Alfons and Ester Shehata; Mervat Beshay; Amir Shehata and Fiby Atta; George and Mariam Shehata; John and Sally Saad; Albert and Marian Wassef; Michael Gad; Michael Ibrahim; Salam Gabran; Michael Ayoub and Sally Daif; Mina Farid; Marco Amir; Mikhail Shenouda; George Soliman; Monica El Gamal; Mina AbdelMalek; Mina AbdelMaseh; George and Christen Shaker; Ayad and Evon Barsoum; George Morkos; Olivia Mikhael; Hany Lewis; Engy Michel.

Finally, I would like to express my endless love and thankfulness to all my family and friends in my home country, Egypt, for their great love, extreme care, permanent presence, and true sharing for every moment in my journey:

> Father, mother, and sister; Grandfather and Grandmother; Father Kirollos Naiem; Sylvia Remon; Nader Fawzy; Martina Ibrahim; Marian Yehia; Joseph Moftah; Nermeen Fathy; Marian Moheb; Peter Ehab; Christen Motie; Rania Noshy; Sherin Nabil; Marian Reda; Fady Boushra; Caroline Gamal; Marina Joseph; Mirna Magdy; Mina Michel; Maria Iskander; Gamal Gaber; Peter Bahgat.

Dedication

To my father, mother, and sister

To my Grandfather and Grandmother

To the Memory of my brother, Mina Fawzy

To the Soul of my advisor, Wael Nabil

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

The smart distribution power systems are the future trend of employing the concepts and features of the smart grid in the traditional distribution power systems such that they could gain the ability of acting smartly. Automation, self-healing, decentralized control, and the injection of the distributed generation and renewable energy sources are the main features required to be achieved in the future smart distribution power systems. These smart systems should employ all the functions employed by the traditional ones with the same, or even better, efficiency and reliability. A key factor of the employment of these functions in the smart distribution power systems is that they should be employed taking into account the new system features mentioned above.

1.1 Research Motivations

One of the main challenges facing the implementation of the smart grid in the distribution power systems is the need to build a system having the powerful capabilities of the distributed processing and decentralized control. This system is expected to perform all the functions and applications required to be performed in the distribution power systems in an online mode in which the decision is made on the spot and in a decentralized fashion. In this thesis, the reconfiguration application has been selected to be performed in the smart distribution power systems via a novel reconfiguration methodology which is deigned to be carried on by the proposed novel multi-agent system designed specially to have the powerful capabilities of the distributed processing, decentralized control, and on the spot decision making.

1.2 Thesis Highlights

As discussed in the research motivations section, one of the main challenges facing the implementation of the smart grid is the design and implementation of a system handling the communications layer functionalities through which several nodes can communicate with each other, share different types of information, and make a decision on their own without the need for a central node. In other words, decentralized control is required instead of the centralized control. One of the suggested approaches for implementing this communication layer is the utilization of the multi-agents systems. A multi-agent system is a system employing a group of intelligent agents, each agent has the ability to share information with the surrounding agents; and make a decision on their own depending on the information gathered from the surrounding environment. This multi-agent system should be able to perform any power systems application through the employment of the concepts of the distributed processing and decentralized control.

In this thesis, the reconfiguration problem in distribution power systems has been selected to be solved by the designed multi-agent system to test the powerfulness of this system and its ability to perform any power application in the smart distribution power systems with better efficiency than that of the traditional systems. For this purpose, the genetic algorithm and the fuzzy logic are used in a hybrid algorithm designed to solve the reconfiguration problem. The proposed reconfiguration methodology has to be first tested in the traditional distribution power systems to make sure of its high efficiency and fast convergence rate; which consequently allows its application in the smart distribution power systems. The genetic algorithm is used to find the optimal network configuration to minimize the power loss in the power system subject to system constraints such as the voltage limits, the current limits, and the radial topology preservation. Three different versions of the genetic algorithm are tested with three different types of mutation functions; the Gaussian mutation function, the uniform mutation function, and the adaptive mutation function. The fuzzy logic is used to build two different fuzzy controllers. The first one targets the rejection of any network topology that violates the radial configuration of the system, while the second one targets the adaptive mutation function control and the proper choice of the mutation rate of the genetic algorithm.

Three different scenarios have been performed in this thesis on the 16-node test distribution power system to evaluate the capabilities of the proposed reconfiguration methodology. First, only the genetic algorithm is applied to solve the reconfiguration problem in the three different cases of the utilized mutation functions. Second, the first fuzzy controller is applied on these three different cases of the mutation functions employed to select the mutation rate in the genetic algorithm. Finally, the proposed methodology with the two fuzzy controllers along with the genetic algorithm is applied to solve the reconfiguration problem in the case of the adaptive mutation function controlled by the second fuzzy controller. In order to prove the powerfulness and effectiveness of the proposed methodology and its suitability to be applied to any distribution power system, it has been applied to three different test distribution power systems; the 33-node test system, the 69-node test system, and the IEEE 123-node test system. This proposed hybrid algorithm is believed to achieve a better performance, a higher efficiency, and a faster convergence rate which encourages its utilization to solve the reconfiguration problem in the smart distribution power systems.

In order to achieve that purpose, a novel multi-agent system has been designed in which several intelligent agents are employed to share the information gathered from the surrounding environment, do a part of the job, and finally make their own decision in a decentralized fashion based on the retrieved information and the calculations performed by each one of them. For the sake of implementing that multi-agent system, the distributed processing and object-oriented programming paradigm has been mapped to the multi-agent paradigm in such a way that simplifies the system design and at the same time enables the systems to have all the functionalities of the intelligent agents. Eight different scenarios have been performed to test the performance of the designed multi-agent system after being built and applied to the IEEE 123-node test system. These eight scenarios involve different loading and switching conditions in order to calculate the total savings achieved by each one of them due to saving in the total power loss. The effect of integrating distributed generations on the total power loss and the total savings achieved has been studied as well.

1.3 Thesis Outline

The first chapter highlights the problem addressed in the thesis and introduces a brief idea about the smart grid, the multi-agent systems, and the reconfiguration methodology proposed in the thesis. The smart grid; its definitions, characteristics, and the challenges facing it are first discussed in Chapter Two, followed by a discussion of the reconfiguration problem of the distribution power systems along with a literature review of the problem and the methodologies utilized for solving it. The tools used as the building blocks in the proposed reconfiguration methodology are then presented. The genetic algorithm is discussed along with the details of the different mutation functions used; the Gaussian mutation function, the uniform mutation function, and the adaptive mutation function, and then the fuzzy logic concepts and the fuzzy inference systems are discussed as well. A discussion of the multi-agent systems; their characteristics, architecture, and applications in power systems is then provided. Finally, a brief idea about the distributed generations, the motivations of their integration into the smart distribution power systems, and the impact of their installation on the performance of these systems is presented.

The proposed methodology suggested for tackling the reconfiguration problem is discussed in Chapter Three along with the design of the two fuzzy controllers used for rejecting the infeasible system configurations and controlling the adaptive mutation rate of the genetic algorithm. The implementation of the three different experiments performed on the 16-node distribution power system along with the simulation results obtained are provided in the Chapter Four, followed by the simulation results for the other three test systems; the 33-node test system, the 69-node test system, and the IEEE 123-node test system. These results are discussed to verify the effectiveness of the proposed reconfiguration methodology. Chapter Five introduces the design steps of the novel multi-agent system in details along with the contributions of that design. The application of the designed multi-agent system to the IEEE 123-node test system along with the simulation results for the eight suggested scenarios are then presented in Chapter Six, followed by the analysis and comparison of these results. Finally, the conclusions and the future work are discussed in Chapter Seven.

Chapter 2
Literature Survey

2.1 Introduction

This chapter introduces a literature survey for the different topics and tools tackled in the thesis. The first section introduces the concept of the smart grid, its definitions, characteristics, and the challenges facing it. This is followed by, a discussion of the distributed processing; one of the most important aspects of the smart grid is provided. The third section tackles the reconfiguration problem of the distribution power systems and provides a literature survey for the problem and the approaches used for solving it. The tools utilized in this thesis for solving the reconfiguration problem for the smart distribution power systems; the genetic algorithm, the fuzzy logic and the fuzzy inference systems, and the multi-agent systems are then surveyed in the fourth, fifth, and sixth sections, respectively. The genetic algorithm and the fuzzy logic are used in the proposed hybrid algorithm to solve the reconfiguration problem in any distribution power system, while the multi-agent approach is used to build a multi-agent system having the capabilities of the distributed computing and decentralized decision making in order to apply the proposed reconfiguration methodology in the smart distribution power systems. The motivations of the distributed generations' integration into the smart distributed power systems are briefly surveyed in the seventh section. Eventually, a brief conclusion for the chapter is provided in the eighth section.

2.2 Smart Grid

This section introduces the definitions of the smart power grid or the smart distribution power system as well as its characteristics. The challenges facing the smart grid are discussed, in addition to discussing one of the most important aspects of the smart grid; the distributed processing, which is considered to be the backbone of the system proposed in this thesis for solving the reconfiguration problem in the smart distribution power systems.

2.2.1 Smart Grid Definitions

Smart grid is the new trend of improving the operation and performance of the distribution power systems by involving complex technologies, which arises in several definitions for the smart grid published by many authorities. Each definition is focusing on a certain characteristic of the smart grid and emphasizing some of its advantages. According to the U.S. department of energy (DOE), smart grid is *“an integrated energy system consisting of interconnected loads and distributed energy resources, which as integrated system can operate in parallel with the grid or in an intentional island mode”* [1]. This definition deals with the distributed property of the smart grid and its ability to integrate different energy resources operating in parallel with the grid. In May 2009, Miles Keogh [2] presented a different definition stating that *“the smart grid takes the existing electricity delivery system and makes it ‘smart’ by linking and applying seamless communications systems that can: gather and store data and convert the data to intelligence; communicate intelligence Omni-directionally among components in the ‘smart’ electricity system; and allow automated control that is responsive to that intelligence”*. The latter definition addresses the role of communications and automation in the new grid, which can make the grid act as being smart.

Another definition is introduced by the Ontario smart grid forum [3], which states that *“a smart grid is a modern electric system. It uses communications, sensors, automation and computers to improve the flexibility, security, reliability, efficiency, and safety of the electricity system. It offers consumers increased choice by facilitating opportunities to control their electricity use and respond to electricity price changes by adjusting their consumption”*. This statement not only defines the smart grid, but also states its resources and its characteristics along with an important aspect which is customer interaction. A similar definition also introduced by the U.S. DOE [4] states that *“an automated, widely distributed energy delivery network, the smart grid will be characterized by a two-way flow of electricity and information and will be capable of monitoring everything from power plants to customer preferences to individual appliances. It incorporates into the grid the benefits of distributed computing and communications to deliver real-time information and enable the near-instantaneous balance of supply and demand at the device level”*.

2.2.2 Smart Grid Characteristics

According to the above mentioned definitions, the characteristics of the smart grid can be identified as, but not limited to [1][5]:

- Self-healing which enables the grid to act automatically when any imperfection happens.
- Bidirectional power flow which authorizes the distributed generation integration so that power could flow from generation stations to customers and vice versa.
- Two way information flow which aims at monitoring the operation of the grid.

- Customer interaction which allows the customers to monitor and control their consumption and expenses through known as demand side management.
- Power quality improvement which copes up with the needs of the modernistic life such as the electric vehicles.
- Assets optimization which enables the grid to efficiently supervise the whole system.

2.2.3 Challenges Facing the Smart Grid

In order to fulfill these characteristics, many challenges have to be faced including [6]:

1. Environmental challenges: The integration of renewable energy sources becomes a must due to the shortage of traditional fuel and energy sources. These renewable sources are considered to be clean sources of energy and also act as alternatives to the traditional sources that generate excessive amounts of Green House Gaseous (GHG) in air resulting in the global warming, which threatens the whole world of a global catastrophe.
2. Market challenges: Customer satisfaction must be maintained under the balance between the quality of service offered and the price paid by the customers.
3. Infrastructure challenges: The power systems reliability improvement is a customer requirement. However, the infrastructure is suffering from components aging in the lack of sufficient investments besides the unprecedented load demand increase.
4. Innovative technologies challenges: The new technologies required for launching the smart grid are not mature enough, and at the same time, the existing power systems are not completely compatible to the new technology installation.

2.2.4 Distributed Processing

One of the most important technological challenges is the implementation of the communications layer, which is responsible for the information flow throughout the whole grid. This could be achieved by applying the concept of distributed processing. In classical distribution power systems, central processing is utilized through the supervisory control and data acquisition (SCADA) systems [1], which have the role of gathering information regarding different parts of the systems from some remote terminal units (RTUs) and making a decision based on the analysis of the gathered information. On the other hand, implementation of the smart distribution power systems requires a kind of distributed processing, which has the capability of managing the systems whose processing units are scattered over a huge physical area, which in turns arises the problem of the large communication delays between these units in the case of using a traditional central processing scheme [5]. Consequently, there is need for a distributed processing and a decentralized decision making schemes. Any distributed processing scheme is characterized by the following [7][8]:

1. The system information is distributed among many processors not only one processor, which limits the memory requirements to a great extent, since there is no need for a single processor with huge memory to accommodate the whole system information.
2. Each processor makes its own decision depending on its local information and this decision is broadcasted to all interconnected nodes in its area zone. Thus, the decision making scheme takes place in a decentralized way.
3. In the case of malfunction in any of the system processors, only the customers connected to that area are affected, and the rest of the system still performs efficiently.

After investigating the distributed processing mechanism; its properties and advantages, it is obvious now that its utilization is a necessity for the smart grid, and its implementation, taking into consideration its complexity, is a real challenge. The pressing need for a powerful communication layer implemented on top of the existing distribution power systems arises to face that challenge. One of the most powerful approaches suggested to implement that communication layer by applying the concept of the distributed processing is the insertion of the multi-agent systems discussed in the sixth section of this chapter.

2.3 Reconfiguration of the Distribution Power Systems

One of the main applications associated with the operation of the distribution power systems is the network reconfiguration in which the status of sectionalizing switches and tie switches is changed in order to optimize a certain objective function including reliability improvement [9], power losses minimization [10], service provision during faulty conditions, voltage profile improvement [11], and overloading prevention by load balancing. Since the reconfiguration problem is considered to be an optimization problem, it has been first tackled using different traditional techniques such as the heuristic search techniques including discrete branch and bound method [12][13], and switch exchange type heuristic technique [14][15][16][17]. Another traditional techniques used for solving the reconfiguration problem is the exhaustive search technique [18] and the simple branch exchange technique implemented in [19]. The previously mentioned techniques have faced many problems such as the large computational time and the convergence to a local optimal solution instead of the global optimal one; such problems required the interjection of soft computing and artificial intelligence techniques including the neural networks, the genetic algorithm, the fuzzy logic, and the swarm intelligence techniques. These techniques are believed to achieve a better performance with respect to a less computational time, a faster convergence rate, and a convergence to the global optimal solutions.

2.3.1 Problem Description

Power systems are mainly classified into three different types of systems; generation, transmission, and distribution power systems. The generation power systems are responsible for generating the electricity using electric generators, while the transmission power systems are responsible for transmitting the electricity from the generation stations to the distribution areas via transmission lines. The traditional distribution power systems have the role of distributing the electricity on the different customers connected to the system, while the smart ones will play the role of power supplying with their interconnected DGs. A distribution power system consists of many electric components including feeders, cables, lines, sectionalizing switches, tie switches, and transformers [20]. The distribution power systems are mainly designed to be radial systems in which no loops exist to prevent power circulation. This could be achieved through the opening and closing of the system switches, which have many roles such as determining the path in which the electric power flows from the feeder to the load, transferring the loads from one feeder to another to achieve load balancing and prevent overloading of feeders, and minimizing the electric power losses resulting from the electric current flowing in the cables. The amount of losses in the whole distribution system could be calculated by performing a load power flow.

The problem of distribution power systems reconfiguration is an optimization problem in which the states of the sectionalizing switches and tie switches are determined in order to optimize a certain objective function. This objective function could be the minimization of the power losses in the system, the balancing of the loads connected to each feeder to prevent the overloading of a certain feeder on the detriments of the others, the improvement of the system reliability, the improvement of the voltage profile of the system, or the provision of the service during faulty conditions. The objective function has to be optimized subject to some constraints and bounds including the maximum and minimum voltage levels, the maximum permissible current level, and the radial constraints that ensure that the radial topology of the system is maintained for all configurations. The optimal reconfiguration problem is not an easy one for two main reasons [20]. First, with respect to the computational load, the process requires a very heavy computational load. A system having N switches has 2^N different network configurations, and in order to determine the optimal configuration with respect to the objective function, a power flow has to be performed for all the possible network configurations. As the system gets bigger to accommodate more customers, the number of the switches increases, and the number of the possible configurations increases consequently; this requires more power flow calculations. Second, in spite of the heavy computational load associated with the tackled optimization problem, the process of determining the optimal network configuration has to be performed in real time in such a way that makes the distribution power system act immediately to any change or any fault happening.

2.3.2 Methodologies used for tackling the problem

The reconfiguration problem has been tackled by many researchers seeking for the optimization of different objective functions by using different methodologies. Haughton and Heydt [9] discussed the importance of having a rapid reconfiguration algorithm in the smart distribution systems which has the roles of the average interruption duration reduction and the un-served energy minimization. Genetic algorithm has a large share in the literature in being used for solving the problem of distribution power systems reconfiguration. Farahani *et al.* [10] used a discrete genetic algorithm in order to optimize the sequence of the loop selection, and then used the simple branch exchange method proposed in [19] to minimize the real power loss in every loop subject to the voltage limits and the maximum permissible current carried by the conductors. This method has the advantage of the fast convergence rate since the genetic algorithm is only used for optimizing the loop sequence selection. A modified genetic algorithm depending on reducing the population size by rejecting the infeasible solutions contradicting the radial topology of the system was used by Ming *et al.* [11] to minimize the real power loss, balance the system loads for overloading prevention, improve the voltage profile, and provide service to the customers at faulty conditions or during planned outages. The problem was solved subject to the voltage and current limits with less population size and less chromosome length. The work done by Radha *et al.* [21] aimed at minimizing the real power loss subject to the radial topology, the power source limits, the node voltage limits, and the branch current thermal stability constraints using a modified genetic algorithm with real valued genes and an adaptive mutation rate. Ravibabu *et al.* [22] implemented an improved genetic algorithm in order to balance the loads, and minimize the real power losses resulting from the faults and feeder overloading subject to the voltage and the current limits constraints. The concepts of accentuated crossover and directed mutation were introduced by Mendoza *et al.* [23] to minimize the real power losses subject to the voltage limits, the current limits, the radial system topology, and the load balancing constraints in a restricted population genetic algorithm. Minimizing the energy losses as well as minimizing the cost of the energy losses was the target of Karegar *et al.* [24] who implemented a multi-objective binary genetic algorithm with adaptive mutation to solve the reconfiguration problem subject to the voltage and current limits constraints. Spanning trees were used with the genetic algorithm in the work done by Torres-Jimenez *et al.* [25] to minimize the power losses subject to the radial topology, the voltage limits, the current limits, and the maximum power limits constraints.

The fuzzy logic has been used in tackling the reconfiguration problem as well, and it has either been used alone or used along with the genetic algorithm in hybrid approach. Sarfi and Solo [26] used a hybrid fuzzy system to prevent the violation of the network operational constraints. They used the fuzzy antecedents of recent temperature trend, line section loading, transformer aging, and voltage level guidelines to obtain the fuzzy consequent of a standardized degree of desirability in order to minimize the real power losses subject to the radial system topology, the acceptable fault current limits, the voltage limits, the current capacity limits, the service priority for critical customers, and the transformer aging constraints. A fuzzy mutated genetic algorithm was used by Prasad *et al.* [27] to minimize the real power losses and improve the power quality by minimizing the voltage deviation index subject to the radial topology, the voltage limits, and the current limits constraints. They used the fuzzy antecedents of the standard deviation of the fitness distribution and the average fitness to control the fuzzy consequent of the standardized degree of desirability. The crossover and mutation probabilities of the genetic algorithm are controlled by two fuzzy controllers in the work done by Ah King *et al.* [28] to minimize the power losses subject to the radial topology, the power source limits, the node voltage limits, and the branch current thermal stability constraints.

The usage of the soft computing and artificial intelligence techniques has extended to include the neural networks such as the work done by Bouchard *et al.* [29] when they used the Hopfield neural networks to minimize the total line power losses. The multilayer perceptron neural networks have been used along with some clustering techniques in the work done by Salazar *et al.* [30] to minimize the active power loss subject to the source power limits, the voltage limits, and the current limits constraints. A modified particle swarm optimization technique has been used by Abdelaziz *et al.* [31]

to minimize the power losses subject to the nodal voltage limits, the line current limits, and the radial topology constraints.

2.3.3 The proposed methodology for tackling the problem

The genetic algorithm and the fuzzy logic have been chosen to be utilized in a hybrid approach in order to solve the reconfiguration problem in the distribution power systems. The genetic algorithm has been chosen for its high accuracy in reaching the global optimal solution for most of the optimization problems, and the fuzzy logic is selected in order to build two different fuzzy controllers which have a very important role in speeding the convergence rate of the genetic algorithm and decreasing its computational time to a great extent such that the global optimal configuration could be achieved in the least time with the fastest convergence rate. After that, the multi-agent approach is selected in order to build a multi-agent system having the powerful capabilities of distributed processing and decentralized control such that the reconfiguration methodology could be applied to solve the problem in the smart distribution power systems. Brief reviews for each of the genetic algorithm; the fuzzy logic and the fuzzy inference systems; and the multi-agent systems are introduced in the next three sections.

2.4 Genetic Algorithm

Genetic algorithm is considered to be one of the oldest evolutionary computation techniques found in the literature. Evolution can be defined as the process of the life adaptation to the surrounding environmental changes in which the characteristics of the new offspring of a certain creature is a mixture of some characteristics from the parents and some other new characteristics that show up due to the effect of the environmental evolution [32]. The evolution theory introduced by Charles Darwin in 1859 is the cornerstone of all the research held in the field of the evolutionary computation in which Darwin’s principle “survival of the fittest” has been the cell upon which all the theories and algorithms of that field were built [33][34].

A straightforward explanation of the stages of any evolutionary computation technique including the genetic algorithm can be found in [32]. First, an encoding mechanism is chosen to encode the population generated in which a chromosome is a member of the population, a gene is a string in the chromosome, a locus is the position of the gene within the chromosome, and alleles are the possible values of the genes. Binary encoding, floating-point encoding, and gray encoding are among the most widely-known encoding mechanisms. Second, an initial population is initialized to solve the tackled objective function or the so called fitness function. After that, the evolutionary operators such as: reproduction, selection, crossover, and mutation come into play. Reproduction is the ability of individual genes to transfer from a generation to another leading to the increase in the population number and the gain of better characteristics [35]. Selection is the process of picking some population members up to be involved in the reproduction process [32]. Crossover is the process of combining the genes of the two parents in a random fashion resulting in the formation of the new genetic structure of the children, while mutation is the process of randomly changing the alleles of certain genes in the chromosome leading to introducing completely new population members and guaranteeing the impossibility of errors replication [32][35]. The evolutionary operators are the core of the evolutionary computation since they are the tools by which the offspring gain better characteristics that allow them to approach the global optimum solution. Finally, working parameters including the population size and the chromosome length are determined.

The history of the evolutionary computation techniques goes back to the second half of the 20th century in concurrent with the emergence of the computers and their utilization in modeling, analyzing, and simulating the biological systems including the work done by Bremermann in 1958 [36][37]. However, the work done by John Holland [38] at the University of Michigan in the late 1960s and the early 1970s launched the spark of genetic and evolutionary algorithms.

Genetic algorithm was first introduced by Holland [38] in 1975 [33] in which fixed-length binary strings with binary mutation and binary crossover were used, while the real-coded genetic algorithm, was implemented by Goldberg [39] in 1991 [36]. Mutation and crossover probabilities were optimized and viewed as a controlled Markov process in the work done by Cao and Wu [40] in 1999, and an extended multi-objective genetic algorithm was introduced by Rodriguez-Vazquez *et al.* [41] in 2004 [36]. The genetic algorithm theory depends on the schemata theory, and its operation is inspired from the biological evolution of the living organisms. The procedure of the genetic algorithm is simply discussed in the flowchart shown in Figure 2.1 [42][43][44].

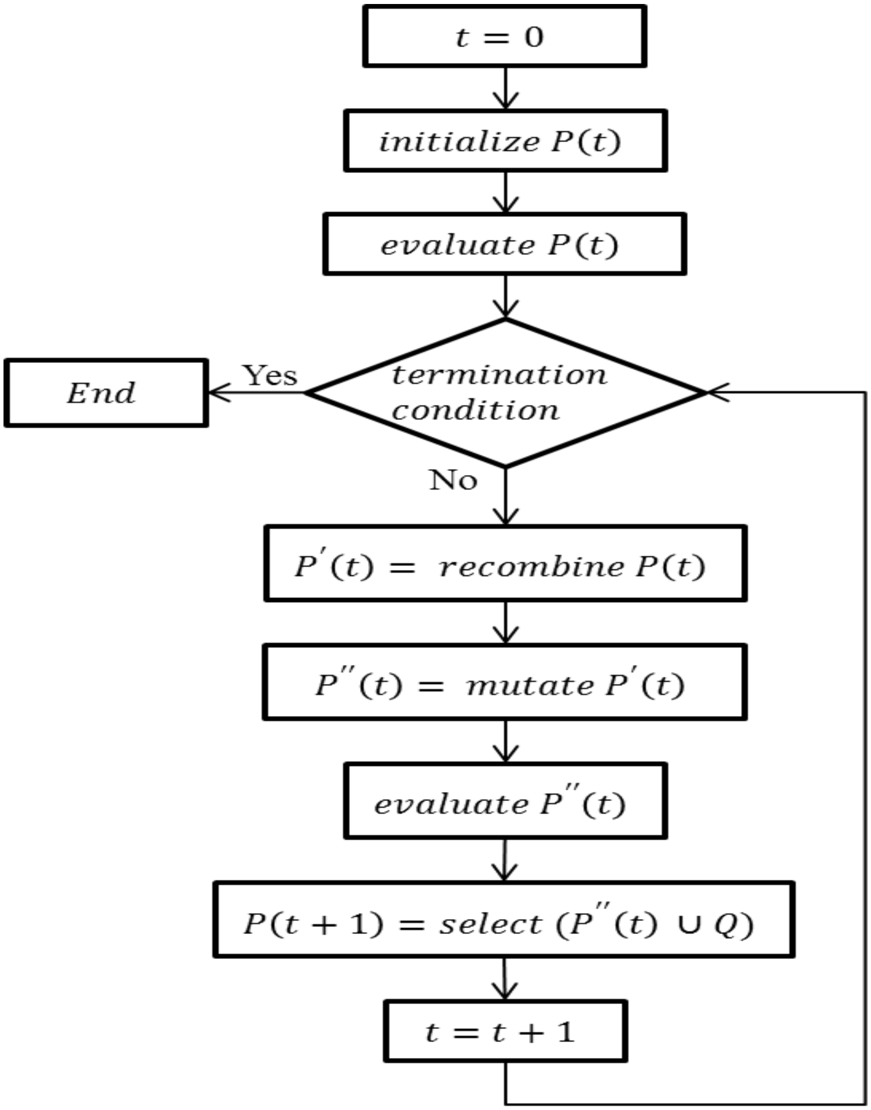

Figure 2.1 Genetic algorithm flowchart.

where $P(t)$ is the population of μ individuals at generation t, Q is a special set of individuals that might be considered during the selection process, e.g. $Q = P(t)$ or $Q = \emptyset$, and $P''(t)$ is the offspring with population $\lambda \geq \mu$ generated via the combination and mutation of selected individuals from the original population $P(t)$.

A more detailed discussion of the genetic algorithm can be summarized in the following steps [32][35][42][45][46][47][48][49][50]:

1. The problem to be solved is well-modeled, and the objective or fitness function that reflects the fitness of each population individual is well-defined.

2. The encoding mechanism to be used in the encoding of each population individual is chosen. Each individual has to be encoded as a vector “chromosome”v_i $(i = 1,2, \dots, p)$, where p is the population size.

3. The population of the chromosomes representing all possible solutions in the search space is initialized according to the encoding mechanism chosen in the previous step.

4. The fitness value of each chromosome $f(v_i)$ is evaluated.

5. The total fitness of all chromosomes in the population generated is calculated according to the equation:

$$F = \sum_{i=1}^{p} f(v_i). \tag{2.1}$$

6. A probability of selection p_i of each chromosome indicating the possibility of that chromosome to be selected for the new generation formation is calculated according to the equation:

$$p_i = \frac{f(v_i)}{F}. \tag{2.2}$$

7. According to the calculated probability of selection p_i, some chromosomes are selected for the reproduction process. Roulette wheel selection mechanism is one of the simplest and most common selection mechanism utilized.

8. Crossover is applied on some chromosomes according to the crossover probability or crossover rate p_c, which determines how many individuals out of the whole population size have to be resulting from crossover. Crossover is responsible for generating the child chromosomes sharing the characteristics of their parents.

9. Mutation is applied on some chromosomes according to the mutation probability or the mutation rate p_m. There are many mutation functions that could be used to determine that mutation rate including the Gaussian mutation function, the uniform mutation function, and the adaptive mutation function as discussed later in this section. Mutation is responsible for generating the child chromosomes having completely new characteristics in order to widen the search space by introducing new members to the population.

10. The new offspring, generated after applying the genetic operators (selection, crossover, and mutation), is set to be the new population. Generally, a small number of chromosomes is passed from the current generation to the next generation without applying any of the genetic algorithm operators. Another number of chromosomes of the new generation is formed by applying the crossover operator. This number is determined according to the crossover rate as discussed in step 8. The rest of the chromosomes of the new generation are formed by applying the mutation operator with a certain mutation rate as discussed in step 9.
11. The process is stopped if the required fitness value is achieved or the maximum number of generations is reached; otherwise, the process is repeated from step 4.

One of the most important parameters of the genetic algorithm that has to be carefully determined is the mutation rate, which determines the probability of each gene in the chromosome to be mutated. In the proposed reconfiguration methodology, three different mutation functions are used to select the mutation rate and apply the mutation operator on the number of chromosomes to be mutated beside the fuzzy controller designed to control the adaptive mutation as discussed in the next chapter. The three different mutation functions used are discussed in the following subsections.

2.4.1 Gaussian Mutation Function

A child chromosome X' is generated by applying the mutation operator on the parent chromosome X chosen to be mutated. Mutation is achieved by adding a random variable from a Gaussian distribution with a zero mean and a predetermined standard deviation to each component of the parent chromosome X as expressed in the following equation:

$$x_i' = x_i + N(0, \sigma_i). \qquad (2.3)$$

2.4.2 Uniform Mutation Function

The uniform mutation function generates the mutated child chromosome by selecting a certain fraction of the parent chromosome to be mutated according to the mutation rate which determines the probability of each gene in the chromosome to be mutated. The mutation rate is considered to be an input to the uniform mutation function that is used to determine which genes in the chromosomes are to be mutated and which are not, and it remains constant throughout the whole genetic algorithm execution. This means that all the chromosomes selected for the mutation in every generated generation have the same mutation rate. This function doesn't take into account the improvement or the deterioration of the fitness function.

2.4.3 Adaptive Mutation Function

The adaptive mutation function is similar to the uniform mutation function with one major difference; that is the mutation rate is no more constant but adaptive. The adaptive mutation rate means that the mutation rate is changing for every generated generation according to a certain function such as the function in [21] and [51]:

$$\rho_m(t+1) = \begin{cases} \rho_m(t) - \rho_{m_{step}} & if\ f_{best}(t+1) = f_{best}(t) \\ \rho_m(t) & if f_{best}(t+1) < f_{best}(t) \\ \rho_{m_{end}} & if\ \rho_m(t) - \rho_{m_{step}} < \rho_m(end) \end{cases} \tag{2.4}$$

$$\rho_{m_{step}} = 0.001 \tag{2.5}$$

$$\rho_{m_{end}} = 0.05 \tag{2.6}$$

$$\rho_{m_{initial}} = 1 \tag{2.7}$$

where $\rho_m(t+1)$ and $f_{best}(t+1)$ are the mutation rate and the best fitness value at the $t+1$ generation, $\rho_m(t)$ and $f_{best}(t)$ are the mutation rate and the best fitness value at the t generation, $\rho_{m_{step}}$ is the step by which the mutation rate changes after each iteration, $\rho_{m_{end}}$ is the final permissible value for the mutation rate, and $\rho_{m_{initial}}$ is the initial value of the mutation rate for the first generation, which is equivalent to $\rho_m(0)$.

In the adaptive mutation function shown above, the mutation rate is given an initial value for the first generation, and then is decreased from one generation to the next by the predetermined mutation step if the best fitness value in those two generations doesn't change, which means that the there is no improvement in the best fitness value. On the other hand, the mutation rate remains the same if the best fitness value in the new generation is smaller than that of the previous generation, which means that the best fitness value is improving. According to the shown adaptive mutation function, the mutation rate is not allowed to become smaller than a certain permissible value called the final mutation rate.

2.5 Fuzzy Logic and Fuzzy Inference Systems

2.5.1 Fuzzy Logic

Fuzzy logic is the platform that enables the representation of approximate reasoning information that can’t be represented by the crisp logic or by the Boolean algebra [32]. Zadeh was the first one introducing the fuzzy logic in the mid-1960s for the sake of representing some approximate information. Instead of representing the information by crisp values in the crisp logic or by 0 and 1 values in the Boolean logic, the information is represented in the fuzzy logic, by what’s called the membership functions such that every input has a membership degree that represents how much this input belongs to a certain membership function. This means that a certain input could belong to a certain membership function with a certain membership degree, and belong to another membership function with another membership degree at the same time. The fuzzy logic uses some operators the same way that any other logic do including the complement “negation”, the union “disjunction”, and the intersection “conjunction”. In addition, the famous properties including the commutative property, the associative property, the distributive property, the absorption property, and DeMorgan’s laws can be applied on the fuzzy logic as well.

2.5.2 Fuzzy Inference Systems

A fuzzy inference system is a rule based system that is mainly used as a controller [28]. It uses if-then rules to control the output of the system to be controlled according to certain given inputs. The inputs and outputs of the system have to be crisp in order to be utilized in the controllers of the real life; however, the fuzzy inference system is based on the fuzzy logic and doesn’t use crisp logic. Thus, a fuzzy inference system is designed in such a way that can tackle this problem. A typical fuzzy inference system is shown in Figure 2.2 [28].

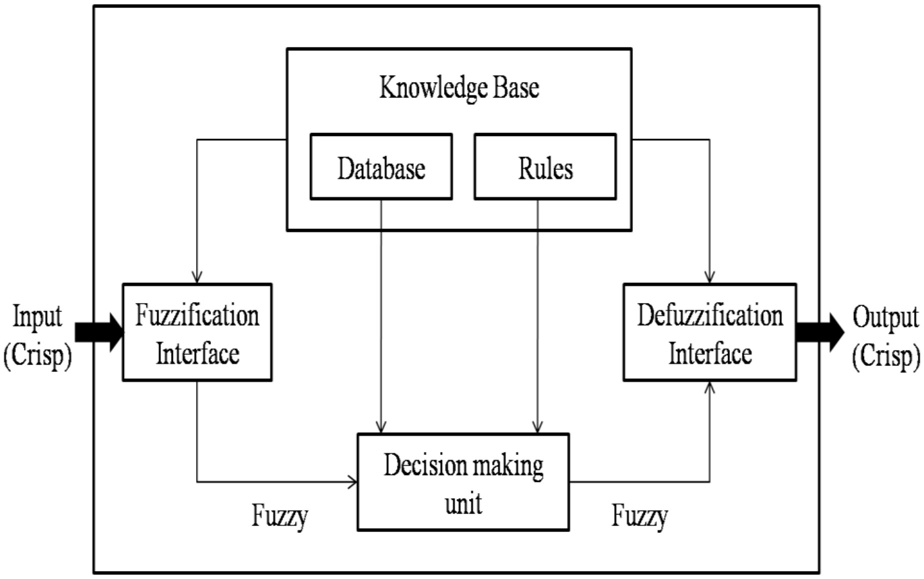

Figure 2.2 Fuzzy inference system.

As shown in the above figure, the fuzzy inference system consists of a fuzzification interface, a knowledge base unit consisting of a database and some rules, a decision making unit, and a defuzzification interface. First, the membership functions of the inputs and the outputs of the system are designed, and the universe of discourse defining the set of the permissible values for the system inputs and outputs is determined. Second, the set of the if-then rules linking the inputs to the outputs is designed in the way that forces the fuzzy inference system to perform the desired controlling function. After that, the crisp input is passed through the fuzzification interface which has the role of converting the input from being crisp to being fuzzy. This requires determining the membership grade of the input that relates that input to each one of the membership functions. In other words, the crisp input is mapped onto the input membership functions to determine how much that input belongs to every one of those functions. There are many membership functions used for the sake of fuzzifying the crisp inputs including the triangular, the trapezoidal, the Gaussian, and the bell-shaped functions [32]. The next step is to use the knowledge base unit containing the system rules and database along with the decision making unit to make the proper decision that controls the system output according to the coming input. The resulting fuzzy output is the area under the output membership functions arising from the combination of the areas formed after applying each of the fuzzy rules. Finally, the defuzzification interface is used to convert the fuzzy output resulting from the previous step to a crisp output that can be used in the real life controllers. The defuzzification step implies the extraction of the crisp output from the resultant fuzzy area by a certain defuzzification method including the centroid, the bisector, the mean of maximum, the largest of maximum, and the smallest of maximum defuzzification methods [32].

In the proposed reconfiguration methodology, the discussed concepts of the fuzzy logic and the fuzzy inference systems are utilized in designing two different fuzzy controllers. The first one is to reject any infeasible configurations that violate the radial topology of the system, and the second one is to control the adaptive mutation rate of the genetic algorithm. The details of the methodology proposed for tackling the discussed reconfiguration problem and the design of the two fuzzy controllers are discussed in the next chapter.

2.6 Multi-agent Systems

The multi-agent approach is the approach by which the proposed reconfiguration methodology could be applied to the smart distribution power systems due to the distributed processing and decentralized decision making capabilities that feature the multi-agent systems. This section introduces the definitions of the intelligent agents and the multi-agent systems, as well as stating the characteristics of the intelligent agents and surveying the applications of the multi-agent approach in the power systems. Finally, the general steps that have to be followed in order to build any multi-agent system are then discussed.

2.6.1 Definitions

Russell and Norvig [52] defined an agent as *"anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors"*. A rational agent is the one working on a certain performance optimization. According to those two definitions, agents could include humans, robots, and software programs [53]. This definition was extended by Wooldridge [54] to include the intelligent agent; the one that could display flexible autonomy.

A multi-agent system (MAS) is a system consisting of several intelligent agents, each of which has its own goal, and they are all communicating together via an agent communication language (ACL) which is discussed in Appendix A.

2.6.2 Characteristics

An intelligent agent is characterized by the following [1][55]:

1. Autonomy: The ability of independent decision making according to the gathered information without the need of a controlling center or a human interference.

2. Reactivity: The ability to react to various surrounding environmental such that the action taken is compatible with the agent’s function.
3. Pro-activity: The ability to “take the initiative” as stated by Wooldridge, which requires a dynamic change in the behavior subject to any surrounding change, which by the turn results in an ultimate goal achievement.
4. Sociality: The ability of social interaction among intelligent agent which goes beyond the simple task of transmitting and receiving data to a more complex task of data discussion and decision making.

2.6.3 Applications in Power Systems

MAS has been recently used for various applications in power systems such as monitoring and diagnostics; distributed control; modeling and simulation; and protection [55]. Among these applications [5]:

- Reactive power dispatching control by Baran and El-Markabi [56] in 2007.
- SCADA system data real-time monitoring by Davidson *et al.* [57] in 2006.
- Power transformers condition monitoring by McArthur *et al.* [58] in 2004.
- Substation Automation by Buse *et al.* [59] in 2003.
- Power system protection by introducing the concept of relay agent by Tomita *et al.* [60] in 1998.
- Power systems marketing by Krishna and Ramesh [61][62] in 1998.
- Economic reliability test by Koesrindartoto [63] in 2005.
- Transmission system cost allocation to users by Zolezzi and Rudnick [64] in 2002.
- Real-world market modeling by Koritarov [65] in 2004.
- Micro-grid control by Dimeas and Hatziargyriou [66] in 2005.
- A distributed smart machine tool service system by Kao and Chen [67] in 2010.
- Distribution power system reconfiguration by Belkacemi and Feliachi [68] in 2010.

- Managing a power distribution system with plug-in hybrid electrical vehicles by Logenthiran and Srinivasan [69] in 2011.
- Power distribution grid self-optimization by Merdan *et al.* [70] in 2011.
- Power distribution system load management by Biabani *et al.* [71] in 2012.
- Power networks reliability modeling by Prymek *et al.* [72] in 2011.
- Facilitating smart distribution networks operation through the integration of agent-based functions by Nguyen *et al.* [73] in 2011.

2.6.4 General Steps for Building MAS

In order to build an MAS, general guidelines should be followed as suggested in [1]:

1. Agent Specification: Specifying the agents used in the system to be built in addition to identifying the role of each one.
2. Application Analysis: Analyzing the problem to be solved and assigning responsibilities to the agents specified in the first step.
3. Application Design: The problem to be solved has to be modeled in order to design the general platform of the agents and the knowledge used by each of them.
4. Application Realization: This step is the bridge between the designing stage performed in the previous steps and the implementation stage to be carried out after this step including agent creation, task configuration, and code generation.
5. Application Implementation: This is the implementation stage of what has been designed, modeled, and configured in the previous steps.

2.7 Distributed Generations

Distributed generations (DGs) have made various changes in the distribution power systems since they are used for improving the system voltage profile, power quality, losses, and reliability [74]. DGs involve the utilization of small power generating units installed in specific locations in the distribution power systems [75]. One of the main features of the smart grid is the interjection of various types of DGs in the system in order to take the advantages of their installation as discussed later in this section. DGs can be classified into several types according to the nature of the generating energy source. The most common DGs are the wind turbines, the photovoltaic systems, the small hydro power systems, the micro-turbines, the diesel DGs, the natural gas-based DGs, and the fuel cell [5]. In this section, the motivations of installing the DGs are briefly surveyed.

2.7.1 Motivations of Distributed Generation Utilization

The motivations of using and installing the DGs in the smart distribution power systems can be listed as follows [5]:

- The continuous increase in the load demand due to the 21st century's life style requires the installation of distributed generation units near the loads centers.
- The leakage of the traditional energy sources requires looking for other forms of renewable energy sources.
- The environmental pollution is threatening the whole world with catastrophic consequences. Thus, the interjection of clean energy sources is a must.

2.8 Conclusion

The smart power grid is the future trend of applying the smart grid concepts in the distribution power systems in which autonomy, self-healing, and decentralized control are highly required to be applied. One of the most important challenges facing the smart grid realization is the distributed processing in which the information can be handled through different processing nodes, and the decision is made in a decentralized fashion after each node has shared its information with the other nodes and performed its share in the computations.

In order to solve the reconfiguration problem in the distribution power systems, the genetic algorithm and the fuzzy logic have been selected to be utilized in a hybrid approach in which the genetic algorithm is used to search for the optimal system configuration that optimizes a certain objective function, and the fuzzy logic is used to build two different fuzzy controllers in order to enhance the performance of the genetic algorithm.

For the sake of applying this proposed reconfiguration methodology in the smart distribution power systems, a decentralized distributed approach has to be utilized. Thus, the multi-agent systems show up to take this responsibility in such a way that ensures the distributed processing and the decentralized decision making, and at the same time guarantees the high performance of the reconfiguration methodology in the presence of distributed generations in the system. A detailed discussion of the proposed reconfiguration methodology is provided in the next chapter, while the design of the proposed multi-agent system is discussed in Chapter Five.

Chapter 3
The Proposed Reconfiguration Methodology

3.1 Introduction

As discussed in the previous chapter, the reconfiguration methodology proposed in this thesis is based on the utilization of two different fuzzy controllers along with the genetic algorithm in a hybrid algorithm in order to solve the reconfiguration problem in distribution power systems. The target of this approach is to find the optimal reconfiguration that optimizes a certain objective function subject to some constraints which have the role of ensuring that the system is working properly. The purpose of the two fuzzy controllers is to control certain parameters that affect the performance of the system, while the role of the genetic algorithm is to optimize the desired objective function. The first fuzzy controller is called the infeasible configurations fuzzy rejector and has the role of rejecting any infeasible configurations coming out as solutions suggested by the genetic algorithm and violating the radial topology of the system, while the second fuzzy controller is called the adaptive mutation fuzzy controller and it is targeting the adaptation of the mutation rate of the genetic algorithm in order to be able to reach the global optimal solution with the fastest convergence rate consuming the least computational time.

In this chapter, the proposed reconfiguration methodology is discussed in details starting from the choice of the system to be studied, passing through the problem formulation, the design of the genetic algorithm, the design of the two fuzzy controllers, and ending by the application of this proposed methodology to the distribution power system chosen to be studied. A complete flowchart for the design steps is shown in Figure 3.1 and a detailed description for each design stage is provided in the following sections.

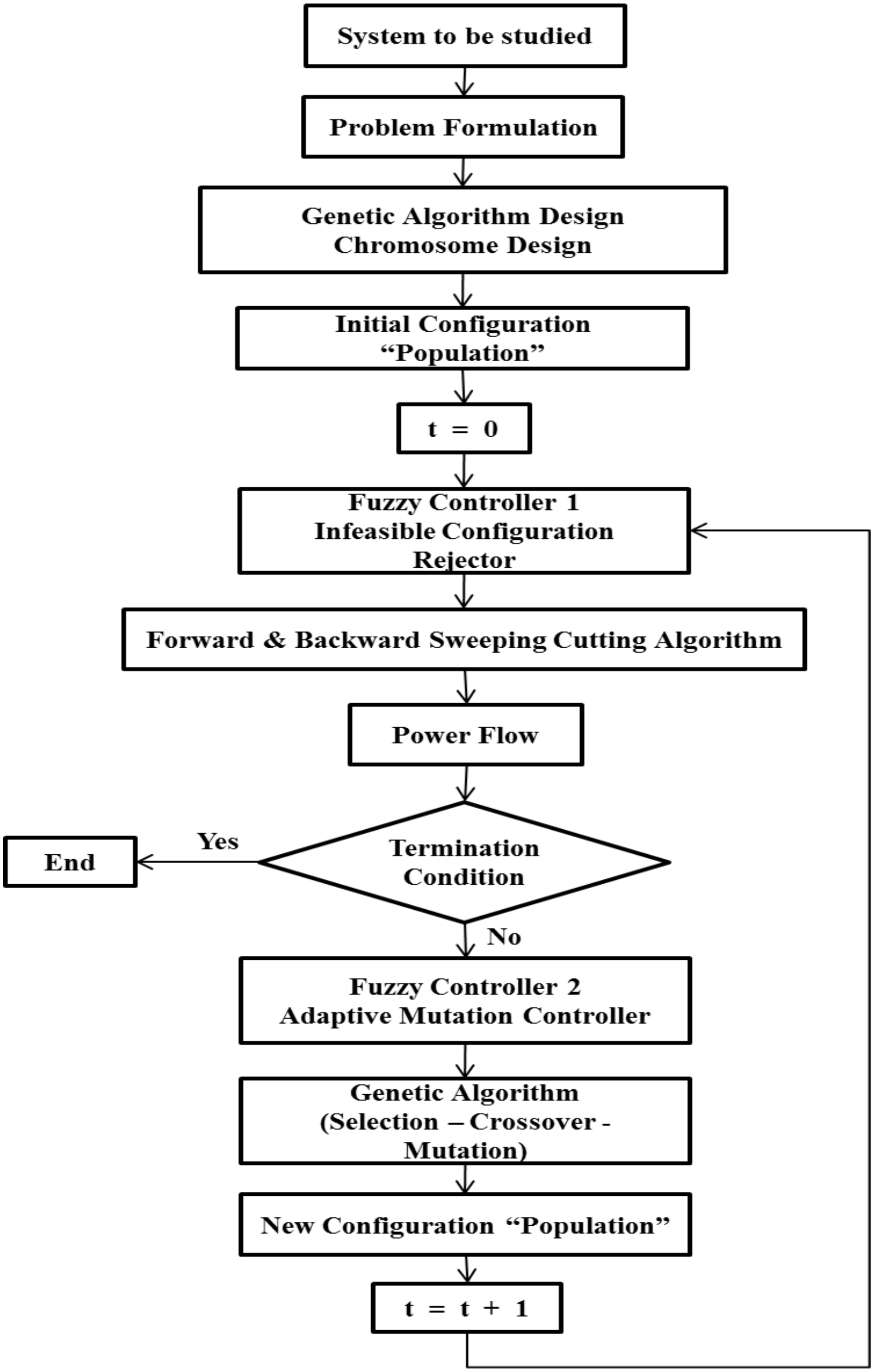

Figure 3.1 A flowchart for the proposed reconfiguration methodology.

3.2 The System Under Study

The proposed methodology is first applied to solve the reconfiguration problem in the 16-node distribution power system shown in Figure 3.2 [20][21][22][28][51][76][78]. This particular system is selected because it is a small distribution power system with a small number of nodes and switches. Therefore, it facilitates the discussion of the different steps of the proposed reconfiguration methodology and at the same time facilitates the application of the different scenarios suggested to demonstrate the effectiveness and the capabilities of the proposed methodology over the traditional methodologies as discussed in the next chapter. After that, the proposed reconfiguration methodology is applied to another three large distribution power systems as presented in the next chapter as well.

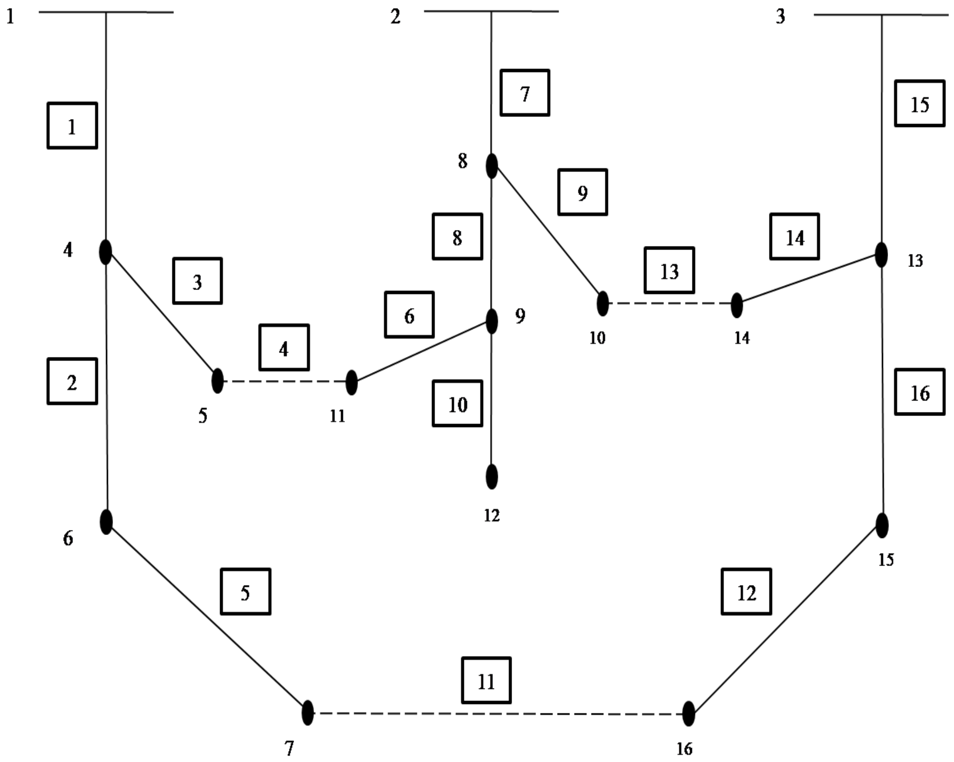

Figure 3.2 The 16-node distribution power system to be studied.

The system shown above has 16 nodes and three feeders at nodes 1, 2, and 3. For the sake of studying the performance of the proposed reconfiguration methodology on this system, each link of the 16 links in the system is supposed to have a switch. Thus, the system has 16 switches indicated by the numbers in squares in the figure shown above. In the initial configuration of the system, the three open switches are indicated by the dotted lines in the above figure.

3.3 The Problem Formulation

The proposed reconfiguration methodology is targeting the minimization of the real power loss in all lines of the system subject to the constraints that ensure the proper performance of the system. The objective function can be formulated as:

$$Minimize \sum_{i=1}^{N} S_i I_i^2 R_i \tag{3.1}$$

Subject to:

1. $V_{min} \leq V_i \leq V_{max}$ (3.2)
2. $I_i \leq I_{max}$ (3.3)
3. The radial topology has to be preserved.
4. Every load has to be connected to one feeder only.
5. All feeders have to be in service.

Where S_i is the state of the i^{th} switch i.e. $S_i = 0$ if the switch is opened, and $S_i = 1$ if the switch is closed, I_i is the current passing in the i^{th} line, R_i is the real resistance of the i^{th} line, V_i is the voltage of the i^{th} node, N is the total number of the lines, V_{min} and V_{max} are the minimum and the maximum permissible voltage levels for any node, respectively, and I_{max} is the maximum permissible current to flow in any line.

The objective function is seeking the finding of the optimal configuration that minimizes the total real power losses in the system such that all nodes voltages lie within the permissible voltage limits, and all line currents don’t exceed the maximum permissible limit. The radial topology constraint ensures that the radial topology of the system is always preserved, which requires the rejection of any infeasible configurations that violates this condition such as any configuration leading to loop formulation in the system. The last two constraints ensure that every load is connected to only one feeder such that all the loads are served and no load is disconnected, and that all the feeders are in service. For the realization of these constraints, only three switches in the given system have to be open, while the other switches have to be closed.

3.4 The Genetic Algorithm Design

The most important aspects in the design of the genetic algorithm which is utilized to optimize the objective function discussed in the last section are the chromosome design, and the mutation function design. First, the chromosome is designed to represent the three switches chosen to be open in the system. Since the given system has 16 switches and only three switches have to be open, the chromosome is designed to be a bit string having 12 bits in which each consecutive four bits are used to represent an open switch. To illustrate the design concept, the initial configuration "open switches" of the given system is shown in Figure 3.3 below.

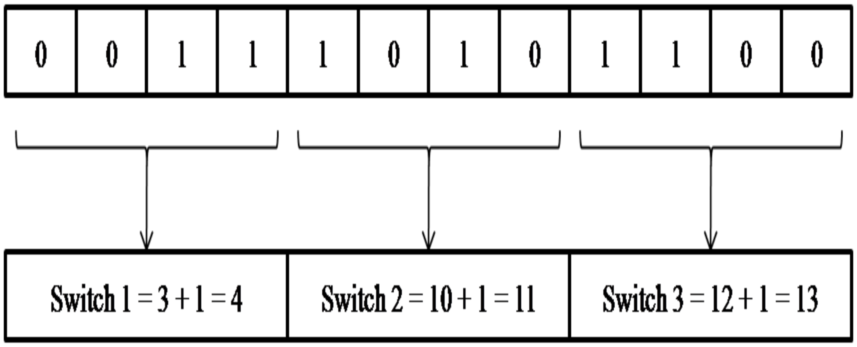

Figure 3.3 The initial configuration "open switches" of the system under study.

This initial configuration is chosen to be the initial population of the genetic algorithm. As mentioned before in the second chapter regarding the mutation function design, three different mutation functions are used to determine the mutation rate; the Gaussian mutation function, the uniform mutation function, and the adaptive mutation function. In addition, an adaptive mutation fuzzy controller is designed as discussed in the seventh section of this chapter.

3.5 The Infeasible Configurations Fuzzy Rejector

For the sake of rejecting all the infeasible configurations that might show up in the population of the genetic algorithm, a fuzzy controller is designed such that every configuration is passed through that controller; and according to the controller rules, any infeasible configuration is rejected and converted to be a feasible one. The design of the infeasible configurations fuzzy rejector can be more illustrated by showing the input and output membership functions in Figure 3.4 below, where all the inputs and the outputs have the same membership functions as discussed below.

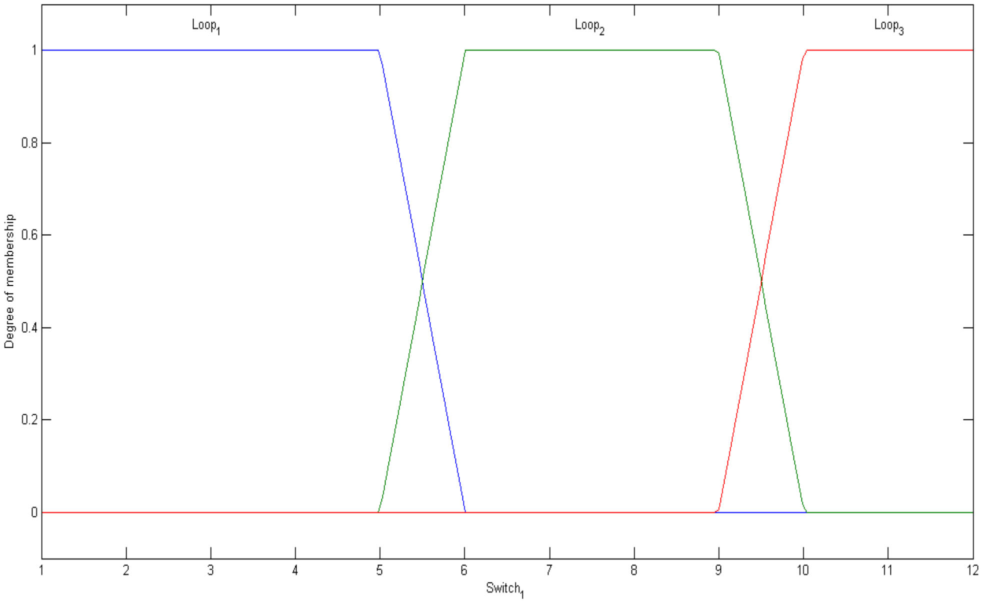

Figure 3.4 The first fuzzy controller membership functions.

The three membership functions shown above represent three groups of system switches; each one has the group of switches that could result in a loop in the case of being all closed at the same time. The numbers shown in the above figure under every loop are not the switches numbers as indicated in the system figure, but they are the indices of every loop switches as indicated below:

Loop 1: [2 5 11 12 16]

Loop 2: [3 4 6 8]

Loop 3: [9 13 14]

Thus, the index number 1 under Loop 1 in the figure above is corresponding to the switch number 2 that links loads number 4 and 6. To fulfill all the requirements and the constraints of the system, switches number 1, 7, and 15 have to be always closed such that all the three feeders are always in service. Switch number 10 also has to be always closed such that the load number 12 is always connected. In addition, only one switch from each of the three switch groups shown above has to be open. In case two or three switches of the same switch group are open at the same time, one of the two other switch groups will have all its switches closed resulting in a loop formation. Thus, the infeasible configuration fuzzy rejector has the role of ensuring that only one switch from each switch group is open through the controller rules shown in Table 3.1.

Table 3.1 Infeasible Configurations Fuzzy Rejector Rules

Rule	Input			Output	
	Switch 1	***Switch 2***	***Switch 3***	***Switch 2***	***Switch 3***
1	Loop 1	Loop 1	Loop 1	Loop2	Loop 3
2	Loop 1	Loop 1	Loop2	Loop 3	X
3	Loop 1	Loop 1	Loop 3	Loop2	X
4	Loop 1	Loop2	Loop 1	X	Loop 3
5	Loop 1	Loop 3	Loop 1	X	Loop2
6	Loop2	Loop 1	Loop 1	X	Loop 3
7	Loop 3	Loop 1	Loop 1	X	Loop2
8	Loop 2	Loop 2	Loop 2	Loop 1	Loop 3
9	Loop 2	Loop 2	Loop 1	Loop 3	X
10	Loop 2	Loop 2	Loop 3	Loop 1	X
11	Loop 2	Loop 1	Loop 2	X	Loop 3
12	Loop 2	Loop 3	Loop 2	X	Loop 1
13	Loop 1	Loop 2	Loop 2	X	Loop 3
14	Loop 3	Loop 2	Loop 2	X	Loop 1
15	Loop 3	Loop 3	Loop 3	Loop 1	Loop2
16	Loop 3	Loop 3	Loop 1	Loop2	X
17	Loop 3	Loop 3	Loop 2	Loop 1	X
18	Loop 3	Loop 1	Loop 3	X	Loop2
19	Loop 3	Loop 2	Loop 3	X	Loop 1
20	Loop 1	Loop 3	Loop 3	X	Loop2
21	Loop 2	Loop 3	Loop 3	X	Loop 1

As shown in the above table, every population generated by the genetic algorithm is passed though the fuzzy controller as a combination of the three switches chosen to be open. All the combinations having more than one open switch from each group are examined. Then, the outputs are determined as shown in the above table to ensure that there is only one open switch from each switch group. The first switch state is always unchanged. Thus, only the second and the third switches are shown in the output. The symbol X in the table above indicates that this switch state is not changed. If the three switches resulting from the genetic algorithm are already from the three groups, they are passed through the fuzzy controller without any change.

3.6 The Forward & Backward Sweeping Cutting Algorithm

After the three open switches are determined by the genetic algorithm and ensured to be from the three different switch groups by the infeasible configurations fuzzy rejector, the forward & backward sweeping cutting algorithm is designed in order to cut the system into three different trees such that the load power flow could be applied on every tree. The algorithm is described in the following steps:

1. The open switches resulting from the fuzzy controller are inputted to the algorithm, and used to form a set of forbidden links.
2. The number of feeder nodes is inputted to the algorithm.
3. The first feeder node is selected to start forming its tree.
4. All the forward links are swept starting from that feeder node to all other nodes
5. If that forward link is not from the set of the forbidden links, that link is considered to be from the tree belonging to that feeder. If the forward link is from the set of the forbidden links, that link is not considered and another forward link is swept.
6. If a forward link is selected in the previous step, the next node is considered and all the forward links from that node to all other nodes are swept the same way in steps 4 and 5.
7. After sweeping all the forward links, steps 4, 5, and 6 are repeated but on the backward links such that they are all swept in the same way.
8. If all the forward and backward links are swept, the tree of the first feeder is formed starting from that feeder node, passing through all the links connected to that feeder node, and ending by the leaf nodes.
9. Another feeder node is selected and the steps from 4 to 8 are repeated until the tree of every feeder is formed.
10. All the resultant trees are passed to the function performing the power flow and the total power loss is calculated. After that, the termination condition of the genetic algorithm is tested such that the process is terminated if the minimum power loss is achieved or the maximum number of generations is reached.

3.7 The Adaptive Mutation Fuzzy Controller

Instead of using the adaptive mutation function discussed in the second chapter, a second fuzzy controller is designed to control the adaptive mutation rate depending on the change of the best fitness value through the different generations. The inputs of the fuzzy controller are the best fitness difference ΔF_{best} and the delta best fitness difference $\Delta^2 F_{best}$ shown in the equations below, while the output of the fuzzy controller is the mutation rate step [28][76][77]. The inputs are normalized in the range of $[-1 : 1]$ and have the same membership functions, while the output is normalized in the range of $[-0.01 : 0.01]$. The input and output membership functions are shown in Figure 3.5 and Figure 3.6, respectively, and the rules of the fuzzy controller are shown in Table 3.2.

$$\Delta F_{best}(t) = F_{best}(t) - F_{best}(t-1) \tag{3.4}$$

$$\Delta^2 F_{best}(t) = \Delta F_{best}(t) - \Delta F_{best}(t-1) \tag{3.5}$$

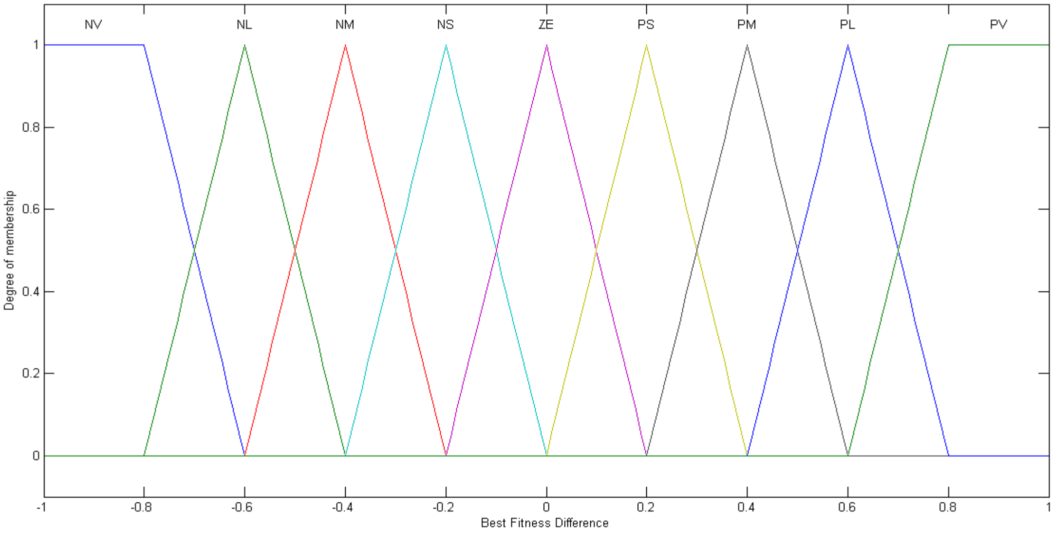

Figure 3.5 The second fuzzy controller input membership functions.

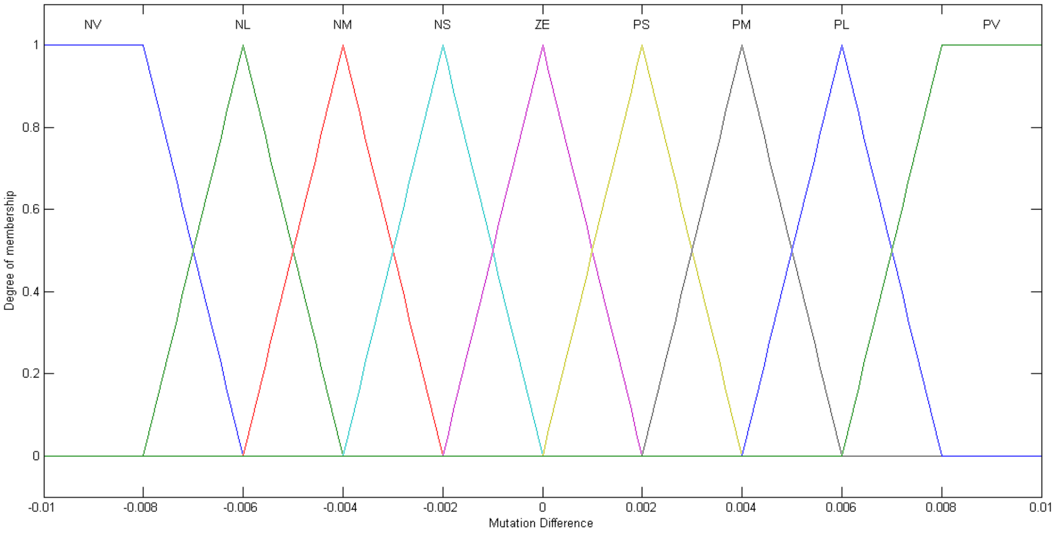

Figure 3.6 The second fuzzy controller output membership functions.

Table 3.2 Mutation Fuzzy Controller Rules

		Best Fitness Difference								
		NV	***NL***	***NM***	***NS***	***ZE***	***PS***	***PM***	***PL***	***PV***
Delta Best Fitness Difference	***NV***	NV	NL	NL	NM	NM	NS	NS	ZE	ZE
	NL	NL	NL	NM	NM	NS	NS	ZE	ZE	PS
	NM	NL	NM	NM	NS	NS	ZE	ZE	PS	PS
	NS	NM	NM	NS	NS	ZE	ZE	PS	PS	PM
	ZE	NM	NS	NS	ZE	ZE	PS	PS	PM	PM
	PS	NS	NS	ZE	ZE	PS	PS	PM	PM	PL
	PM	NS	ZE	ZE	PS	PS	PM	PM	PL	PL
	PL	ZE	ZE	PS	PS	PM	PM	PL	PL	PV
	PV	ZE	PS	PS	PM	PM	PL	PL	PV	PV

where NV stands for very large negative, NL stands for large negative, NM stands for medium negative, NS stands for small negative, ZE stands for zero, PS stands for small positive, PM stands for medium positive, PL stands for large positive, and PV stands for very large positive.

3.8 Conclusion

The proposed reconfiguration methodology is discussed in this chapter with a detailed description for each design step. Minimizing the total real power loss in the system is chosen to be the objective function to be optimized subject to the constraints ensuring that the voltage and the current limits are not exceeded and the radial topology of the system is always preserved. The genetic algorithm is designed such that the chromosome is a bit string representing the open switches in the system. Two fuzzy controllers are designed to enhance the performance of the genetic algorithm with respect to the convergence rate and the computational time. The first fuzzy controller is targeting the rejection of any infeasible configurations that might show up in the genetic algorithm population and violate the system radial topology, while the second one is targeting the adaptation of the mutation rate in such a way that minimizes the computational time, and maximizes the convergence rate. The forward and backward sweeping cutting algorithm is responsible for cutting the system into trees according the switches selected to be opened by the genetic algorithm. Then, a power flow is performed on these trees to calculate the total power loss in the system in order to be minimized.

In the next chapter, the proposed reconfiguration methodology is first applied to the 16-node test system with three different scenarios to demonstrate the effectiveness of the proposed reconfiguration methodology over the traditional methodologies. The three scenarios involve the utilization of the genetic algorithm alone, the genetic algorithm in addition to the first fuzzy controller, and the genetic algorithm with the two designed fuzzy controllers, respectively. The simulation results of each of the three scenarios are provided in the next chapter in details. After that, the proposed reconfiguration methodology is applied to solve the reconfiguration problem in another three large distribution power systems; the 33-node test system, the 69-node test system, and the IEEE 123-node test system. Thus, the proposed reconfiguration methodology proves its generalization capability to solve the optimization problem in large distribution systems with different topologies and conditions.

Chapter 4
Applications of the Proposed Reconfiguration Methodology

4.1 Introduction

As mentioned in the previous chapter, the 16-node test system is selected for two reasons. First, it is selected because of its small size which makes the discussion of every design step more clear and simple. Second, its small size facilitates the testing of the three different scenarios suggested to demonstrate the effectiveness of the proposed reconfiguration methodology. The first scenario involves the genetic algorithm alone for solving the optimization problem without interjecting any of the two designed fuzzy controllers, while the second scenario involves both the genetic algorithm and the first fuzzy controller for solving the optimization problem and rejecting the infeasible configurations at the same time. In both scenarios, the three different mutation functions discussed in chapter two are tested. The third scenario involves the genetic algorithm to solve the optimization problem, the first fuzzy controller to reject any infeasible configurations violating the radial topology of the system, and the second fuzzy controller to adapt the mutation rate of the genetic algorithm. The details of the three different scenarios and the simulation results for each one of them are discussed in the second section.

After that, the proposed reconfiguration methodology is applied to solve the reconfiguration problem in another three large distribution power systems; the 33-node test system, the 69-node test system, and the IEEE 123-node test system. The simulation results of each of those three systems are provided in the third, fourth, and fifth sections, respectively.

4.2 The 16-Node Test System

4.2.1 Genetic Algorithm Alone

In this scenario, three different tests are performed using the genetic algorithm alone to solve the reconfiguration problem for the 16-node test system. Every test uses one of the mutation functions discussed in Chapter Two; the Gaussian mutation function, the uniform mutation function, and the adaptive mutation function. The simulation results of the genetic algorithm in the three different tests are shown in Figure 4.1, Figure 4.2, and Figure 4.3, respectively.

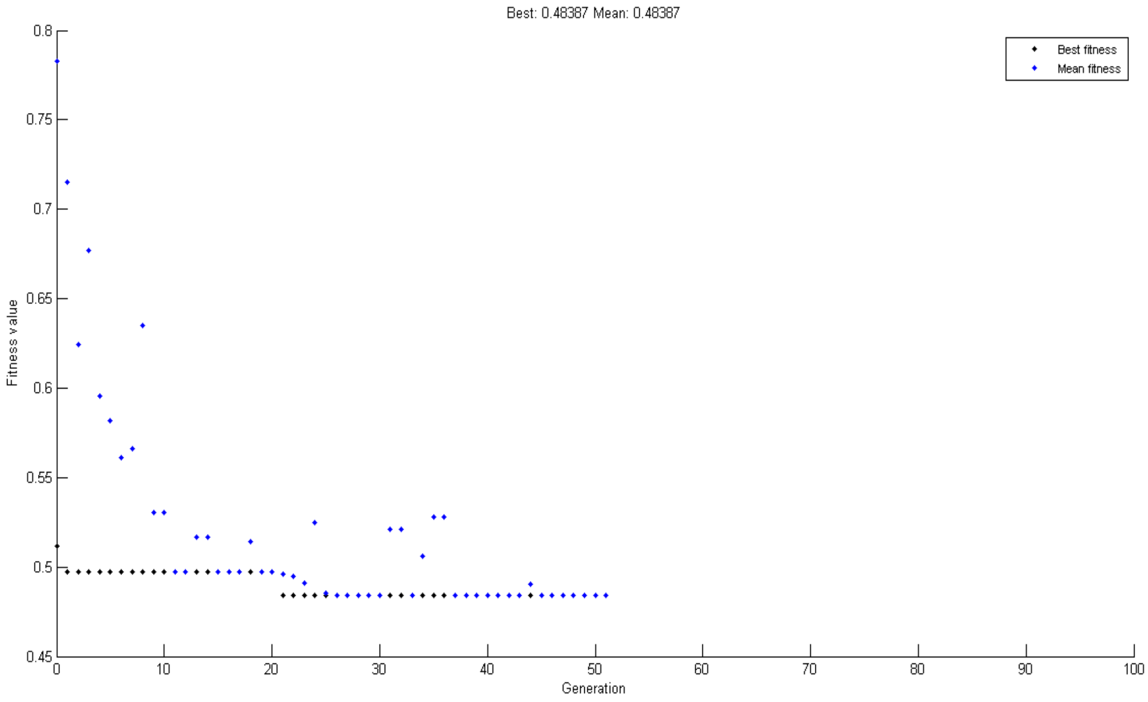

Figure 4.1 Genetic algorithm alone with the Gaussian mutation function.

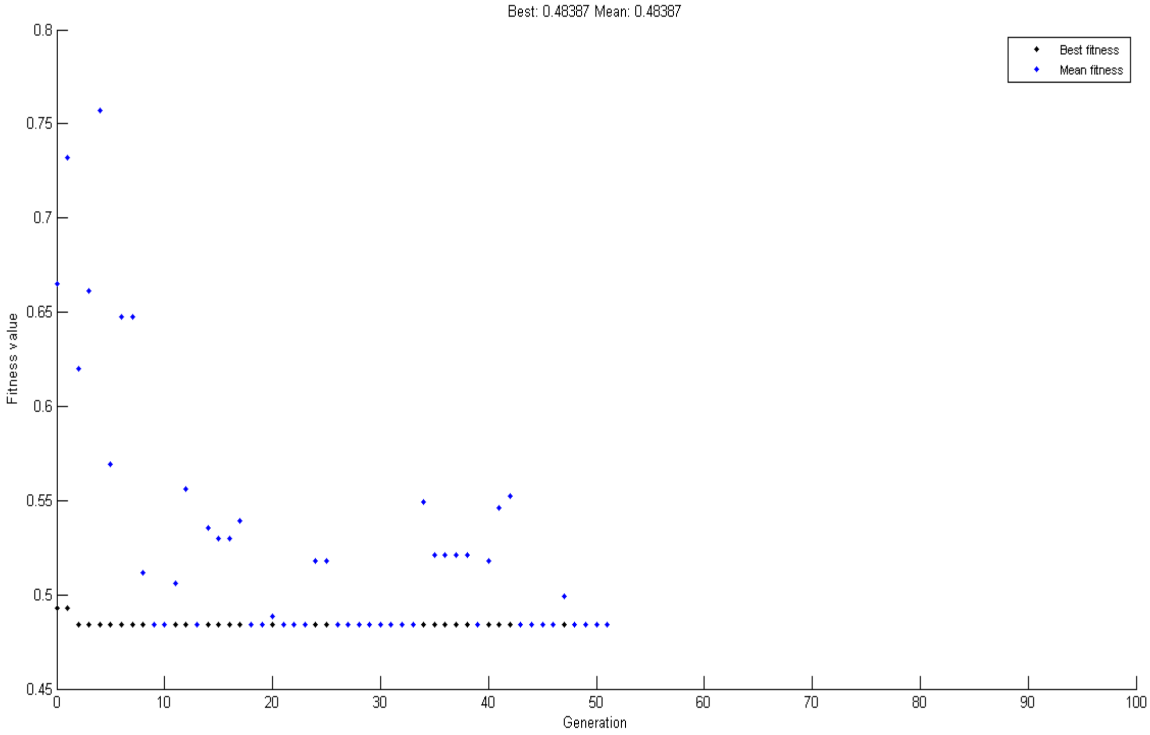

Figure 4.2 Genetic algorithm alone with the uniform mutation function.

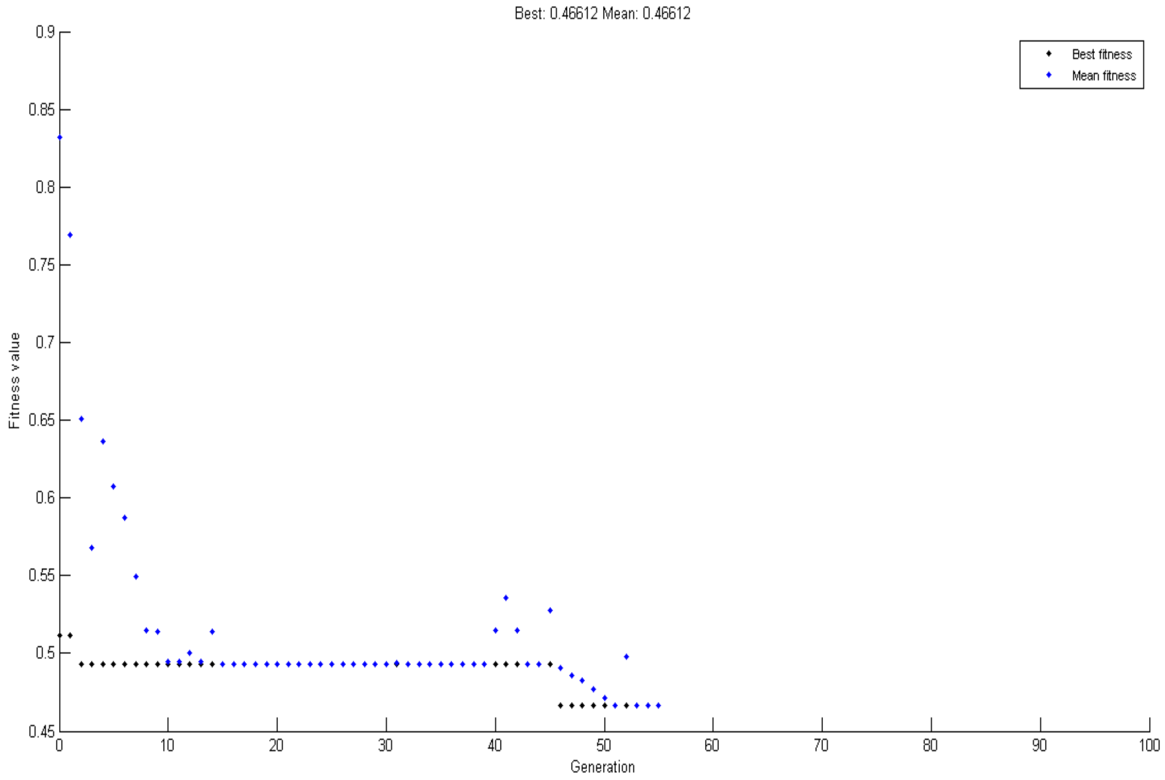

Figure 4.3 Genetic algorithm alone with the adaptive mutation function.

4.2.2 Genetic Algorithm and the First Fuzzy Controller

In this scenario, the same three tests mentioned in the first scenario are performed using the genetic algorithm along with the first fuzzy controller to reject any infeasible configurations violating the system radial topology. The simulation results of the genetic algorithm in the three different tests are shown in Figure 4.4, Figure 4.5, and Figure 4.6, respectively.

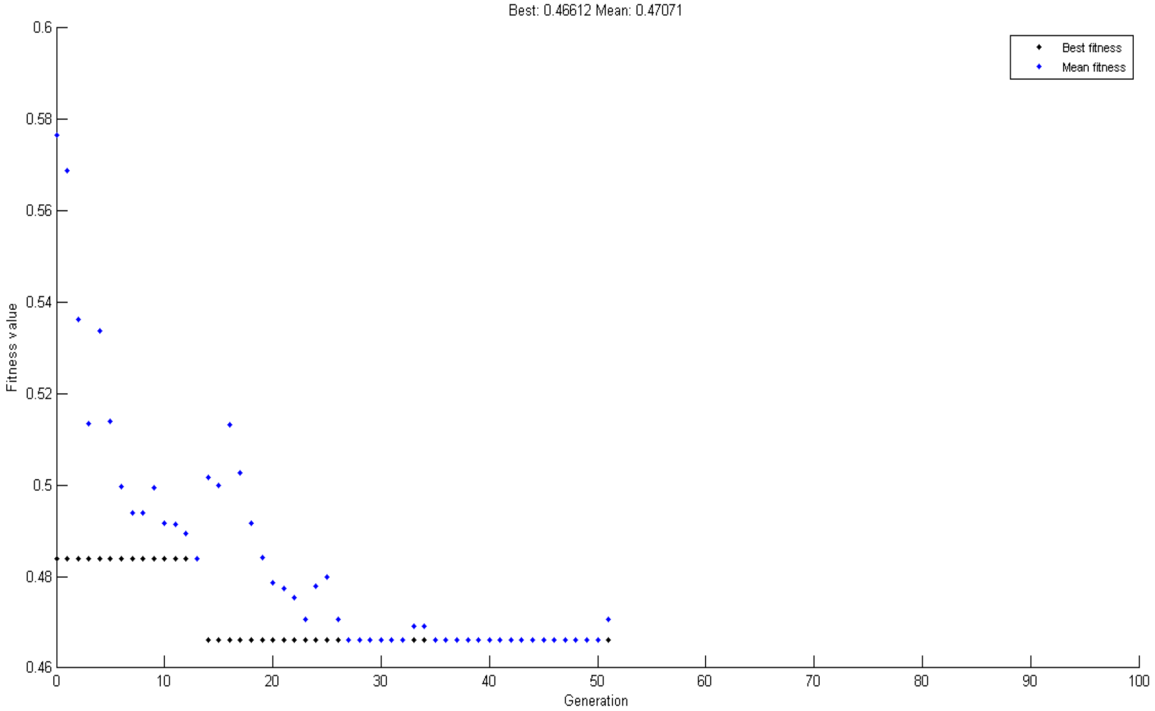

Figure 4.4 Genetic algorithm & first fuzzy controller with the Gaussian mutation function.

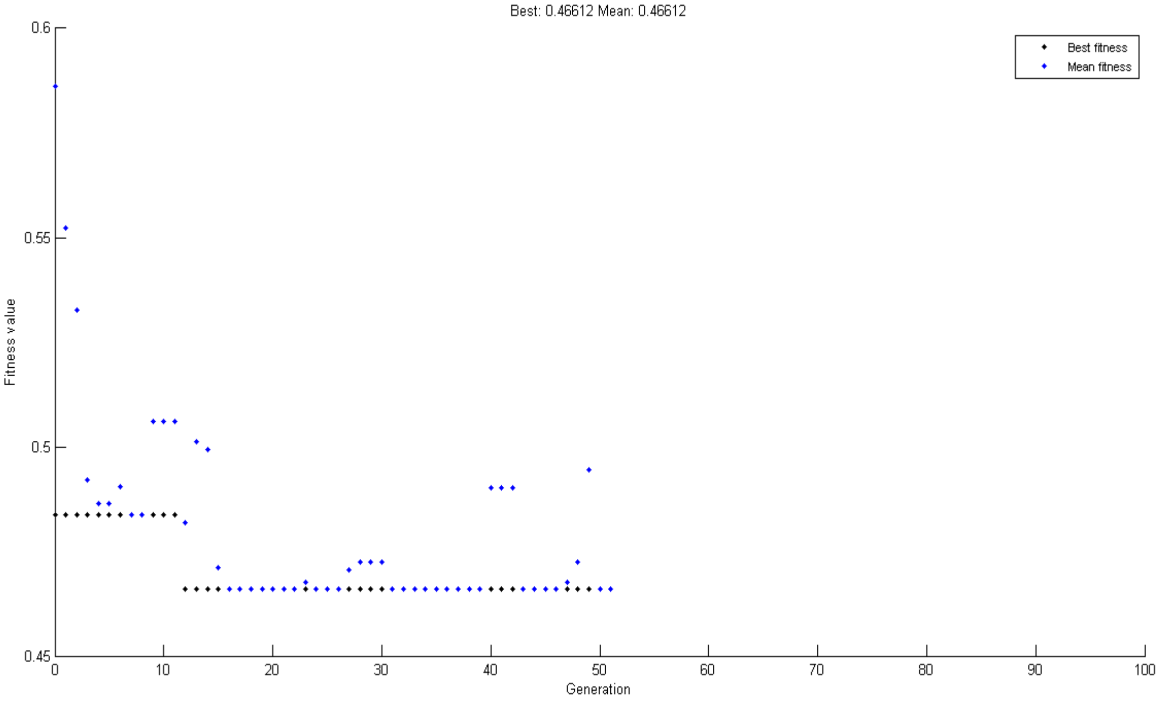

Figure 4.5 Genetic algorithm & first fuzzy controller with the uniform mutation function.

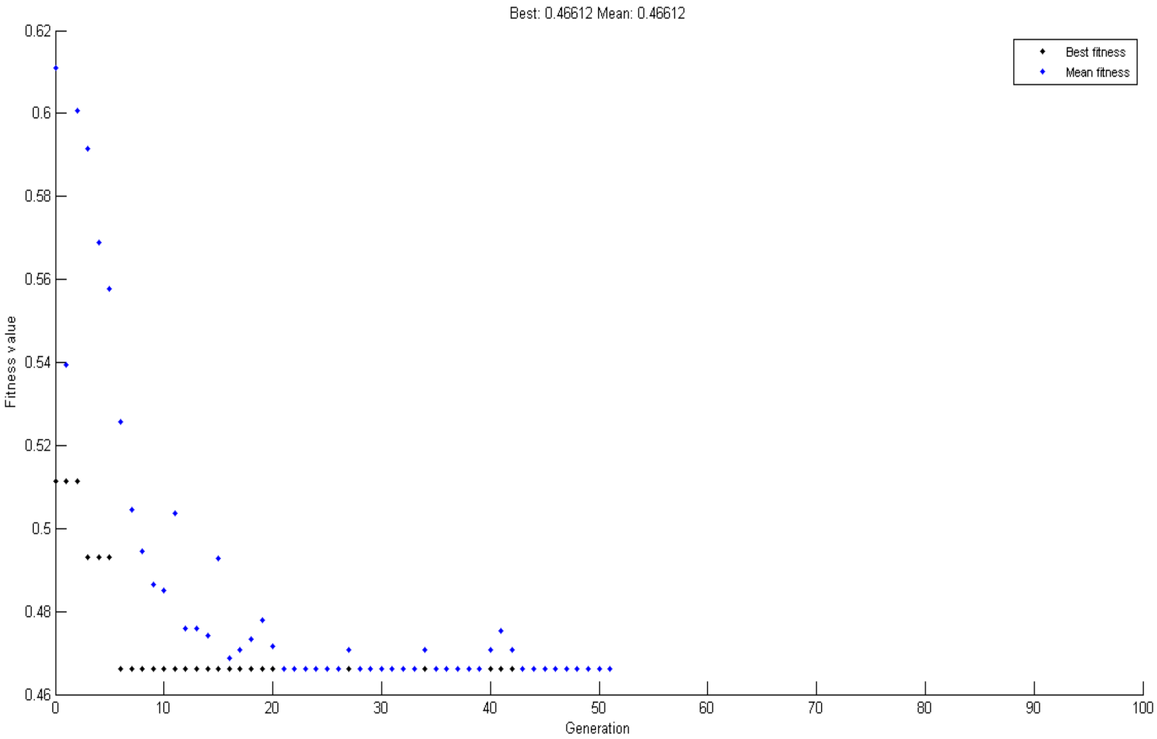

Figure 4.6 Genetic algorithm & first fuzzy controller with the adaptive mutation function.

4.2.3 Genetic Algorithm and the Two Fuzzy Controllers

In this scenario, only one test is performed using the genetic algorithm with the two designed fuzzy controllers to reject any infeasible configurations violating the system radial topology, and control the adaptive mutation rate as well. Only one test is performed because the second fuzzy controller plays the role of adapting the mutation rate of the genetic algorithm. Thus, it replaces the adaptive mutation function, and there is no need to use the Gaussian and the uniform mutation functions. The simulation result of this test is shown in Figure 4.7, and a comparison between the results of all the tests performed in the three scenarios is shown in Table 4.1.

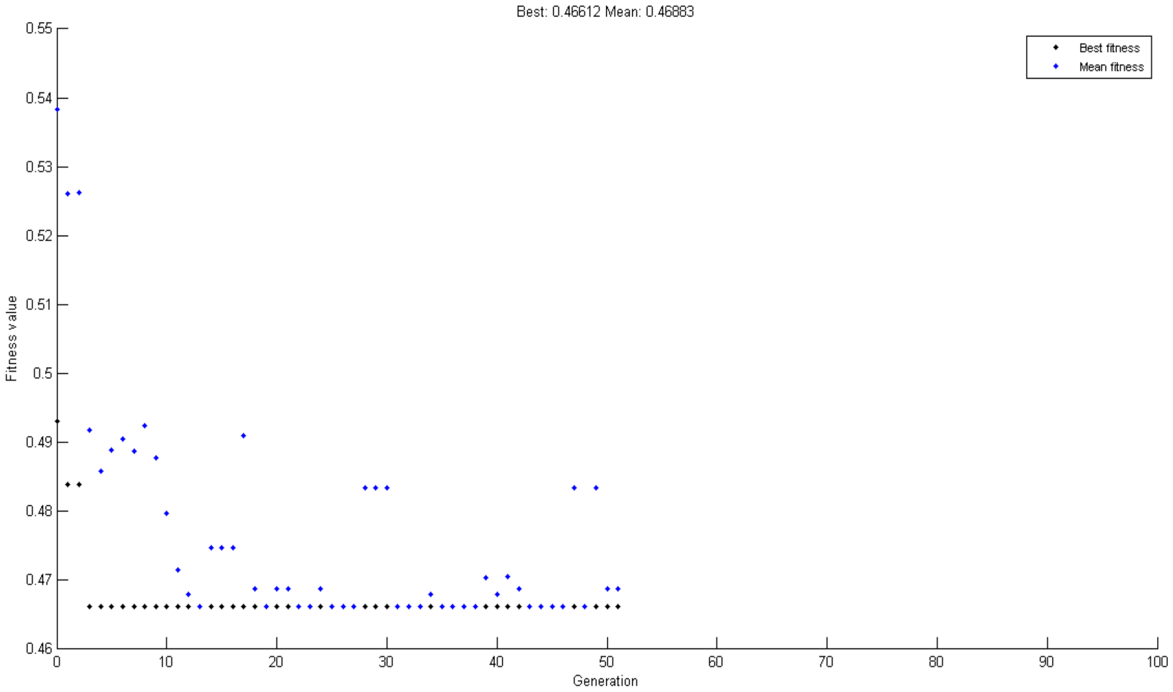

Figure 4.7 Genetic algorithm and the two fuzzy controllers.

Table 4.1 The Results Obtained in all Cases for the 16-Node System

			Open Switches	**Power Loss**
		Initial System	4, 11, 13	0.5114 MW
Scenario 1	***Genetic Algorithm Alone***	Gaussian Mutation	4, 9, 11	0.4838 MW
		Uniform Mutation	4, 9, 11	0.4838 MW
		Adaptive Mutation	6, 9, 11	0.4661 MW
Scenario 2	***Genetic Algorithm & First Fuzzy Controller***	Gaussian Mutation	6, 9, 11	0.4661 MW
		Uniform Mutation	6, 9, 11	0.4661 MW
		Adaptive Mutation	6, 9, 11	0.4661 MW
Scenario 3	***Genetic Algorithm & Two Fuzzy Controllers***	Adaptive Mutation	6, 9, 11	0.4661 MW

In the initial configuration of the system, the switches number 4, 11, and 13 are open and the total power loss is 0.5114 MW. From the results of the three scenarios shown in the figures and the table above, it could be concluded that for both of the two cases of using the genetic algorithm alone with the Gaussian mutation and the uniform mutation functions, the system doesn’t converge to the global optimal solution. The optimal configuration obtained in both cases chooses the switches number 4, 9, and 11 to be open with a total power loss of 0.4838 MW, which is less than that of the initial configuration of the system but doesn’t equal to the minimum power loss that could be obtained. The difference between those two cases as shown in the figures above is that the case with the uniform mutation function converges faster than the case with the Gaussian mutation function.

For all the other cases, the system converges to the global optimal configuration with the switches number 6, 9, and 11 are open, and the total power loss is 0.4661 MW. The difference between those cases is the rate of convergence of the genetic algorithm in each case. In the case of using the genetic algorithm alone with the adaptive mutation function, the genetic algorithm converges to the global optimal solution after 45 generations, while in the case of using the genetic algorithm with the infeasible configurations fuzzy rejector, the genetic algorithm converges after 13, 11, and 5 generations in the cases of the Gaussian, uniform, and adaptive mutation functions, respectively. In the final experiment of using the genetic algorithm with the two designed fuzzy controllers; the infeasible configurations fuzzy rejector and the adaptive mutation fuzzy controller, the best results are obtained since the genetic algorithm converges after only 2 generations.

From the discussion of the results above, it could be concluded that the results obtained when using the adaptive mutation functions are better than those of the uniform mutation function, and both of them are better than those of the Gaussian mutation function. It could also be concluded that the introduction of the infeasible configurations fuzzy rejector leads to better results and faster convergence rate since it doesn’t allow the genetic algorithm to search through infeasible configurations. Finally, the best results and the fastest convergence rate are obtained when the two designed fuzzy controllers are used with the genetic algorithm.

After the proposed reconfiguration methodology has been tested on the 16-node test system, it is tested again on three different systems in order to proof its performance accuracy, high efficiency, and compatibility with any distribution power system. Since the 16-node test system, used to demonstrate the effectiveness of the proposed methodology, is considered to be a small system, the new three systems are selected such that each one has a larger number of nodes and links than the preceding one. Those three systems are the 33-node test system, the 69-node test system, and the IEEE 123-node test system. The details of applying the proposed reconfiguration methodology to find the optimal configuration of each of the three mentioned systems in which the minimum power loss is achieved and the optimization constraints are maintained are shown in the coming sections.

4.3 The 33-Node Test System

The 33-node test system is shown in Figure 4.8 [21][28][51][77][79]. The system consists of 33 nodes and 37 links numbered in circles. The analysis of this system is carried out using the same procedure that has been used to analyze the 16-node test system. Every indicated link is assumed to have a sectionalizing switch, and the links shown in dotted lines are the open switches in the initial configuration of the system.

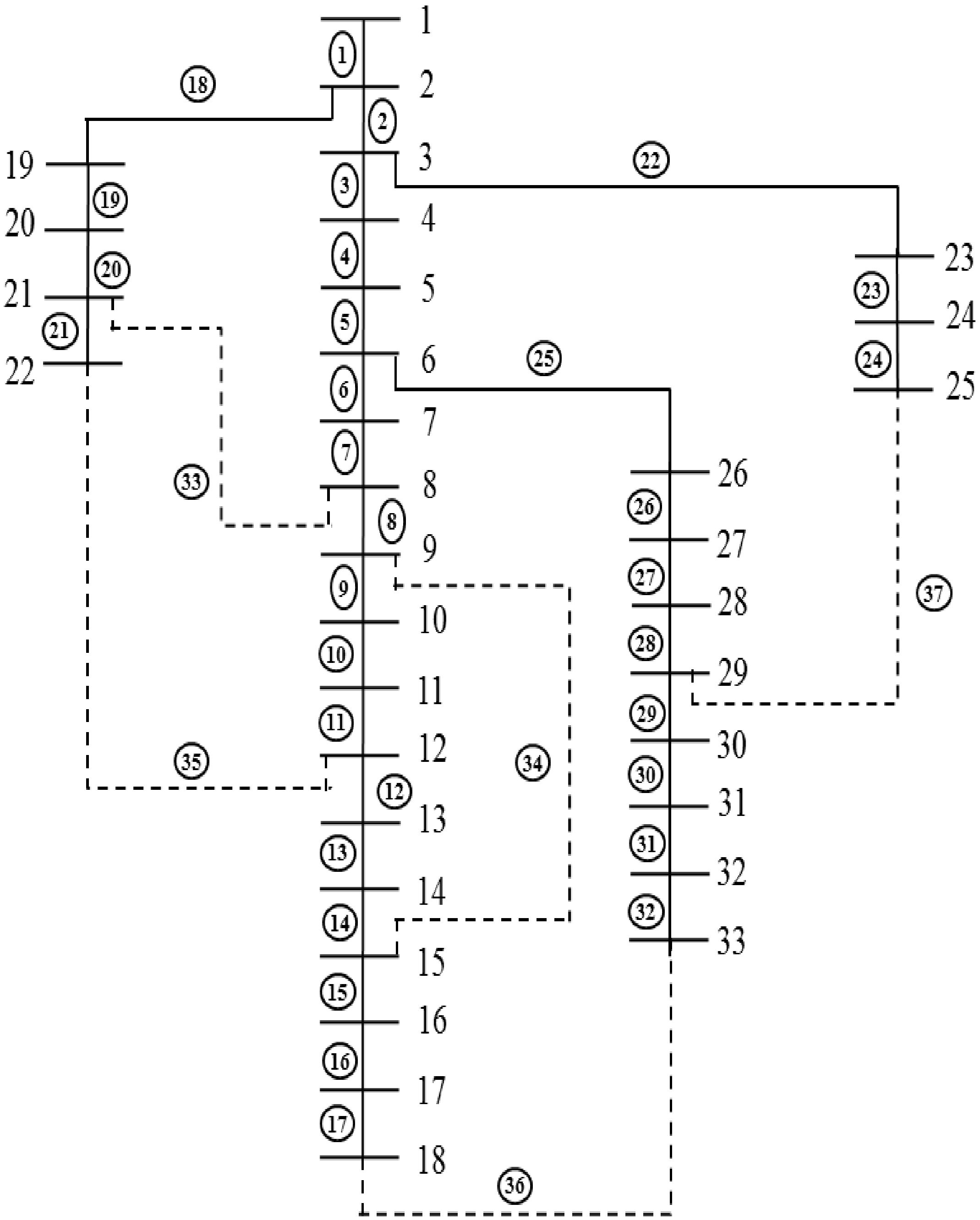

Figure 4.8 The 33-Node Test System.

The results of applying the genetic algorithm along with the two designed fuzzy controllers to obtain the optimal system configuration in which the minimum power loss is achieved are shown in Table 4.2.

Table 4.2 The Results of Applying the Proposed Methodology on the 33-Node System

	Open Switches	Power Loss
Initial System	33, 34, 35, 36, 37	0.2024 MW
The Proposed Reconfiguration Methodology ***Genetic Algorithm & Two Fuzzy Controllers***	7, 9, 14, 32, 37	0.13936 MW

As shown in the above table, the proposed reconfiguration methodology proves its capability of solving the optimization problem and reaching the global optimal configuration of the 33-node test system. The power lost is decreased from 0.2024 MW (in the case of the initial configuration of the system) to 0.13936 MW (in the case of the proposed methodology) with a 31% improvement in the power loss reduction.

4.4 The 69-Node Test System

The 69-node test system is shown in Figure 4.9 [78][80][81]. It has 69 nodes and 73 links whose numbers shown in circles in the figure below. The same procedure followed to analyze both of the 16-node and the 33-node test systems is followed again for the 69-node test system. Every link is assumed to have a switch, which can be opened or closed in order to minimize the total power losses in the system and maintain the system radial topology as well. The system has 5 open switches in its initial configuration indicated by the dotted lines.

The results obtained after applying the proposed reconfiguration methodology in order to obtain the optimal configuration of the 69-node system are shown in Table 4.3.

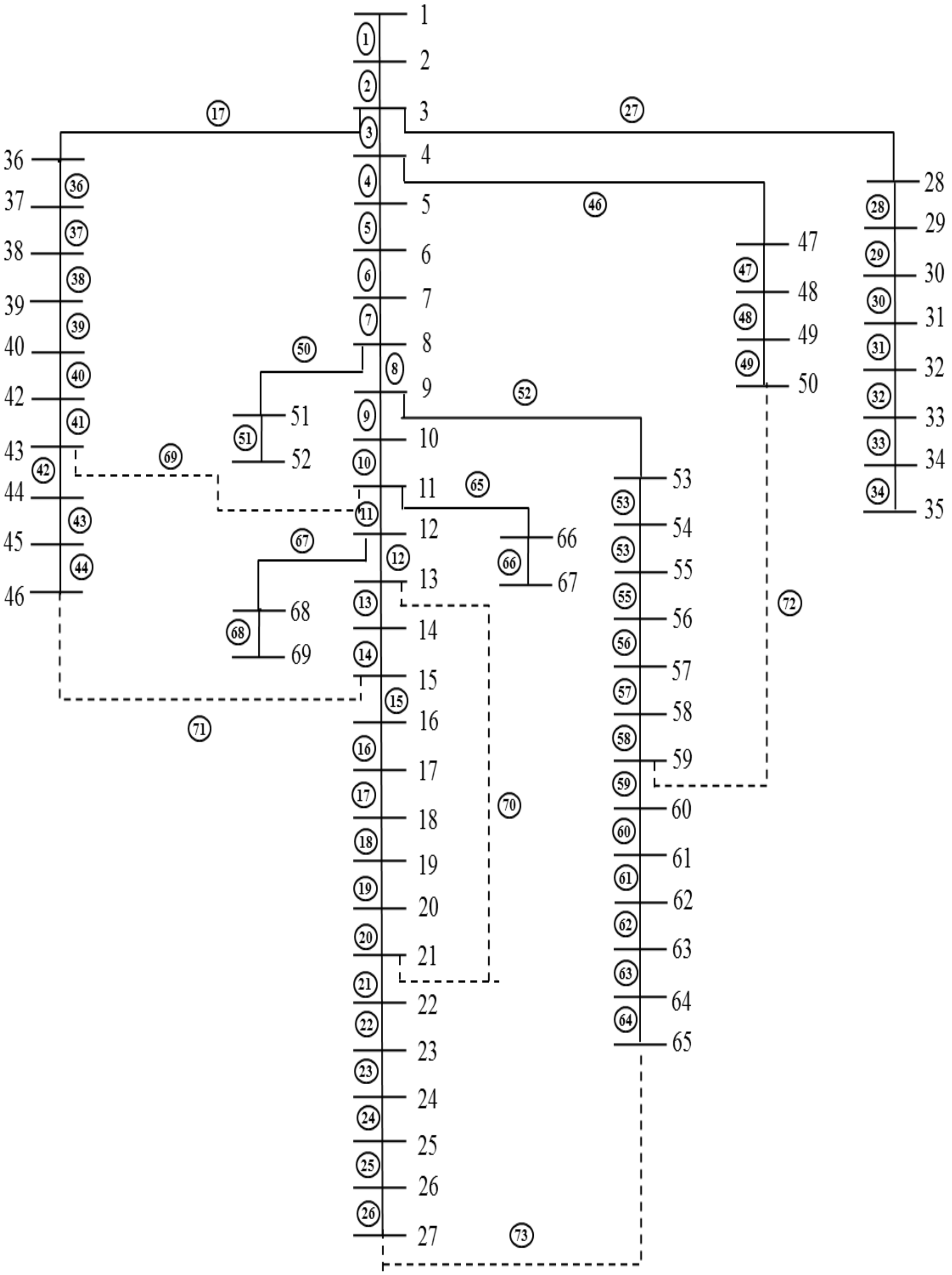

Figure 4.9 The 69-Node Test System.

Table 4.3 The Results of Applying the Proposed Methodology on the 69-Node System

	Open Switches	**Power Loss**
Initial System	69, 70, 71, 72, 73	0.2247 MW
The Proposed Reconfiguration Methodology Genetic Algorithm & Two Fuzzy Controllers	13, 20, 58, 63, 69	0.105526 MW

The proposed methodology succeeds in finding the global optimal configuration of the 69-node system, minimizing the total power losses in the system from 0.2247 MW to 0.105526 MW with a 53% improvement in the power loss reduction, and maintaining the radial topology of the system by refusing any infeasible configurations which is the role of the first designed fuzzy controller; the infeasible configurations fuzzy rejector.

4.5 The IEEE 123-Node Test System

The last step to emphasize on the capabilities and powerfulness of the proposed reconfiguration methodology is to test its performance on the IEEE 123-node test system. The topology of the system is shown in Figure 4.10 [82]. The numbers in circles in the figure below indicates the switches in the system. In the initial configuration of the system, the switches whose numbers lying between 1 and 6 are closed, while the switches carrying the numbers 7 to 12 are open.

For the sake of studying the operation of the system under the control of the proposed reconfiguration methodology, the system is assumed to be balanced using the method discussed in [83]. This assumption is taken into consideration in order to simplify the calculations and ensure that the system is operating in an efficient and effective manner.

The results obtained after applying the proposed reconfiguration methodology to the IEEE 123-node test system are shown in Table 4.4 and the genetic algorithm performance during the different generations is shown in Figure 4.11.

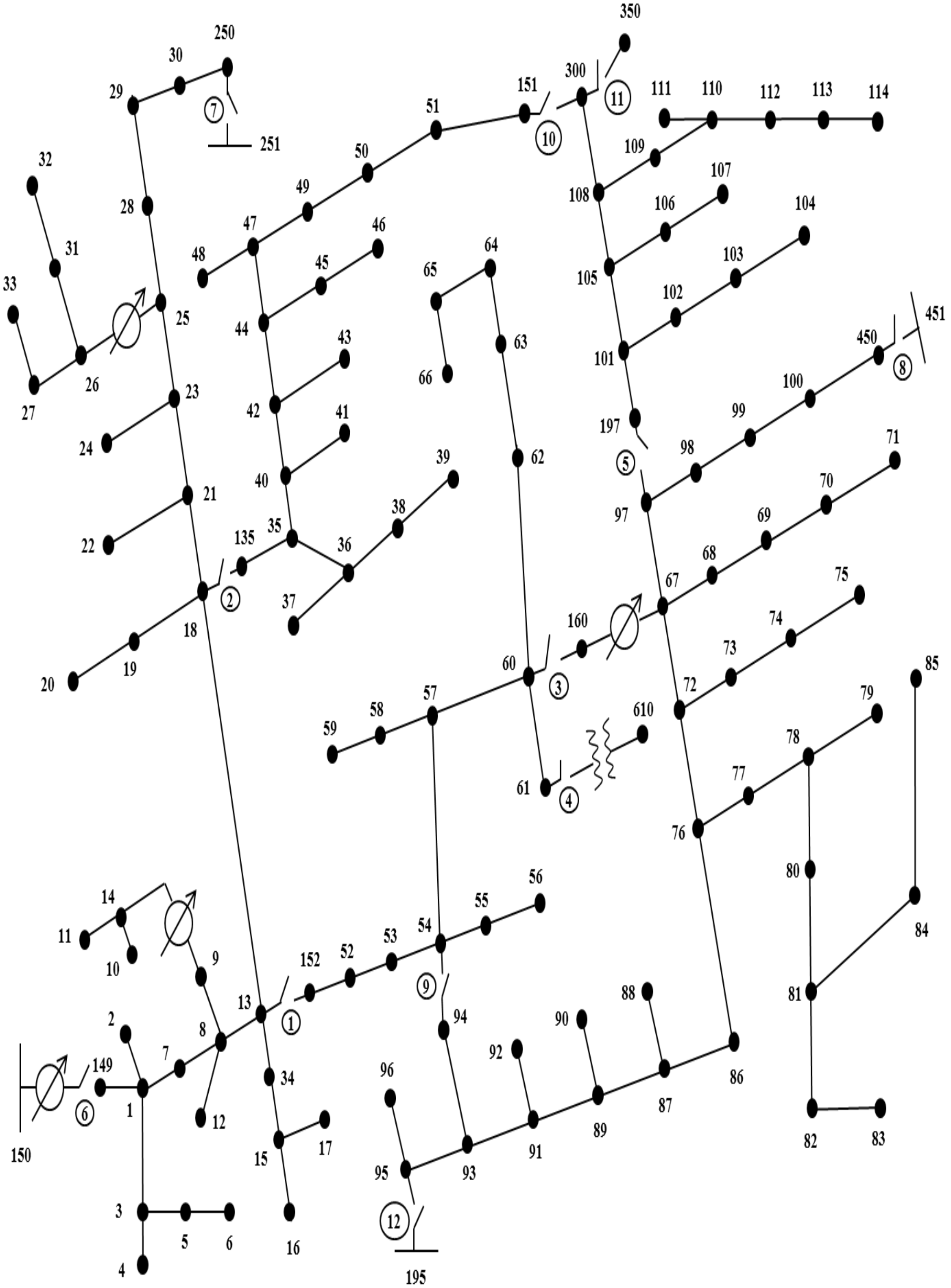

Figure 4.10 The IEEE 123-Node Test System.

Table 4.4 The Results of Applying the Proposed Methodology on the IEEE 123-Node system

	Open Switches	Power Loss
Initial System	7, 8, 9, 10, 11, 12	0.1065 MW
The Proposed Reconfiguration Methodology Genetic Algorithm & Two Fuzzy Controllers	3, 5, 7, 8, 11, 12	0.097056 MW

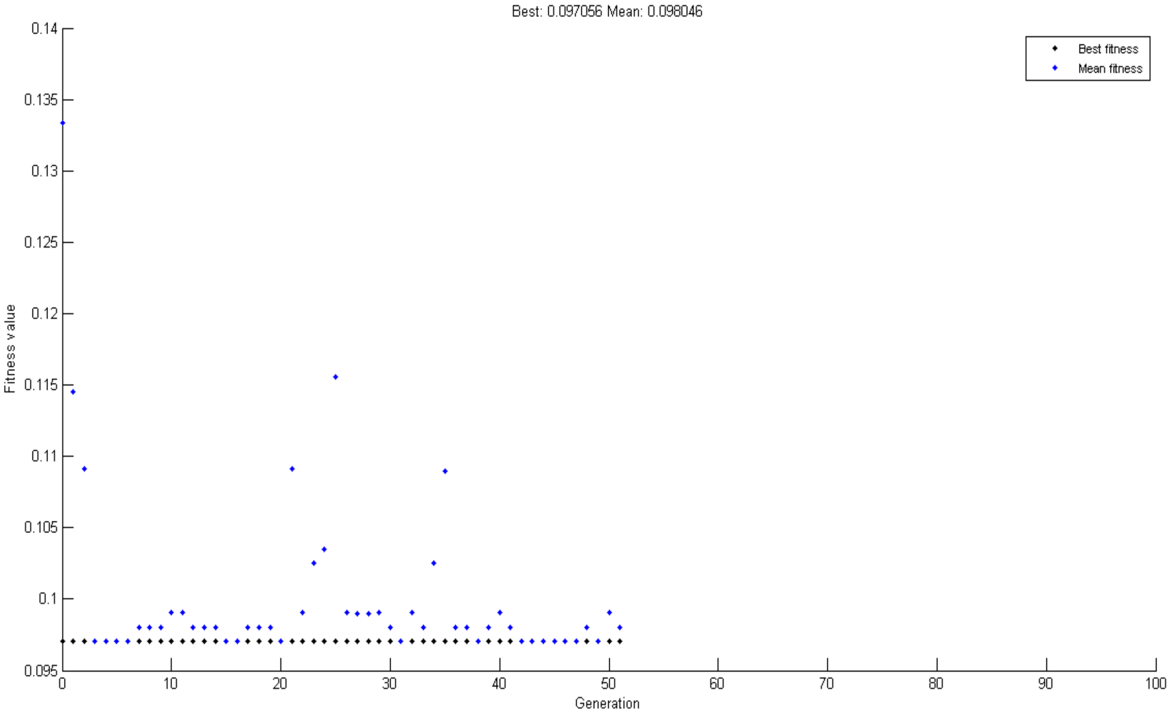

Figure 4.11 The genetic algorithm performance for the IEEE 123-node test system.

As shown in the above table and figure, the proposed reconfiguration methodology succeeds in minimizing the total power losses in the system from 0.1065 MW to 0.097056 MW with a 9% improvement in the power loss reduction, and the genetic algorithm converges to the global optimal solution from the first generation. Thus, the high efficiency and the fast convergence rate of the proposed algorithm are verified.

4.6 Conclusions

The proposed reconfiguration methodology has been applied to four different distribution power systems; the 16-node test system, the 33-node test system, the 69-node test system, and the IEEE 123-node test system. For the 16-node test system, three different scenarios have been performed to show the effect of each part of the proposed reconfiguration methodology on the performance of the system. The three scenarios involve the utilization of the genetic algorithm; once alone, once with the infeasible configurations fuzzy rejector only, and once with both the infeasible configurations fuzzy rejector and the adaptive mutation fuzzy controller. The last scenario has proved to have the best performance since it was able to reach the global optimal solution with the fastest convergence rate consuming the least computational time. Thus, the third scenario has demonstrated the powerful capabilities of the proposed reconfiguration methodologies.

To proof the high efficiency of the proposed reconfiguration methodology and its compatibility with any distribution power system other than the 16-node system, it has been applied to the three other test systems. For all systems, the proposed reconfiguration methodology has reached the global optimal configuration that minimizes the total power loss subject to the desired constraints. A summary for the minimum power loss realized in each of the four systems along with the percentage of the power loss saving in each case is shown in Table 4.5.

Table 4.5 A Summary for the Minimum Power Loss Realized in the Four Systems

	Initial Power Loss (MW)	**Minimum Power Loss (MW)**	**Percentage of Power Loss Saving**
The 16-Node System	0.5114	0.4661	8.858 %
The 33-Node System	0.2024	0.13936	31.146 %
The 69-Node System	0.2247	0.105526	53.0369 %
The IEEE 123-Node System	0.1065	0.097056	8.867 %

Chapter 5
The Novel Multi-agent System Design

5.1 Introduction

As discussed in the previous chapters, the proposed reconfiguration methodology has been designed using a hybrid fuzzy genetic approach in which the genetic algorithm has been used to reach the optimal system configuration having the minimum system power loss, and the fuzzy logic has been used to build two different fuzzy controllers; the function of the first one is to reject any infeasible configurations that violates the radial topology of the system, while the second one is used to adapt the mutation rate used in the genetic algorithm in order to achieve the fastest convergence rate. This proposed methodology has been tested on four different test systems; the 16-node test system, the 33-node test system, the 69-node test system, and the IEEE 123-node test system. As presented in the previous chapter, the proposed methodology has proved its high capability of minimizing the power losses in each of the four test systems by reaching the global optimal configuration with the fastest convergence rate consuming the least computational time.

In order to employ this reconfiguration methodology in the smart distribution power systems, decentralized control is needed instead of the centralized control. Thus, the multi-agent approach manifests having the required capabilities of performing the job in a decentralized fashion in which several intelligent agents are employed in a multi-agent system to share information, do a part of the job, and make their own decision without any need to a centralized control. As discussed in Chapter Two, a multi-agent system consists of a group of intelligent agents having the capabilities of autonomy, reactivity, pro-activity, and sociality. Many approaches have been suggested to implement the multi-agent systems including the emergence of the agent communication languages and the agent platforms and toolkits discussed in Appendix A and Appendix B, respectively. These approaches are very efficient but have the drawbacks of design complexity and design limitation. To overcome these drawbacks, a novel approach is introduced in this thesis to design and build the multi-agent system as discussed in the coming sections.

5.2 Contributions of the Novel Multi-agent Approach

A novel multi-agent approach is proposed in this thesis in which the distributed computing and object-oriented programming concepts are used to design and build the multi-agent system required to perform the proposed reconfiguration methodology in the smart distribution power systems. The main contributions of that novel approach are discussed as follows:

1. Taking the advantage of the design simplicity of the distributed computing and object-oriented programming as well as achieving the main functionalities of the intelligent agents in any multi-agent system.
2. Making a new miniature version of the multi-agent paradigm by mapping the distributed computing and object-oriented programming paradigm to the multi-agent paradigm.
3. Acquiring the ability of communicating up to deep level of details with different disciplines since the object-oriented programming has the powerful ability of abstracting what is happening in the real world in the form of objects; each has its own properties and its own behavior.
4. Facilitating the implementation varieties by documenting the design methodology in an appropriate way that could be understood and converted to a real implementation.
5. Designing the suitable framework for each of the sequential processing and parallel processing applications. This means that the designed multi-agent system could offer the framework needed to implement any power systems application, not only the reconfiguration application.

5.3 The Novel Multi-agent System Design

As mentioned above, the object-oriented programming has the advantage of being able to abstract the design components in the form of objects. Each object is an independent entity whose unique properties and methods. The object properties are used to describe the object states or characteristics, while the object methods are used to describe the object behavior and the actions that can be performed by that object. Any object-oriented design can be described using two diagrams; the class diagram and the state diagram. The first one is representing each class in the form of its properties and methods, while the second one is describing the state flow of the system designed. A detailed description and discussion of both diagrams of the designed multi-agent system is provided below.

5.3.1 The Class Diagram

The class diagram of the designed multi-agent system is shown in Figure 5.1. The first row of blocks contains the class name, the second row lists the class properties, and the third one is describing the class methods. A detailed discussion of each class, its properties and methods is given as follows:

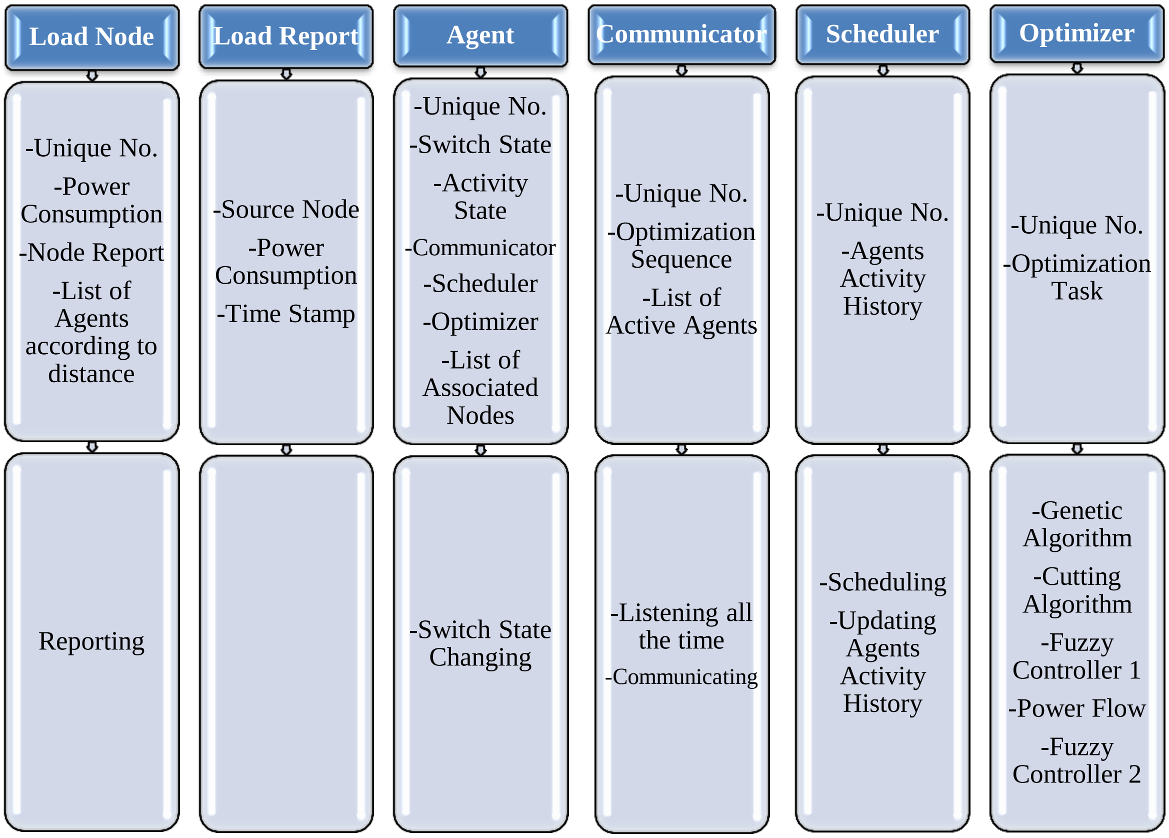

Figure 5.1 The class diagram of the designed multi-agent system.

1. Load Node Class

This class represents the load nodes in the distribution power system. Each object or instance of this class represents a unique load node in the system with the following properties and methods:

A. Properties

a. Unique Number: This represents a unique identifier to each load node in the distribution power system.

b. Power Consumption: This represents the power consumed by this particular node at a certain time instant.

c. Node Report: This is the report generated by each load node having its unique number, its power consumption, and the time stamp at which the power consumption is calculated. This property is an object of the second class which means that each object of the first class “Load Node” has an object of the second class “Load Report” as one of its properties.

d. List of Agents according to distance: This is a list of the agents in the system sorted according to their closeness to that load node.

B. Methods

Reporting: Every load node in the system reports a load report describing the power consumption state of that node at a particular time. Thus, every object of the load node class “a load node” generates an object of the load report class “a load report”, keeps that object as one of its properties, and sends that report to the nearest active agent. After sending the report, the load node waits for an acknowledgement from the agent. In case the acknowledgement has not been received within a certain time, the load node has to send its report to the next nearest active agent until receiving an acknowledgement.

2. Load Report Class

This class represents the load reports generated by the load nodes in the distribution power system. Each object of this class represents a unique load report belonging to a certain load node in the system. The class properties and methods are discussed as follows:

A. Properties

a. Source Node: This is the number of the source node generating the load report.

b. Power Consumption: The power consumption of the source node at a particular time instant.

c. Time Stamp: The time at which the power consumption of the source node is calculated.

B. Methods

This class has no methods since it is just representing a report having certain properties as discussed above, but has no actions to do.

3. Agent Class

This class represents the intelligent agents installed in the distribution power system to carry out the operations required for a certain application. Each object of this class represents a unique agent in the designed multi-agent system. The agent is the only piece of hardware to be installed in the system and it might be a computer or a micro-controller. The class properties and methods are discussed as follows:

A. Properties

a. Unique Number: This is the unique number carried by each agent in the system and it acts as the identifier that identifies every agent from the other agents.

b. Switch State: Every agent is installed at a certain switch in the system. Thus, each agent needs to keep track of its switch state; either closed or opened.

c. Activity State: This property indicates whether the agent is active "working" or inactive "not working". Active agents should be taken into account while performing the reconfiguration application, while inactive agents should be excluded until they become active again.

d. Communicator: This is an object of the communicator class and it is responsible for all the message handlings and communications from and to the agent.

e. Scheduler: It is an object of the scheduler class and it is in charge of the scheduling process in which the work sequence is determined.

f. Optimizer: It is an object of the optimizer class and it is responsible for the all the steps of the optimization application.

g. List of Associated Nodes: It a list indicating the nodes that are supposed to send their reports to the agent.

B. Methods

Switch State Changing: This function is in charge of changing the switch state according to the output of the optimization process.

4. Communicator Class

This class represents the communicator existing in each agent and responsible for the message handling and communications of each agent with the other agents in the system. Each object of this class acts as an employee which is employed by the agent to take care of sending and receiving messages. It has the following properties and methods:

A. Properties

a. Unique Number: This acts as a unique identifier carrying the same identifier of the employing agent.

b. Optimization Sequence: This is the sequence of the optimization process determining the part of the job to be performed by each agent. This sequence is determined by the scheduler and sent to the communicator in order to be able to organize the work among the different agents in the system.

c. List of Active Agents: It is a list having the active agents that could participate in the optimization process at a particular time. Agents not existing in this list are not active at that time and have to be excluded from taking a part of the job until they recover.

B. Methods

a. Listening all the time: One of the most important functions of the communicator is to keep listening all the time such that as soon as it receives any new data, it takes the appropriate action according to the received data.

b. Communicating: Handling all messages sent from and to the agent.

5. Scheduler Class

This class represents the scheduler existing in each agent and responsible for the scheduling process. Each object of this class acts as an employee that is employed by the agent to take care of determining when the agent should participate in the optimization process and when it shouldn’t. It has the following properties and methods:

A. Properties

a. Unique Number: This acts as a unique identifier carrying the same identifier of the employing agent.

b. Agent Activity History: The scheduler has to keep track of the agent activity history which tracks the number of times the agent has performed each part of the job. The purpose of this property is to allow the scheduler to determine if the agent has to participate in the optimization process in the next time or not and which part of the job it should take. This acts as a guarantee that all agents have equal workloads such that no agent is working more than the other agents and no agent is performing the same task every time.

B. Methods

a. Scheduling: The main function of the scheduler is to schedule the optimization sequence and determine if the agent has to take a part of the job or not and which part to take. The scheduling strategy used in the designed system ensures that all agents have equal workloads in a such a way that prevents any agent from being overloaded by working more than the other agents in the system, and at the same time allows the agent to perform different tasks instead of performing a fixed task every time.

b. Updating Agents Activity History: This method targets updating the agent activity history in case it is selected to participate in the optimization process.

6. Optimizer Class

This class represents the optimizer existing in each agent and in charge of performing the different tasks of the optimization algorithm aiming to solve the reconfiguration problem. Each object of this class acts as an employee that is employed by the agent to perform a certain task as a part of the different optimization tasks discussed in the proposed reconfiguration methodology. As discussed in Chapter Three, the proposed reconfiguration methodology divided the optimization process into five main tasks; the genetic algorithm, the forward & backward sweeping cutting algorithm, the first fuzzy controller, the power flow, and the second fuzzy controller. These five methods are included inside the optimizer class in addition to its properties as discussed below.

A. Properties

a. Unique Number: This acts as a unique identifier carrying the same identifier of the employing agent.

b. Optimization Task: This is the number of the optimization task being performed by the agent at the current time.

B. Methods

a. Genetic Algorithm: The main optimizer that searches for the optimal configuration in order to minimize the total power loss in the system.

b. Cutting Algorithm: The forward & backward sweeping cutting algorithm targets the formation of the tree of nodes starting from the root node or the feeder node and ending with the leaf nodes taking into account the switches to be opened.

c. First Fuzzy Controller: The infeasible configurations fuzzy rejector targets the rejection of any infeasible configurations showing up in the population of the genetic algorithm and validating the radial topology of the system, and converts them to feasible configurations.

d. Power Flow: The function performing the power flow in the trees resulting from the cutting algorithm after rejecting the infeasible configurations.

e. Second Fuzzy Controller: The adaptive mutation fuzzy controller adapts the mutation rate of the genetic algorithm in order to reach the global optimal solution with the fastest convergence rate consuming the least computational time.

5.3.2 The State Diagram

The state diagram of the designed multi-agent system is shown in Figure 5.2. It is describing in more details how the different objects of the different classes in the system interact with each other, and how they communicate and socialize to share information, distribute the tasks among themselves, and finally make their own decision in order to achieve the ultimate objective. A more detailed description of the state diagram is provided in the following steps:

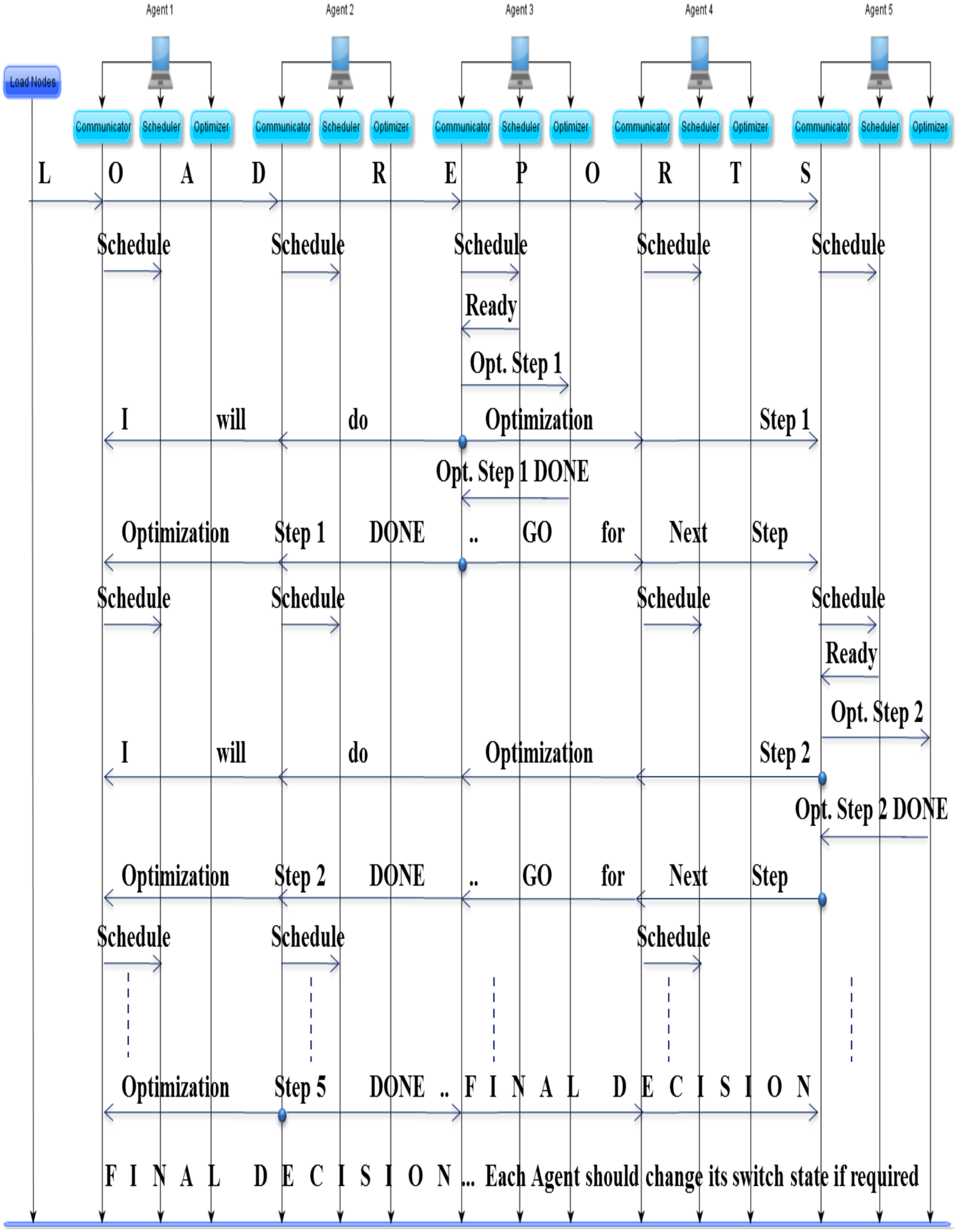

Figure 5.2 The state diagram of the designed multi-agent system.

1. The load nodes send their load reports to their nearest active agent. As discussed in the class diagram, one of the main functions of the communicators of the agents is to keep listening all the time to any incoming data. Thus, as soon as the communicators of the active agents receive the load reports and make sure all the load reports have been received, they have to take an action.

2. Every communicator sends a message to the scheduler of the same agent asking it to schedule the optimization sequence. In other words, it asks the schedule to determine if the agent is ready to take a part of the job or not.

3. The scheduler of the first ready agent sends a message to its communicator informing it that the agent is ready to perform the first part of the job. In the state diagram shown above, agent 3 is assumed to be the first ready agent. Thus, its scheduler sends a message to its communicator saying *"I am ready"*.

4. The communicator of agent 3 sends a message to the optimizer of the same agent asking it to start doing the first optimization task, and at the same time it sends a message to all other communicators of the other active agents informing them that agent 3 is doing the first task so that no other agent does the same task.

5. After the optimizer of agent 3 finishes doing the first optimization task, it sends a message to the communicator of the same agent informing it that the first task has been completed.

6. The communicator of agent 3 sends a message to all other communicators of the other active agents informing them that the first optimization task has been completed, and asking them to proceed to the second task.

7. Every communicator of the active agents except that of agent 3 sends a message to the scheduler of the same agent asking it to determine if the agent is ready to take the next part of the job or not.

8. The scheduler of agent 5 sends a message to the communicator of the same agent informing it that agent 5 is ready to take the next part of the job.

9. The communicator of agent 5 sends a message to the optimizer of the same agent asking it to start doing the next optimization task, and at the same time sends a message to the other communicators of the other active agents to let them know that agent 5 is doing that task.

10. After the optimizer of agent 5 finishes doing the second optimization task, it sends a message to the communicator of the same agent informing it that the second task has been completed.

11. The communicator of agent 5 sends a message to all other communicators of the other active agents informing them that the second optimization task has been completed, and asking them to proceed to the third task.

12. Every communicator of the active agents except those of agents 3 and 5 sends a message to the scheduler of the same agent asking it to determine if the agent is ready to take the next part of the job or not.

13. The same procedure is followed until the five optimization tasks discussed in the optimizer class are completed. In the state diagram shown above, agent 2 is assumed to be the last agent performing the last optimization task. Thus, the communicator of agent 2 sends a message to all other communicators of the other agents informing them that the last optimization task has been done and asking them to make the final decision.

14. The agents participating in the optimization process share their information and make their own final decision determining which switches have to be opened and which have to be closed.

15. The final decision is broadcasted to all agents, and each agent changes its switch state either by opening or closing, if required.

5.4 Conclusions

From the discussion of the class and state diagrams of the proposed multi-agent design, the main design contributions are clarified. The proposed design combines both of the design simplicity and the high functionality at the same time. The four main characteristics of the intelligent agents employed in a multi-agent system are all achieved in the proposed design as discussed below:

a. The designed multi-agent system can operate automatically and all the agents are automated in such a way that allows them to take automatic actions as soon as they receive any data from the surrounding environment without the need to any external orders or instructions.

b. All agents have the ability to take a reaction in response to a certain action or a certain piece of information retrieved from the surroundings.

c. The intelligent agents in the designed system have the capability of pro-activity such that they are capable of taking the initiative of performing a certain action or changing their behavior when they are ready to do that and when the surrounding circumstances require.

d. The sociality property is highly achieved in the designed system in which all the agents could communicate with each other, share the information retrieved from the surroundings, and make their own decision that could realize their ultimate objective.

In the next chapter, the designed multi-agent system is built for the IEEE 123-node test system and used to perform the proposed reconfiguration methodology to reach the optimal configuration of the system that minimizes the total power loss. The reconfiguration application is performed in a distributed and decentralized fashion such that each agent in the system perform a certain task of the optimization process, share its information with the other agents employed in the system, and collaborate with these agents to make their own final decision in order to realize the ultimate objective they are designed for.

Chapter 6

Applications of the Novel Multi-agent System

6.1 Introduction

In this chapter, the designed multi-agent system has been built and applied to the IEEE 123-node distribution test system as shown in Figure 6.1. The system has 8 agents installed at the 8 switches present in the system as indicated in the figure below. The initial and optimal configurations for the system in the case of the maximum loading of all nodes are shown in Table 6.1. In the initial system, a total power of 106.5 kW is lost when the switches 6, 7, and 8 are open, while the optimal configuration reached by the proposed reconfiguration methodology minimizes the total power loss to 97.056 kW by opening the switches 3, 5, and 8. Thus, 8.867% of the total power loss due to the initial configuration of the system is saved.

Table 6.1 The Initial and Optimal Configurations of the IEEE 123-Node System

	Open Switches	**Power Loss**	**Percentage of Power Loss Saving**
Initial System	6, 7, 8	106.5 kW	
The Proposed Reconfiguration Methodology ***Genetic Algorithm & Two Fuzzy Controllers***	3, 5, 8	97.056 kW	8.867 %

Eight different scenarios are employed to study the performance of the designed multi-agent system under different conditions. The different scenarios are employing different loading conditions, switching conditions, and distributed generations' conditions. First, no distributed generations are installed in the system, and the total number of switching allowed per day is varied while varying the loading conditions. Second, distributed generations are installed in different locations in the system, and the total number of switching allowed per day is varied again while the loading conditions are varied between light loading and heavy loading conditions. Each scenario and its obtained simulation results are discussed in details in the coming sections.

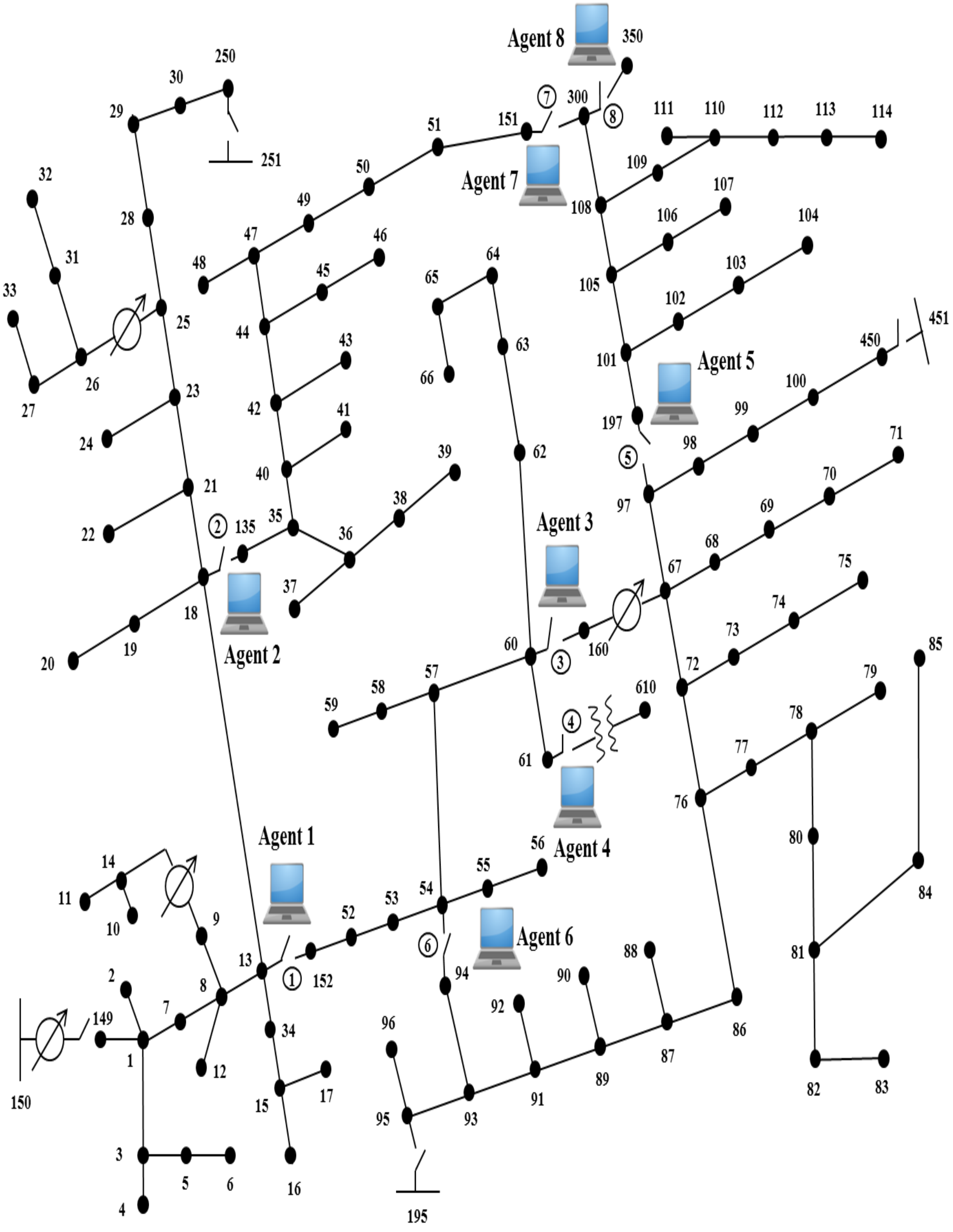

Figure 6.1 The designed multi-agent system applied to the IEEE 123-node test system.

6.2 Scenario 1: Light Loading and Switching Every Hour

In this scenario, every load in the system is allowed to take a random loading value between 0 % and 100 % of its maximum loading value. Thus, no overloading is allowed and all the loads can't exceed their maximum permissible loading values. The proposed reconfiguration methodology is assumed to be performed every hour searching for the optimal system configuration that minimizes the total power loss in the system, and the switches that have to be opened or closed are determined based on that optimal configuration. Thus, the switching is allowed to be changed every hour.

In Table 6.2, the simulation results for consecutive 24 hours are shown. For every hour, the number of the agent that performed every optimization task is given and the switches that were chosen to be opened in order to minimize the total power loss are shown as well. In addition, the initial power loss, the minimum power loss realized, and the percentage of the power loss saved for every hour are shown. Finally, the cost of the power loss savings is calculated for every hour given that the energy cost is 11 cents/KWhr. At the bottom of the table, the total energy loss and the total savings are calculated per day and per year. From the results shown in this table, a total energy of 470.04335 MWhr is lost per year and 5006.89 $ are saved every year due to the savings in the total system energy loss.

Table 6.2 The Simulation Results for 24 Hours in Scenario 1

Hour	**Number of Agent Performing Each Optimization Task**					**Open Switches**			**Initial Power Loss (kW)**	**Minimum Power Loss (kW)**	**Percentage of Power Loss Saving (%)**	**Cost of Energy Loss Savings ($)**
	T1	**T2**	**T3**	**T4**	**T5**							
1	1	2	3	4	5	3	4	5	77.68218	70.16215	10.71807	0.82720
2	2	1	6	7	8	3	4	5	50.14696	45.19734	10.95113	0.54445
3	3	4	5	6	7	3	4	5	55.89442	51.25684	9.047729	0.51013
4	4	3	1	2	8	3	5	8	50.23763	45.76816	9.765451	0.49164
5	5	6	7	1	8	3	4	5	76.44624	69.45304	10.06896	0.76925
6	6	5	2	3	4	3	5	8	52.48106	48.52149	8.160428	0.43555
7	7	2	3	8	1	3	5	8	69.29286	64.15068	8.015774	0.56563
8	8	7	4	5	6	3	4	5	57.62197	52.33195	10.10859	0.58190
9	1	3	5	4	2	3	4	5	56.39238	51.15918	10.22925	0.57565
10	2	1	6	7	8	3	5	8	46.81706	42.46926	10.23753	0.47825
11	5	4	7	6	3	3	4	5	69.47971	63.61036	9.227029	0.64562
12	3	8	1	2	4	3	5	8	54.01975	48.17006	12.14384	0.64346
13	7	6	8	1	5	3	4	5	69.60711	63.0603	10.38182	0.72014
14	4	5	2	3	6	3	4	5	44.67726	40.84377	9.385721	0.42168
15	8	2	3	7	1	3	5	8	60.11654	55.19931	8.908146	0.54089
16	6	8	4	5	7	3	4	5	48.05253	43.80404	9.69887	0.46733
17	1	3	5	4	2	3	4	5	69.27517	63.45907	9.165124	0.63977
18	2	1	6	7	8	3	5	8	46.89323	43.32921	8.22544	0.39204
19	5	4	7	6	3	3	4	5	67.58834	61.3964	10.08519	0.68111
20	3	8	1	2	4	4	5	6	53.78845	50.68667	6.119511	0.34119
21	7	6	8	1	5	3	5	8	56.44454	51.08145	10.49908	0.58993
22	4	5	2	3	6	3	4	5	46.55714	42.25096	10.1919	0.47368
23	8	2	3	7	1	3	4	5	75.47294	68.42725	10.29661	0.77502
24	6	7	4	5	8	3	4	5	57.50928	52.00137	10.59187	0.60587
Total Energy Loss per Day					**1.28779 MWhr**				***Total Cost of Energy Loss Savings per Day***		**13.72 $**	
Total Energy Loss per Year					**470.04335 MWhr**				***Total Cost of Energy Loss Savings per Year***		**5006.89 $**	

The agents' performance in this scenario is shown in Table 6.3, in which the total number of each agent performance out of 24 is given, as well as the number of performing each optimization task. The same results are plotted in the diagram in Figure 6.2. As shown in the table and the figure below, each agent worked 15 times out of the 24 times, and all the agents have performed all of the 5 optimization tasks during their work. Thus, the balanced loading of all agents is verified in the proposed multi-agent system such that no agent is working more than the other agents do, and at the same time every agent is performing a different optimization task every time it works.

Table 6.3 The Agents Performance in Scenario 1

Agent Number	Number of Performance of Each Optimization Task					Total Number of Performance out of 24
	Task 1	Task 2	Task 3	Task 4	Task 5	
1	3	3	3	3	3	15
2	3	4	3	3	2	15
3	3	3	4	3	2	15
4	3	3	3	3	3	15
5	3	3	3	3	3	15
6	3	3	3	3	3	15
7	3	2	3	5	2	15
8	3	3	2	1	6	15

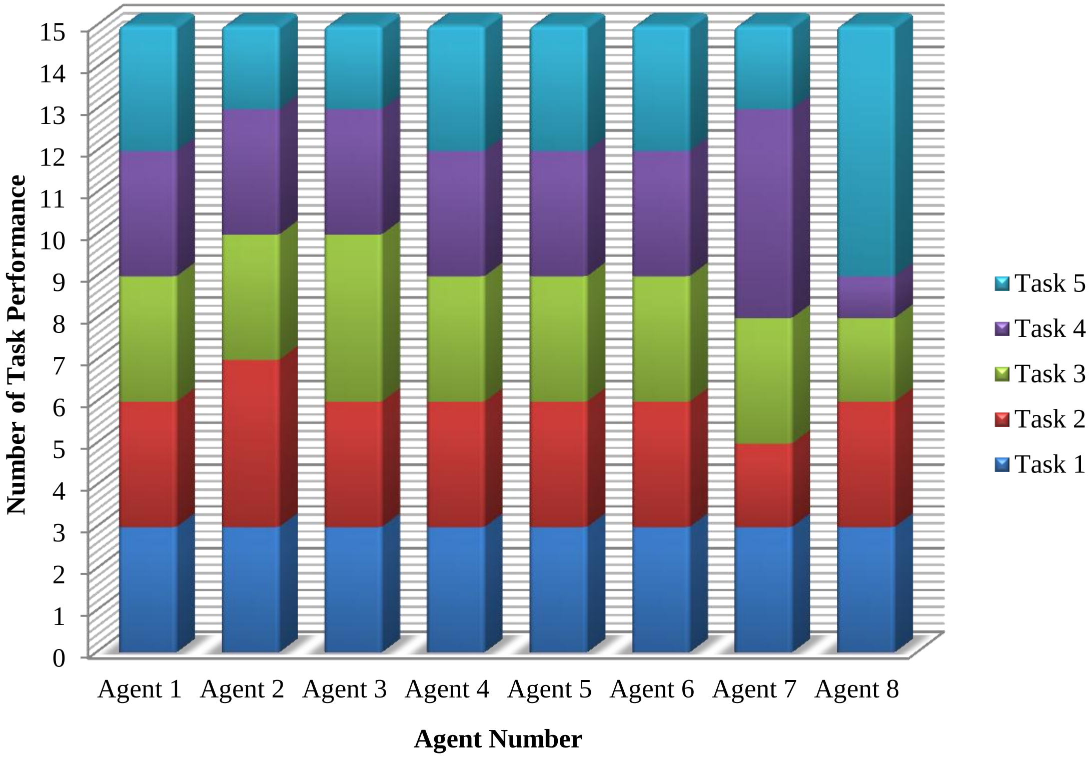

Figure 6.2 Agent Performance in Scenario 1.

6.3 Scenario 2: Light Loading and Switching Every 6 Hours

In this scenario, the same loading conditions are preserved such that every load is assigned a random loading value from 0 % to 100 % of its maximum loading value. However, the switching is carried out every 6 hours. The purpose of decreasing the number of switching per day is to elongate the life time of the system switches as will be discussed in a later section in this chapter. The simulation results of this scenario are shown in Table 6.4. In this Scenario, a total energy of 513.2265 MWhr is lost per year, while 4017.45 $ are saved per year due to the savings in the total energy loss. It can be noticed that the total energy lost per year in this scenario is more than that of the previous scenario, and at the same time the total savings per year achieved by this scenario is less than those achieved by the first one. Thus, it can be concluded that for the same loading conditions, the total savings per year are increased as the number of switching per day is increased, but that is on the expense of the switches lifetime that is definitely decreased as the number of switching per day is increased.

The agents' performance in this scenario is shown in Table 6.5 and Figure 6.3. The first four agents have worked 3 times out of 4, while the second four agents have worked only 2 times out of 4. Thus, the balanced agents' workload is still preserved in this scenario. It can be concluded that the balanced agents' workload is always realized in the proposed multi-agent system independent to the number of switching allowed per day.

Table 6.4 The Simulation Results for 24 Hours in Scenario 2

Hour	**Number of Agent Performing Each Optimization Task**					**Open Switches**			**Initial Power Loss (kW)**	**Minimum Power Loss (kW)**	**Percentage of Power Loss Saving (%)**	**Cost of Energy Loss Savings ($)**
	T1	**T2**	**T3**	**T4**	**T5**							
1	1	2	3	4	5	3	4	5	56.45834	52.17287	8.213986	2.82841
7	2	1	6	7	8	3	4	5	59.82982	54.79617	9.186138	3.32221
13	3	4	5	6	7	5	6	8	77.15868	72.58776	6.297096	3.01680
19	4	3	1	2	8	4	5	6	57.58554	54.79874	5.085523	1.83928
Total Energy Loss per Day					**1.4061 MWhr**				***Total Cost of Energy Loss Savings per Day***		**11.01 $**	
Total Energy Loss per Year					**513.2265 MWhr**				***Total Cost of Energy Loss Savings per Year***		**4017.45 $**	

Table 6.5 The Agents Performance in Scenario 2

Agent Number	**Number of Performance of Each Optimization Task**					**Total Number of Performance out of 4**
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**	
1	1	1	1	0	0	3
2	1	1	0	1	0	3
3	1	1	1	0	0	3
4	1	1	0	1	0	3
5	0	0	1	0	1	2
6	0	0	1	1	0	2
7	0	0	0	1	1	2
8	0	0	0	0	2	2

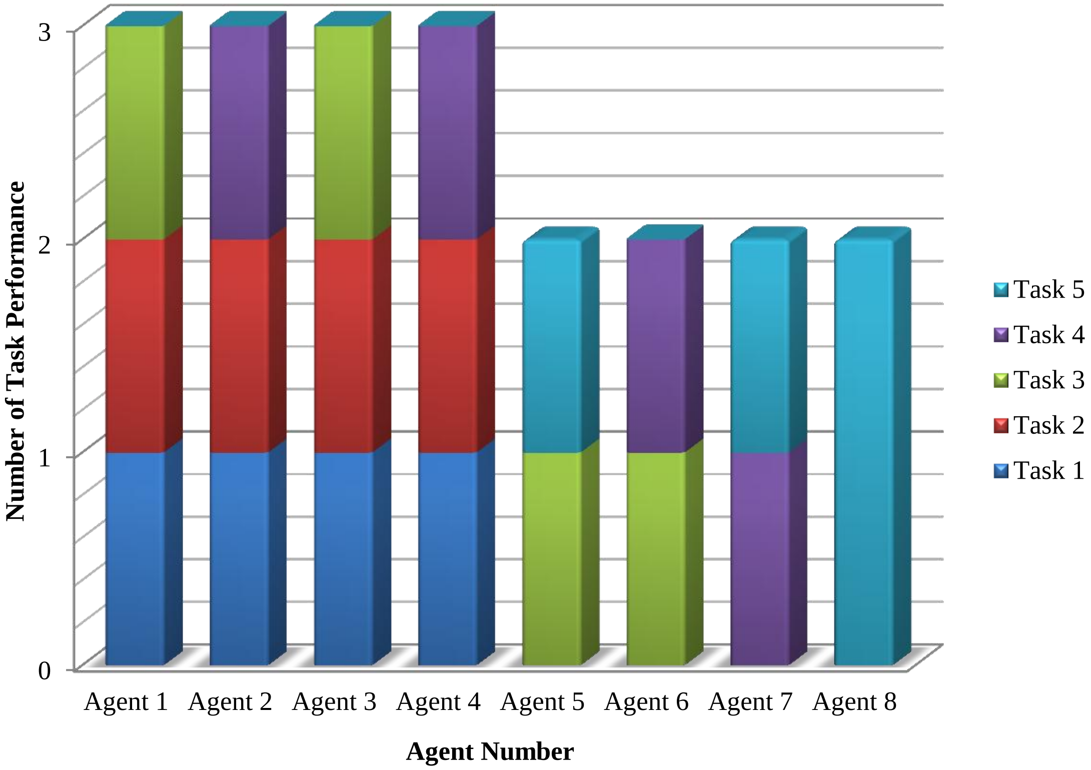

Figure 6.3 The Agents Performance in Scenario 2.

6.4 Scenario 3: Heavy Loading and Switching Every Hour

This scenario is involving every hour switching as the first scenario, but overloading is allowed here such that every load is assigned a random loading value from 70 % to 110 % of its maximum permissible loading value. The simulation results of this scenario are shown in Table 6.6. A total energy of 767.18985 MWhr is lost per year, while a total savings of 8061.54 $ is achieved per year. Comparing this scenario to the first one, it can be concluded that for the case of heavy loading, the total energy lost is increased and the total savings is increased as well.

Table 6.6 The Simulation Results for 24 Hours in Scenario 3

Hour	**Number of Agent Performing Each Optimization Task**					**Open Switches**			**Initial Power Loss (kW)**	**Minimum Power Loss (kW)**	**Percentage of Power Loss Saving (%)**	**Cost of Energy Loss Savings ($)**
	T1	**T2**	**T3**	**T4**	**T5**							
1	1	2	3	4	5	3	5	8	96.22508	87.56768	9.886521	0.95231
2	2	1	6	7	8	3	4	5	93.81163	85.19726	10.11109	0.94758
3	3	4	5	6	7	3	4	5	99.66809	91.09284	9.413738	0.94327
4	4	3	1	2	8	3	5	8	97.35382	88.46739	10.04487	0.97750
5	5	6	7	1	8	3	4	5	97.07473	88.48777	9.704118	0.94456
6	6	5	2	3	4	3	5	8	90.31641	82.44881	9.5424	0.86543
7	7	2	3	8	1	3	4	5	98.84073	90.24774	9.521552	0.94522
8	8	7	4	5	6	3	4	5	91.20237	83.17968	9.645014	0.88249
9	1	3	5	4	2	3	4	5	97.85962	89.22686	9.675069	0.94960
10	2	1	6	7	8	4	5	6	91.41409	86.28207	5.947954	0.56452
11	5	4	7	6	3	3	4	5	99.77892	90.8397	9.840656	0.98331
12	3	8	1	2	4	3	4	5	96.40665	87.90218	9.674916	0.93549
13	7	6	8	1	5	3	5	8	99.05394	90.19443	9.822677	0.97454
14	4	5	2	3	6	3	4	5	91.40392	83.71123	9.189551	0.84619
15	8	2	3	7	1	3	4	5	97.65236	89.05103	9.65888	0.94614
16	6	8	4	5	7	3	4	5	95.16768	86.76696	9.681935	0.92407
17	1	3	5	4	2	3	5	8	97.0819	88.3282	9.910425	0.96290
18	2	1	6	7	8	3	4	5	93.31317	85.35052	9.329346	0.87589
19	5	4	7	6	3	3	4	5	98.34042	89.71978	9.608411	0.94827
20	3	8	1	2	4	3	4	5	95.55302	87.28922	9.467148	0.90901
21	7	6	8	1	5	3	4	5	97.52148	88.91512	9.679305	0.9467
22	4	5	2	3	6	3	4	5	94.5304	86.04399	9.862874	0.93350
23	8	2	3	7	1	3	4	5	100.617	91.58885	9.85727	0.99309
24	6	7	4	5	8	3	4	5	92.48959	83.99201	10.11713	0.93473
Total Energy Loss per Day					**2.10189 MWhr**				***Total Cost of Energy Loss Savings per Day***			**22.09 $**
Total Energy Loss per Year					**767.18985 MWhr**				***Total Cost of Energy Loss Savings per Year***			**8061.54 $**

6.5 Scenario 4: Heavy Loading and Switching Every 6 Hours

This scenario is the same as the previous one, but the switching is performed every 6 hours. The simulation results are shown in Table 6.7. In this scenario, a total energy of 842.43825 MWhr is lost per year, and a total savings of 7289.55 $ is achieved per year. Comparing this scenario to the second one, it can be concluded again that for the case of heavy loading both of the total energy loss and the total savings are increased per year. Comparing this scenario to the previous scenario, it can be concluded that as the number of switching allowed per day is decreased, the total savings achieved per year is decreased as well.

Table 6.7 The Simulation Results for 24 Hours in Scenario 4

Hour	**Number of Agent Performing Each Optimization Task**					**Open Switches**			**Initial Power Loss (kW)**	**Minimum Power Loss (kW)**	**Percentage of Power Loss Saving (%)**	**Cost of Energy Loss Savings ($)**
	T1	**T2**	**T3**	**T4**	**T5**							
1	1	2	3	4	5	3	4	5	100.9231	91.65842	10.10788	6.11471
7	2	1	6	7	8	5	6	8	94.0204	88.7208	5.973348	3.49773
13	3	4	5	6	7	3	5	8	91.5961	83.54662	9.634724	5.31266
19	4	3	1	2	8	3	4	5	90.47202	82.82615	9.231223	5.04627
Total Energy Loss per Day					**2.0805 MWhr**				***Total Cost of Energy Loss Savings per Day***			**19.97 $**
Total Energy Loss per Year					**842.43825 MWhr**				***Total Cost of Energy Loss Savings per Year***			**7289.55 $**

6.6 Scenario 5: DG, Light Loading, and Switching Every Hour

In the next four scenarios, four DGs are installed in the system in order to minimize the total power loss, improve the voltage profile, and improve the system reliability. The same four previous scenarios are performed again with the presence of the DGs in order to study the effect of the DGs installation on the performance of the system. All DGs are assumed to have a unity power factor. Thus, they are supplying active power only. The positions of the DGs are randomly chosen, and their ratings are calculated to be 20% of the summation of the maximum powers consumed by all nodes in the feeder at which each DG is installed. The data of the four DGs installed in the system is given in Table 6.8, and the simulation results of this scenario are shown in Table 6.9.

A total energy of 423.35109 MWhr is lost per year, and a total savings of 2567.11 $ is achieved per year. Comparing this scenario to the first one in which no DGs are installed, it can be concluded that both of the total energy loss and the total savings per year are decreased after DGs installation.

Table 6.8 The Data of the DGs Installed in the System

DG Number	The Load Point at which the DG is Installed	The Active Power Supplied by the DG (KW)
1	85	64.333
2	114	28
3	66	59.333
4	96	52

Table 6.9 The Simulation Results for 24 Hours in Scenario 5

Hour	Number of Agent Performing Each Optimization Task					Open Switches			Initial Power Loss (kW)	Minimum Power Loss (kW)	Percentage of Power Loss Saving (%)	Cost of Energy Loss Savings ($)
	T1	T2	T3	T4	T5							
1	1	2	3	4	5	3	5	8	37.16739	35.22408	5.517004	0.21376
2	2	1	6	7	8	3	4	5	52.74268	50.66553	4.099733	0.22848
3	3	4	5	6	7	3	5	8	60.92201	57.10206	6.689701	0.42019
4	4	3	1	2	8	3	4	5	58.39231	55.52954	5.155409	0.31490
5	5	6	7	1	8	3	4	5	41.75548	39.81698	4.868508	0.21323
6	6	5	2	3	4	4	5	6	57.39123	54.93065	4.479432	0.27066
7	7	2	3	8	1	3	4	5	34.80002	33.02283	5.381686	0.19549
8	8	7	4	5	6	3	4	5	53.29184	50.08278	6.407503	0.35299
9	1	3	5	4	2	3	4	5	58.87553	55.82415	5.466051	0.33565
10	2	1	6	7	8	3	4	5	53.01626	50.16898	5.675375	0.31320
11	5	4	7	6	3	3	5	8	38.99213	37.44171	4.140886	0.17054
12	3	8	1	2	4	3	5	8	57.38533	54.23401	5.810605	0.34664
13	7	6	8	1	5	4	5	6	42.01784	40.38824	4.034848	0.17925
14	4	5	2	3	6	3	4	5	52.07962	48.82412	6.667825	0.35810
15	8	2	3	7	1	3	4	5	59.48127	55.94778	6.315688	0.38868
16	6	8	4	5	7	3	4	5	55.08451	52.17135	5.58384	0.32044
17	1	3	5	4	2	4	5	6	39.61068	38.53109	2.801868	0.11875
18	2	1	6	7	8	3	4	5	57.58845	53.98796	6.669063	0.39605
19	5	4	7	6	3	3	4	5	44.55783	42.69439	4.36458	0.20497
20	3	8	1	2	4	3	4	5	57.11246	53.56809	6.616577	0.38988
21	7	6	8	1	5	3	5	8	56.83204	53.0817	7.06522	0.41253
22	4	5	2	3	6	3	5	8	60.35956	56.68711	6.478445	0.40396
23	8	2	3	7	1	3	4	5	37.42691	36.20677	3.369933	0.13421
24	6	7	4	5	8	3	4	5	56.92127	53.7349	5.929795	0.35050
Total Energy Loss per Day					**1.159866 MWhr**				***Total Cost of Energy Loss Savings per Day***			**7.03 $**
Total Energy Loss per Year					**423.35109 MWhr**				***Total Cost of Energy Loss Savings per Year***			**2567.11 $**

6.7 Scenario 6: DG, Light Loading, and Switching Every 6 Hours

This scenario has the same loading and switching conditions as those of the second scenario except for the presence of the DGs in the system. The simulation results of this scenario are shown in Table 6.10. In this scenario, a total energy of 392.1195 MWhr is lost per year which is much less than that lost in the second scenario, and a total savings of 2464.17 $ is achieved per year which is less than that achieved in the second scenario as well. Comparing this scenario to the previous one, it can be concluded that the decreasing the total number of switching per day has a great effect on decreasing the total energy loss per year, while the total savings per year is not affected too much and it is almost the same in the two scenarios.

Table 6.10 The Simulation Results for 24 Hours in Scenario 6

Hour	**Number of Agent Performing Each Optimization Task**					**Open Switches**			**Initial Power Loss (kW)**	**Minimum Power Loss (kW)**	**Percentage of Power Loss Saving (%)**	**Cost of Energy Loss Savings ($)**
	T1	**T2**	**T3**	**T4**	**T5**							
1	1	2	3	4	5	3	5	8	59.73396	56.12277	6.434445	2.38338
7	2	1	6	7	8	3	4	5	40.87521	39.00484	4.795202	1.23443
13	3	4	5	6	7	3	4	5	41.59136	39.84534	4.381999	1.15237
19	4	3	1	2	8	3	4	5	47.07868	44.07722	6.809545	1.98096
Total Energy Loss per Day					**1.0743 MWhr**				***Total Cost of Energy Loss Savings per Day***			**6.75 $**
Total Energy Loss per Year					**392.1195 MWhr**				***Total Cost of Energy Loss Savings per Year***			**2464.17 $**

6.8 Scenario 7: DG, Heavy Loading, and Switching Every Hour

This scenario resembles the third scenario in the loading and switching conditions with the only difference of the presence of the DGs installed in the system. The simulation results for 24 hours are given in Table 6.11. A total energy of 662.037 MWhr is lost per year and a total savings of 4766.85 $ is realized per year. Thus, the effect of the DGs installation appears in decreasing both of the total energy loss and the total savings per year. Comparing this scenario to the fifth one, both of the total energy loss and the total savings per year are increased due to the effect of the overloading.

Table 6.11 The Simulation Results for 24 Hours in Scenario 7

Hour	Number of Agent Performing Each Optimization Task					Open Switches			Initial Power Loss (kW)	Minimum Power Loss (kW)	Percentage of Power Loss Saving (%)	Cost of Energy Loss Savings ($)
	T1	**T2**	**T3**	**T4**	**T5**							
1	1	2	3	4	5	3	4	5	76.13632	71.0496	7.159397	0.55954
2	2	1	6	7	8	3	5	8	81.32375	76.4825	6.329867	0.53253
3	3	4	5	6	7	3	4	5	82.08237	76.95595	6.661504	0.56390
4	4	3	1	2	8	3	4	5	79.37792	74.49879	6.549265	0.53670
5	5	6	7	1	8	3	5	8	80.06197	74.91313	6.873085	0.56637
6	6	5	2	3	4	3	4	5	80.49655	75.70003	6.336217	0.52761
7	7	2	3	8	1	3	5	8	77.77932	73.0379	6.491723	0.52155
8	8	7	4	5	6	3	4	5	83.0833	77.66038	6.982877	0.59652
9	1	3	5	4	2	3	4	5	83.35517	78.10354	6.723933	0.57767
10	2	1	6	7	8	3	5	8	83.65515	78.41991	6.675905	0.57587
11	5	4	7	6	3	3	4	5	75.70864	71.28234	6.209533	0.48689
12	3	8	1	2	4	3	5	8	82.37057	77.021	6.945592	0.58845
13	7	6	8	1	5	3	4	5	75.25694	70.60147	6.594018	0.51210
14	4	5	2	3	6	3	4	5	82.54389	77.34003	6.728544	0.57242
15	8	2	3	7	1	3	4	5	81.84027	76.99309	6.295614	0.53319
16	6	8	4	5	7	3	4	5	82.8053	77.44274	6.924547	0.58988
17	1	3	5	4	2	3	5	8	77.51405	72.95539	6.248561	0.50145
18	2	1	6	7	8	3	5	8	81.03845	76.22312	6.317416	0.52968
19	5	4	7	6	3	3	5	8	76.32914	71.70019	6.455983	0.50918
20	3	8	1	2	4	3	4	5	81.39094	76.4201	6.504624	0.54679
21	7	6	8	1	5	3	4	5	85.79048	80.41855	6.679965	0.59091
22	4	5	2	3	6	5	6	8	83.57223	79.73758	4.809088	0.42181
23	8	2	3	7	1	3	4	5	76.58913	71.82607	6.631387	0.52393
24	6	7	4	5	8	3	4	5	82.42842	77.02076	7.021035	0.59484
Total Energy Loss per Day					**1.8138 MWhr**				***Total Cost of Energy Loss Savings per Day***			**13.06 $**
Total Energy Loss per Year					**662.037 MWhr**				***Total Cost of Energy Loss Savings per Year***			**4766.85 $**

6.9 Scenario 8: DG, Heavy Loading, and Switching Every 6 Hours

The last scenario has the same loading and switching conditions as those of the forth scenario with the only difference of the presence of the DGs installed in the system. The simulation results for 24 hours are shown in Table 6.12. A total energy of 647.2399 MWhr is lost per year and a total savings of 4517.35 $ is achieved per year. It is obvious that both of the total energy loss and the total savings per year are less than those of the fourth scenario due to the effect of the DGs installation.

Comparing this scenario to the previous scenario, it can be concluded that the total energy loss per year is decreased a little bit, while the total savings per year is almost the same. Thus, it can be concluded that in the case of the heavy loading with the DGs installed in the system, the total savings per year is not affected too much while the total number of switching permitted per day is changed. On the other hand, decreasing the total number of switching per day has a great impact on the switches lifetime, which by its turns has a great impact on the savings that could be gained in the case of elongating the switches lifetime as discussed in the next section.

Table 6.12 The Simulation Results for 24 Hours in Scenario 8

Hour	**Number of Agent Performing Each Optimization Task**					**Open Switches**			**Initial Power Loss (kW)**	**Minimum Power Loss (kW)**	**Percentage of Power Loss Saving (%)**	**Cost of Energy Loss Savings ($)**
	T1	**T2**	**T3**	**T4**	**T5**							
1	1	2	3	4	5	3	4	5	82.66951	77.81275	6.241604	3.20546
7	2	1	6	7	8	3	5	8	78.7935	73.74122	6.851357	3.33450
13	3	4	5	6	7	3	4	5	75.0492	70.51621	6.428305	2.99177
19	4	3	1	2	8	3	4	5	77.78332	73.47339	5.865964	2.84454
Total Energy Loss per Day					**1.77326 MWhr**				***Total Cost of Energy Loss Savings per Day***			**12.38 $**
Total Energy Loss per Year					**647.2399 MWhr**				***Total Cost of Energy Loss Savings per Year***			**4517.35 $**

6.10 Results Analysis and Comparison

In order to analyze the results of the eight scenarios presented, the number of switching performed per year in each scenario is calculated. Then, the lifetime of the switches is calculated based on the number of switching per year and the mechanical endurance of the switches. For most of the high power switches, the mechanical endurance is approximately 50,000 cycles [84][85]. Thus, by dividing the mechanical endurance of the switches by the number of switching per year, the lifetime of the switches could be obtained in years. For all the eight scenarios performed, the total energy loss per year, the total savings per year, the total number of switching per year, and the life time of the switches in years are all shown in Table 6.13. The total energy loss and the total savings per year for the eight scenarios are plotted in Figure 6.4 and Figure 6.5, respectively.

Table 6.13 Results of the Eight Scenarios

	Scenario	Total Energy Loss per Year (MWhr)	Total Cost of Energy Loss Savings per Year ($)	Number of Switching per year	Lifetime (years)
No DG	***Light Loading & Switching Every Hour***	470.04335	5006.8875	8760	5.7
	Light Loading & Switching Every 6 Hours	513.2265	4017.4455	1460	34.2
	Heavy Loading & Switching Every Hour	767.18985	8061.536	8760	5.7
	Heavy Loading & Switching Every 6 Hours	842.43825	7289.5537	1460	34.2
With DG	***Light Loading & Switching Every Hour***	423.35109	2567.10486	8760	5.7
	Light Loading & Switching Every 6 Hours	392.1195	2464.1734	1460	34.2
	Heavy Loading & Switching Every Hour	662.037	4766.85255	8760	5.7
	Heavy Loading & Switching Every 6 Hours	647.2399	4517.34585	1460	34.2

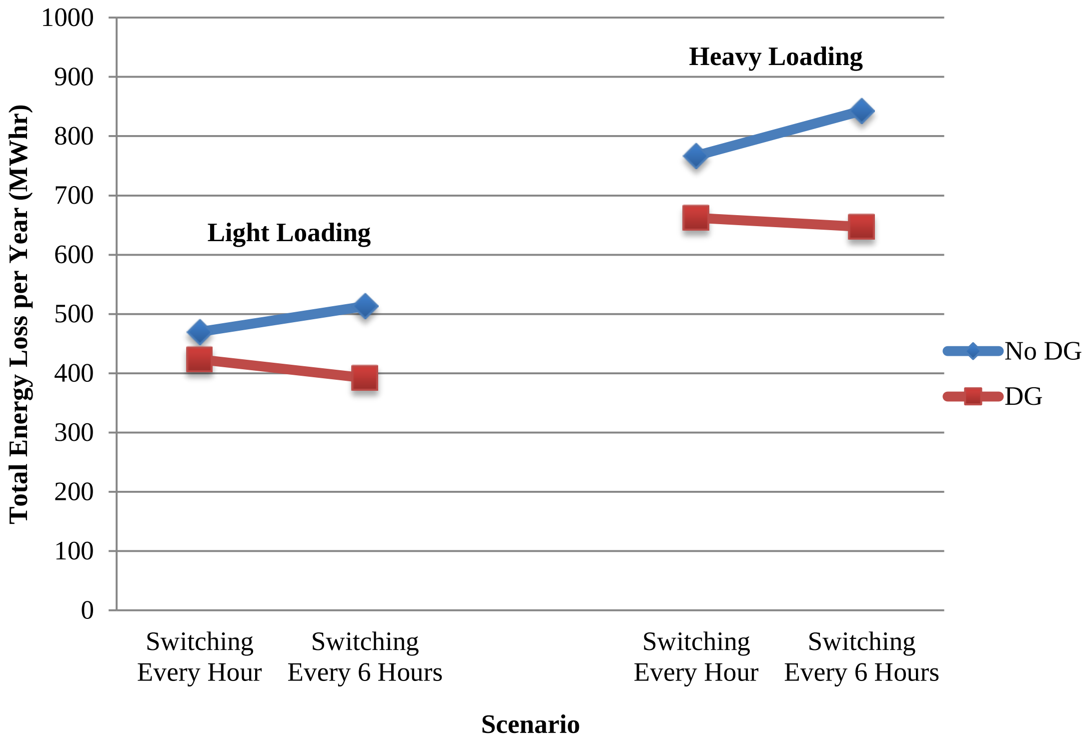

Figure 6.4 The Total Energy Loss per Year for the Eight Scenarios.

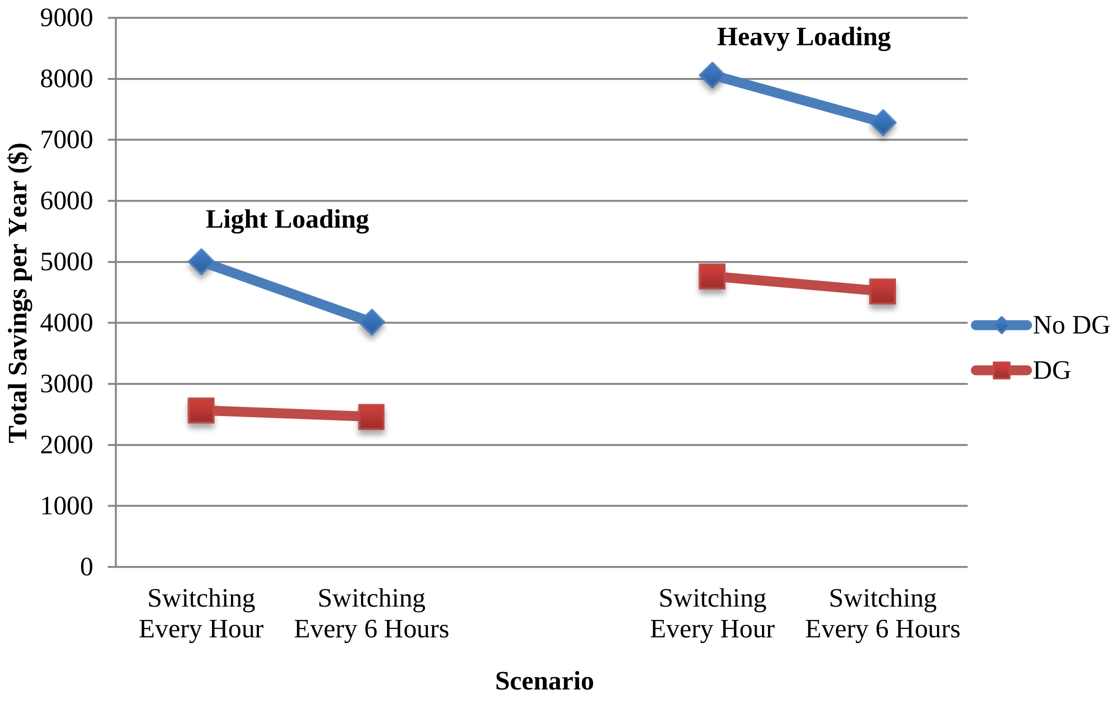

Figure 6.5 The Total Savings per Year for the Eight Scenarios.

The results shown in the table and figures above demonstrate the powerful capabilities of the designed multi-agent system in reaching the optimal configuration of the system in which the total power loss is minimized and the power cost savings are maximized. The maximum savings are achieved in the third scenario with heavy loading, no DGs installed in the system, and the switching is performed every hour. However, the switch lifetime in this scenario is expected to be around 5.7 years. On the other hand, the least savings are achieved in the sixth scenario with light loading, DGs are installed in the system, and the switching is performed every 6 hours. However, the switches lifetime in this scenario is expected to be around 34.2 years which is 6 times the switches lifetime in the third scenario.

It can be concluded that when the switching is performed every 6 hours, the total savings are not affected too much, and only are a little bit less than the cases in which the switching is performed every hour. However, the switches lifetime differ greatly in the two cases. When the switching is performed every 6 hours, the switches lifetime is almost 6 times that of the case when the switching is performed every hour. Thus, in the fourth scenario with heavy loading and no DGs are installed in the system, the total savings achieved per year are 7289.55 $ when the switching is performed every 6 hours, and the total savings in the switches lifetime is around 28.5 years more than the cases in which the switching is performed every hour. In the eighth scenario with heavy loading and DGs are installed in the system, the total savings achieved per year are 4517.35 $ when the switching is performed every 6 hours, and the total savings in the switches lifetime is around 28.5 years as well.

Although installing the DGs in system decreases the total savings per year, it has a great impact on decreasing the total energy loss per year, as well as improving the system reliability and voltage profile. Thus, either with installing the DGs in the system or not installing them, the switching every 6 hours can achieve the highest savings in both cases of light and heavy loading with respect to the total power loss savings and the switches lifetime savings.

From the previous discussion, it can be concluded that the communication delays don’t have to be taken into account since the application tackled in this thesis, the reconfiguration application, could be performed every 6 hours to achieve the maximum savings and elongate the switches lifetimes as discussed. Thus, the proposed multi-agent system could achieve the best results for that particular application from the perspective of the savings achieved, the design simplicity, and the full functionality of the intelligent agents employed in the system.

6.11 Conclusions

The designed multi-agent system has been built for the IEEE 123-Node test system employing eight different agents installed at the eight switches existing in the system. Eight different scenarios have been performed to test the performance of the designed system under different conditions. Light and heavy customer loadings have been tested while performing the switching operations every hour and every 6 hours. The impact of DGs installation on the system performance has been studied as well.

It has been shown that DGs play a vital role in decreasing the total energy loss in the system either for light loading or heavy loading. Besides, the presence of the DGs forces the total savings per year to be almost the same in both cases of switching every hour or every 6 hours. However, switching every 6 hours elongate the switches lifetime by 6 times their lifetime in case of switching every hour. Thus, for the smart distribution power system with DGs installed at various locations, the designed multi-agent system succeeds to fulfill the smart grid requirements as well as enhancing the system performance as discussed in the following points:

a. The distributed processing and decentralized control and decision making required for the implementation of the smart grid are fully achieved by the designed multi-agent system.

b. The system has the ability of the online operation in such a way that enables the reconfiguration and switching process to be performed in a very fast and accurate way.

c. The system succeeds to achieve great savings per year due to the total power loss saved by the proposed reconfiguration methodology which is able to reach the system configuration that minimizes the total power loss in the system while the loading conditions are continuously changing.

d. The system has the ability of decreasing the total number of switching per day to 4 switching operations only without increasing the total energy lost in the system. This results in elongating the switches lifetime which by its turn has a great effect on saving the costs of switch maintenance and replacement.

e. The system employs the distributed generations to minimize the total energy losses without affecting the total savings achieved per year related to saving the total energy loss per year or affecting the savings achieved related to the elongating the switches lifetime.

Chapter 7
Conclusions and Future Work

The need to migrate from the traditional distribution power systems to the smart ones becomes an urgent need because of the increasing gross in the power systems applications and the increasing load demand such as the electric vehicles. In order to achieve this migration, many challenges have to be faced. The implementation of a system handling the communications layer that handles the information flow between the different nodes in the system is one of the main technological challenges. One of the main suggested approaches for the implementation of this system is the multi-agent systems approach in which many intelligent agents could gather information from the surroundings and make a decision based on that retrieved information. This multi-agent system is expected to perform all the functions and applications required to be performed in the distribution power systems. One of those applications is the reconfiguration problem tackled in this thesis.

The reconfiguration problem is considered to be a difficult optimization problem in which the states of the sectionalizing switches in distribution power networks are determined such that the minimum power loss could be achieved subject to the radial topology, the voltage limits, and the current limits constraints. The proposed reconfiguration methodology in this thesis targets the minimization of the total power loss in the distribution power systems subject to the mentioned constraints by the utilization of the genetic algorithm and the two designed fuzzy controllers in a hybrid algorithm. The first fuzzy controller; the infeasible configurations fuzzy rejector, is rejecting any infeasible configuration that violates the system radial topology. This has a great effect in accelerating the convergence rate of the genetic algorithm since it prohibits the genetic algorithm from searching through any infeasible configurations. The second fuzzy controller; the adaptive mutation fuzzy controller, is controlling the adaptive mutation rate, which has a major impact on obtaining better results and faster convergence rate. The proposed reconfiguration methodology has demonstrated its potential of converging to the global optimal solution with the fastest convergence rate consuming the least computational time.

After the proposed reconfiguration methodology has proved its high efficiency in solving the reconfiguration problem in four different distribution power systems; the 16-node test system, the 33-node test system, the 69-node test system, and the IEEE 123-node tests system, the implementation of this methodology is targeted to solve the reconfiguration problem in the smart distribution power systems. This have been done through merging the proposed reconfiguration methodology with the concepts of automation and decentralized control to obtain an automated reconfiguration algorithm that is running on decentralized control bases taking into account the speed performance requirements. In this thesis, a novel multi-agent system has been designed employing the concepts of the distributed processing and object-oriented programming in order to simplify the design, enhance the system performance, and perform the full functionalities of intelligent agents in any multi-agent system. The designed system has been built and tested in the IEEE 123-node test system in which eight scenarios were suggested to test the system performance under different loading and switching conditions. The impact of the distributed generations' installation on the system performance has been tested as well. The designed multi-agent system has proved its powerful capabilities of the distributed processing, the decentralized control, the decentralized and on the spot decision making, the total energy loss minimization especially in the presence of the distributed generations, the total savings maximization due to the saved power loss, and the total savings in the switches maintenance and replacement due to elongating the switches lifetime.

The future work will target the real implementation of the designed multi-agent system on a set of computers or micro-controllers in order to test the system performance under real-time working conditions. The processing speed, communication delays, and system feasibility have to be evaluated and compared to those achieved by another multi-agent system implemented via one of the agents toolkits discussed in appendix B.

Appendix A
Agent Communication Languages

1. The multi-agent system architecture suggested by FIPA

The foundation of intelligent physical agents (FIPA) [86] has started working on the field of MAS in 1997 and has been crowned to be the IEEE computer society standards committee in 2005 [87]. The MAS architecture suggested by the FIPA is shown in Figure A.1 [87][88].

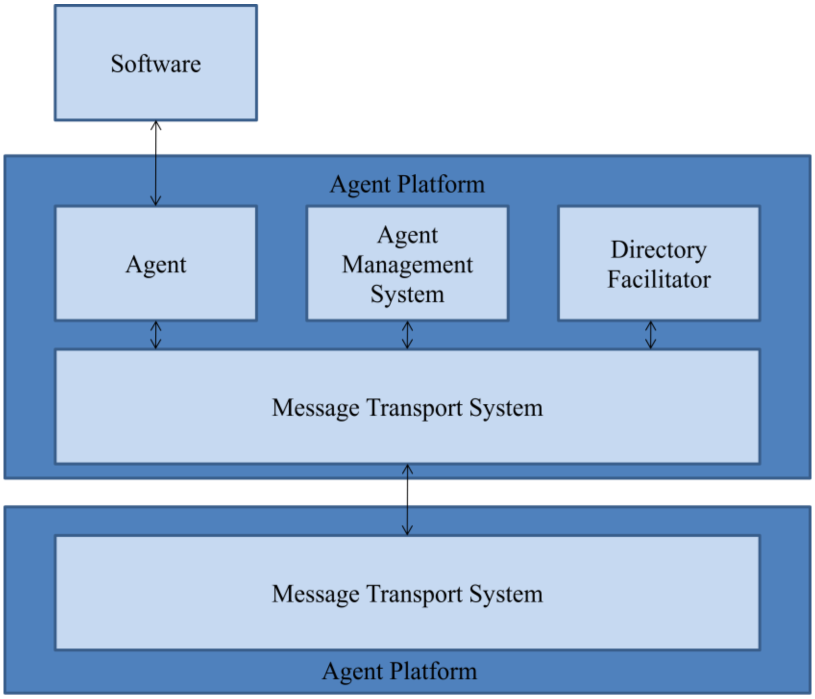

Figure A.1 FIPA agent management reference model.

The FIPA agent management reference model consists of the following components [87][88]:

1. An Agent: It is the basic unit of the MAS architecture, which is designed to perform a certain task and has a unique label according to an agent identifier (AID) to be easily distinguished among other agents in the system.

2. A Directory Facilitator (DF): It is the yellow pages server of the system which keeps the agents directory along with the services that can be offered by each agent to the other agents. Single DF or multiple DFs could be utilized in the same system.

3. An Agent Management System (AMS): It is the unique white pages server of the system that keeps the directory of all AIDs containing the transport address of all the agents which have to register with the AMS in order to join the system.
4. A Message Transport System (MTS): It is the communication channel between different agents; the DF; and the AMS in the same agent platform, and between the agents in different agent platforms.
5. An Agent Platform (AP): It is the physical infrastructure containing the machines, operating system, supporting software, DF, AMS, MTS, and agents.
6. Software: It is a kind of agent support software that could assist the agent to acquire new communication protocols, new negotiation protocols, and new security algorithms.

2. Definitions and Terminologies

For the sake of MAS utilization in real-life applications, standards have to be established to regulate the communication process among the agents, specify a canonical architecture to the MAS, and lay the foundation for agent communications languages (ACLs). In this regard, several foundations have taken the initiative to standardize the MAS rules which resulted in the creation of many ACL such as knowledge query and manipulation language (KQML) [89] and foundation for intelligent physical agents - agent communication language (FIPA-ACL) [90], which are considered as two of the most common ACLs. First, some definitions and terminologies are introduced, and then the two mentioned types of ACLs are discussed in details.

Agent communication language (ACL) is the mean through which various agents in an MAS could communicate with each other, exchange messages, and make a common decision; and it is mainly based on speech-act theory [91][92][93][94][95][96][97][98][99]. ACLs are inspired from the fact that agents could perform better in groups the same way human beings do when they work in teams, and they are implemented in a logical layer above the transport layer [91]. The transport protocols such as TCP/IP, HTTP, and IIOP are concerned with the communication aspects on the data level, while ACLs are concerned with the communications acts on the social and interaction level.

Speech-act theory was first introduced by Austin [100] in 1962, and it is inspired from the linguistic analysis of human communications, since it enables agents to perform each other with actions not only statements so that agents could behave like human beings to a great extent [95][98][99]. Since agents are communicating via some actions, the terms “speech-act”, “message”, or “performative” are used to refer to those actions, while the content of the message itself is called “content” [97]. Agents could communicate efficiently when they share a common agent communication language and protocol, a common communication content format, and a common ontology [96][99].

Ontology can be defined as the way by which the contents of the ACL messages are represented. It gains its importance from the fact that it is the channel that unifies the different expressions and vocabulary used by agents from different MAS so that they all could interact together [99]. The process of unifying the different linguistic expressions utilized by the agents connected to different platforms is called the ontology mapping, and it is a key problem that has to be solved before merging different platforms together so that agents connected to one platform could understand and interact with those connected to another one.

3. KQML

KQML [89] is considered to be a message-based language that provides the agents with a communication mean enabling them to share messages in real-time [92][96][98]. KQML can be modeled as a three-layer language [92][96] as shown in Figure A.2.

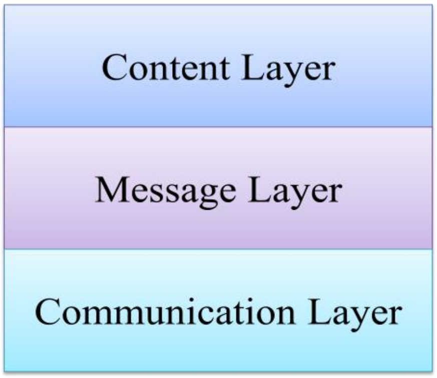

Figure A.2 KQML layered architecture.

1. Communication Layer: The layer addressing the low level communication details such as the sender, the receiver, and the message unique identifier.
2. Message Layer: It is the core of the layered architecture of KQML that is responsible for encoding messages sent between different applications in addition to identifying the message delivery protocol that performs the speech-acts attached to the message content. The content

description language, the content description, and the ontology used are also identified by this layer.

3. Content Layer: The real content of the message including the speech-acts encoded in the agent software's specific language is carried by this layer. The strength of KQML appears in its capability of carrying different messages expressed in any language and providing the protocol transporting the knowledge included in those messages regardless of the language used to express that knowledge.

In order to have a basic and good understanding of how the KQML message is formed, a simple example is shown and discussed as follows [96][97][98]:

```
(ask-one
:sender          agent-A
:receiver        stock-server
:reply-with      ibm-stock
:content        (PRICE IBM ?price)
:language        LPROLOG
:ontology        NYSE
)
```

The KQML message shown above begins with the speech-act or the performative "ask-one" which reflects the meaning of the message and the purpose behind it, followed by some message parameters, each of which begins with a keyword that describes the content of that particular parameter. The message parameters can be divided to three types [96]:

1. Content Parameter: The one carrying the major content of the message and preceded by the keyword "content".
2. Transport Parameters: The ones guiding the transport services through the message transportation process such as the parameters preceded by the keywords "sender" and "receiver" which determine the identity of the sender and receiver.
3. Receiver Parameters: These parameters either assist the receiver to decode the message by identifying the encoding language "language" and the ontology used "ontology", or guide the receiver through the replying process "reply-with".

KQML has a list of performatives from which the performative at the head of the message is chosen, and another list of keywords used for identifying the message parameters. The list of performatives and their meanings can be found in [97]**[**101**]** and is shown in Table A.1 below, while the list of parameters keywords and their meanings can be found in [96][98]**[**102**]** and is shown in Table A.2 below.

Table A.1 A List of KQML Message Performatives

KQML Message Performatives Categories		
Discourse Performatives	***Intervention and Mechanics Performatives***	***Facilitation and Networking Performatives***
ask-if, ask-all, ask-one, stream-all, eos, tell, untell, deny, insert, uninsert, delete-one, delete-all, undelete, achieve, unachieve, advertise, subscribe.	error, sorry, standby, ready, next, rest, discard.	register, unregister, forward, broadcast, transport-address, recommend-one, recommend-all, broker-one, broker-all, recruit-one, recruit-all.

Table A.2 A List of KQML Message Parameters Keywords

KQML Message Parameter Keyword	**Meaning**
sender	The agent sending the message.
receiver	The agent receiving the message.
from	If the message passes through different agents through its journey from the sender to the receiver, the original sender is identified.
to	If the message passes through different agents through its journey from the sender to the receiver, the original intended receiver is identified.
in-reply-to	The original identifier of the message initiating the message submission.
reply-with	The identifier utilized by the message replying to this message
language	The language used for encoding the message content.
ontology	The ontology used for forming the information mentioned in the message content.
content	The real message content carrying the information required to be sent from the sender to the receiver.

4. FIPA-ACL

FIPA-ACL [90] was developed and introduced by the FIPA [86] in 1997, and was inspired by the language introduced by Sadek *et al.* **[**103**]** in 1994 for the ARCOL system [97]. It resembles the KQML in its structure to a great extent, and it can also be modeled as a three layered architecture shown in Figure A.3 [94].

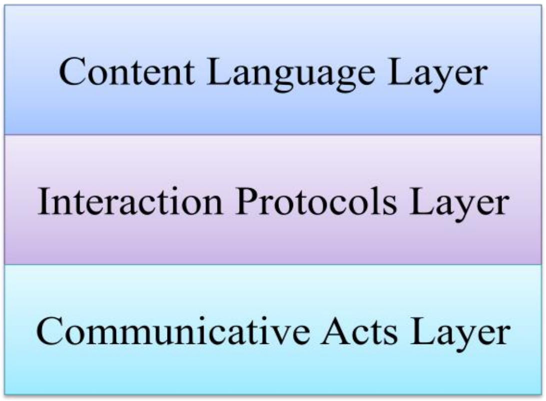

Figure A.3 FIPA-ACL layered architecture.

1. Communication Acts Layer: The layer addressing the low level communications aspects through the FIPA communicative acts Library.
2. Interaction Protocols Layer: It the core of the layered architecture of FIPA-ACL that is responsible for identifying the protocols used for the interaction between the different applications implemented on different agents in the system.
3. Content Language Layer: The Layer taking care of the real content of the message, which can be implemented using various content languages such as:
    a. FIPA-SL Content Language.
    b. Cerner Command Language (CCL).
    c. Knowledge Interchange Format (KIF).
    d. Resource Description Framework (RDF).
    e. EXtensible Markup Language (XML).
    f. Web Ontology Language (OWL).

For the sake of better understanding, two sample FIPA-ACL messages are shown below [96][97][98] written in two different content languages.

```
(inform
:sender         Agent-A
:receiver       Agent-B
:reply-with     bid02
:content        (price (bid good01) 100)
:language       fipa-sl
:ontology       auction
)
```

```
(inform
:sender         Agent-A
:receiver       Agent-B
:reply-with     bid02
:content        “weather (today, raining)”
:language       Prolog
:ontology       auction
)
```

It could be easily noticed that the message structure resembles that of the KQML message, starting with a speech-act or a performative followed by some keywords for the different parameters of the message. FIPA-ACL has a set of 20 performatives shown in Table A.3 [98], and has the same KQML set of parameter keywords described in Table A.2 above.

Table A.3 A List of FIPA-ACL Message Performatives

FIPA-ACL Message Performatives Categories				
Information Passage	***Information Request***	***Negotiation***	***Action Performance***	***Error Handling***
confirm, disconfirm, inform, inform-if, inform-ref	query-if, query-ref, subscribe	accept-proposal, cfp, propose, reject-proposal	agree, cancel, propagate, proxy, refuse, request, request-when, request-whenever	failure, not-understood

Appendix B
Agent Platforms and Toolkits

1. Introduction

The need for applying the new concept of the smart grid requires the implementation of a coherent communication layer with the capabilities of information transformation between the different parts of the system. One promising approach for the communication layer implementation is the MAS. The implementation of the MAS requires the development of the ACL which enable the intelligent agents to understand each others through the information flow between them, and also requires some powerful simulating platforms and toolkits that could provide the system designers with a tool enabling them to design the system and analyze its performance before the satge of the real implementation. Several agent platforms and toolkits have been developped over the past few years to cope up with the huge leap in agent technology, which necessitates the determination of a set of standards and features that must be available in these platforms including [104]:

1. Compitability Standars: Agent platforms have to subject to the agreed upon compitability standards such as the IEEE standards set by the FIPA.
2. Communication Capabilities: Agent platforms have to support variuos communication capabilities between agents in the same platform and among agents in different platforms.
3. Agent Mobility: The ability of the system to migrate the agent code as well as the agent execution state to another system in an efficient way.
4. Security Standards: Security of the information has to be guaranteed.
5. Availability: The platform or the toolkit has to be available to the developers.
6. Usability and Documentations: The satisfaction of the developers could be achieved through the ease and robustness of the toolkit in addition to the clear, neat, and well-organized documentation provided to them.

2. Different Agent Platforms and Toolkits

Many agent platforms and toolkits are presented in the literature including Aglets, Ajanta, Tryllian, FIPA-OS, Grasshopper, JADE, JACK, ZEUS, Voyager, Tracy, Springs, and Skeleton [1][104][105], and many studies have been done to evaluate and compare the performance of each of these toolkits such as the work done by Shakshuki and Jun [106] in 2004 in which they compared the performance of three of the agent toolkits; JADE, ZEUS, and JACK by measuring the time consumed by the agents of each of the three mentioned platforms in sending and receiving messages. In 2005, Shakshuki [107] compared the performance of eight of the agent toolkits according to many criteria including the availability, the environmental familiarity, the development powerfulness, the communication standards, the agent mobility, the security standards, the available documentation, and the message delivery time. Camacho et al. [108] compared the performance of JADE, ZEUS, and Skeleton in 2002. A brief idea about the most common utilized agent platforms is given below.

2.1 Aglets

One of the most famous agent software development kits is the Aglets [109], which was first developed by IBM in 1997 based on Java [104][105]. Each agent is assigned a single thread that limits the tasks running time of the agent. It is an open source that supports a reasonable GUI.

2.2 Ajanta

Ajanta [110] was first introduced by the Computer Science Department in University of Minnesota. It is mainly utilized for developping internet agent based applications and it has a well-organized documentation but its GUI is fragile [104].

2.3 Tryllian

The Homonym company first introduced Tryllian [111] in 2001, which is considered to be a very powerful agent toolkit that is based on the FIPA standards and implemented using Java [104]. Tryllian allows the agents to behave according to two different behaviors; reactive and proactive. The former behavior is in correspondance to the incoming messages to the agent, while the latter one is in correspondance to what is called heart beats [105].

2.4 FIPA-OS

FIPA-OS [112] is a Java-based open source agent platform developed by the FIPA, and accordingly it is based on the FIPA standards and the utilization of FIPA-ACL.

2.5 Grosshopper

In 1999, IKV++ developed and introduced Grosshopper agent toolkit [113], which is based on the FIPA standards, utilizing the FIPA-ACL, and provides a good GUI to be employed in electronic commerce, dynamic information retreival, telecommunications, and mobile computing applications [104].

2.6 JADE

One of the most well-known FIPA compliant agent development toolkits is the JADE [114]. It was first developed by Telecom Italia Lab in July 1998, and was offered as a Java-based open source agnet platform in February 2000 [105]. It has a good GUI, neat documentaion, and high level of customer satisfaction. On the other hand, agent mobility is not considered in JADE which may be a drwaback to some applications depending on the mobility of the agents, while it is not a big deal for non-mobile applications such as the distribution power systems applications.

2.7 JACK

JACK [115] was first intoruced by the Agent Oriented Software Pty. Ltd. as a Java-based agent-oriented development toolkit. It is neither an open source nor a FIPA compliant [104][106].

2.8 ZEUS

The British Telecommunications Lab offered the Java-based open source agent development toolkit ZEUS [116], which is characterized by its excellent GUI. It is a FIPA compliant and utilizes both of the agent communication languages discussed in Appendix A; KQML and FIPA-ACL [104]. The main disadvantage of ZEUS is its weak documentation which results in the lack of the customers satisfaction.

2.9 Voyager

In 1997, Object Space offered Voyager [117] which is neither an open source nor a FIPA compliant. It is mainly used for remote communication management and it doesn’t have a GUI [105].

2.10 Tracy

The University of Jena in Germany developed Tracy [118] which is characterized by its migration capabilities. It is not a well-known agent development toolkit, rarely used by researchers, and have limited applications [105].

2.11 Springs

Springs [119] is an open source agent platform offered by the Distributed Information Systems Group in the University of Zaragoza in Spain. It is a user friendly platform that focuses on the scalability and reliability of the MAS. However, it is not a FIPA compliant and it doesn't have a GUI or a good documentation.

2.12 Skeleton

Skeleton **[**120**]** is developed based on Java in order to build the MAS using two types on agents; control agents and execution agents [108].

3. Comparing Agent Platforms and Toolkits

The study made in [107] offers a detailed agent toolkits evaluation methodology based on many criteria as discussed in Table B.1. From the results shown in this table, it can be concluded that JADE and ZEUS have a similar performance criteria except for the message delivery time criteria since JADE is better than ZEUS in this aspect. It is also noticeable that Aglets is superior in the ease of documentation, mobility, and security criteria. In all the literature surveyed, there is no absolute preference to any of the discussed toolkits, but it is recommended to choose the toolkit based on the application to be implemented.

Regarding the application of implementing the communication layer that handles different smart distribution power systems applications using the multi-agents approach, it is recommended to use the JADE agent development kit since it achieves the best performance for most of the mentioned criteria especially the message delivery time, which is considered to be a key problem in power systems applications in which the minimum delay of message delivery time is required. The mobility criterion doesn't have a great importance in power systems applications since they are considered to be stationary applications in which no agent mobility is considered.

Table B.1 Comparison between the Different Agent Platforms and Toolkits

Criteria	**Best Agent Toolkit Achieving This Criteria**
Availability	JADE and ZEUS
Documentation	Aglets, JADE, and ZEUS
Ease of Installation	Aglets and FIPA-OS
GUI	Aglets, JADE, JACK, and ZEUS
Communication Standards	FIPA-OS, JADE, and ZEUS
Mobility	Aglets
Coordination	FIPA-OS, JADE, JACK, and ZEUS
Security	Aglets
Message Delivery Time	JADE

Bibliography

[1] M. Pipattanasomporn, H. Feroze, S. Rahman, “Multi-agent Systems in a Distributed Smart Grid: Design and Implementation,” Power Systems Conference and Exposition, IEEE, pp. 1 – 8, 2009.

[2] M. Keogh, *The Smart Grid: Frequently Asked Questions for State Commissions.* The National Association of Regulatory Utility Commissioners May 2009. Available Online at http://www.naruc.org/Publications/NARUC%20Smart%20Grid%20Factsheet%205_09.pdf

[3] *Enabling Tomorrow’s Electricity System*. Report of the Ontario Smart Grid Forum. Available Online at http://www.ieso.ca/imoweb/pubs/smart_grid/Smart_Grid_Forum-Report.pdf

[4] *The Smart Grid: An Introduction.* U.S. Department of Energy. Available Online at http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/DOE_SG_Book_Single_Pages%281%29.pdf

[5] M. Elkhatib, *Novel Decentralized Operation Schemes for Smart Distribution Systems.* PhD Thesis, University of Waterloo, Electrical and Computer Engineering, 2012.

[6] F. Li, W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, P. Zhang, “Smart Transmission Grid: Vision and Framework,” IEEE Transactions on Smart Grid, IEEE, Vol. 1, No. 2, pp. 168 – 177, 2010.

[7] M. Shahidehpour, Y. Wang, *Communication and Control in Electric Power Systems: Applications of Parallel and Distributed Processing.* Wiley – IEEE Press, 2003.

[8] A. S. Tanenbaum, M. V. Steen, *Distributed Systems: Principles and Paradigms.* Pearson Prentice Hall, Second Edition, 2007.

[9] D. Haughton, G. T. Heydt, “Smart Distribution System Design: Automatic Reconfiguration for Improved Reliability,” Power and Energy Society General Meeting, IEEE, 2010.

[10] V. Farahani, S. H. H. Sadeghi, H. Askarian, K. Mazlumi, “An Improved Reconfiguration Method for Maximum Loss Reduction Using Discrete Genetic Algorithm,” The 4th International Power Engineering and Optimization Conference, IEEE, pp. 178 – 183, 2010.

[11] Z. Ming, B. Pengxiang, L. Jian, “An Improved Genetic Algorithm for Distribution System Reconfiguration,” International Conference on Power System Technology, IEEE, Vol. 3, pp. 1734 – 1738, 2002.

[12] A. Merlin, H. Back, “Search for a Minimum-Loss Operating Spanning Tree Configuration for an Urban Power Distribution System,” Proceedings of the 5thPowerSystems Computer Conference, Cambridge, U.K., Sept. 1-5, 1975.

[13] D. Shirmohammadi, H. W. Hong, “Reconfiguration of Electric Distribution Networks for Resistive Line Losses Reduction,” IEEE Transactions on Power Delivery, IEEE, Vol. 4, No. 2, pp.1492 – 1498, 1989.

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

[15] S. K. Goswami, S. K. Basu, “A New Algorithm for the Reconfiguration of Distribution Feeders for Loss Minimization,” IEEE Transactions on Power Delivery, IEEE, Vol. 7, No. 3, pp. 1482 – 1491, 1992.

[16] J. Y. Fan, L. Zhang, J. D. McDonald, “Distribution Network Reconfiguration: Single Loop Optimization”, IEEE Transactions on Power Delivery, IEEE, Vol. 11, No. 3, pp. 1643 – 1647, 1996.

[17] M. E. Barain, F. F. Wu, “Network reconfiguration in distribution systems for loss reduction and load balancing,” IEEE Transactions on Power Delivery, IEEE, Vol.4,No. 2, pp. 1401 – 1407, 1989.

[18] A. B. Morton and 1. M. Y. Mareels, “An Efficient Brute-Force Solution to the Network Reconfiguration Problem”, IEEE Transactions on Power Delivery, IEEE, Vol. 15, No. 3, pp. 996 – 1000, 2000.

[19] D. Zhang, Z. Fu, L. Zhang, “Joint Optimization for Power Loss Reduction in Distribution Systems,” IEEE Transactions on Power Systems, IEEE, Vol. 23, No. 1, pp. 161 – 169, February 2008.

[20] D. E. Bouchard, *Towards Loss Minimization in Power Distribution Systems using AI: the WatDist Algorithm.* PhD Thesis, University of Waterloo, Waterloo, Ontario, Canada, 1996.

[21] B. Radha, R. T. F. Ah King, H. C. S. Rughooputh, “A Modified Genetic Algorithm for Optimal Electric Distribution Network Reconfiguration,” The 2003 Congress on Evolutionary Computation, IEEE, Vol. 2, pp. 1472 – 1479, 2003.

[22] P. Ravibabu, M. V. S. Ramya, R. Sandeep, M. V. Karthik, S. Harsha, “Implementation of Improved Genetic Algorithm in Distribution System with Feeder Reconfiguration to Minimize Real Power Losses,” Second International Conference on Computer Engineering and Technology, IEEE, Vol. 4, pp. 320 – 323, 2010.

[23] J. Mendoza, R. Lopez, D. Morales, E. Lopez, P. Dessante, R. Moraga, “Minimal Loss Reconfiguration using Genetic Algorithms with Restricted Population and Addressed Operators:

Real Application," IEEE Transactions on Power Systems, IEEE, Vol. 21, No. 2, pp. 948 – 954, 2006.

[24] H. K. Karegar, S. Jalilzadeh, V. Nabaei, A. Shabani, "Reconfiguration of Deregulated Distribution Network for Minimizing Energy Supply Cost by using Multi-objective BGA," 43rd International Universities Power Engineering Conference, IEEE, pp. 1 – 5, 2008.

[25] J. Torres-Jimenez, J. L Guardado, F. Rivas, S. Maximov, E. Melgoza, "Reconfiguration of Power Distribution Systems using Genetic Algorithms and Spanning Trees," 2010 Electronics, Robotics, and Automotive Mechanics Conference, IEEE, pp. 779 – 784, 2010.

[26] R. J. Sarfi, A. M. G. Solo, "A Hybrid Fuzzy Knowledge-based System for Network Operational Constraints in Power Distribution System Reconfiguration," International Conference on Fuzzy Systems, IEEE, pp. 1038 – 1044, 2006.

[27] K. Prasad, R. Ranjan, N. C. Sahoo, A. Chaturvedi, "Optimal Reconfiguration of Radial Distribution Systems using a Fuzzy Mutated Genetic Algorithm," IEEE Transactions on Power Delivery, IEEE, Vol. 20, Issue 2, Part 1, pp. 1211 – 1213, 2005.

[28] R. T. F. Ah King, R. Radha, H. C. S. Rughooputh, "A Fuzzy Logic Controlled Genetic Algorithm for Optimal Electrical Distribution Network Reconfiguration," Proceedings of the 2004 IEEE International Conference on Networking, Sensing, and Control, IEEE, Vol. 1, pp. 577 – 582, 2004.

[29] D. Bouchard, A. Chikhani, V. L. John, M. M. A. Salama, "Applications of Hopfield Neural Networks to distribution Feeder Reconfiguration," Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems, IEEE, pp. 311 – 316, 1993.

[30] H. Salazar, R. Gallego, R. Romero, "Artificial Neural Networks and Cluttering Techniques Applied in the Reconfiguration of Distributed Systems," IEEE Transactions on Power Delivery, IEEE, Vol. 21, No. 3, pp. 1735 – 1742, 2006.

[31] A. Y. Abdelaziz, S. F. Mekhamer, F. M. Mohammed, M. A. L. Badr, "A Modified Particle Swarm Technique for Distribution Systems Reconfiguration," The Online Journal on Electronics and Electrical Engineering, Vol. 1, No. 2, pp. 121 – 129, 2010.

[32] F. Karray, C. De Silva, *Soft Computing and Intelligent Systems Design.* Addison Wesley, 2004.

[33] S. N. Sivanandam, S. N. Deepa, *Introduction to Genetic Algorithms.* Springer, 2008.

[34] E. Kita, *Evolutionary Algorithms.* In Tech, 2011.

[35] D. B. Fogel, “An Introduction to Simulated Evolutionary Optimization,” IEEE Transactions on Neural Networks, IEEE, Vol. 5, No. 1, pp. 3 – 14, 1994.

[36] W. J. Tang, Q. H. Wu, “Biologically inspired optimization: A review,” Transactions of the Institute of Measurement and Control, pp. 495 – 515, 2009.

[37] H. Bremermann, *The evolution of intelligence. The nervous system as a model if its environment.* Contract No. 477 (17) 1, Department of Mathematics, University of Washington, 1958

[38] J. Holland, *Adaptation in natural and artificial systems.* The University of Michigan Press, 1975.

[39] D. E. Goldberg, *Real-coded genetic algorithms, virtual alphabets, and blocking.* Complex Systems, Vol. 5, pp. 139 – 167, 1991.

[40] Y. J. Cao, Q. H. Wu, *Optimization of control parameters in genetic algorithms: a stochastic approach.* International Journal of Systems Science, Lawrence Erlbaum Associates, Vol. 30, pp. 551 – 559, 1999.

[41] K. Rodriguez-Vazquez, C. M. Fonseca, P. J. Fleming, “Identifying the structure of nonlinear dynamic systems using multi-objective genetic programming,” IEEE Transactions on Systems Man and Cybernetics Part A - Systems And Humans, IEEE, Vol. 34, pp. 531 – 545, 2004.

[42] Z. Michalewicz, M. Michalewicz, “Evolutionary Computation Techniques and Their Applications,” International Conference on Intelligent Processing Systems, IEEE, Vol. 1, pp. 14 – 25, 1997.

[43] T. Back, U. Hammel, H.-P. Schwefel, “Evolutionary Computation: Comments on the History and Current State,” IEEE Transactions on Evolutionary Computation, IEEE, Vol. 1, No. 1, pp. 3 – 17, 1997.

[44] T. Back, H.-P. Schwefel, “Evolutionary Computation: An Overview,” Proceedings of IEEE International Conference on Evolutionary Computation, IEEE, pp. 20 – 29, 1996.

[45] A. Abraham, N. Nedjah, L. Mourelle, *Evolutionary Computation: from Genetic Algorithms to Genetic Programming.* Studies in Computational Intelligence, Springer, pp. 1 – 20, 2006.

[46] D. Whitley, *An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls.* Information and Software Technology 43, pp. 817 – 831, 2001.

[47] G. Jones, *Genetic and Evolutionary Algorithms.* In Encyclopedia of Computational Chemistry; P. V. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer, P. R. Schreiner, John Wiley & Sons: Chichester, UK: , 1998.

[48] X. Yao, “Global Optimization by Evolutionary Algorithms,” Proceedings of Second Aizu International Symposium on Parallel Algorithms/Architecture Synthesis, IEEE, pp. 282 – 291, 1997.

[49] N. Chaiyaratana, A. M. S. Zalzala, “Recent Developments in Evolutionary and Genetic Algorithms: Theory and Applications,” Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, IEEE, pp. 270 – 277, 1997.

[50] M. Mitchell, *An Introduction to Genetic Algorithms.* A Bradford Book, MIT Press, 5th edition, 1999.

[51] J. Z. Zhu, “Optimal reconfiguration of electrical distribution network using the refined genetic algorithm,” Electric Power Systems Research, Vol. 62, pp. 37 – 42, 2002.

[52] S. J. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach.* Englewood Cliffs, New Jersey: Prentice Hall, 1995.

[53] N. Vlassis, *A concise Introduction to Multi-agent Systems and Distributed AI.* Intelligent Autonomous Systems, Informatics Institute, University of Amsterdam, 2003.

[54] M. Wooldridge, G. Weiss, Ed., “Intelligent Agents,” in Multi-agent Systems. Cambridge, MA: MIT Press, pp. 3 – 51, 1999.

[55] D. J. AcArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, T. Funabashi, “Multi-agent Systems for Power Engineering Applications – Part I: Concepts, Approaches, and Technical Challenges,” IEEE Transactions on Power Systems, IEEE, Vol. 22, No. 4, pp. 1743 - 1752, November 2007.

[56] M. E. Baran, I. M. El-Markabi, “A Multi-agent Based Dispatching Scheme for Distributed Generators for Voltage Support on Distribution Feeders,” IEEE Transactions on Power Systems, IEEE, Vol. 22, No. 1, pp. 52 – 59, 2007.

[57] E. M. Davidson, S. D. J. McArthur, J. R. McDonald, T. Cumming, I. Watt, “Applying multi-agent system technology in practice: automated management and analysis of SCADA and digital fault recorder data,” IEEE Transactions on Power Systems, IEEE, Vol. 21, No. 2, pp. 559 – 567, 2006.

[58] S. D. J. McArthur, S. M. Strachan, G. Jahn, “The design of a multi-agent transformer condition monitoring system,” IEEE Transactions on Power Systems, IEEE, Vol. 19, No. 4, pp. 1845 – 1852, 2004.

[59] D. P. Buse, P. Sun, Q. H. Wu, J. Fitch, “Agent-based substation automation,” Power and Energy Magazine, IEEE, Vol. 1, No. 2, pp. 50 – 55, 2003.

[60] Y. Tomita, C. Fukui, H. Kudo, J. Koda, K. Yabe, "A cooperative protection system with an agent model," IEEE Transactions on Power Delivery, IEEE, Vol. 13, No. 4, pp. 1060 – 1066, 1998.

[61] V. Krishna, V. C. Ramesh, "Intelligent agents for negotiations in market games. I. Model," IEEE Transactions on Power Systems, IEEE, Vol. 13, No. 3, pp. 1103 – 1108, 1998.

[62] V. Krishna, V. C. Ramesh, "Intelligent agents for negotiations in market games. 2. Application," IEEE Transactions on Power Systems, IEEE, Vol. 13, No. 3, pp. 1109 – 1114, 1998.

[63] D. Koesrindartoto, S. Junjie, L. Tesfatsion, "An agent-based computational laboratory for testing the economic reliability of wholesale power market designs," IEEE Power Engineering Society General Meeting, IEEE, Vol. 3, pp. 2818 – 2823, 2005.

[64] J. M. Zolezzi, H. Rudnick, "Transmission cost allocation by cooperative games and coalition formation," IEEE Transactions on Power Systems, IEEE, Vol. 17, No. 4, pp. 1008 – 1015, 2002.

[65] V. S. Koritarov, "Real-world market representation with agents," Power and Energy Magazine, IEEE, Vol. 2, No. 4, pp. 39 – 46, 2004.

[66] A. L. Dimeas, N. D. Hatziargyriou, "Operation of a Multi-agent System for Micro-grid Control," IEEE Transactions on Power Systems, IEEE, Vol. 20, No. 3, pp. 1447 – 1455, 2005.

[67] Y. Kao, M. Chen, "An Agent-Based Distributed Smart Machine Tool Service System," International Symposium on Computer, Communication, Control, and Automation, IEEE, Vol. 2, pp. 41 – 44, 2010.

[68] R. Belkacemi, A. Feliachi, "Multi-agent Design for Power Distribution System Reconfiguration based on the Artificial Immune System Algorithm," Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, IEEE, pp. 3461 – 3464, 2010.

[69] T. Logenthiran, D. Srinivasan, "Multi-agent System for Managing a Power Distribution System with Plug-in Hybrid Electrical Vehicles in Smart Grid," 2011 IEEE PES Innovative Smart Grid Technologies, IEEE, pp. 346 – 351, 2011.

[70] M. Merdan, W. Lepuschitz, T. Strasser, F. Andren, "Multi-agent System for Self-optimizing Power Distribution Grids," Proceedings of the 5th International Conference on Automation, Robotics, and Applications, IEEE, pp. 312 – 317, 2011.

[71] M. Biabani, M. A. Golkar, A. Sajadi, "Operation of a Multi-agent System for Load Management in Smart Power Distribution System," The 2012 11th International Conference on Environment and Electrical Engineering, IEEE, pp. 525 – 530, 2012.

[72] M. Prymek, A. Horak, T. Sikora, " Smart Power Network Reliability Modeling using Intelligent Agents: Exploring Economic Aspects of the Power Distribution Network Reliability," The 2011 8th International Conference and Expo on Emerging Technologies for a Smarter World, IEEE, pp. 1 – 6, 2011.

[73] P. H. Nguyen, W. L. Kling, I. G. Kamphuis, P. F. Ribeiro, "Integration of Agent-based functions to facilitate operation of Smart Distribution Networks," The 2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies, IEEE, pp. 1 – 5, 2011.

[74] R. K. Sinha, R. Kumar, M. Venmathi, L. Ramesh, "Analysis of Voltage Sag with Different DG for Various Faulty Conditions," International Journal of Computer Communication and Information System, Vol. 2, No. 1, pp. 189 – 193, December 2010.

[75] C. L. T. Borges, D. M. Falcao, "Impact of Distributed Generation Allocation and Sizing on Reliability, Losses, and Voltage Profile," the 2003 IEEE Bologna Power Tech Conference Proceedings, IEEE, Vol. 2, 2003.

[76] Y. H. Song, G. S. Wang, A. T. Johns, P. Y. Wang, "Distribution network reconfiguration for loss reduction using fuzzy controlled evolutionary programming," IEE Proceedings on Generation, Transmission, and Distribution, IEE, Vol. 144, No. 4, pp. 345 – 350, 1997.

[77] J. Chakravorty, "Network Reconfiguration of Distribution System using Fuzzy Controlled Evolutionary Programming," International Journal of Engineering Science and Advanced Technology, Vol. 2, No. 2, pp. 176 – 182, 2012.

[78] J. Wang, A. Luo, M. Qi, M. Li, "The Improved Clonal Genetic Algorithm and Its Application in Reconfiguration of Distribution Networks," Power Systems Conference and Exposition, IEEE, Vol. 3, pp. 1423 – 1428, 2004.

[79] M. E. Baran, F. F. Wu, "Network Reconfiguration in Distribution Systems For Loss Reduction and Load Balancing," IEEE Transactions on Power Delivery, Vol. 4, No. 2, pp. 1404 – 1407, 1989.

[80] B. Venkatesh, S. Chandramohan, N. Kayalvizhi, R. P. Kumudini Devi, "Optimal Reconfiguration of Radial Distribution System Using Artificial Intelligence Methods," The 2009 IEEE Toronto International Conference on Science and Technology for Humanity, IEEE, pp. 660 – 665, 2009.

[81] T. Lantharthong, N. Rugthaicharoencheep, "Network Reconfiguration for Load Balancing in Distribution System with Distributed Generation and Capacitor Placement," World Academy of Science, Engineering and Technology, Issue 64, pp. 435 – 440, 2012.

[82] IEEE Power and Energy Society, *Distribution Test Feeders*. Available online at http://ewh.ieee.org/soc/pes/dsacom/testfeeders/index.html

[83] N. Mwakabuta, A. Sekar, "Comparative Study of the IEEE 34 Node Test Feeder under Practical Simplifications," The 39th North American Power Symposium, IEEE, pp. 484 – 491, 2007.

[84] *High Power Switches, Expertly engineered for reliable performance.* MERSEN. Available online at http://ep-us.mersen.com/library/AdvisorPDFs/Advisor111-L-HiPwrSwitches-web.pdf

[85] *High Power Switches, Product Data.* BOSNAX (THAILAND) CO., LTD. Available Online at http://www.bosnax.com/catalog/ferraz_hipowsw.pdf

[86] *The Foundation for Intelligent Physical Agents (FIPA).* Available Online at http://www.fipa.org/

[87] D. J. AcArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, T. Funabashi, "Multi-agent Systems for Power Engineering Applications – Part II: Technologies, Standards, and Tools for Building Multi-agent Systems," IEEE Transactions on Power Systems, IEEE, Vol. 22, No. 4, pp. 1753 - 1759, November 2007.

[88] The Foundation for Intelligent Physical Agents (FIPA), *FIPA Agent Management Specification.* Available Online at http://www.fipa.org/specs/fipa00023/XC00023H.pdf

[89] T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, D. McKay, J. McGuire, P. Pelavin, S. Shapiro, C. Beck, *Specification of the KQML Agent Communication Language.* Technical Report EIT TR 92-04, Enterprise Integration Technologies, Palo Alto, CA, 1992.

[90] The Foundation for Intelligent Physical Agents (FIPA), *FIPA-ACL Message Structure Specification.* Available Online at http://www.fipa.org/specs/fipa00061/

[91] B. Chaib-Draa, F. Dignum, "Trends in Agent Communication Language," Computational Intelligence, Vol. 18, No. 2, pp. 89 – 101, 2002.

[92] T. Finin, J. Mayfield, C. Thirunavukkarasu, "A security Architecture for Agent Communication Language," ATAL, 1997.

[93] M. Nodine, D. Chandrasekara, "Agent Communication Languages for Information-centric Agent Communities," Proceedings of the 32nd Annual Hawaii International Conference on System Sciences, IEEE, Vol. 8, pp. 1 – 10, 1999.

[94] S. Khalique, M. Jamshed, H. Suguri, H. F. Ahmad, A. Ali, M. S. Awan, "Assessment of OWL and FIPA-SL as Semantic Language," Proceeding of the IEEE Symposium on Emerging Technologies, IEEE, pp. 536 – 541, 2005.

[95] V. Vasudevan, *Comparing Agent Communication Languages.* OBJS Technical Note, Available online at http://www.objs.com/agility/tech-reports/9807-comparing-ACLs.html, 1998.

[96] H. F. Ahmad, "Multi-agent Systems: Overview of a New Paradigm for Distributed Systems," Proceedings of the 7th IEEE International Symposium on High Assurance Systems Engineering, IEEE, pp. 101 – 107, 2002.

[97] F. Guerin, *Specifying Agent Communications Languages.* PhD Thesis, University of London, Department of Electrical and Electronic Engineering, 2002.

[98] M. Wooldridge, *An Introduction to Multi-agent Systems.* John Wiley and Sons, LTD, 2002.

[99] S. M. El-Ghamrawy, A. I. El-Desouky, M. Sherief, "Dynamic Ontology Mapping for Communication in Distributed Multi-agent Intelligent System," International Conference on Networking and Media Convergence, IEEE, pp. 103 – 108, 2009.

[100] J. L. Austin, *How to do things with words.* Oxford University Press, 1962.

[101] Y. Labrou, *Semantics for an agent communication language*. PhD thesis, Baltimore, MD: University of Maryland Graduate School, 1996.

[102] Y. Labrou, T. Finin, Yun Peng "Agent Communication Language: The Current Landscape", IEEE Intelligent Systems and their Applications, IEEE, Vol. 14, No. 2, pp. 45 – 52, 1999.

[103] M. Sadek, A. Ferrieux, A. Cozannet, *Towards an artificial agent as the kernel of a spoken dialogue system: A progress report.* Proceedings AAAI Workshop of Integration of Natural Language and Speech Processing, 1994.

[104] G. Nguyen, T. T. Dang, L. Hluchy, M. Laclavik, Z. Balogh, I. Budinska, *Agent Platform Evaluation and Comparison.* Technical Report for Pellucid 5FP IST-2001-34519, Institution of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia, 2002.

[105] R. Trillo, S. Ilarri, E. Mena, "Comparison and Performance of Mobile Agent Platforms," Third International Conference on Automatic and Autonomous Systems, IEEE, 2007.

[106] E. Shakshuki, Y. Jun, *Multi-agent Development Toolkits: An Evaluation.* Innovations in Applied Artificial Intelligence, Springer Berlin/Heidelberg, Vol. 3029, pp. 209 – 218, 2004.

[107] E. Shakshuki, "A Methodology for Evaluating Agent Toolkits," International Conference on Information Technology: Coding and Computing, IEEE, Vol. 1, pp. 391 – 396, 2005.

[108] D. Camacho, R. Aler, C. Castro, J. M. Molina, “Performance Evaluation of ZEUS, JADE, and Skeleton Agent Frameworks,” The 2002 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, Vol. 4, 2002.

[109] Aglets Agent Development Toolkit, Available Online athttp://aglets.sourceforge.net/

[110] Ajanta Agent Development Toolkit, Available Online at http://ajanta.cs.umn.edu/

[111] Tryllian Agent Development Toolkit, Available Online at http://www.tryllian.com/home.html

[112] FIPA-OS Agent Development Toolkit, Available Online at http://fipa-os.sourceforge.net/

[113] Grasshopper Agent Development Toolkit, Available Online at http://www.grasshopper.de/

[114] JADE Agent Development Toolkit, Available Online at http://jade.cselt.it/

[115] JACK Agent Development Toolkit, Available Online at http://www.agent-software.com/

[116] ZEUS Agent Development Toolkit, Available Online at http://opensource.org/

[117] Voyager Agent Development Toolkit, Available Online at http://www.recursionsw.com/

[118] Tracy Agent Development Toolkit, Available Online at http://www.mobile-agents.org/

[119] Springs Agent Development Toolkit, Available Online at http://sid.cps.unizar.es/SPRINGS/

[120] D. Camacho, R. Aler, D. Borrajo and J. Molina, “Agent architecture for intelligent gathering system”, AI Communications, Vol. 18, No. 1, pp. 15 – 32, 2005.