Development of intelligent energy management system using natural computing

Cheng Yang

The University of Toledo
A Thesis

entitled

Development of Intelligent Energy Management System Using Natural Computing

by

Cheng Yang

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the

Master of Science Degree in Engineering

______________________________
Dr. Devinder Kaur, Committee Chair

______________________________
Dr. Ezzatollah Salari, Committee Member

______________________________
Dr. Mansoor Alam, Committee Member

______________________________
Dr. Patricia R. Komuniecki, Dean
College of Graduate Studies

The University of Toledo

August 2012
An Abstract of
Development of Intelligent Energy Management System Using Natural Computing
by
Cheng Yang
Submitted to the Graduate Faculty as partial fulfillment of the requirements for the
Master of Science Degree in Engineering
The University of Toledo
August 2012
In this thesis an Intelligent Energy Management System (EMS) for end consumer has
been proposed. This system develops an algorithm for smart meter which is integrated
between distribution grid and end consumers. The smart meter determines when to draw
the energy from the grid or the storage unit for consumption. The first objective of the
intelligent EMS is to save the cost for consumers by shifting the power drawn from the
grid from high cost period to low cost period. The second objective of the intelligent
EMS is to avoid grid overload by shifting the power drawn from the grid from high
demand period to low demand period. The algorithm takes into consideration the hourly
price and load demand of the grid. The algorithm was tested with the real data collected
by ISO New England for the six states of Connecticut, Maine, Massachusetts, New
Hampshire, Rhode Island and Vermont, during the period of Jan 1, 2011 to Dec 31, 2011.
Two approaches based on Fuzzy Logic and Genetic Algorithm (GA) were used. It was
demonstrated the GA based approach outperformed the Fuzzy Logic based approach. The
intelligent approach based on GA resulted in more cost saving as compared to what was
theoretically foreseen and predicted.
Acknowledgements

First, I want to express my most sincere thanks to my advisor, Prof. Devinder Kaur, who lead me into the field of natural computing, and provided continuous support on my research throughout my study in University of Toledo. Besides, I would like to thank all the professors in thesis committee: Prof. Mansoor Alam and Prof. Ezzatollah Salari, for their careful review and insightful comments.

I am grateful to all my colleagues and friends. Their helps, inspirations and friendships make my life in Toledo enjoyable and memorable.

I want to thank my parents and girlfriend for their constant love and support. The encouragement and support from my girlfriend have always been my motivation to go further. Her considerate care also ensures my dedication on the research. Also, I cannot finish this work without the guidance and support from my parent, who gave me the greatest love since I was born.
Table of Contents

Abstract ........................................................................................................................................ iii

Acknowledgements ................................................................................................................ iv

Table of Contents ....................................................................................................................... v

List of Tables .......................................................................................................................... ix

List of Figures ........................................................................................................................... x

List of Abbreviations ............................................................................................................... xiii

1. Introduction .......................................................................................................................... 1
  1.1 General Introduction ................................................................................................. 1
  1.2 Design Introduction ................................................................................................. 4
    1.2.1 Problem Definition ............................................................................................ 4
    1.2.2 Statement of Proposed Design ........................................................................ 4
    1.2.3 Overview of All Chapters ................................................................................. 5

  2.1 Management of Energy Resources ........................................................................... 7
    2.1.1 Management of Conventional Energy Resources ....................................... 7
    2.1.2 Management of Renewable Energy .............................................................. 8
  2.2 Criteria of Energy Storage Technology ...................................................................... 9
  2.3 Causes of Power Outage ........................................................................................... 10
  2.4 State of the Art in Smart Grid and Energy Management System ............................. 11
2.5 Trends and Requirement in Energy Management System

2.5.1 Power Generation Due To Renewable Energy Fluctuates

2.5.2 Deregulation Trends for Aging Infrastructure

2.5.3 Trends for Distributed Generation Control

2.5.4 Supplier – Consumer Interaction

2.6 Overview of Intelligent Approaches in Design of Energy Management System

3. Proposed Energy Management System and Explanation of Datasets Used

3.1 Explanation of Datasets

3.2 Energy Management System Scheme for Proposed Design

3.2.1 Framework of Proposed Design

3.2.2 Intelligent Smart Meter in the Proposed Design

3.2.3 Energy Storage Unit in the Proposed Design

3.2.4 Specification of Working States of ESU

3.2.5 Cost-Centric and Safety-Centric

3.3 Classic Boolean Strategy in Energy Management System Design

4. Fuzzy Logic Based Strategy in Energy Management System Design

4.1 Background Information of Fuzzy Logic

4.2 Proposed Strategy Based on Mamdani's Fuzzy Inference System

4.3 Design Details of Mamdani’s Fuzzy Inference System

5. Genetic Algorithm Based Strategy in Energy Management System Design

5.1 Background Information of Genetic Algorithm

5.1.1 Origins of Genetic Algorithm

5.1.2 Basic Concepts
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.3</td>
<td>Flowchart of Genetic Algorithm</td>
<td>43</td>
</tr>
<tr>
<td>5.1.4</td>
<td>Advantages and Disadvantages of Genetic Algorithm</td>
<td>45</td>
</tr>
<tr>
<td>5.2</td>
<td>Proposed Strategy Based on Genetic Algorithm</td>
<td>45</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Framework of Proposed Genetic Algorithm Model</td>
<td>46</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Gene Specification</td>
<td>47</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Encoding to Chromosome</td>
<td>48</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Fitness Function</td>
<td>48</td>
</tr>
<tr>
<td>5.2.5</td>
<td>Constraints</td>
<td>48</td>
</tr>
<tr>
<td>5.3</td>
<td>Design Details of Genetic Algorithm</td>
<td>49</td>
</tr>
<tr>
<td>6.</td>
<td>Methodology and Implementation</td>
<td>51</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Overview of the Matlab and Toolboxes</td>
<td>51</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Fuzzy Logic Toolbox</td>
<td>53</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Genetic Algorithm Toolbox</td>
<td>57</td>
</tr>
<tr>
<td>6.3</td>
<td>Development of Fuzzy Logic Model in Matlab</td>
<td>63</td>
</tr>
<tr>
<td>6.4</td>
<td>Development of GA Model in Matlab</td>
<td>68</td>
</tr>
<tr>
<td>7.</td>
<td>Results Comparison and Performance Evaluation</td>
<td>73</td>
</tr>
<tr>
<td>7.1</td>
<td>Household Power Consumption without Intelligent EMS</td>
<td>73</td>
</tr>
<tr>
<td>7.2</td>
<td>Power Consumption of Classic Boolean strategy</td>
<td>77</td>
</tr>
<tr>
<td>7.3</td>
<td>Power Consumption of Mamdani’s FIS Strategy</td>
<td>79</td>
</tr>
<tr>
<td>7.4</td>
<td>Power Consumption of GA Strategy</td>
<td>81</td>
</tr>
<tr>
<td>7.5</td>
<td>Comparison of Saving Cost of Three Strategies</td>
<td>84</td>
</tr>
<tr>
<td>8.</td>
<td>Conclusion</td>
<td>86</td>
</tr>
</tbody>
</table>
8.1 Summary and Conclusion of work ......................................................... 86
8.2. Contribution to this work ................................................................. 86
8.3. Future work ................................................................................. 87
References ......................................................................................... 88
Appendix A ......................................................................................... 93
Source Code of Fitness Function ......................................................... 93
Source Code of Main Function ............................................................ 94
List of Tables

7.1: Quarterly saving rates of Classic Boolean ................................................................. 78
7.2: Quarterly saving rates of Mamdani’s FIS ................................................................. 80
7.3: Quarterly saving rates of GA_I .................................................................................. 82
7.4: Quarterly saving rates of GA_II ................................................................................ 82
7.5: Quarterly saving rates of GA_III ............................................................................. 83
List of Figures

3-1: Datasets sample 1 collected from ISO New England........................................ 17
3-2: Datasets sample 2 collected from ISO New England........................................ 19
3-3: Proposed intelligent Energy Management System........................................... 21
3-4: Working states of ESU with Classic Boolean strategy on Jan 1, 2011................. 29
4-1: An example of fuzzy inference process ............................................................. 31
4-2: Membership Functions of seasons..................................................................... 32
4-3: Framework of a Mamdani’s Fuzzy Inference System based EMS...................... 34
4-4: Rules of proposed Mamdani’s Fuzzy Inference System.................................... 36
4-5: Block diagram of proposed Mamdani’s FIS .................................................... 37
5-1: Flowchart of genetic algorithm ........................................................................ 44
5-2: Framework of proposed design based on genetic algorithm............................... 46
5-3: Chromosome in proposed design ..................................................................... 47
6-1: Main desktop of Matlab 7.0................................................................................ 52
6-2: FIS Editor ....................................................................................................... 54
6-3: Membership Function Editor............................................................................ 55
6-4: Rule Editor....................................................................................................... 55
6-5: Rule Viewer....................................................................................................... 56
6-6: Surface Viewer................................................................................................ 57
6-7: GUI of Genetic Algorithm Toolbox.................................................................. 58
6-8: Calculation of Classic Boolean strategy .......................................................... 62
6-9: Mamdani’s FIS based model in FIS Editor ....................................................... 63
6-10: Membership Functions of Price in Membership Function Editor .................. 64
6-11: Membership Functions of Load in Membership Function Editor .................. 65
6-12: Membership Functions of ESU in Membership Function Editor .................. 66
6-13: Rule Editor for proposed Mamdani’s FIS based model ................................. 66
6-14: Rule Viewer for proposed Mamdani’s FIS based model ............................... 67
6-15: Surface Viewer for proposed Mamdani’s FIS based model ......................... 67
6-16: Settings of the GA model in GA Toolbox ....................................................... 68
6-17: Results of simulation of GA_I for Jan 1, 2011 ................................................. 70
6-19: Results of simulation of GA_II for Jan 1, 2011 .............................................. 71
7-1: Power price curve of household on Jan/1/2011 .............................................. 74
7-2: Grid load curve of household on Jan/1/2011 ................................................. 74
7-3: Cost of power consumption curve of household on Jan/1/2011 ..................... 75
7-4: Monthly grid load of Year 2011 ...................................................................... 76
7-5: Monthly cost of power consumption of Year 2011 ....................................... 76
7-6: Load curve of Classic Boolean strategy on Jan/1/2011 .................................. 77
7-7: Hourly cost of Classic Boolean strategy on Jan/1/2011 ................................. 78
7-8: Quarterly cost of Classic Boolean ................................................................... 78
7-9: Load curve of Mamdani’s FIS on Jan/1/2011 ................................................. 79
7-10: Hourly cost of Mamdani’s FIS on Jan/1/2011 .............................................. 79
7-11: Quarterly cost of Mamdani’s FIS ................................................................. 80
7-12: Load curve of GA on Jan/1/2011 ................................................................. 81
7-13: Hourly cost of GA on Jan/1/2011 .................................................................. 81
7-14: Quarterly cost of GA_I, GA_II and GA_III ....................................................... 83
7-15: Quarterly saving rates of three strategies ....................................................... 84
List of Abbreviations

CC ..............................Congestion Component
DA ..............................Day Ahead
EC ..............................Energy Component
EMS .............................Energy Management System
ESU ............................Energy Storage Unit
FIS ..............................Fuzzy Inference System
GA ..............................Genetic Algorithm
LMP ............................Locational Marginal Price
MLC .............................Marginal Loss Component
PTF .............................Pool Transmission Facilities
RT ..............................Real Time
Chapter 1

Introduction

1.1 General Introduction

During the recent decades, both domestic and global consumption of energy is growing rapidly. According to 2010 report of the International Energy Agency, in United States, total consumption of energy in 2008 was 26.6 TWh, compared with 22.3 TWh back in 1990. This represents a growth of 20%. The global energy demand is even more dramatic. In 1990, the global demand was 102.3 TWh and in 2008 it was 142.3 TWh which increased by 39%. The global demand of energy is rising more rapidly than in the United States [1]. It is estimated that annual rate of energy consumption will increase to the tune of 5% from 2010 [2]. The increasing amount of energy consumption is often due to various reasons that come along with rise of living standard. For industrialized countries, mostly in North America and West Europe, demand for energy steadily recovered from economic crisis of 2008. In the emerging countries like "BRIC" (Brazil, Russia, India and China), demand for all forms of energy continued to grow at a very fast speed. To avoid potential energy crisis, it is imperative to generate more energy and
at the same time, increase the energy efficiency, so that we can do more work with less energy.

Conventional energy resources typically include coal, fuel oil, natural gas and nuclear power. Materials to generate conventional energy are cheap and the power stations in which these resources are handled can be built almost anywhere. However, drawbacks of conventional energy solution are apparent. First, the quantity of material for generating conventional energy is limited and it could cost thousands of years for earth to cycle and regenerate these resources. Therefore, these resources will stop being available eventually. Second, the process of generation itself in conventional way often causes damage to surrounded environments and can lead to further serious pollution. In 1986 the infamous nuclear leak disaster occurred in Chernobyl Nuclear Power Plant in Ukraine which caused at least 200,000 cancer cases in that area [3]. Scientists and consumers are aware of the pollution caused by conventional energy and environmental degradation of these resources. Therefore, it is imperative to come up with alternative energy solutions which are sustainable and renewable and have lower carbon footprint on the environment.

To increase the supply, besides conventional energy, renewable energy is being increasingly used to generate essential power for daily consumption. Renewable energy is generated from natural resources such as sunlight, wind, rain, tides and geothermal heat, which are naturally replenished compared with conventional energy source. Due to natural aspect of renewable energy, it has much lower environmental effects than conventional energy like coal. In addition, being generated from sun directly or
indirectly, source of renewable energy regenerates itself much easier than conventional energy.

Management of energy has been complex and imperative since various sources are to be integrated to a single management system. Therefore, Smart Grid is being invented. Typical electrical grid is made up of four parts: generation, transmission, distribution and customers [4]. Smart grid is supposed to make everything better by establishing necessary 2-way data communications among components within the electrical grid. Therefore the management of energy is becoming more interactive. The possible benefits of smart grid are reported to be reduced finance costs, fewer environmental damages and possibly eliminating power outages.

Most efforts for improvement of electrical grid are fulfilled in generation and transmission grid. However, for end-consumers, applications like solar powered houses and micro wind turbines are being developed and utilized. These are still experimental approaches. There are three primary disadvantages which prevent these technologies from flourishing. The first one is that these approaches usually require infrastructural and architectural modification. The second one is that these approaches cost much more money compared with paying for electricity purchased from the grid. The third one is that energy efficiency and grid reliability of these technologies have not yet been achieved to a satisfactory level.

In conclusion, there is not yet a prominent and economical method in optimization of the Energy Management System, especially for end consumer. My thesis is concentrated in addressing the Energy Management System to make it more efficiently
and reduce the cost for consumers. In addition, improvement of the grid on avoiding possible overload is another great concern in my research.

1.2 Design Introduction

1.2.1 Problem Definition

As has been introduced in the Section 1.1, there is not yet a prominent and economical method for optimization of the Energy Management System available for end consumers. There are two critical requirements in developing it. Firstly, the Energy Management System must be embedded with the ability to reduce cost for end consumers and offers robust performance in a consistent way. Secondly, the Energy Management System is supposed to help the grid avoid possible overload or blackout.

1.2.2 Statement of Proposed Design

In this thesis a design of intelligent Energy Management System is being proposed. The proposed design is to reduce cost for end consumers and avoid possible overload or blackout for electrical grid. Two intelligent approaches were attempted to design the Energy Management System.

The first approach was based on Mamdani’s Fuzzy Inference System. The results are satisfactory for most parts of datasets. However, in very few days, which are with certain pattern in daily power consumption, the results are not satisfactory.

The second approach was based on genetic algorithm. For all tested datasets, the results are satisfactory and outperform Mamdani’s Fuzzy Inference System.
The proposed Energy Management System is made up of integrated Smart Meter and Energy Storage Unit, which are cooperating with other household appliances such as television, air conditioner, refrigerator etc. Smart Meter is empowered by intelligent strategy. Energy Storage Unit is under the control of Smart Meter. It is responsible for storage of surplus electricity. The proposed model was tested on the real-time datasets collected in Massachusetts, New Hampshire, Vermont, Connecticut, Maine and Rhode Island during the year of 2011 by ISO New England. The simulation results indicated the proposed design offers robust performance in cost saving and helps avoid overload or blackout for the grid.

1.2.3 Overview of All Chapters

This thesis is organized as follows.

Chapter One: This chapter briefly introduces the background and the objective of this research.

Chapter Two: This chapter gives information about current practices in energy management. It also discusses trends and requirements in this field. An overview of intelligent approach in design of Energy Management System is presented in the chapter.

Chapter Three: This chapter gives information about preparation of the proposed design. The physical framework will be discussed in this chapter. Datasets for testing the proposed model will also be introduced and specified in this chapter. In addition, a mathematical strategy based Energy Management System will be presented in this chapter.

Chapter Four: This chapter presents the Mamdani's Fuzzy Inference System based approach in design of Energy Management System.
Chapter Five: This chapter presents a genetic algorithm based approach in design of Energy Management System.

Chapter Six: This chapter discusses the methodology in developing the proposed model. It also introduces the implementation of the proposed model, working platform and two toolboxes that help complete the design.

Chapter Seven: This chapter compares the results of all three different models. It also gives the in-depth evaluation of the performance for all three approaches.

Chapter Eight: This chapter gives a summary of this research and outlines the intellectual contribution. It also discusses the possible improvements that could be made in the future.
Chapter 2


In this chapter, current practices in development of Energy Management System will be discussed. Section 2.1 discusses the current techniques in management of conventional and renewable energy resources. Section 2.2 presents an overview of the criteria of energy storage technologies. Section 2.3 discusses the usual causes of power outage. Section 2.4 gives an overview of the state of art for Smart Grid. Section 2.5 discusses the trends and requirements for power transmission and energy storage. Section 2.6 presents a literature review which summarized the intelligent approaches taken in the design of Energy Management System.

2.1 Management of Energy Resources

2.1.1 Management of Conventional Energy Resources

The management of conventional energy resources is through electrical grids, which bridge an interconnected and one-way network which delivers electricity from suppliers to consumers. The paradigm is complex. Generating plants are usually quite large in order to take advantage of the "economies of scale" [5]. They are often located far away from the residential area and fairly close to a source of water. The generated electrical
power is converted to a much higher voltage, at which it connects to the transmission network. The transmission network will move the power long-distance, often across state lines, and sometimes even across international boundaries, until it reaches its wholesale customers [6]. Upon arrival at the substation, the electricity will be tuned down in voltage, to fit into the distribution grid. After it exits the substation, it enters the distribution power transmission wiring. Finally, the electricity wires through service location and will be converted to the required service voltages. It is 110 volt in the United States.

2.1.2 Management of Renewable Energy

The energy management is significantly different for renewable and conventional energy resources. As has been mentioned in the Chapter 1, electrical grid is developed to offer infrastructural support to the following distinct operations: generation, transmission and distribution. All three operations bring challenges for management of renewable energy.

For generation, the sources of renewable energy geographically spread across wide distances. As an example, there are two types of wind turbines: onshore and offshore. Both of them require to be built in locations with constant high-speed winds. Therefore, they have to be set up in various places to collect wind power.

For power transmission, the renewable energy uses low voltage line as the primary transmission medium. In comparison, the conventional energy is transmitted through high voltage lines for many reasons.

For electricity distribution, due to geographical reasons, the distribution of renewable energy is more complicated than the conventional. Therefore, Smart Grid has
been developed to address the distribution of renewable energy, which is non-centralized and asymmetric. In comparison, the distribution of conventional energy is designed to feed the group of consumers who live in a certain residential area. Therefore the topology of conventional energy distribution is centralized and well-balanced.

2.2 Criteria of Energy Storage Technology

Energy storage is defined as a method to store some forms of energy in order to perform certain operations at a later time [7]. The storage forms involve chemical, biological, electrochemical, electrical, mechanical, thermal and fuel conservation storage etc. The development of energy storage allows both power suppliers and end consumers to balance the supply and the demand.

The research on energy storage technology draws consistent attention of government agencies and global corporations. The 2009 Stimulus Plan proposed an industrial standard for energy storage technology and its integration with Smart Grid [8]. The utilization of renewable energy requires the development on energy storage technology. Otherwise it would not be possible to utilize renewable energy in a reliable, stable and sustainable way. U.S. Department of Energy has funded a specified program which is concentrated in energy storage technologies. The program is named Energy Storage Systems Program [9].

Batteries are one of the approaches available for energy storage. Batteries are electrochemical storage devices. Until 2005, due to comparatively small capacity and high cost, the use of batteries has been very limited. In 2005, revolutionary development on battery technology is able to offer large capability of storage at a greatly reduced cost.
In New York and California, companies are exploring to build tremendous electrical storage facility that allows "arbitrage", which implies buying electricity at a low price and selling it later at a higher price [10].

The in-house utilization of energy storage devices has been advocated as one of the primary methods to save energy and reduce cost. However, if thousands of in-house energy storage devices are charging at the same time, that will result in higher and sometimes peak demand for distribution grid. Therefore, power suppliers have to develop redundant generation capacity to feed the need. That will cause more carbon emissions, and in the worst case, can result in power outages due to over-demand. Therefore, the in-house energy storage system with a sophisticated strategy to relieve peak demand will benefit both power suppliers and end consumers. This thesis will propose a strategy based on intelligent approach to address this challenge.

2.3 Causes of Power Outage

Power outage is caused by malfunctioning in power stations, damage to power transmission line and transmission overloading etc. A transient fault is a momentary loss of power typically caused by a temporary fault on transmission line. Power is automatically restored once the fault is cleared. Therefore, it is less harmful to the business owners and end consumers than a blackout. A brownout is a sudden decline in voltage at power supply. It often results in poor performance of business facilities and household appliances. A blackout implies a total loss of power in residential or commercial area. It is the most severe form of power outage [11]. In this thesis, a great
concern is to improve stability and sustainability of the electrical grid, which will potentially reduce the chances of all forms of outages.

2.4 State of the Art in Smart Grid and Energy Management System

Smart Grid has been popular during the past decade due to its great potential to modernize power generation, transmission and distribution grid. Smart Grid relies on multiple power suppliers and networks. It is imperative for Smart Grid to integrate all the components within the grids. Smart Grid is designed to improve energy efficiency and grid reliability by increasing the connectivity, automation and coordination between power suppliers and end consumers.

Existing implementations of Smart Grid offer a wide range of features. These features include load adjustment, demand response support, decentralization of power generation and price signaling to end consumers. These functions are fulfilled through integrated communications, sensing and measurement, Smart Meters and phasor measurement unit etc. The Energy Management System is the brain unit inside the Smart Grid. It is responsible for directing proper power supply to end consumers. In addition, it helps harmonized usage of household appliances through an energy efficient way. In application level, it is a series of computer-aided tools operating manually or automatically to monitor, control, and optimize the performance of generation, transmission and distribution grid.

2.5 Trends and Requirement in Energy Management System

2.5.1 Power Generation Due To Renewable Energy Fluctuates
Unlike the management of conventional energy generation, system operators of renewable energy have very few controls upon availability and quantity of the renewable energy such as wind and solar resources. Weather variations dictate the output intermittency of renewable energy. Therefore, fast-response power generators and energy storage systems are required to be employed to supplement the load-following capability of a power system [12].

2.5.2 Deregulation Trends for Aging Infrastructure

The trends of deregulation have become apparent. In large power systems, all three components of an electrical grid can be found, which are generation, transmission, and distribution. A conventional power system is completely self-sufficient. In comparison, a modernized power system is able to buy power from or sell power to neighboring power systems.

Deregulation of infrastructure is inspired by the competition in free markets. Conventional infrastructure usually requires generation, transmission and distribution grid to be integrated and operated by a single operator. In free markets, the idea of deregulation implies the separation of deployment for all three grids. Trends in deregulation offer new opportunities to integrate more features into the Smart Grid.

2.5.3 Trends for Distributed Generation Control

Conventional power is being generated in centralized power plants. The techniques have been developed for micro power generation. Small engines, micro wind turbines and fuel cells are being deployed to generate power remotely, which indicates a decentralized trend for power generation.
As the number of distributed generation units keeps increasing, the management of
these units is becoming critical. There has not been an industrialized standard established
for the management of distributed generation units. However, the potential standard must
ensure dependable communication between distributed resources and power demands.

2.5.4 Supplier – Consumer Interaction

The relationship between suppliers and consumers has changed greatly since Smart
Meter was invented. Smart Meter communicates with both power suppliers and end
consumers and provides real-time power information on a two-way basis.

It was tough for power suppliers to deal with peak-load in their electrical grid. Peak-
load is often due to high consumer demand. Severe cooling needs on a summer night and
warming needs on a winter night can raise the demand on power. Peak-load response
brings new challenges for the design of Energy Management System. These challenges
can be addressed by the following two strategies.

Strategy 1: The first strategy is a passive strategy, which is developed to reduce the
impact of peak-load by affecting behaviors of end consumers. For example, a consumer
is able to get immediately alerted through a Smart Meter when a peak-load rate is being
priced. Therefore they will be able to make a wise choice to switch off appliances such as
dryers and postpone their use until a lower rate is available.

Strategy 2: The second strategy is an active strategy. This can be realized by having
energy storage devices at household of end consumers. Integrated energy storage devices
are able to fill the gap of consumption while power price is high, without affecting end
consumers behaviors. Therefore, the active strategy is considered as a better solution
compared with the passive strategy.
While peak-loads are being handled properly, power suppliers are able to manage their power generation capacity, power transmission capacity and power distribution capacity in a more efficient way. In addition, power industry will be more environment friendly due to reduced emission on pollutions.

The proposed design is based on the ability of supplier-consumer interaction of Smart Meter. Smart Meters are communicating through a smart network. It allows Energy Management System to implement a variety of features, such as real-time pricing and real-time power consumption display [13].

2.6 Overview of Intelligent Approaches in Design of Energy Management System

Intelligent approaches started to play an important role in maturation of Energy Management System. In previous researches, various applications of artificial intelligence were developed to address challenges for a renewable Energy Management System. Artificial Intelligence has been used in the design of solar energy power system. It was used for design and modeling of a solar steam generating plant, for the estimation of a parabolic trough collector intercept factor and local concentration ratio. It was also used for the modeling and performance prediction of solar water heating systems. In addition, it was used for the estimation of heating loads in buildings, for the prediction of air flow in a naturally ventilated test room and for the prediction of the energy consumption of a passive solar building [14].

Traditional ways in time series load forecasting are either linear auto regressive moving average (ARMA) or nonlinear auto regressive moving average (NARMA)
models for time series. Linear models for regression are auto regresses and ARMA with infinite variance data [15]. A detailed statistical study of a large number of long, fine grain load traces from a variety of real machines leads to consideration of the Box-Jenkins models [16]. Many types of nonlinear models have been presented in the literature, see for example MARS [17], CART [18], auto regressive models [19], and bilinear models [20].

Intelligent approaches have been used for forecasting of load, price and other parameters to develop efficient Energy Management Systems. Load and price forecasting has been a central and integral process in the planning and operation of electric utilities. Load forecasting, with lead times from a few minutes to several months, helps the system operator to schedule power reserve allocation effectively. Price forecasting, on the other hand, helps the power suppliers come up with the profit-centric strategy [21].

There are basically three groups of intelligent approaches: expert system, artificial neural networks and genetic algorithm. Artificial neural network has been found well-suited for time series forecasting and has better performance than ARMA and NARMA [22]. The expert systems as well as the artificial neural network have been found to be one of the most dependable approaches for this field [23]. The advantage of these techniques over statistical models is the ability to model a multivariate problem without making complex dependency assumptions among input variables [24]. Furthermore, the artificial neural network extracts the implicit non-linear relationship among input variables by learning from training data [25].
Chapter 3

Proposed Energy Management System and Explanation of Datasets Used

In this chapter, datasets that were used to test the proposed model will be introduced in section 3.1. The basic conception of the proposed design will be introduced in section 3.2. In this section the framework of proposed design will be demonstrated. In section 3.3, a possible strategy for optimization of Energy Management System will be proposed. This strategy is named Classic Boolean Strategy and is constructed on pure mathematical method. The output of Classic Boolean Strategy will be considered to be an experimental group and will be compared with other two proposed intelligent strategies.

3.1 Explanation of Datasets

Datasets are collected from ISO New England Inc. [26]. It is an independent, non-profit Regional Transmission Organization, serving Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island and Vermont. ISO New England oversees the operation of New England's bulk electric power system and transmission lines, generated and transmitted by its member utilities, as well as Hydro-Québec, NB Power, the New York Power Authority and utilities in New York state, when the need arises. ISO New England is responsible for reliably operating New England's 32,000-megawatt bulk electric power
generation and transmission system. One of its major duties is to provide tariffs for the prices, terms, and conditions of the energy supply in New England [27].

The datasets for proposed design are real-time hourly price and load of 2011 in Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, and Vermont. According to ISO New England, the datasets were collected and archived from power stations during year of 2011. Therefore, the simulation based on these datasets is promising and convincing.

Figure 3-1 is a sample captured from the datasets collection. It includes statistics of essential power components collected from Connecticut on Jan 1st, 2011. It contains two categories of power parameters: day ahead parameters and real time parameters. The day ahead parameters are used in a financial market where market participants purchase and sell energy at financially binding day-ahead prices for the following day. Therefore, day ahead parameters are from forecasting. Read time parameters are real data collected when the power systems are running.

Figure 3-1: Datasets sample 1 collected from ISO New England
The first column is Date in the format MM/DD/YYYY, in this sample; all rows contain data for 1/1/2011.

The second column is Hour. Hour 1 means 1:00 am. In this sample, the range of the second column is from 1:00AM to 24:00PM.

The third row is called DA_DEMD, which is the day-ahead demand consisting of fixed and price sensitive demand bids plus decrement bids and increment offers. The unit of the DA_DEMD is reported as MWh which stands for megawatts hours.

The fourth column is DEMAND. It is the Non-PTF Demand as determined by metering (PTF – Pool Transmission Facilities). The unit is also MWh. Non-PTF Demand is the load used in the settlement process and is calculated as Equation 3-1.

\[
\text{Non-PTF Demand} = [\text{non-dispatchable} + \text{unmetered} + \text{station service}] \tag{3-1}
\]

From the fifth column DA_LMP to the twelfth column RT_MLC lists all prices. The units are reported as $/MWh, which stands for dollar per megawatts hours.

The fifth column DA_LMP is the day ahead locational marginal price.

The sixth column DA_EC is the energy component of the day ahead price.

The seventh column DA_CC is the congestion component of the day ahead price.

The eighth column DA_MLC is the marginal loss component of the day ahead price.

The ninth column RT_LMP is the real time locational marginal price.

The tenth column RT_EC is the energy component of the real time price.

The eleventh column RT_CC is the congestion component of the real time price.

The twelfth column RT_MLC is the marginal loss component of the real time price.

The thirteenth column DryBulb is the dry bulb temperature in degrees Fahrenheit for the corresponding weather station.
The fourteenth column DewPt is dew point temperature in degrees Fahrenheit for the corresponding weather station.

Figure 3-2 is the other sample captured from the datasets collection. This sample is called ISO New England Control Area Worksheet, which is the aggregation of all power parameters collected in Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island and Vermont. Most columns of figure 3.2 are same compared with figure 3.1. The additional two columns are SYSLoad and RegCP. SYSLoad is the actual grid load in the unit of MWh which stands for megawatts hours. RegCP is the Regulation Clearing Price in the unit of $/MWh which stands for dollar per Megawatts Hours.

SYSLoad is determined by metering. It is used for day-ahead and long-term forecasting and reporting purposes and is calculated as Equation 3-2:

\[
\text{grid load} = \text{losses} + \text{non-dispatchable} + \text{unmetered} \\
= \text{generation} - \text{pumping} + \text{net interchange}
\]
grid load is only used for planning purposes and is not used in settlement. Hourly loads are subject to re-settlement by the ISO New England.

The proposed design is focused on real time hourly price and grid load, which are critical parameters for the research on Energy Management System. An effective and efficient strategy for arbitrage in the market must be constructed on real time power price. At the same time, grid load draws a big picture of end consumer behaviors on power consumption. Therefore, RT_LMP, SYSLoad, Date and Hour are the four parameters which are imperative for the proposed design.

3.2 Energy Management System Scheme for Proposed Design

3.2.1 Framework of Proposed Design

As has been discussed in Chapter 2, a sophisticated Energy Management System for end consumer is supposed to have two crucial abilities. The first is to lower monthly bill. The second is to help the grid avoid overload or blackout. To achieve these goals, a Smart Meter is integrated between the distribution grid and end consumers. The Smart Meter has the ability of intercommunication between the power suppliers and the end consumers. It informs end consumer with real time price and grid load from power supplier. The framework of the proposed design is illustrated in Figure 3-3.
Figure 3-3 shows a modernized household equipped with Smart Meter and Energy Storage Unit, along with all other household appliances such as air conditioner, television, etc. The house in Figure 3-3 is physically connected to the distribution grid. The owner of the house pays for the power consumption on a monthly basis in the conventional way. However, there is something different called Power Smart Pricing applied in the proposed design. Power Smart Pricing indicates that the price customers pay for electricity varies from hour to hour, depending on the actual market price. The prices are set a day ahead by the hourly wholesale electricity market run by the power supplier. Power Smart Pricing has been applied by a few power suppliers such as ISO New England and Midwest Independent System Operator (MISO) [28].

3.2.2 Intelligent Smart Meter in the Proposed Design
Smart Meter is the core intelligent unit that plays the most significant role in proposed design. Firstly, it maintains an interactive connection between households of end consumers and distribution grid. Through a two way communication, consumers are able to get real time information about power price and grid load from their power suppliers. Power suppliers will be informed with the hourly power consumption of every single consumer, which is being called household Load. So that power supplier gets informed about household Load of every single family on a real time basis. After all these information being collected and organized, power supplier will get an accurate grid load for some certain area of its coverage. This information will be helpful in every aspect of grid management and price determination.

Secondly, Smart Meter is solely responsible for management of power input incoming from the distribution grid and redirecting power to household appliances and Energy Storage Unit.

Smart Meter is in charge of two switches. One switch controls when to draw power from the grid and is named "Grid Switch". The other switch controls when the Energy Storage Unit charge or release power so it is called "Storage Switch".

In Figure 3-3, household appliances and Energy Storage Unit are interconnected via Smart Meter to receive power input from the distribution grid. Energy Storage Unit has two basic states: "Charge" and "Release". State of "Charge" implies the Energy Storage Unit is charging itself with the power comes from distribution grid. State of "Release" implies the Energy Storage Unit is releasing the storage power of itself to supply other household appliances. Therefore, Storage Switch is needed to determine which states the
Energy Storage Unit is working on. Storage Switch is under control of Smart Meter integrated in household of end consumers.

The strategy inside Smart Meter must obey the following two rules, otherwise it is not an effective strategy.

**Rule 1. household appliances must be given needed power at any time.**

**Rule 2. Energy Storage Unit must stop charge immediately when the storage power hits its maximum capacity.**

Rule 1 is easy to understand. No one wants the power to be cut off from their household appliances. A design that would not let climate control work in the hottest summer day is miserable. A valid design of Smart Meter must ensure uninterruptible and adequate power supply to all household appliances.

Rule 2 ensures that we do not charge the Energy Storage Unit beyond its maximum capacity. When the Energy Storage Unit hits the designated maximum capacity, it will not be able to charge anymore. In this situation, Smart Meter must switch the state of Energy Storage Unit to any other states but not "Charge".

### 3.2.3 Energy Storage Unit in the Proposed Design

Energy Storage Unit is the second important power component within the proposed design. It consists of battery unit such as fuel cells [29]. In the real world, there is some sort of cost associated with the maintenance of the storage unit. In proposed design, the cost of maintenance will be considered to be zero to simplify the problem. However, for this research, maintenance cost is assumed to be zero. Although the storage device loses power storage capacity over time and considerable research is focused on the design of
low-loss batteries [30][31]. However, for this research, it is assumed that Energy Storage Unit will maintain its maximum storage capacity regardless of time shifting.

As has been introduced in Section 3.2.2, the Energy Storage Unit has states of "Charge" and "Release" which are under control of a Smart Meter. Just two states are not enough in considering the complexity of the situation in the real world. Therefore, some other states have been taken into consideration and have been added the proposed model. For example, "Charge 24%" means Energy Storage Unit is supposed to charge to 24% of its designed capacity. "Release 56%" means Energy Storage Unit is supposed to release 56% of its designed capacity.

By allowing Energy Storage Unit to works in a more accurate way, the proposed design can fit in some more complex situation. This fact offers great potential to develop an intelligent strategy based Energy Management System. In next two chapters, two intelligent strategies will be presented.

3.2.4 Specification of Working States of ESU

The parameter of power consumption can be presented in the following way:

\[ P_{G}[n] = E[n] + H[n] - R[n] \]  (3-3)
The working states of ESU is represented as $x[n]$, while $n$ is the hour of a day ranges from 1 to 24. The specification of $x[n]$ is introduced in following conditions.

While $x[n] > 0$, ESU is in charging, so that $R[n] = 0$

**Condition 1**: $x[n] = 1$

$$E[n] = 1 \times H[n] = H[n] \quad (3-4)$$


It means ESU is charging at 100% of the household appliances demand and power demand of household appliances is being drawn 100% from the grid.

**Condition 2**: $x[n] = 0.7$

$$E[n] = 0.7 \times H[n] \quad (3-6)$$

$$PG[n] = E[n] + H[n] - R[n] = 0.7 \times H[n] + H[n] - 0 = 1.7 \times H[n] \quad (3-7)$$

It means ESU is charging at 70% of the household appliances demand and power demand of household appliances is being drawn 100% from the distribution grid.

While $x[n] < 0$, ESU is releasing, so that $E[n] = 0$

**Condition 3**: $x[n] = -1$

$$R[n] = 1 \times H[n] = H[n] \quad (3-8)$$


It means ESU is releasing to feeding 100% of the power demand of household appliances.

**Condition 4**: $x[n] = -0.7$

$$R[n] = 0.7 \times H[n] \quad (3-10)$$

$$PG[n] = E[n] + H[n] - R[n] = 0 + H[n] - 0.7 \times H[n] = 0.3 \times H[n] \quad (3-11)$$
It means 70% of power demand of household appliances is being met by ESU and remaining 30% of the demand is being drawn from the grid.

While \( x[n] = 0 \), ESU is neither charging nor releasing, so that \( E[n] = 0 \), \( R[n] = 0 \)

**Condition 5:** \( x[n] = 0 \)

\[
\] (3-12)

It means ESU is not working at all and 100% power demand of household appliances is being drawn from the grid.

For example, value "0.5" implies Energy Storage Unit is charging itself until half of amount of current power load has being stored. In this case, total power demand to distribution grid would be Equation 3-13.

\[
\text{Power Load} = (1+0.5) \times \text{Original Power Load} = 1.5 \times \text{Original Power Load}
\] (3-13)

Another example would be when value equals to "-0.2", which means Energy Storage Unit is releasing itself to supply 20% of current power load. In this case, total power demand to distribution grid would be Equation 3-14.

\[
\text{Power Load} = (1-0.2) \times \text{Original Power Load} = 0.8 \times \text{Original Power Load}
\] (3-14)

For example, in a specific household the total power requirement of the household appliances is 1000Wh. Not all the appliances are being used at any given time. Let us suppose from 8:00PM to 9:00PM the total power demand of appliances is 600Wh and \( x[n] \) is 0.7 at the same time. Then from Equation 5 it is known that \( PG[n] = 1.7 \times H[n] \), which implies that 100% of the power demand of household is been drawn from the grid and in addition to that ESU is charging at the rate of 70% of \( H[n] \) which is 420 Wh. The total demand to the grid is 1020Wh.

3.2.5 Cost-Centric and Safety-Centric
Strategies can be either cost-centric or safety-centric. When a strategy is cost-centric, it is solely concerned with monthly cost. For example, extra power is stored when the real time price is low. When the price is turning high, the household can partly or completely rely on the "extra" power that has been stored earlier, which might potentially lower monthly bill. This concern is for only money saving purpose, so that this strategy is cost-centric.

When a strategy is safety-centric, household Energy Management System works in a way that potentially prevents the grid from being overloaded and terminates power outages. For instance, extra power is stored while the grid load is low. When the grid load is turning high, the household can partly or completely rely on the "extra" power that has been stored earlier, to stop the grid from being overloaded. It is more concentrated on helping the grid. So that it is safety-centric.

In section 3.3, a possible strategy named Classic Boolean Strategy will be proposed and introduced. It is cost-centric. In Chapter 4, a fuzzy logic based strategy will be presented. It is cost and safety centric. In Chapter 5, an genetic algorithm based strategy will be presented. This strategy is proposed to reduce the cost and help avoid overload and blackout so that it is cost and safety centric.

3.3 Classic Boolean Strategy in Energy Management System Design

Classic Boolean Strategy is a theoretical strategy for Energy Management System. It is a simple mathematical design to reduce monthly bill and it is cost-centric.
In a Classic Boolean Strategy based Household, ESU has only two states, either "Charge" or "Release". In twelve hours of a day, the Energy Storage Unit works in the state of "Charge". In the rest twelve hours, it works in the state of "Release".

Figure 3-4 demonstrates how Classic Boolean Strategy works. The lowest hourly power price on Jan 1, 2011 was 21.95, in the hour of 14:00 PM - 15:00 PM. The highest hourly power price on Jan 1, 2011 was 43.09, in the hour of 17:00 PM - 18:00 PM. In twelve hours with lower power price, Energy Storage Unit was charging itself with the power drawn from the grid. As shown in Figure 3-4, these hours were 3:00 AM - 10:00 AM, 12:00 PM - 16:00 PM and 23:00 PM - 00:00 AM. In the rest twelve hours, Energy Storage Unit was releasing the stored power to meet the demand of household appliances.

To ensure Energy Storage Unit keeps certain level of power in store, Classic Boolean Strategy base household EMS charges the same amount of power exactly as what it releases on a daily basis. For example, in Figure 3-6 during the twelve hours of "Charge" the Energy Storage Unit stores certain amount of power. Suppose that amount to be "N", during the rest twelve hours of "Release", the Energy Storage Unit releases same amount of power, which is "N". This rule enables Classic Boolean Strategy to work at any time in a sustainable way.
Figure 3-4: Working states of ESU with Classic Boolean strategy on Jan 1, 2011

The result of Classic Boolean Strategy shows a cost saving rate which ranges from 9.7% to 17.8% during the days that have been tested within 2011. However, Classic Boolean Strategy is completely dedicated for cost saving to end consumers and does not take into consideration the grid load. Therefore, it may bring challenges for maintaining the grid. For example, in a residential area where all houses with Classic Boolean Strategy applied, in twelve hours of a day the power demand will double and in the rest twelve hours the power demand will be zero. Power supplier must be aware of such situation and upgrade distribution grid to accommodate these houses.

To address the issue caused by Classic Boolean Strategy, two intelligent strategies were proposed and will be presented in Chapter 4 and Chapter 5.
Chapter 4

Fuzzy Logic Based Strategy in Energy Management System Design

In this chapter, an intelligent Energy Management System based on fuzzy logic is proposed. Section 4.1 gives background information of fuzzy logic. Section 4.2 presents the framework of proposed design. Section 4.3 discusses the details of proposed Mamdani’s FIS strategy.

4.1 Background Information of Fuzzy Logic

The purpose of Fuzzy Logic is to map an input space to an output space with a human-thinking like mechanism. This mechanism is consisted of a series of if-then statements called rules. All rules are evaluated in parallel and the order of the rules does not affect the result. Figure 4-1 shows an example for the fuzzy inference process. In general, fuzzy inference is a method that interprets the values in the input space and, based on some set of rules, assigns values to the output space.

Fuzzy Logic starts with the concept of a fuzzy set. Compared with a classical set, a fuzzy set is a set without crisp and clearly defined boundary. For example, in Figure 4-1 temperature is a typical fuzzy set. There are no clear boundaries between high and medium, or between medium and low. Another example is determined to what extent
March is winter or spring. March can be either feel like winter or feel like spring depends on the weather and temperature associated. A warm March feels like more spring and less winter, while a cold March feels like more winter and less spring.

![Fuzzy Logic Diagram]

*Figure 4-1: An example of fuzzy inference process*

In fuzzy logic, the truth of any statement becomes a matter of degree. The following two statements about season can be both true. What matters is not whether these statements are true or not, but how true these statements are.

*Statement 1:* "March is spring."

*Statement 2:* "March is winter."
Membership function is a curve to describe the degree of truth for a statement. It defines how each point in the input space is mapped to a membership value. Figure 4-2 illustrates membership function of four seasons.

![Membership Functions of seasons](image)

Figure 4-2: Membership Functions of seasons

If \( X \) is the universe of discourse and its elements are denoted by \( x \), then a fuzzy set \( A \) in \( X \) is defined as a set of ordered pairs. It is shown in Equation 4-1.

\[
A = \{x, \mu_A(x) \mid x \in X\} \tag{4-1}
\]

\( \mu_A(x) \) is the membership function of \( x \) in \( A \). The membership function maps each element of \( X \) to a membership value between 0 and 1.

Through membership function, statements can be fuzzified. A single input fuzzy if-then rule is the following form.

*If \( x \) is \( A \) then \( y \) is \( B \)*

Both "\( x \) is \( A \)" and "\( y \) is \( B \)" are fuzzified statements. In this example, "\( x \) is \( A \)" is called the antecedent or premise, while the "\( y \) is \( B \)" is called the consequent or conclusion. An example of such a rule might be

---

32
If temperature is high then climate control is cooling

A single if-then rule can also have two input parameters such as

*If* $x$ *is* $A$ *and* $y$ *is* $B$ *then* $z$ *is* $C$

In this situation, "and" is called fuzzy operator. Fuzzy operators can be "and", "or" and "not". Operation "A and B" will return the minimum value between A and B, operation "A or B" will return the maximum value between A and B, operation "not A" will return the value "$1 - A". Through logical operation by fuzzy operator, two or more input spaces can be fuzzified and mapped to a single output space.

The process of formulating the mapping from a given input to an output using fuzzy logic is called Fuzzy Inference. The process of fuzzy inference involves all of the pieces that have been discussed: Membership Function, Logical Operations and If-Then Rules.

There are two types of fuzzy inference systems: Mamdani's Fuzzy Inference System and TSK Fuzzy Inference System [32] [33] [34]. Fuzzy inference systems have been successfully applied in the fields such as automatic control, data classification, decision analysis, expert systems and computer vision.

Mamdani's Fuzzy Inference System is a preferred model to put knowledge into if-then rules. It was proposed in 1975 by Ebrahim Mamdani [33]. Mamdani's effort was on the basis of Lotfi Zadeh's paper on fuzzy algorithms for complex systems and decision processes [35].

In this thesis, one of the proposed intelligent strategies is on the basis of Mamdani's Fuzzy Inference System.

In the next section, the proposed design based on Mamdani's Fuzzy Inference System will be presented and all these steps will be demonstrated in a detailed way.
4.2 Proposed Strategy Based on Mamdani’s Fuzzy Inference System

A household intelligent Energy Management System based on Mamdani’s Fuzzy Inference System is proposed in this section. It was designed to reduce cost for end consumers and avoid grid overload for power supplier. Figure 4-3 illustrates the framework of the Mamdani’s FIS based EMS.

Figure 4-3: Framework of a Mamdani’s Fuzzy Inference System based EMS
As shown in Figure 4-3, Smart Meter connects Distribution Grid, Energy Storage Unit and other household Appliances. Smart Meter receives power information from Distribution Grid which contains power price and grid load. The integrated Mamdani's Fuzzy Inference System will process the input of price and load, and generate output which will be sent as control signal to operate Grid Switch and Storage Switch. The control signal sent to Grid Switch will determine how Smart Meter directs electricity from the Distribution Grid to Energy Storage Unit and other household Appliances. The control signal sent to Storage Switch will operate Energy Storage Unit and it will decide whether Energy Storage Unit is charging or releasing.

4.3 Design Details of Mamdani’s Fuzzy Inference System

In this section, the details of the proposed design based on Mamdani's Fuzzy Inference System will be presented. Details include fuzzification of input and defuzzification for output, fuzzy operators, and if-then rules. The whole process simulates the process of decision making of human mind.

Figure 4-4 illustrates the rules of Mamdani's Fuzzy Inference System. It was proposed as a two-input, one-output and six-rule system.

As shown in Figure 4-4, Input 1 is power price and input 2 is grid load. Output is the signal that controls Energy Storage Unit. Six rules represent the knowledge base which determines how FIS works. These rules are based on consideration of cost reduction for end consumers and grid stabilization for power supplier. Out of the two considerations, cost reduction weighted much higher than grid stabilization. There are four trends which can be seen from six rules.
Trend 1: When power price is cheaper, ESU is charging more

Trend 2: When power price is more expensive, ESU is releasing more

Trend 3: When grid load is lower, ESU is charging more

Trend 4: When grid load is higher, ESU is releasing more

These four trends precisely simulate the process of decision making in a human mind. Trend 1 and Trend 2 are focused on cost reduction, while Trend 3 and Trend 4 are focused on grid stabilization.
Figure 4-5 shows the block diagram of proposed Mamdani’s FIS. The process consists of the following five steps.

Step 1: Fuzzification of Inputs

In this step, input of power price and input of grid load are being fuzzified through membership function. Before the fuzzification, inputs of power price ranged from 19.94 to 558.55 ($/MWh), while inputs of grid load were from 8294 to 27707 (MWh). After the fuzzification, input parameter price was represented with three fuzzy set: "expensive", "medium" and "cheap". Input parameter load was represented with two fuzzy sets: "high" and "low".

Step 2: Application of Fuzzy Operator
Three fuzzified values for power price and two fuzzified values for grid load were
generated from Step 1 and ranged from 0 to 1. In Step 2, fuzzy operators apply to these
values through six rules. For example, Rule 1 is "if price is cheap and load is low, the
ESU is charging at 100%". When 0.6 is the fuzzified value of cheap and 0.4 is the
fuzzified value of low, the result of fuzzy operator is 0.4 which is the small value
between 0.6 and 0.4.

Step 3: Implication Method Application

The consequent is reshaped using a function associated with the antecedent. The
input for the implication process is a single number given by the antecedent, for example
0.4, and the output is a fuzzy set. Due to the fact that every rule needs its own
implication, a total of six implications will be made in the proposed design.

Step 4: Aggregation of Output of Rules

Because decisions are based on the testing of all of the rules in a Mamdani's Fuzzy
Inference System, the rules must be combined in some manner in order to make the
decision. Aggregation is the process by which the fuzzy sets that represent the outputs of
each rule are combined into a single fuzzy set. The input of the aggregation process is the
list of truncated output functions returned by the implication process for each rule.
Therefore there are six inputs for this step in the proposed design. The output of the
aggregation process is one fuzzy set for each output variable. Thus there is one output
after this step for the proposed design.

Step 5: Defuzzification

The input for the defuzzification process is the fuzzy set from aggregate in Step 4
and the output is the single number x[n] which represents the state that Energy Storage
Unit should be working on. Specification of $x[n]$ has been discussed in Section 3.2.4. The aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set. The most popular defuzzification method is the centroid calculation, which returns the center of area under the curve. Centroid calculation has been applied in the proposed design.
Chapter 5

Genetic Algorithm Based Strategy in Energy Management System Design

In this chapter, a genetic algorithm based strategy is proposed to outperform the Mamdani’s FIS strategy proposed in Chapter 4. Section 5.1 gives background information of genetic algorithm. Section 5.2 presents the framework of proposed design. Section 5.3 discusses the details of proposed genetic algorithm strategy.

5.1 Background Information of Genetic Algorithm

5.1.1 Origins of Genetic Algorithm

Genetic algorithms are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. The idea of genetic algorithm is inspired by Darwin's theory about evolution and Mendel’s theory about genetics. Idea of evolutionary computing was introduced in the 1973 by I. Rechenberg in his work "Evolution strategies" [35]. His idea was then developed by other researchers. Genetic Algorithms (GA) were invented by John Holland and developed by him and his students and colleagues. This lead to Holland's book "Adaption in Natural and Artificial Systems" published in 1975 [37].

In 1992 John Koza used genetic algorithm to evolve programs to perform certain tasks. He called his method "genetic programming" (GP). LISP programs were used,
because programs in this language can be expressed in the form of a "parse tree", which is the object the GA works on [38].

5.1.2 Basic Concepts

**a. Chromosome**

In genetic algorithm, the problem to be solved is represented as a chromosome. The chromosome is a string of either real numbers or binary numbers such as 0 and 1.

**b. Encoding**

Representing the problem as a chromosome is called Encoding, and it is a major challenge is GAs. One common approach is to use binary strings. These strings are sequences of 1 and 0, where the digit at each position represents the value of some aspect of the solution.

For example, while Gene represents some data such as eye color and hair color, it looks like "11010110" in binary form of genetic algorithm. Chromosome consist of an array of genes, it looks like "11010110, 10010110, 10111010, 11100101", which represents "Gene1, Gene2, Gene3, Gene4".

There are many other ways of encoding, e.g. Permutation Encoding Value Encoding Tree Encoding. The method of Encoding chosen is very much depended on the problem.

**c. Search Space**

To solve a problem with genetic algorithm, some solutions are needed. The space of all feasible solutions is called search space. Each point in the search space represent one feasible solution. Each feasible solution can be "marked" by its value or fitness for the problem. The purpose of genetic algorithm is to search the solution space to find the optimal.
d. Fitness Function

In genetic algorithm, a fitness function is a particular type of objective function that is used to determine how close a given design solution is to achieving the desired solution [39]. It varies greatly from one type of program to others. For example, if one were to create a genetic program to correct a digital scale, the fitness function would simply be the difference between the actual weight of the object and the one predicted by the digital scale, e.g. the actual weight of the object is 3 kg and weight shown by digital scale is 2 Kg, then the fitness function is 3 - 2 = 1 Kg. In conclusion, fitness function quantifies the optimality of a solution so that a particular solution may be ranked against all the other solutions.

e. Selection

Chromosomes are ranked according to the fitness and there is higher probability for the chromosome of higher fitness to be picked for reproducing. According to Darwin's natural selection theory the best ones should survive and create new offspring. This process is being called selection. There are many methods how to select the best chromosomes, eg. roulette wheel selection, Boltzman selection, tournament selection, rank selection, steady state selection and some others [40]. Roulette wheel selection is the most common one that has been used.

f. Crossover

In genetic algorithm, crossover is a genetic operator that combines two chromosomes, which are called parents, to produce a new chromosome which is called offspring. The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover
occurs during evolution according to a user-defined crossover probability. It selects genes from parent chromosomes and creates a new offspring. There are many types of crossover operators. The simple one is One-Point crossover. This method is to select a single crossover point on chromosome of both parents. All data beyond that point in either chromosome is swapped between the two parent chromosomes. Other ways of crossover involves two point, uniform, arithmetic, and heuristic crossovers.

**g. Mutation**

Mutation is a genetic operator used to maintain genetic diversity from one generation to the next. It usually takes place after a crossover is performed. Mutation alters one or more gene values in a chromosome from its initial state, which can result in entirely new gene values being added to the gene pool. With the new gene values, the genetic algorithm may be able to arrive at better solution than what was previously possible. Mutation is intended to prevent the search falling into a local optimum of the whole state space. The simple way of doing mutation is called Flip Bit. It is performed by simply inverting the value of the chosen gene. Other ways of doing mutation are Boundary, Non-Uniform, Uniform, and Gaussian.

5.1.3 Flowchart of Genetic Algorithm

According to flowchart of genetic algorithm shown in Figure 5-1, genetic algorithm is executed with the following steps.

Step 1: *Start* - Random population of n chromosomes is generated.

Step 2: *Fitness* - Fitness f(x) of each chromosome x in the population is evaluated.

Step 3: *New population* - A new population is created by repeating following steps until the new population is equal to the original population.
Step 3.1: [Selection] - Two parent chromosomes are selected from a population according to their fitness. Usually chromosome with the better fitness has a greater chance to be selected.
Step 3.2: [Crossover] - Selected parents crossover their genes to produce offsprings according to the crossover probability. If no crossover was performed, offspring is an exact copy of parents. In this case it is called self cloning.

Step 3.3: [Mutation] - With a mutation probability, algorithm mutates new offspring at each position in chromosome.

Step 3.4: [Accepting] - In this step a new offspring is placed in a new population.

Step 4: [Test] - If the termination condition is satisfied, the algorithm will stop, and return the best solution in current population. If not, the algorithm will go back to step 2.

5.1.4 Advantages and Disadvantages of Genetic Algorithm

Genetic algorithm is a heuristic algorithm which estimates a solution instead of calculating one. It has certain advantages. It can solve every optimization problem which can be described with the chromosome encoding. It solves problems with multiple solutions. The inductive nature of genetic algorithm means that it doesn't have to know any rules of the problem. This is very useful for complex or loosely defined problems. Genetic algorithm does not demand the knowledge of mathematics. Genetic algorithms are easily transferred to existing simulations and models [41].

There are some drawbacks of genetic algorithm. There is no absolute assurance that a genetic algorithm will find a global optimum. Like other artificial intelligence techniques, the genetic algorithm cannot assure constant optimization response times [42].

5.2 Proposed Strategy Based on Genetic Algorithm
5.2.1 Framework of Proposed Genetic Algorithm Model

![Image of Genetic Algorithm Framework]

Figure 5-2: Framework of proposed design based on genetic algorithm

Compared with the proposed design based on Mamdani’s Fuzzy Inference System, the GA was a completely different strategy. Input of power price and input of grid load are to be sent to genetic algorithm to generate output signal which controls the Grid Switch and Storage Switch. The purpose of the GA strategy is to outperform Mandani’s FIS strategy in cost saving. Therefore, the fitness function is determined to be daily cost in power consumption.
5.2.2 Gene Specification

In the proposed design genes are the states that Energy Storage Unit is working on in each hour. The gene is defined as $x(n)$ while the "n" implies the hour. Figure 5-3 shows the Chromosome in proposed design.

The positive value means Energy Storage Unit is charging itself with the power drawn from the distribution grid. In the meantime, other household appliances are also drawing power from the distribution grid. In this case, the power demand of the proposed household is higher than that of conventional household. The negative value means Energy Storage Unit is releasing the stored power. In the meantime, other household appliances should be working partly or completely with power released from Energy Storage Unit. In this case, the power demand of the proposed household is lower than that of conventional household. The specification of value for $x(n)$ has been given in Section 3.2.4. The default setting of $x(n)$ ranges from -1 to 1. It is defined as GA_I strategy. Another two setting of $x(n)$ ranged from -1 to 1.5 and from -1 to 2 have also been tested and defined as GA_II and GA_III. The performance of GA_I, GA_II and GA_III will be compared and evaluated in Chapter 7.
5.2.3 Encoding to Chromosome

Each gene is a double vector in the interval [-1,1]. Chromosome is defined as [ x(1) x(2) x(3) x(4) x(5) x(6) x(7) x(8) x(9) x(10) x(11) x(12) x(13) x(14) x(15) x(16) x(17) x(18) x(19) x(20) x(21) x(22) x(23) x(24) ] according to Figure 5-3. As an example, a typical chromosome is [-0.8261 -0.3799 -0.4024 0.7101 -0.1284 -0.1990 -0.3180 -0.8835 0.8835 0.2337 0.9682 -0.4254 0.1618 -0.4382 0.7074 0.8814 0.9189 -0.3746 -0.8960 -0.9767 -0.4577 0.6675 -0.0385 0.8043 -0.7543], which implies states of Energy Storage Unit in 24 hours of a day.

5.2.4 Fitness Function

The fitness function is the daily cost of the energy for household consumers and is given by the following function shown in Equation (5-1)

\[ f(x) = \sum_{n=1}^{24} [1 + x(n)] \cdot L(n) \cdot P(n) \]  

(5-1)

f(x) is the fitness function. x (n) represents the states of Energy Storage Unit at hour n. L(n) is the household load at hour n. P(n) is the power price at hour n.

5.2.5 Constraints

Energy Storage Unit must store certain amount of power to meet the demand of other household appliances. This is to ensure that it may not run out of power to meet the demand. Therefore, in proposed design Energy Storage Unit charges same amount of power as it releases on a daily basis.
The constraint for GA can be presented as the following formulation shown in Equation 5-2

\[
\sum_{n=1}^{24} [1 + x(n)] \ast L(n) = \sum_{n=1}^{24} L(n)
\]  
(5-2)

The formulation indicates that household load after applied with GA strategy remains the same as that before applying GA strategy on a daily basis.

This formulation can be simplified to Equation 5-3.

\[
\sum_{n=1}^{24} x(n) \ast L(n) = 0
\]  
(5-3)

5.3 Design Details of Genetic Algorithm

1. Initial population are generated randomly with the designated size. Initial population ranges from either [-1,1] or [-1,1.5]. Population size is 1000.

2. Scores each member of the current population by computing the fitness value using fitness function listed in Formula 5.1. In this step, individuals that do not obey the constraint (Formula 5.3) will be given a very large value, which ensure it scores very low compared with those valid individuals who met the constraint.


4. Some of the individuals in the current population that have more ability of cost saving are chosen as elite. These elite individuals are passed to the next population.

5. Perform crossover and mutation to produce offsprings. Crossover function is set to be scattered and mutation function is set to be gaussian.

6. Replaces the current population with the children to form the next generation.

7. If stop criteria is not met, go back to 2. Otherwise, result will be generated.
The best result or chromosome will be divided to 24 control signals to operate ESU daily. The fitness value associated with the best chromosome is the lowest daily cost came from genetic algorithm.
Chapter 6

Methodology and Implementation

In Chapter 6, details of the methodology and implementation of the design is presented. In Section 6.1, background information of the simulation platform is introduced. In Section 6.2, implementation of the Classic Boolean strategy is discussed. In Section 6.3, details in development of Mamdani's Fuzzy Inference System based Energy Management System are presented. In Section 6.4, details in the development of genetic algorithm strategy based Energy Management System are presented.

6.1. Overview of the Matlab and Toolboxes

6.1.1 Introduction of Matlab

Matlab is a commercial "Matrix Laboratory" package which operates as an interactive programming environment. It is a product of Mathworks. Inc and is also available for PC, Macintosh and may be found on the CIRCA VAX [43]. Matlab is well known as a specific tool for doing numerical computation with matrices and vectors. It is also well adapted to numerical experiments since the underlying algorithms for Matlab’s built-in functions and supplied m-files are based on the standard libraries LINPACK and EISPACK [44]. Therefore, Matlab is a great tool for programming and algorithm
development, all phases of data analysis, from data acquisition to visualization, application and deployment.

Figure 6-1 shows the main desktop of Matlab 7.0.

![Main desktop of Matlab 7.0](image)

Figure 6-1: Main desktop of Matlab 7.0

In the development environment of Matlab, many programming tools such as file management, variable management and application and so on are provided. On a Matlab desktop, following windows are integrated:

1. The Command Window
2. The Command History Window
3. Launch Pad
4. The Edit/Debug Window

5. Workspace Browser and Array Editor

6. Help Browser

The command window is the most important window in Matlab. Users can type commands in the command window and press enter button to run that command. In addition, m-files can be activated from the command window by typing the name of the m-file.

The history command window saves the history commands users typed in. All the history commands are listed in a descending order. By double clicking the command in this window, these commands can be activated too.

The launch pad is a unique tool which offers useful documentations for demos, related development files and applications.

The edit/debug window can be used to generate new m-files or edit existing m-files. In addition, it provides basic debug function for programming development.

Work space and array editor display related information of current values of variables, including single value, array value and matrices value. In array editor, current values of variables can be inspected and edited directly.

The help browser can be activated by clicking "help" in the menu. The help browser provides a group of information from getting started to function specification of Matlab.

6.1.2 Fuzzy Logic Toolbox

Fuzzy Logic Toolbox is an integrated development tool of Matlab that provides Matlab functions, graphical tools and a Simulink block for analyzing, designing, and simulating systems based on fuzzy logic.
Fuzzy Logic Toolbox consists of the following five GUI tools to build, edit and view fuzzy inference systems:

1. *Fuzzy Inference System (FIS) Editor*: FIS Editor is used to define the fuzzy inference system by specifying input and output variables. Figure 6-2 shows a FIS Editor.

![FIS Editor](image)

Figure 6-2: FIS Editor

2. *Membership Function Editor*: Membership Function Editor is used to define the shapes of all the membership functions associated with each variable. Figure 6-3 shows a Membership Function Editor.
3. **Rule Editor**: Rule Editor is used to edit the list of rules that defines the behavior of the fuzzy inference system. Figure 6-4 shows a Rule Editor.
4. Rule Viewer: Rule Viewer shows all the rules which fire along with their rule strength. It also shows the aggregation of the rules and gives the defuzzified output. Figure 6-5 shows the snapshot of the Rule Viewer.

5. Surface Viewer: Surface Viewer generates and plots an output surface map for the system. It gives the performance of the FIS over the entire universe of discourse. Figure 6-6 shows the snapshot of the Surface Viewer.

These five GUI tools are dynamically linked. Any changes made to the fuzzy inference system through one GUI tool will affect other GUI tools.
6.1.3 Genetic Algorithm Toolbox

Genetic Algorithm Toolbox is an integrated tool of Matlab which solves optimization problems by mimicking the principles of biological evolution, performing crossover and mutation [45]. Figure 6-7 shows Graphical User Interface (GUI) of Genetic Algorithm Toolbox.

Genetic Algorithm Toolbox provides the following configurable options.

1. *Fitness function*: Fitness function is the objective function to minimize

2. *Number of variables*: Number of variables is the number of independent variables for the fitness function.
3. **Plot Functions**: Plot functions plots various aspects of the genetic algorithm as it is executing. Each one will draw in a separate axis on the display window.

4. **Plot interval**: Plot interval specifies the number of generations between successive updates of the plot.

5. **Best fitness**: Best fitness plots the best function value in each generation.

6. **Expectation**: Expectation plots the expected number of children versus the raw scores at each generation.
7. **Score diversity**: Score diversity plots a histogram of the scores at each generation.

8. **Stopping**: Stopping plots stopping criteria levels.

9. **Best individual**: Best individual plots the vector entries of the individual with the best fitness function value in each generation.

10. **Genealogy**: Genealogy plots the genealogy of individuals.

11. **Scores**: Scores plots the scores of the individuals at each generation.

12. **Distance**: Distance plots the average distance between individuals at each generation.

13. **Range**: Range plots the minimum, maximum, and mean fitness function values in each generation.

14. **Selection**: Selection plots a histogram of the parents.

15. **Population Options**: Population options specify options for the population of the genetic algorithm; it includes population type, population size, creation function, initial population and initial range. Population type specifies the type of the input to the fitness function. Population type can be set to be Double vector, or Bit string, or Custom. Population size specifies the number of individuals in each generation. Creation function specifies the function that creates the initial population. The default creation function is Uniform which creates a random initial population with a uniform distribution. Initial population specifies an initial population for the genetic algorithm. Initial scores specify scores for initial population. Initial range specifies lower and upper bounds for the entries of the vectors in the initial population.
16. **Fitness Scaling Options:** The scaling function converts raw fitness scores returned by the fitness function to values in a range that is suitable for the selection function. Scaling function specifies the function that performs the scaling.

17. **Selection Options:** The selection function chooses parents for the next generation based on their scaled values from the fitness scaling function. Following functions are provided: Stochastic, Uniform, Roulette and Tournament.

18. **Reproduction Options:** Reproduction options determine how the genetic algorithm creates children at each new generation. Elite count specifies the number of individuals that are guaranteed to survive to the next generation. Crossover fraction specifies the fraction of individuals in the current generation which will be picked up for crossover to produce offsprings for the next generation. The remaining individuals, other than elite individuals, in the next generation are produced by mutation.

18. **Mutation Options:** Mutation functions make small random changes in the individuals in the population, which provide genetic diversity and enable the GA to search a broader space. Two functions are provided: Uniform, Gaussian.

19. **Crossover Options:** Crossover combines two individuals, or parents, to form a new individual, or child, for the next generation. Following functions are provided: One point, Two point, Scattered, Intermediate and Heuristic.

19. **Migration Options:** The following three parameters control how migration occurs. Direction controls whether Migration to be unidirectional or bidirectional. Fraction controls how many individuals move between subpopulations. Interval controls how many generations pass between migrations.
20. **Hybrid Function Options:** Hybrid Function specifies another minimization function that runs after the genetic algorithm terminates.

21. **Stopping Criteria Options:** Generations specify the maximum number of iterations for which the genetic algorithm runs. Time limit specifies the maximum time in seconds the genetic algorithm runs before stopping. Fitness limit implies if the best fitness value is less than or equal to the value of Fitness limit, the algorithm stops. Stall generations determine if there is no improvement in the best fitness value for the number of generations specified by Stall generations, the algorithm stops. Stall time limit means if there is no improvement in the best fitness value for an interval of time in seconds specified by Stall time limit, the algorithm stops.

22. **Output Function Options:** History outputs the iterative history of the algorithm to a separate window. Interval specifies the number of generations between successive outputs.

23. **Display to Command Window Options:** Level of display specifies the amount of information displayed in the MATLAB Command Window when the genetic algorithm is running.

24. **Vectorize Option:** The vectorize option specifies whether the computation of the fitness function is vectorized.

6.2. **Implementation of Classic Boolean Strategy**

Classic Boolean strategy is a simple mathematical strategy. It can be implemented through Excel 2010. Figure 6-8 shows the process of calculation for Jan 1, 2011.
As shown in Figure 6-8, a day was divided into two parts. Each part contains 12 hours. From Row 2 to Row 13 lists 12 hours with comparatively lower price, while the rest 12 hours with comparatively higher price are listed within Row 15 to Row 26. Column E shows the original cost for each hour of the day.

In 12 hours with higher price household appliances are getting power from the ESU and no power is drawn from distribution grid. Hence in these 12 hours (Row 15 - Row 26), the cost is zero. In the remaining hours (Row 2 - Row 13), household appliances are drawing power from the distribution grid at 100%, and ESU is also drawing power from
the distribution grid. Therefore, the total cost is sum of the cost of power consumption of appliances and the cost of power for charging ESU. The cost of power consumption of appliances is showed in F14. The cost of power for charging ESU is showed in G15. The total cost of Classic Boolean household is showed in G29, which indicates a saving rate of 9.7% compared with conventional household.

6.3. Development of Fuzzy Logic Model in Matlab

Figure 6-9 shows Mamdani's FIS based model in FIS Editor. Inputs are powe price and grid loads. Outputs are states of ESU. "And method" is set to be min. "Or method" is set to be max. "Implication" is set to be min. "Aggregation" is set to be max. "Defuzzification function" is set to be centroid.

![Mamdani's FIS based model in FIS Editor](image)

Figure 6-9: Mamdani's FIS based model in FIS Editor
Figure 6-10 shows membership functions of price in Membership Function Editor. There are three membership functions for price: cheap, medium and expensive. All three membership functions are set to be Gaussian type.

![Figure 6-10: Membership Functions of Price in Membership Function Editor](image)

Figure 6-11 shows membership functions of load in Membership Function Editor. There are two membership functions for price: low and high. Both membership functions are set to be trapezoid type.
Figure 6-11: Membership Functions of Load in Membership Function Editor

Figure 6-12 shows membership functions of ESU in Membership Function Editor. There are six membership functions for ESU: releasing high, releasing medium, releasing low, charging low, charging medium and charging high. All membership functions are set to be triangle type.

Figure 6-13 shows six rules in Rule Editor of the proposed Mamdani’s FIS based model.

Figure 6-14 shows a snapshot of the rules which fire for a given input of price 37.1 $/Mwh and grid load of 1300 MWh. The output computed after defuzzification is 42.9 for proposed FIS.

Figure 6-15 shows the proposed Mamdani's FIS based model in Surface Viewer.
Figure 6-12: Membership Functions of ESU in Membership Function Editor

Figure 6-13: Rule Editor for proposed Mamdani's FIS based model
Figure 6-14: Rule Viewer for proposed Mamdani's FIS based model

Figure 6-15: Surface Viewer for proposed Mamdani's FIS based model
6.4. Development of GA Model in Matlab

The proposed GA model was developed based on Genetic Algorithm Toolbox of Matlab. Figure 6-16 shows the settings for simulation of the proposed GA model. There are 24 variables represented as x[n]. n ranges from 1 to 24. The interval of x[n] is [-1,1] in default setting which is called GA_I. Population size has been set to 1000. Simulation stops at generation 500.

Figure 6-16: Settings of the GA model in GA Toolbox
Creation function is set to be Uniform. Fitness scaling function is set to be Rank. Selection function is set to be stochastic uniform. Reproduction elite count is set to be 2. Mutation function is set to be Gaussian with scale 1.0 and shrink rate 1.0. Crossover function is set to be scattered. Migration function is set to be forward.

The fitness function $Y=f(x)$ is the daily cost and has been discussed in Section 5.2.4. The pseudo code of the fitness function of GA_I is given below.

```plaintext
function Y = fitnessfunc(x[n])
if x[n]<-1 or x[n]>1
    return a maximum value
else if constraint for GA is not met
    then return a maximum value
else Y = cost of consumption of appliances + cost for charging ESU
return Y
```

Line 1 defines the input of the fitness function. Line 2 and Line 3 enable the fitness function to drop the invalid $x[n]$ value away from the solution space. Line 4 and line 5 guarantees that the final minimum value is being met by the constraints of GA. Line 6 gives the total power cost as predicted by GA for household appliances and ESU. Line 7 returns the valid value and quit the simulation.

Figure 6-17 shows the results of simulation of GA_I for Jan 1, 2011. The upper part displays the best fitness after 500 generations of GA simulation which is 98.7. The lower
part displays chart of the best individuals associated with best fitness. The actual values of $x[n]$ are stored in an auto-generated array "ans" in Matlab. In command window the results are displayed. Figure 6-18 shows the results in command window.

Figure 6-17: Results of simulation of GA_I for Jan 1, 2011

Figure 6-18: Actual value of $x[n]$ from $x[1]$ to $x[24]$
The pseudo code of the fitness function of GA_II is as below:

```
function Y = fitnessfunc(x[n])
if x[n]<-1 or x[n]>1.5 then return a maximum value
else if constraint for GA is not met then return a maximum value
else Y = cost of consumption of appliances + cost for charging ESU
return Y
```

The difference from pseudo code of GA_I is line 2 which indicates larger capacity of ESU compared with GA_I. Figure 6-19 shows the results of simulation of GA_II for Jan 1, 2011.

![Graph](image)

Figure 6-19: Results of simulation of GA_II for Jan 1, 2011
The result shows a lower cost at 96.3 compared with 98.7 of GA_I, which suggests that larger capacity of ESU can result in better saving rate for consumers.
Chapter 7

Results Comparison and Performance Evaluation

In Chapter 7, comparison of results and performance evaluation will be discussed. In Section 7.1, statistics of household without proposed intelligent EMS are presented. In Section 7.2, performance of power consumption for Classic Boolean strategy is presented. Section 7.3 details the performance of power consumption for Mamdani’s FIS strategy. Section 7.4 illustrates the performance of power consumption for GA strategy. Section 7.5 compare and analyze the performance of all three strategies. All the results in this chapter were based on the test of 104 days during the year of 2011. To be more specific, every Monday and every Thursday were tested. Not every day of 2011 was tested due to high cost in simulation.

7.1 Household Power Consumption without Intelligent EMS

Figure 7-1 shows hourly power price curve of a typical day. The highest hourly price was at 18:00 PM. Figure 7-2 shows hourly grid load curve of a typical day. Hourly price were significantly affected by the hourly grid load. Therefore Figure 7-2 shows the highest grid load also occurred at 18:00 PM. Figure 7-3 shows the hourly cost of power
consumption curve of a typical day. The Equation 7-1 shows the connection among cost, price and load.

\[
\text{Cost} = \text{Price} \times \text{Load}
\]  

(7-1)

In figure 7.3, the lowest hourly cost was at 15:00, when people were usually at work. The highest hourly cost was at 18:00, when people have already come home and started household activities.

Figure 7-1: Power price curve of household on Jan/1/2011

Figure 7-2: Grid load curve of household on Jan/1/2011
Figure 7-3: Cost of power consumption curve of household on Jan/1/2011

Figure 7-4 shows monthly grid load of year 2011. The highest monthly load was in Aug when people have strong cooling need. The second highest monthly load occurred in Feb when people have strong heating needs. The lowest monthly load occurred in May, when the climate is often moderate and very few cooling or heating needs are required.

Figure 7-5 shows monthly cost of power consumption of year 2011. It is very clear that the trend indicated by monthly cost curve is the same compared with monthly grid load curve. The highest cost occurred in Aug and Feb, while the lowest cost occurred in Jan and May.
Figure 7-4: Monthly grid load of Year 2011

Figure 7-5: Monthly cost of power consumption of Year 2011
7.2 Power Consumption of Classic Boolean strategy

Figure 7-6 shows load curve of household that applied Classic Boolean strategy. In the 12 hours with lower power price of a day, the load of Classic Boolean Strategy was twice as the original load. In the remaining 12 hours with higher power price of a day, the load of Classic Boolean Strategy remained zero. Figure 7-7 shows hourly cost of Classic Boolean Strategy.

![Hourly Load Curve of Classic Boolean Strategy](image)

Figure 7-6: Load curve of Classic Boolean strategy on Jan/1/2011

Table 7.1 shows the quarterly saving rates of Classic Boolean strategy. It is shown that the saving rate of Classic Boolean strategy is from 16.9% to 31.8%. Figure 7-8 gives the quarterly cost of power consumption of Classic Boolean Strategy.
Figure 7-7: Hourly cost of Classic Boolean strategy on Jan/1/2011

Table 7.1: Quarterly saving rates of Classic Boolean

<table>
<thead>
<tr>
<th>Months</th>
<th>Original Cost ($)</th>
<th>Classic Boolean Cost ($)</th>
<th>Classic Boolean Saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan-Mar</td>
<td>371.1</td>
<td>457.65</td>
<td>19.9%</td>
</tr>
<tr>
<td>Apr-Jun</td>
<td>457.18</td>
<td>376.15</td>
<td>19.5%</td>
</tr>
<tr>
<td>Jul-Sep</td>
<td>682.91</td>
<td>470.15</td>
<td>29.1%</td>
</tr>
<tr>
<td>Oct-Dec</td>
<td>409.68</td>
<td>323.35</td>
<td>21.1%</td>
</tr>
</tbody>
</table>

Figure 7-8: Quarterly cost of Classic Boolean
7.3 Power Consumption of Mamdani’s FIS Strategy

Figure 7-9 shows hourly load curve of Mamdani’s FIS strategy in a typical day. Figure 7-10 shows hourly cost of power consumption of Mamdani’s FIS strategy in a typical day.

![Hourly Load Curve of Mamdani's FIS Strategy](image)

Figure 7-9: Load curve of Mamdani’s FIS on Jan/1/2011

![Hourly Cost Curve of Mamdani's FIS Strategy](image)

Figure 7-10: Hourly cost of Mamdani’s FIS on Jan/1/2011
Table 7.2 shows the quarterly saving rates of Mamdani's FIS strategy. The saving rates are from 10.0% to 31.3%. In first three quarters, the saving rate of Mamdani's FIS is close and even sometimes better than Classic Boolean. However, in the fourth quarter of 2011 the saving rate of Mamdani's FIS is 10.0%, which is much worse compared with 16.9% of Classic Boolean. The bad performance in that quarter is due to certain pattern of power consumption. Therefore, Mamdani's FIS based approach usually resulted in cost saving better than predicted theoretically by Classic Boolean approach. However, it had some glitches and in those days its performance did not outperform the Classic Boolean prediction. Figure 7-11 shows quarterly cost of Mamdani’s FIS strategy.

Table 7.2: Quarterly saving rates of Mamdani’s FIS

<table>
<thead>
<tr>
<th>Months</th>
<th>Original Cost ($)</th>
<th>Mamdani’s FIS Cost ($)</th>
<th>Mamdani’s FIS Saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan-Mar</td>
<td>571.1</td>
<td>483.85</td>
<td>15.3%</td>
</tr>
<tr>
<td>Apr-Jun</td>
<td>467.18</td>
<td>396</td>
<td>15.2%</td>
</tr>
<tr>
<td>Jul-Sep</td>
<td>582.91</td>
<td>463.2</td>
<td>30.1%</td>
</tr>
<tr>
<td>Oct-Dec</td>
<td>469.88</td>
<td>363.05</td>
<td>11.4%</td>
</tr>
</tbody>
</table>

Figure 7-11: Quarterly cost of Mamdani’s FIS
7.4 Power Consumption of GA Strategy

Figure 7-12 shows hourly load curves of GA_I, GA_II and GA_III. Figure 7-13 shows hourly cost curves of GA_I, GA_II and GA_III.

Figure 7-12: Load curve of GA on Jan/1/2011

Figure 7-13: Hourly cost of GA on Jan/1/2011
Table 7.3 shows quarterly saving rates of GA_I in the year of 2011, when range of x[n] is from -1 to 1. It is shown that the saving rate of GA_I strategy is from 18.1% to 25.7%. The saving rates of GA_I are very close but a little bit worse than the Classic Boolean strategy.

Table 7.3: Quarterly saving rates of GA_I

<table>
<thead>
<tr>
<th>Months</th>
<th>Original Cost ($)</th>
<th>GA_I Cost ($)</th>
<th>GA_I Saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan-Mar</td>
<td>371.1</td>
<td>467.61</td>
<td>18.1%</td>
</tr>
<tr>
<td>Apr-Jun</td>
<td>467.18</td>
<td>380.55</td>
<td>18.5%</td>
</tr>
<tr>
<td>Jul-Sep</td>
<td>662.91</td>
<td>488.24</td>
<td>26.3%</td>
</tr>
<tr>
<td>Oct-Dec</td>
<td>409.68</td>
<td>334.12</td>
<td>18.4%</td>
</tr>
</tbody>
</table>

Table 7.4 shows quarterly saving rates of GA_II in the year of 2011, when range of x[n] is from -1 to 1.5. It is shown that the saving rate of GA_II strategy is from 18.1% to 25.7%. From Table 7.4 it is shown that for quarter the GA_II is able to achieve the better result than Classic Boolean and GA_I. In addition, GA_II does not have certain glitches like Mamdani's FIS strategy.

Table 7.4: Quarterly saving rates of GA_II

<table>
<thead>
<tr>
<th>Months</th>
<th>Original Cost ($)</th>
<th>GA_II Cost ($)</th>
<th>GA_II Saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan-Mar</td>
<td>371.1</td>
<td>451.69</td>
<td>20.9%</td>
</tr>
<tr>
<td>Apr-Jun</td>
<td>467.18</td>
<td>375.56</td>
<td>19.6%</td>
</tr>
<tr>
<td>Jul-Sep</td>
<td>662.91</td>
<td>469.55</td>
<td>29.2%</td>
</tr>
<tr>
<td>Oct-Dec</td>
<td>409.68</td>
<td>319.33</td>
<td>22.1%</td>
</tr>
</tbody>
</table>

Table 7.5 shows quarterly saving rates of GA_III in the year of 2011, when range of x[n] is from -1 to 2. It is shown that the saving rate of GA_III strategy is from 19.6% to 28.5%. The performance of saving rates of GA_III is best among all three GA strategies. It indicates that in summer the GA strategy works the best compared with other three seasons.
Table 7.5: Quarterly saving rates of GA_III

<table>
<thead>
<tr>
<th>Months</th>
<th>Original Cost ($)</th>
<th>GA_III Cost ($)</th>
<th>GA_III Saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan-Mar</td>
<td>571.1</td>
<td>440.38</td>
<td>22.9%</td>
</tr>
<tr>
<td>Apr-Jun</td>
<td>467.18</td>
<td>371.14</td>
<td>20.6%</td>
</tr>
<tr>
<td>Jul-Sep</td>
<td>562.91</td>
<td>459.49</td>
<td>30.7%</td>
</tr>
<tr>
<td>Oct-Dec</td>
<td>409.68</td>
<td>311.96</td>
<td>23.9%</td>
</tr>
</tbody>
</table>

Figure 7-14 shows comparison of quarterly cost of GA_I, GA_II and GA_III. The GA_III gives better result than GA_II, while the GA_II gives better result than GA_I.

Therefore, it is concluded that the larger capacity of ESU can lead to better performance in cost saving for GA strategy.

Figure 7-14: Quarterly cost of GA_I, GA_II and GA_III
7.5 Comparison of Saving Cost of Three Strategies

Figure 7-15 shows the comparison of results of all three strategies. In the third quarter, the power price and grid load fluctuates greatly, all three strategies are capable of achieving better result of cost saving compared with other quarters.

Figure 7-15: Quarterly saving rates of three strategies

Mamdani's FIS based approach resulted in close cost saving compared with predicted theoretically by Classic Boolean approach. However, it had some glitches and in those days its performance did not come close to Classic Boolean prediction. GA_I
approach, when $x(n) \in [-1, 1]$, is very close to the predicted saving. GA_II approach, when
$x(n) \in [-1, 1.5]$, slightly outperformed the cost saving of Classic Boolean and GA_I for all
quarter of the data for year of 2011. GA_III approach, when $x(n) \in [-1, 2]$, is able to give
best results among all strategies and has no glitches at all.
Chapter 8

Conclusion

8.1 Summary and Conclusion of work

In this thesis an infrastructure of household intelligent EMS was proposed. Three approaches based on the infrastructure were developed and analyzed. Classic Boolean approach has proved to have ability for cost saving to some extent. Mamdani’s FIS approach shows better performance than Classic Boolean in certain circumstance but not uniformly. GA_III approach was the best and resulted in most cost saving during the days tested in the year of 2011.

8.2. Contribution to this work

The major contribution of this research is highlighted below.

A novel Energy Storage Unit has been proposed and has been integrated to household Energy Management System

In this thesis, smart meter has been designed using natural computing and has been demonstrated that this is a better strategy than the conventional boolean method for cost saving for end consumers.
8.3. Future work

Future work will focus on other heuristic algorithms on cost saving based on the proposed design, which may lead to best cost saving rates for end consumers and better performance in peak demand response for power suppliers.
References


[6] "Electric power transmission", Wikipedia, 


[9] Sandia National Laboratories, "Energy Storage Systems Program (ESS)."


Appendix A

Source Code of Genetic Algorithm Based Strategy

The program was developed on Matlab platform. The fitness function calculates the daily cost and the main function minimizes the fitness. The main function is based on the integrated Global Optimization Toolbox provided by Matlab platform.

Source Code of Fitness Function

function Y = ems_fitness_func(x)

    global M;
    global Original_Cost;
    global Simu_Hour;
    global ESU_Cap;

    Y = 0;

    %%%Caculate ESU Cost
    for i=1:Simu_Hour

        Y = M(i,4)*x(i)/100000+Y;

    end
%%Total Cost = ESU Cost + Appliances Cost
Y = Y + Original_Cost;

%%Dispose Invalid Inputs
for j = 1:Simu_Hour
    if (x(j)>ESU_Cap||x(j)<-1)
        Y=999999999;
        return
    end
end

%%Apply Constraints
T = 0
for i = 1:Simu_Hour
    T = M(i,3)*x(i)/10000+T;
end
if (T<0)
    Y = Y-99999999*T;
end
return

Source Code of Main Function

function [X,FVAL,REASON,OUTPUT,POPULATION,SCORES] =
my_fun_test_elite_GA
%% This is an auto generated M file to do optimization with the Genetic Algorithm and
%% Direct Search Toolbox. Use GAOPTIMSET for default GA options
%% structure. Patched By Cheng Yang
%% At the end of running, type "xlswrite('FileName.xls',ans)" to get
%% output to a xls file.

global M;
global Original_Cost;
global Simu_Hour;
global ESU_Cap;

% Setting Capacity of ESU, Default Value is 1
ESU_Cap = 1.0;
Simu_Hour = 24;
% Read Data From Excel File, Setting the Arrange to The Day of Simulation
M = xlsread('Data_2011.xls',1,'A2:E25');
% Caculate Original Cost
Original_Cost = 0;
for i=1:Simu_Hour
    Original_Cost = M(i,4)/100000+Original_Cost
end

%%Fitness function
fitnessFunction = @ems_fitness_func;

%%Number of Variables
nvars = 24;
%Start with default options
options = gaoptimset;
%Modify some parameters
options = gaoptimset(options,'PopInitRange',[-1;ESU_Cap]);
options = gaoptimset(options,'PopulationSize',2000);
options = gaoptimset(options,'StallGenLimit',5000);
options = gaoptimset(options,'StallTimeLimit',2000);
options = gaoptimset(options,'MigrationFraction',0.3);
options = gaoptimset(options,'Generations',500);
options = gaoptimset(options,'MutationFcn',{ @mutationgaussian 0.1 1 });
options = gaoptimset(options,'Display','off');
options = gaoptimset(options,'PlotFcns',{ @gaplotbestf @gaplotbestindiv });

%%Run GA
[X,FVAL,REASON,OUTPUT,POPULATION,SCORES] =
ga(fitnessFunction,nvars,options);

Source Code of Genetic Algorithm Function

function [x,fval,exitFlag,output,population,scores] = ga(FUN,genomeLength,options)
%GA    Genetic algorithm solver.
%   X = GA(FITNESSFCN,NVARS) finds the minimum of FITNESSFCN using
%   GA. NVARS is the dimension (number of design variables) of the
%   FITNESSFCN. FITNESSFCN accepts a vector X of size 1-by-NVARS,
%   and returns a scalar evaluated at X.
%
%   X = GA(FITNESSFCN,NVARS,OPTIONS) finds the minimum for
%   FITNESSFCN with the default optimization parameters replaced by values
%   in the structure OPTIONS. OPTIONS can be created with the GAOPTIMSET
%   function.
%
%   X = GA(PROBLEM) finds the minimum for PROBLEM. PROBLEM is a structure
%   that has the following fields:
%   fitnessfcn: <Fitness Function>
%   nvars: <Number of design variables>
%   options: <Options structure created with GAOPTIMSET>
randstate: <Optional field to reset rand state>

randnstate: <Optional field to reset randn state>

[X, FVAL] = GA(FITNESSFCN, ...) returns FVAL, the value of the fitness function FITNESSFCN at the solution X.

[X, FVAL, REASON] = GA(FITNESSFCN, ...) returns the REASON for stopping.

[X, FVAL, REASON, OUTPUT] = GA(FITNESSFCN, ...) returns a structure OUTPUT with the following information:

randstate: <State of the function RAND used before GA started>

randnstate: <State of the function RANDN used before GA started>

generations: <Total generations, excluding HybridFcn iterations>

funccount: <Total function evaluations>

message: <GA termination message>

[X, FVAL, REASON, OUTPUT, POPULATION] = GA(FITNESSFCN, ...) returns the final POPULATION at termination.

[X, FVAL, REASON, OUTPUT, POPULATION, SCORES] = GA(FITNESSFCN, ...) returns the SCORES of the final POPULATION.

There are several steps to the GA:

population generation

scoring

loop

fitness

scaling

selection
% crossover
% mutation
% scoring
% migration
% output
% termination testing
% end loop
%
% Each of these steps can be controlled by the options structure created
% by GAOPTIMSET.
%
% Example:
% Minimize 'rastrignsfcn' fitness function of numberOfVariables = 2
% x = ga(@rastrignsfcn,2)
% Display plotting functions while GA minimizes
% options = gaoptimset('PlotFcns',
%                    {@gaplotbestf,@gaplotbestindiv,@gaplotexpectation,@gaplotstopping});
% [x,fval,reason,output] = ga(@rastrignsfcn,2,options)
%
% See also GAOPTIMSET, FITNESSFUNCTION, PATTERNSEARCH, @.

% Copyright 2004 The MathWorks, Inc.
% $Revision: 1.28.4.1 $ $Date: 2004/03/09 16:15:33 $

% If the first arg is not a gaoptimset, then it's a fitness function followed by a genome
% length. Here we make a gaoptimset from the args.
defaultopt = struct('PopulationType', 'doubleVector', ...
                    'PopInitRange', [0;1], ...
                    'PopulationSize', 20, ...
                    'EliteCount', 2, ...
                    'CrossoverFraction', 0.8, ...
                    'MigrationDirection','forward', ...}
'MigrationInterval',20, ...
'MigrationFraction',0.2, ...
'Generations', 100, ...
'TimeLimit', inf, ...
'FitnessLimit', -inf, ...
'StallGenLimit', 50, ...
'StallTimeLimit', 20, ...
'InitialPopulation',[], ...
'InitialScores', [], ...
'PlotInterval',1, ...
'FitnessScalingFcn', @fitscalingrank, ...
'SelectionFcn', @selectionstochunif, ...
'CrossoverFcn', @crossoverscattered, ...
'MutationFcn', @mutationgaussian, ...
'PlotFcns', [], ...
'Display', 'final', ...
'OutputFcns', [], ...
'CreationFcn', @gacreationuniform, ...
'HybridFcn', [], ...
'Vectorized','off');

msg = nargchk(1,3,nargin);
if ~isempty(msg)
    error('gads:GA:numberOfInputs',msg);
end

% If just 'defaults' passed in, return the default options in X
if checkinputs && nargin == 1 && nargout <= 1 && isequal(FUN,'defaults')
    x = defaultopt;
    return
end
%if 3 arg, options is passed
if nargin>=3
    if ~isempty(options) && ~isa(options,'struct')
        error('gads:GA:thirdInputNotStruct','Invalid Input for GA.);
    else
        FitnessFcn = FUN;
        GenomeLength = genomeLength;
    end
    if isempty(options)
        options = gaoptim;
    end
end
end

%If 2 args, use default options for GA
if nargin==2
    options = gaoptimset;
    FitnessFcn = FUN;
    GenomeLength = genomeLength;
end
if nargin == 1
    try
        options = FUN.options;
        GenomeLength = FUN.nvars;
        FitnessFcn = FUN.fitnessfcn;
        if isfield(FUN, 'randstate') && isfield(FUN, 'randnstate') && ...
            isa(FUN.randstate, 'double') && isequal(size(FUN.randstate),[35, 1]) && ... 
            isa(FUN.randnstate, 'double') && isequal(size(FUN.randnstate),[2, 1])
            rand('state',FUN.randstate);
            randn('state',FUN.randnstate);
        end
    end
catch
    error('gads:GA:invalidStructInput',lasterr);
end
end

x =[];fval =[];exitFlag=";population=[];scores=[];user_options = options;
%Remember the random number states used
output.randstate  = rand('state');
output.randnstate = randn('state');
output.generations = 0;
output.funccount   = 0;
output.message   = ";

%Validate arguments
[GenomeLength,FitnessFcn,options] = validate(GenomeLength,FitnessFcn,options);
%Create initial state: population, scores, status data
state = makeState(GenomeLength,FitnessFcn,options);
%Give the plot/output Fcns a chance to do any initialization they need.
state = gaplot(FitnessFcn,options,state,'init');
[state,options] = gaoutput(FitnessFcn,options,state,'init');

%Print some diagnostic information if asked for
if strcmpi(options.Display,'diagnose')
    gadiagnose(FUN,GenomeLength,user_options);
end
%Setup display header
if any(strcmpi(options.Display, {'iter','diagnose'}))
    fprintf("n       Best       Mean       Stall\n");
    fprintf('Generation f-count f(x)     f(x)     Generations\n');
end

exitFlag = ";
while isempty(exitFlag)
    state.Generation = state.Generation + 1;
    offset = 0;
    totalPop = options.PopulationSize;
    for pop = 1:length(totalPop)
        populationSize = totalPop(pop);
        thisPopulation = offset:(offset + populationSize - 1));
        population = state.Population(thisPopulation,:);
        score = state.Score(thisPopulation);
        [score,population,state] =
        stepGA(score,population,options,state,GenomeLength,FitnessFcn);
        state.Population(thisPopulation,:) = population;
        state.Score(thisPopulation) = score;
        offset = offset + populationSize;
    end

    scores = state.Score;
    best = min(state.Score);
    generation = state.Generation;
    state.Best(generation) = best;
    if((generation > 1) && finite(best))
        if(state.Best(generation-1) > best)
state.LastImprovement = generation;
state.LastImprovementTime = cputime;
end
end

% do any migration
state = migrate(FitnessFcn,GenomeLength,options,state);
% update the Output
state = gaplot(FitnessFcn,options,state,'iter');
[state,options] = gaoutput(FitnessFcn,options,state,'iter');

% check to see if any stopping criteria have been met
exitFlag = isItTimeToStop(options,state);
end %End while loop

% find and return the best solution
[fval,best] = min(state.Score);
x = state.Population(best,:);

%Update output structure
output.generations = state.Generation;
output.message = exitFlag;
output.funccount = state.Generation*length(state.Score);

% load up outputs
if(nargout > 4)
    population = state.Population;
    if(nargout > 5)
        scores = state.Score;
    end
end
end
% give the Output functions a chance to finish up
gaplot(FitnessFcn,options,state,'done');
gaoutput(FitnessFcn,options,state,'done');

%A hybrid scheme. Try another minimization method if there is one.
if(strcmpi(options.PopulationType,'doubleVector') && ~isempty(options.HybridFcn))
    % Who is the hybrid function
    if isa(options.HybridFcn,'function_handle')
        hfunc = func2str(options.HybridFcn);
    else
        hfunc = options.HybridFcn;
    end

    % Inform about hybrid scheme
    if any(strcmpi(options.Display, {'iter','diagnose','final'}))
        fprintf('%s%s
','Switching to the hybrid optimization algorithm
(upper(hfunc),.');
    end

% Determine which syntax to call
switch hfunc
    case 'fminsearch'
        if isempty(options.HybridFcnArgs)
            [xx,ff,e,o] = feval(options.HybridFcn,FitnessFcn,x,[],options.FitnessFcnArgs{:});
        else % cell expansion {:} will change the number of input args and hence we need two syntax.
            [xx,ff,e,o] = feval(options.HybridFcn,FitnessFcn,x,options.HybridFcnArgs{:},options.FitnessFcnArgs{:});
        end
output.funccount = output.funccount + o.funcCount;
output.message = [output.message sprintf('nFMINSEARCH:
'), o.message];
case 'patternsearch'
    [xx,ff,e,o] = feval(options.HybridFcn,{FitnessFcn,options.FitnessFcnArgs{:}},x,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[], [],