

1998

# Profiting from competition: financial tools for electric generation companies

Charles William Richter Jr.  
*Iowa State University*

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Artificial Intelligence and Robotics Commons](#), [Finance Commons](#), [Finance and Financial Management Commons](#), [Oil, Gas, and Energy Commons](#), and the [Operational Research Commons](#)

## Recommended Citation

Richter, Charles William Jr., "Profiting from competition: financial tools for electric generation companies" (1998). *Retrospective Theses and Dissertations*. 12520.  
<https://lib.dr.iastate.edu/rtd/12520>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

## INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

**The quality of this reproduction is dependent upon the quality of the copy submitted.** Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

# UMI

A Bell & Howell Information Company  
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA  
313/761-4700 800/521-0600



**Profiting from competition: Financial tools for electric generation companies**

**by**

**Charles William Richter, Jr.**

**A dissertation submitted to the graduate faculty  
in partial fulfillment of the requirements for the degree of  
DOCTOR OF PHILOSOPHY**

**Major: Electrical Engineering (Electric Power)**

**Major Professor: Gerald B. Sheblé**

**Iowa State University**

**Ames, Iowa**

**1998**

**Copyright © Charles William Richter, Jr., 1998. All rights reserved.**

UMI Number: 9911636

Copyright 1998 by  
Richter, Charles William, Jr.

All rights reserved.

---

UMI Microform 9911636  
Copyright 1999, by UMI Company. All rights reserved.

This microform edition is protected against unauthorized  
copying under Title 17, United States Code.

---

**UMI**  
300 North Zeeb Road  
Ann Arbor, MI 48103

**Graduate College**  
**Iowa State University**

**This is to certify that the Doctoral dissertation of**  
  
**Charles William Richter, Jr.**  
  
**has met the dissertation requirements of Iowa State University**

Signature was redacted for privacy.

**Major Professor**

Signature was redacted for privacy.

**For the Major Program**

Signature was redacted for privacy.

**For the Graduate College**

This dissertation is dedicated to my wonderful family (Hi Mom!)

who had to live with me, in spite of their best efforts to hide.

I guess there was simply nowhere to hide. Just kidding!

They were there to support and encourage me, to argue with me, to comfort me,

and to teach me the value of hard work, love, friendship, and education.

## TABLE OF CONTENTS

<b>LIST OF FIGURES.....</b>	<b>VII</b>
<b>LIST OF TABLES .....</b>	<b>x</b>
<b>ACKNOWLEDGMENTS.....</b>	<b>xi</b>
<b>ABSTRACT.....</b>	<b>xii</b>
<b>1 INTRODUCTION.....</b>	<b>1</b>
1.1 History of the electric industry .....	1
1.2 The shift toward deregulation .....	2
1.3 Choosing a competitive framework.....	3
1.4 Preparing the utility of the future for competition .....	4
1.5 Contents of this dissertation.....	5
<b>2 BASIC CONCEPTS, METHODS AND ASSUMPTIONS.....</b>	<b>6</b>
2.1 Chapter summary .....	6
2.2 Developing a coherent bidding strategy .....	6
2.3 Genetic algorithms (GAs).....	9
2.3.1 Parent selection.....	11
2.3.1.1 Roulette selection .....	11
2.3.1.2 Rank selection.....	12
2.3.1.3 Tournament selection.....	12
2.3.2 Crossover.....	13
2.3.3 Mutation.....	13
2.3.4 Fitness evaluation .....	14
2.3.5 A GA example: The string evolver.....	14
2.4 Market structure.....	15
2.4.1 Using the markets to manage risk .....	19
2.4.1.1 Futures contracts.....	20
2.4.1.2 Futures options contracts.....	20
2.4.2 Auctions .....	21
2.5 Limitations of the model.....	23
<b>3 EVOLVING BIDDING STRATEGIES WITH GENETIC ALGORITHMS AND GP-AUTOMATA</b>	<b>24</b>
3.1 Chapter overview.....	24
3.2 Introduction.....	24
3.3 Review of recent work .....	25
3.4 Evolving non-adaptive bidding strategies with a genetic algorithm.....	27
3.4.1 The marketplace .....	27
3.4.2 The basics of genetic algorithms .....	27
3.4.3 Using genetic algorithms for trading agent representation and evolution.....	27
3.4.4. Developing the GENCO offer .....	29
3.4.5 Results.....	31
3.4.6 Conclusions of non-adaptive strategy evolution.....	32
3.5 Adaptive bidding strategies with genetic programming and finite state automata .....	35
3.5.1 The basics of genetic programming .....	35

3.5.2	The basics of GP-Automata.....	37
3.5.3	Using the strategies in an auction.....	38
3.5.4	Evolving and testing the strategies: A first attempt.....	39
3.5.5	An experiment to test the effect of GP-Automata tree size and state number.....	43
3.6	Chapter conclusions and future research.....	51
<b>4</b>	<b>FUZZY LOGIC IN THE COMPETITIVE ENVIRONMENT.....</b>	<b>52</b>
4.1	Chapter overview.....	52
4.2	Using fuzzy logic to build bidding strategies that handle uncertainty.....	52
4.2.1	Introduction.....	52
4.2.2	The marketplace.....	53
4.2.3	Evolving bidding strategies with genetic algorithms.....	53
4.2.4	Fuzziness and uncertainty.....	54
4.2.5	Building bidding strategies.....	57
4.2.5.1	Generating the fuzzy rule-sets manually.....	58
4.2.5.2	The search for the "optimal" fuzzy rule-set.....	58
4.2.5.3	Using a GA to evolve fuzzy rules for bidding.....	59
4.2.5.4	Using a GA to extract expert-system bidding rules from a historical database.....	59
4.2.6	Fuzzifying decision analysis.....	59
4.3	Automatic generation control with a fuzzy logic controller.....	61
4.3.1	Introduction.....	61
4.3.2	Automatic generation control.....	62
4.3.3	Fuzzy logic control (FLC).....	64
4.3.4	Fuzzy AGC.....	65
4.3.5	Results of the FLC AGC.....	67
4.3.6	Extensions.....	70
4.4	Chapter summary and future research.....	72
<b>5</b>	<b>PROFIT-BASED UNIT COMMITMENT FOR THE COMPETITIVE ENVIRONMENT.....</b>	<b>74</b>
5.1	Chapter overview.....	74
5.2	Introduction.....	74
5.3	Updating unit commitment.....	75
5.4	Genetic-Based UC algorithm.....	81
5.4.1	The basics of genetic algorithms.....	81
5.4.2	GA for price-based UC.....	82
5.5	Price-Based UC-GA results.....	84
5.6	UC and bidding strategies.....	88
5.7	Handling uncertainty in price and load forecasts.....	89
5.8	Conclusions and future research.....	90
<b>6</b>	<b>IMPROVING BIDDING STRATEGIES THROUGH INTELLIGENT DATA MINING.....</b>	<b>92</b>
6.1	Chapter overview.....	92
6.2	Introduction.....	92
6.3	Applications of expert systems.....	93
6.4	The basics of machine learning.....	94
6.5	Genetic algorithms.....	97
6.6	Expert systems.....	97
6.7	Using GAs to evolve expert system rules.....	98
6.7.1	Data storage and preprocessing.....	98
6.7.2	A data structure to encode the solutions.....	99
6.7.3	Calculating the fitness.....	100
6.7.4	Creating new solutions through evolution.....	101

6.8 Results.....	103
6.8.1 An example .....	104
6.9 Conclusion and future research.....	106
<b>7 CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH .....</b>	<b>108</b>
7.1 Commentary.....	108
7.2 Future market framework and assumptions.....	108
7.3 Evolving bidding strategies.....	109
7.4 Fuzziness in the competitive environment .....	109
7.5 Unit commitment .....	110
7.6 Inferring rules by intelligently mining a database .....	110
7.7 Additional suggestions for future research .....	111
7.7.1 Evolving multi-market electric commodity portfolios and bidding strategies.....	111
7.7.2 Forecasting electric prices and demand with uncertainty .....	111
<b>REFERENCES.....</b>	<b>112</b>

## LIST OF FIGURES

Figure 2.1 An overview of bidding strategy development.....	7
Figure 2.2 Determining competitor models.....	7
Figure 2.3 Using the competitor models in the auction markets.....	8
Figure 2.4 Determining the single period aggregate supply curve.....	8
Figure 2.5 Determining the forward demand and price curves.....	9
Figure 2.6 Using unit commitment as an input to bid generator.....	10
Figure 2.7 A simple genetic algorithm.....	11
Figure 2.8 Roulette and rank probability wheels.....	12
Figure 2.9 Single point crossover.....	13
Figure 2.10 GA mutation.....	14
Figure 2.11 The string evolver example.....	15
Figure 2.12 Brokerage system model.....	16
Figure 2.13 Interconnection between the markets.....	17
Figure 2.14 Methods of using options.....	18
Figure 2.15 Different risks with the same expected profit.....	19
Figure 2.16 Hedging with futures and futures options contracts.....	20
Figure 2.17 The auction process.....	22
Figure 3.1 A population of agents/bidding strategies.....	28
Figure 3.2 Developing the offer from the data structure.....	29
Figure 3.3 The GA agent evolution process.....	30
Figure 3.4 Bid multipliers of the best agent at the first and last generation.....	33
Figure 3.5 MWs sold by best agent each generation.....	33
Figure 3.6 Fitness and trading results.....	34
Figure 3.7 Sample GPs.....	36
Figure 3.8 A four state GP-Automaton.....	37
Figure 3.9 Case 1: Competitor buy bids at \$15/MWhr and sell offers at \$5/MWhr.....	40
Figure 3.10 Case 2: Competitor bids distributed as shown.....	41
Figure 3.11 Case 3. Competitors' bids/offers moving toward together over cycles of the auction.....	42
Figure 3.12 Results for the control case.....	45
Figure 3.13 An individual run selected at random from the control case.....	46
Figure 3.14 Case 2, averaged across 40 runs. GENCOs are limited to one node GPs.....	47

Figure 3.15 An individual run selected at random from the one node GP GENCO case (case 2).....	48
Figure 3.16 Case 3 averaged across 40 runs. GENCOs are limited to having only one state.....	49
Figure 3.17 An individual run selected at random from the single state GENCO case (case 3). .....	50
Figure 4.1 Fuzzy membership functions.....	55
Figure 4.2 Three input fuzzy rule set to determine bid.....	56
Figure 4.3 Fuzzy system architecture.....	57
Figure 4.4 Using the rule-set.....	58
Figure 4.5 Evaluating the alternatives with decision analysis.....	60
Figure 4.6 Traditional Automatic Generation Control Block Diagram [Wood and Wollenberg, 1996].....	63
Figure 4.7 Fuzzy input membership functions.....	64
Figure 4.8 Using two inputs to determine a fuzzy output.....	65
Figure 4.9 FLC block diagram.....	66
Figure 4.10 FLC Block Diagram.....	66
Figure 4.11 ACE ranges and associated actions and conditions.....	67
Figure 4.12 Fuzzy rules for the AGC controller.....	68
Figure 4.13 Ranges defining membership functions of area 1 controller.....	69
Figure 4.14 Ranges defining membership functions of area 2 controller.....	70
Figure 4.15 Area 1 Frequency and Ptie Deviation.....	71
Figure 4.16 Area 1 and Area 2 Control Error.....	71
Figure 4.17 Accumulated Time and Interchange Errors.....	72
Figure 5.1 The market as an additional generator and an additional load.....	77
Figure 5.2 The UC problem.....	79
Figure 5.3 Defining a pseudo $\lambda$ .....	80
Figure 5.4 Example of fixed and transition costs allocation.....	81
Figure 5.5 GA UC block diagram.....	82
Figure 5.6 A population of UC schedules.....	83
Figure 5.7 Two point crossover on UC schedules.....	84
Figure 5.8 Data for 10 unit, 48 hour case.....	86
Figure 5.9 Max., min., and avg. fitness vs. GA generations for the 2 generator, 14 hour case.....	88
Figure 5.10 Expected prices with error bars.....	90
Figure 6.1 A general learning structure.....	95
Figure 6.2 A sample decision tree for patient treatment or auction bidding.....	96
Figure 6.3 The GA “intelligently” searches through the database for rules.....	98
Figure 6.4 Database information.....	99
Figure 6.5 Sample GA population.....	100

Figure 6.6 The GA searches through the database for good rules .....	102
Figure 6.7 Five bit "active" input data .....	104
Figure 6.8 Single bit input data .....	105
Figure 6.9 Sample output data.....	105
Figure 6.10 Data removed by the compression function.....	106

**LIST OF TABLES**

Table 2.1 Long call example .....	19
Table 2.2 Example of auction bid matching.....	21
Table 3.1 Valid operators and terminals for the GP.....	36
Table 3.2 Assignment of buyers and sellers to an auction.....	44
Table 4.1 Parameter settings for plant simulation.....	67
Table 5.1 Forecasted demand and prices (2 generator case).....	85
Table 5.2 Unit data for 2 generator case.....	85
Table 5.3 GA control parameters.....	86
Table 5.4 Comparing UC-GA with exhaustive search.....	87
Table 5.5 The best UC-GA schedules of the population.....	87
Table 6.1 Adjustable parameters used in the program.....	101

## ACKNOWLEDGMENTS

I would like to acknowledge the guidance and support of my major professor and mentor, Dr. Gerald B. Sheblé, a insightful professor who always knew where the industry was heading, and what research should have top priority. I am deeply grateful to friend and committee member Dr. Dan Ashlock for providing creative input on various projects (the strafing-run project comes to mind) during my time as a researcher at Iowa State University. I would like to thank the other members of my committee, Dr. Vijay Vittal, Dr. Jennifer Davidson, and Dr. David Hennessy, for all of their time and efforts on my behalf.

I thank the National Science Foundation and members of the power affiliates program whose contributions help make it possible for the graduate students in the power program at ISU to focus on research twenty-four hours per day. Without financial assistance, we graduate students would have to spend more of our time foraging for food and building shelter.

Last and certainly not least, I would also like to acknowledge the support of several graduate students who provided me with their time and input. I thank Jayant, Giat, Gilsoo, Madhura, Mona, Qinghua, Sundar, Vincent, and many others for everything they did for me. You were there with me in the trenches and know that I couldn't have done it without you. (Wait a minute... I remember you telling me that I couldn't do it even with your help... Just kidding!) I hope I can continue to count on your support wherever life's roads may lead.

## ABSTRACT

Regulations governing the operation of electric power systems in North America and many other areas of the world are undergoing major changes designed to promote competition. This process of change is often referred to as deregulation. Participants in deregulated electricity systems may find that their profits will greatly benefit from the implementation of successful bidding strategies. While the goal of the regulators may be to create rules which balance reliable power system operation with maximization of the total benefit to society, the goal of generation companies is to maximize their profit, i.e., return to their shareholders. The majority of the research described here is conducted from the point of view of generation companies (GENCOs) wishing to maximize their expected utility function, which is generally comprised of expected profit and risk. Strategies that help a GENCO to maximize its objective function must consider the impact of (and aid in making) operating decisions that may occur within a few seconds to multiple years.

The work described here assumes an environment in which energy service companies (ESCOs) buy and GENCOs sell power via double auctions in regional commodity exchanges. Power is transported on wires owned by transmission companies (TRANSCOs) and distribution companies (DISTCOs). The proposed market framework [Kumar and Sheblé, 1996a] allows participants to trade electrical energy contracts via the spot, futures, options, planning, and swap markets.

An important method of studying these proposed markets and the behavior of participating agents is the field of experimental/computational economics. For much of the research reported here, the market simulator developed by Kumar and Sheblé and similar simulators has been adapted to allow computerized agents to trade energy. Creating computerized agents that can react as rationally or irrationally as a human trader is a difficult problem for which we have turned to the field of artificial intelligence. Some of our work uses GP-Automata, a technique which combines genetic programming and finite state machines, to represent adaptive agents. We use a genetic algorithm to evolve these adaptive agents (each with its own bidding strategy) for use in a double auction. The agent's strategies may be judged by the amount of profit they produce and are tested by computerized agents repeatedly buying and selling electricity in an auction simulator. In addition to the obvious profit-maximization strategies, one can also design strategies which exhibit other types of trading behaviors. The resulting strategies can be used directly in on-line trading, or as realistic models of competitors in a trading simulator.

In addition to developing double auction bidding strategies, we investigate and discuss methods of minimizing an energy trader's risk. This can be done using such financial vehicles as futures and options contracts or through the inclusion of risk while judging strategies used in the market simulations described above. We discuss the role of fuzzy logic in the competitive electric marketplace, including how it can be

applied in developing bidding strategies. Since competition promises to drive the power system closer to its operating limits, improvements in measurement and system control will be important. We provide an example of using fuzzy logic to do automatic generation control and discuss extensions that would make it superior to traditional controllers. Since the GENCO's forte is primarily generating electricity, we examine unit commitment and discuss how to update it for the competitive environment. We discuss the role of unit commitment in developing bidding strategies, as well as, the role of bidding strategies in solving the unit commitment problem. Depending on the market structure adopted by a particular location, large amounts of bidding data may be available to regulators or market participants. Ideally, regulators could use this data to verify that the market is efficient. Market participants with access to this data might gain an advantage over their competitors if they could somehow determine their competitor's bidding strategy. We outline methods of automatically inferring other participants' trading rules based on historical data. Much of the work described here should aid in the design of effective operating procedures, trading strategies and profitable portfolios for energy producers.

# 1 INTRODUCTION

## 1.1 History of the electric industry

The electric utility industry which most of us take for granted began over one hundred years ago with the electrical pioneers of the late 1800s. The electric light bulb had just been invented, but wasn't going to be a big hit until people had a place to plug it in. According to a famous humorist [Barry, 1985, p 223]:

The greatest electrical pioneer of them all was Thomas Edison, who was a brilliant inventor despite the fact that he had little formal education and lived in New Jersey. ...Edison's greatest achievement came in 1879 when he invented the electric company. His design was a brilliant adaptation of the simple electrical circuit. The electric company sends electricity through a wire to a customer, then immediately gets the electricity back through another wire, then, (this is the brilliant part) sends it right back to the customer again. This means that an electric company can sell a customer the same batch of electricity thousands of times a day and never get caught, since very few customers take the time to examine their electricity closely. In fact, the last year any new electricity was generated was 1937. The electric companies have been merely reselling it ever since, which is why they have so much time to apply for rate increases.

Most people in the electric utility industry would agree that it is more complex than Dave Barry makes it out to be. Hundreds of brilliant people have been working throughout the past century to make our power system what it is today. Having worked in a electric generating plant, I can say that at least some electricity has been generated after 1937. The days when electric companies had so much time to apply for rate increases are gone, and the forces of supply and demand will be setting US electric rates in the very near future.

For decades, electric consumers in the US had only their local vertically integrated utility as a source of electricity. In exchange for the guarantee to be the only electricity provider within a given service territory, the electric utility had the obligation to serve everyone within its territory. The electric utility was sole producer, transporter, and distributor of electric energy to the customer. The electricity rates it decided to charge were subject to a review by regulatory bodies (e.g., a Public Utilities Commission) to prevent price gouging. The Public Utilities Commission (PUC) allowed the rates to cover the utility's cost, plus a respectable return on investment. This system of vertically integrated monopolies evolved to prevent expensive duplication of transmission and distribution by competing companies. Economies of scale meant that a single large power plant operated by the monopoly utility could produce electricity more efficiently than two smaller power plants operated by competing utilities. The efficiencies gained from economies of scale outweighed the deadweight losses associated with monopoly operation and system over-building. For most of the 20th century, the electric industry was viewed as a natural monopoly.

## 1.2 The shift toward deregulation

The high energy prices during the 1970's oil embargo focused attentions on further efficiencies that might come with competition. The country began to take small steps toward a competitive electricity system. Independent power producers were granted the right to produce and sell power to the local electric utility at a price which represented costs avoided by the utility associated with not having to produce an equivalent amount of electrical energy at utility-owned plants. While, for the most part, the electric energy industry was still hanging on to its monopolistic structure, other industries including the natural gas, airline, and communication industries in the US were being deregulated. Countries throughout the world in which the government owned many industries began to see the benefits of privatization.

Attention on monopolistic inefficiencies due to high fuel prices was probably not the only reason deregulation began to happen in the late 20th century. It may be that the countries around the world needed to wait until it was mature enough to meet the prerequisites to competition. A recent article [Clayton and Mukerji, 1996] lists the following intuitive prerequisites to competition without which it would be difficult to implement competition successfully:

- Mature physical system
- Stable national economy
- Trust in the sanctity of contracts
- Relatively high and/or diverse prices
- Regulated market imperfection

The regulators decided it was time for increased competition in the US electrical system. Increasing competition via re-regulation of the electrical system, they would increase power system efficiencies and see benefits for electric consumers. With the passage of the Energy Policy Act in 1992, entities that did not own transmission-lines were granted the right to use the transmission system. This was termed *open access* and US electric utilities began to see limited competition in power production. Countries outside of North America (e.g., United Kingdom, Norway, Chile) had already changed, or were in the process of changing their regulations, (commonly referred to as deregulating), to allow a more competitive electric marketplace. The Federal Energy Regulatory Commission (FERC), in various Notices of Proposed Regulation (NOPRs), announced its intent to expand competition in the US electric marketplace. Attitudes toward these changes in regulation still vary from region to region. Many electric utilities in the US have been reluctant to change from the status quo. Regulating bodies and consumers in regions with high electrical rates have more receptive to change. In 1998, California became the first state in the union to adopt a competitive structure, and utilities in other states are observing the outcome. As California's markets continue to evolve, electric utilities in other areas of the country continue to form their opinions on what market and operational structures would suit them best.

### **1.3 Choosing a competitive framework**

There are many electricity market frameworks that could be used to introduce competition to the electric utilities of the US. Almost every country embracing competitive markets for their electric system has done so in a different manner. The research described here assumes an electric marketplace derived from commodities exchanges like the Chicago Mercantile Exchange, Chicago Board of Trade, and New York Mercantile Exchange (NYMEX) where commodities (other than electricity) have been traded for many years. The fact that in 1996, NYMEX actually added electricity futures to their offerings supports the predictions [Kumar and Sheblé, 1996a; Richter and Sheblé, 1997a; Sheblé, 1996; Sheblé, 1994b] regarding the framework of the coming competitive environment. The framework we are assuming has some similarities to the Norwegian electric system. The details of our framework and some of its major differences from the emerging power markets/pools will be described in detail later.

Many believe the ultimate competitive electric industry environment is one in which retail consumers have the ability to choose their own electric supplier. Often referred to as retail access, this is quite a contrast to the vertical integrated monopoly of the past which served the typical electricity consumer as it saw fit. Soon, telemarketers will be contacting consumers asking to speak to the person in charge of making decisions about the electricity service. Depending on consumer preference and the installed technology it may be possible to do this on an almost real-time basis. Real-time pricing, where electricity is priced as it is used, is getting closer to being a reality as information technology advances. However, at present, customers in most regions of the US lack the sophisticated metering equipment necessary to implement retail access at this level. Whether or not retail access with real-time pricing becomes a reality in the near future may have big implications for energy service companies (ESCOs). With an ESCO or distribution company (DISTCO) standing between the consumers and the GENCO, the implementation of retail access is less important from the perspective of the electric generation company (GENCO).

Charging rates that were deemed fair by the public utilities commission, the average monopolistic electric utility of the old environment met all consumer demand while attempting to minimize their costs. In times of need or disaster, neighboring utilities might cooperate without charging for their assistance. The costs were passed on to the rate payers. The electric companies in a region were all members of one big happy family. The GENCOs and DISTCOs of the future competitive environment will also be happy to help out in times of disaster, but each dollar spent will be accounted for. No longer guaranteed a rate of return, buyers and sellers participating in the "new and improved" competitive electric utility industry must be profit driven. The competitive GENCO will attempt to maximize its profit.

#### 1.4 Preparing the utility of the future for competition

In the summer of 1998 electric energy prices in the midwest rose to more than \$4500/MWhr due to a combination of high temperatures and several units on forced outage for various reasons. Many midwestern electric utilities bought energy at that high price, and then sold it to consumers for the normal rate. Unless these companies thought they were going to be fined heavily, or lose all customers for a very long time, it may have behooved them to think twice before doing such a thing.

Under highly competitive scenarios, the successful GENCO will recover its incremental costs as well as the fixed costs through the prices it charges. For a short time, producers may sell below their costs, but will need to make up the losses during another time. Economic theory says that eventually, under perfect competition, ESCOs and GENCOs will arrive at a point where their economic profit is zero. It should be noted that economic profit is different from accounting profit, which is used commonly in business. This point of zero economic profit is the point at which all companies surviving participants can break even. Supply exactly equals demand. At that point, the best the producer can do in an auction, ignoring fixed costs, is to bid his incremental cost. Since the fixed and transition costs associated with generating electricity are quite large, ignoring fixed costs is something that we in the electrical industry are not able to do reasonably. In contrast to the assumptions of some theoretical economic research, perfect competition is not often found in the real world, and there are things that the competitive producer can do to increase the odds of surviving and remaining profitable.

The operational tools used and decisions made by GENCOs operating in a competitive environment are dependent on the structure and rules of the power system operation. In each of the various market structures, the GENCO's goal is to maximize his profit. Entities such as an independent system operator (ISO), or a national grid operator, are responsible for ensuring that the system operates in a secure manner. The rules of operation themselves should be designed by regulators prior to implementation to be fair. Fairness depends on what the policy makers consider fair. It could call for maximization of social welfare (i.e., maximize the total happiness of everyone) or perhaps maximization of consumer surplus (i.e., make the customers happy).

Changing regulations are affecting the GENCO's way of doing business and to remain profitable GENCOs need new tools to help them make the transition from operation in the old environment to the competitive world of the future. This dissertation describes and develops methods and tools that are designed for the competitive GENCO. Some of these tools include bidding strategy developers, software to infer the bidding rules of others, and updates of common tools like economic dispatch and unit commitment.

## 1.5 Contents of this dissertation

Chapter 2 presents the reader with general information and methods that are used and referred to in subsequent chapters. Much of our research relies heavily on the use of genetic algorithms, therefore, the details of the basic genetic algorithm are presented in this chapter. Chapter 2 also describes the market framework and auction framework assumed for our research. Within that market framework, we discuss how we can create energy trader portfolios that combine the spot market contracts with options and futures contracts to achieve increased profitability and decreased risk.

An important method of studying the proposed markets and the behavior of participating agents is the field of experimental/computational economics. Chapter 3 presents some of our work with GA and with GP-Automata, a technique which combines genetic programming and finite state machines, to represent adaptive agents. We present work on evolving agents (each with its own bidding strategy) for use in a double auction. The first part of the chapter describes work with fixed string/non-adaptive bidding strategies, and the remainder of the chapter discusses an attempt to evolve adaptive strategies with GP-Automata.

In chapter 4 we discuss the role of fuzzy logic in the competitive electric marketplace, including how it can be applied in developing bidding strategies. Since competition promises to drive the power system closer to its operating limits, improvements in measurement and system control will be important. We provide an example of using fuzzy logic to do automatic generation control and discuss extensions that would make it superior to traditional controllers.

Since the GENCO's forte is primarily generating electricity, in chapter 5 we examine unit commitment and discuss how to update it for the competitive environment. We discuss the role of unit commitment in developing bidding strategies, as well as the role of bidding strategies in solving the unit commitment problem.

Depending on the market structure adopted by a particular location, large amounts of bidding data may be available to regulators or market participants. Ideally, regulators could use this data to verify that the market is efficient. Market participants with access to this data might gain an advantage over their competitors if they could somehow determine their competitor's bidding strategy. In chapter 6 we outline methods of automatically inferring other participants' trading rules based on historical data.

Finally, in chapter 7 we present some general commentary on the upcoming competitive environment. We present conclusions for each the chapter topics in this dissertation, and ideas for future research.

## **2 BASIC CONCEPTS, METHODS AND ASSUMPTIONS**

### **2.1 Chapter summary**

Within a competitive framework the objective of electric energy producers is to maximize their profit, subject to a particular risk level. The bidding strategies are designed to maximize a trader's utility (a function of risk and profit). Although much of the research described here focuses on double auction bidding strategies for the spot market, it is also independently applicable for the futures, options, and forwards markets. This chapter shows how the research in the following chapters can be used together to come up with bids for the various markets. This chapter presents the reader with general information and methods that are used and referred to in subsequent chapters. Many of the later chapters use genetic algorithms and genetic programming; to prevent duplication, the details of the basic genetic algorithm are presented in this chapter. The market framework and auction framework assumed for this research is described here. Within that market framework, this chapter discusses how to create energy trader portfolios that combine the spot market contracts with options and futures contracts with the overall goal of increasing profitability and decreasing risk.

### **2.2 Developing a coherent bidding strategy**

Much of the research reported in this dissertation relies heavily on computational economics. This section describes how computational economics can be used to develop sensible bidding strategies and to influence a GENCO's position in the markets. Figure 2.1 provides an overview of the process this research proposes. In the figure, simulated auction markets are used to develop the forward price and demand curves. To obtain realistic data, these simulated markets are populated with computerized trading agents. The computerized agents have been seeded with models of the competing GENCOs and of the ESCOs to whom they wish to sell electricity. Expected prices and demands are developed from the forward curves are given to the unit commitment scheduler. Based on the predicted price and demand information, the unit commitment scheduler attempts to provide the GENCO with the schedule of generating units that maximizes expected profit. The GENCO then uses this information to develop bids and take market positions with the aid of the double auction bidding strategy developer. It should be noted that the models are microeconomical, constructed for the firm, and are not necessarily consistent with macroeconomic assumptions.

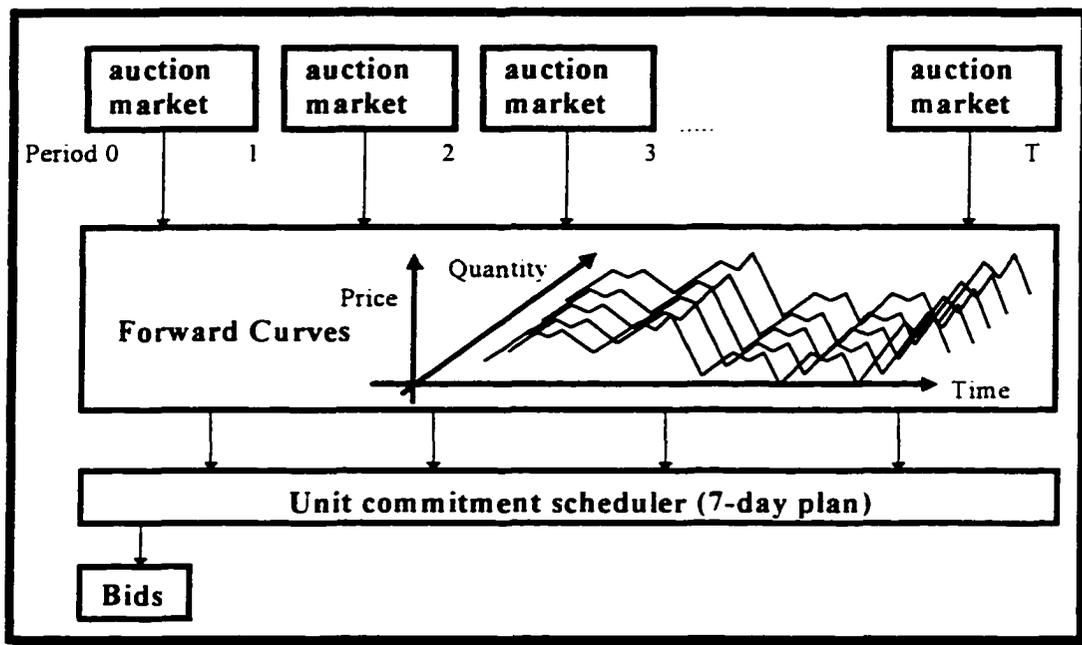


Figure 2.1 An overview of bidding strategy development.

Developing the competitor models is an involved process. Under the market framework assumed for this research, the results of the auctions are public information, similar to the Australian electric power market. As shown in Figure 2.2, a database of auctions from previous periods contains competitor (GENCO and ESCO) bidding information, and can be intelligently mined to determine the general rules that the competitors are using. This information can be combined with any additional information known about the system or about the competitors and tuned to develop a model of the competition that will be used in computerized agents in the simulated markets. Chapter 6 describes the intelligent data mining process.

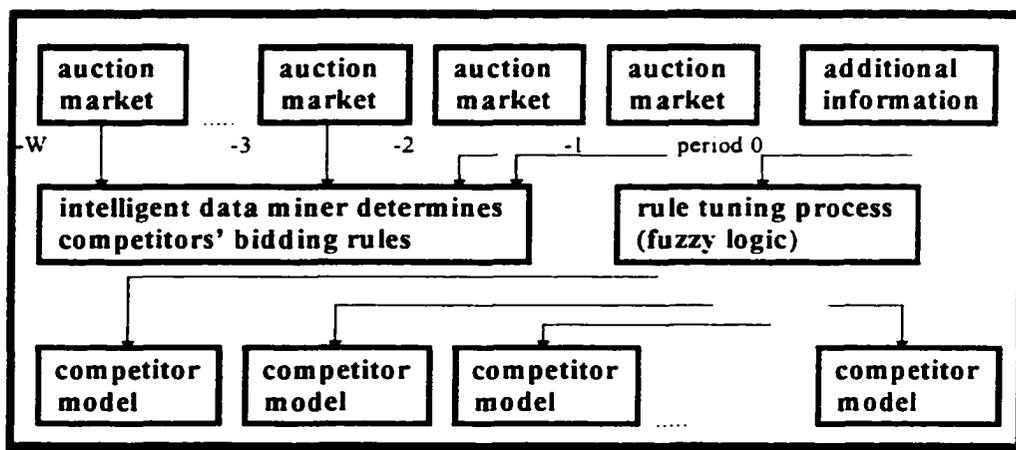


Figure 2.2 Determining competitor models.

The competitor models developed as described above are used to populate computerized agents participating in simulated auctions for each period of interest. The competitor models may represent an aggregated GENCO competitor and an aggregated ESCO, or individual GENCOs and ESCOs. Important information like weather, unit outage information, status of transmission system, and time of day, which will influence the bidding process in markets is fed into the competitor model as shown in Figure 2.3.

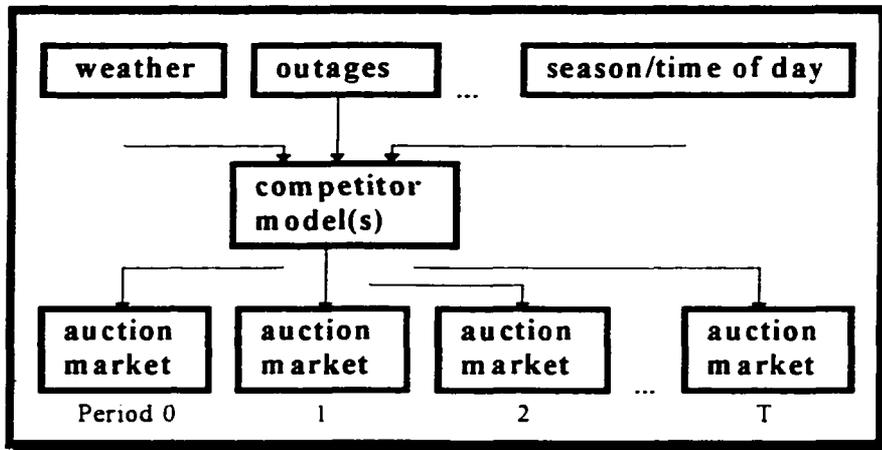


Figure 2.3 Using the competitor models in the auction markets.

For each auction market period, it is possible to determine a curve relating the price to the quantity demanded and a curve relating the price to the quantity supplied. For an individual market, the process of determining this relationship uses a horizontal summation of each GENCO's supply curves. Figure 2.4 shows how an aggregate demand can be determined from three individual GENCO supply curves. Each of the GENCO curves is bounded on the left by the minimum generation level, and on the right by the maximum generation level. The aggregated ESCO curve can be found in a similar manner. These curves from each time period are then put together to get the forward price and demand curves shown in Figure 2.5.

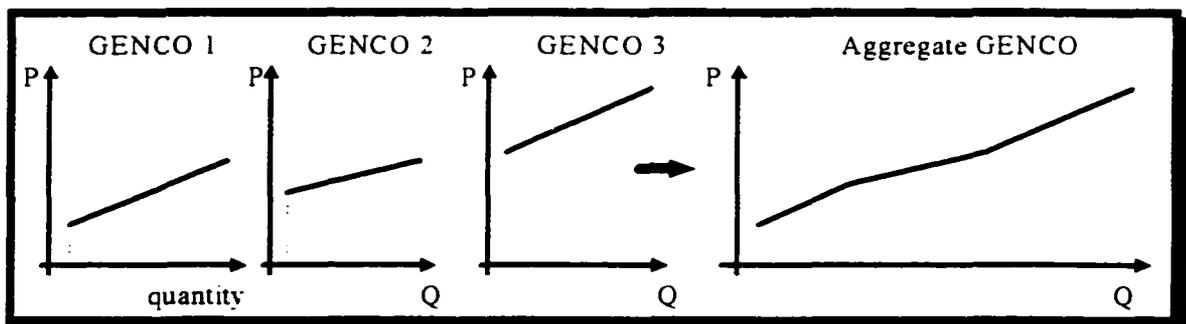


Figure 2.4 Determining the single period aggregate supply curve.

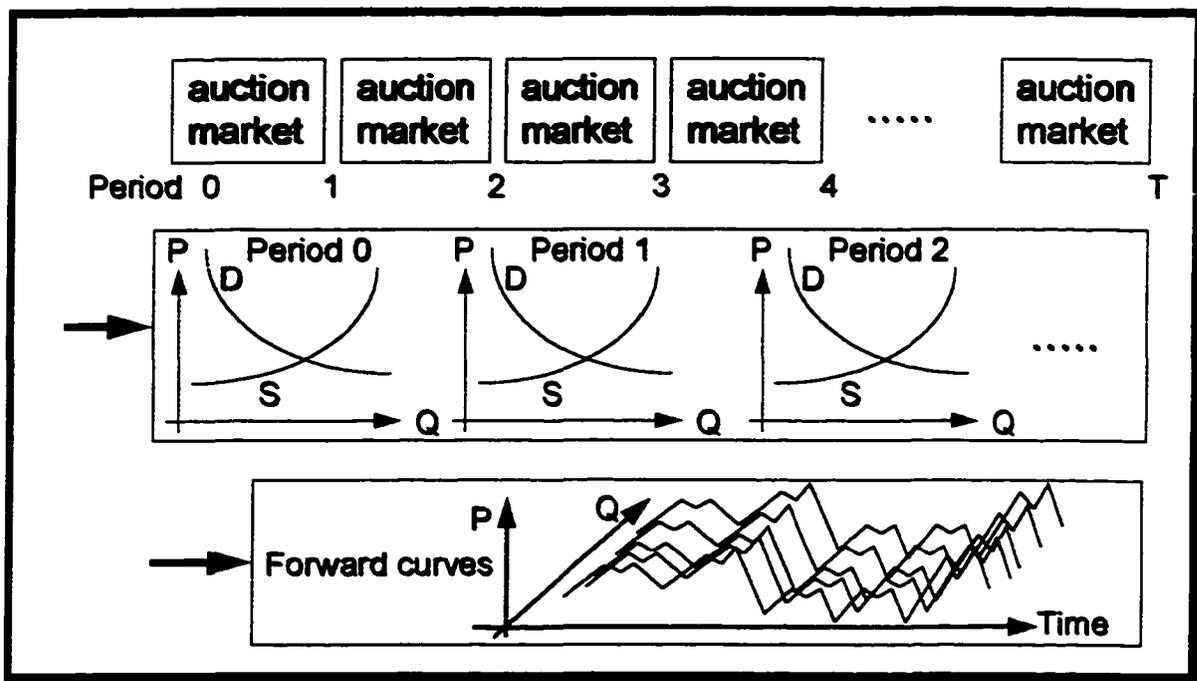


Figure 2.5 Determining the forward demand and price curves.

Once the forward curves have been calculated these can be used to determine a unit commitment (UC) schedule that maximizes a GENCO's expected profit. For simplification, the UC algorithm in chapter 5 does not use the entire forward curve. Instead it uses a single expected price and quantity for each period as shown in Figure 2.6. An hourly UC schedule is typically calculated for the following week. Every period, based on the new auction results, the market information is updated and the UC is run again with the latest information.

### 2.3 Genetic algorithms (GAs)

Developed by John Holland, genetic algorithms (GAs) are non-traditional general-purpose search techniques inspired by the biological model of evolution. One of the first books on genetic algorithms was *Adaptation in Natural and Artificial Systems* written by Holland in 1975, but only recently have they been gaining wider popularity among researchers. Holland's formulation was motivated by the observation that sexual reproduction, in conjunction with the pressure of natural selection, could result in the development of highly adapted species.

GAs typically involve the use of a population of data structures which are capable of containing the solution to the problem in question. The contents of the data structures evolve according to natural selection, the Darwinian principle of survival of the fittest [Goldberg, 1989]. GAs are often able to find solutions to problems which can not be handled by more traditional optimization techniques. GAs have been shown to

work on problems like the unit commitment [Kazarlis et al., 1995; Maifeld and Sheblé, 1996; Kondragunta, 1997] and the economic dispatch [Walters and Sheblé, 1992] problems in the electrical power systems field. GAs can search solution spaces in a parallel fashion. An entire population of candidate solutions (data structures with a form suitable for solving for the problem being studied) is "randomly" initialized and evolves according to GA rules. Although they could consist of anything, the data structures often consist of strings of binary numbers which are mapped onto the solution space for evaluation. Each solution (often termed a creature) is assigned a fitness, which is simply a heuristic measure of its quality. During the evolutionary process, those creatures having higher fitness are favored in the parent selection process and are allowed to procreate. The parent selection is essentially a random selection with a fitness bias. The type of fitness bias is determined by the parent selection method. Following the parent selection process, the processes of crossover and mutation are utilized and new creatures are developed which ideally explore a different area of the solution space. These new creatures replace less fit creatures from the existing population. Figure 2.7 shows a block diagram of the general GA. The following subsections provide additional detail on the different processes involved in the genetic algorithm.

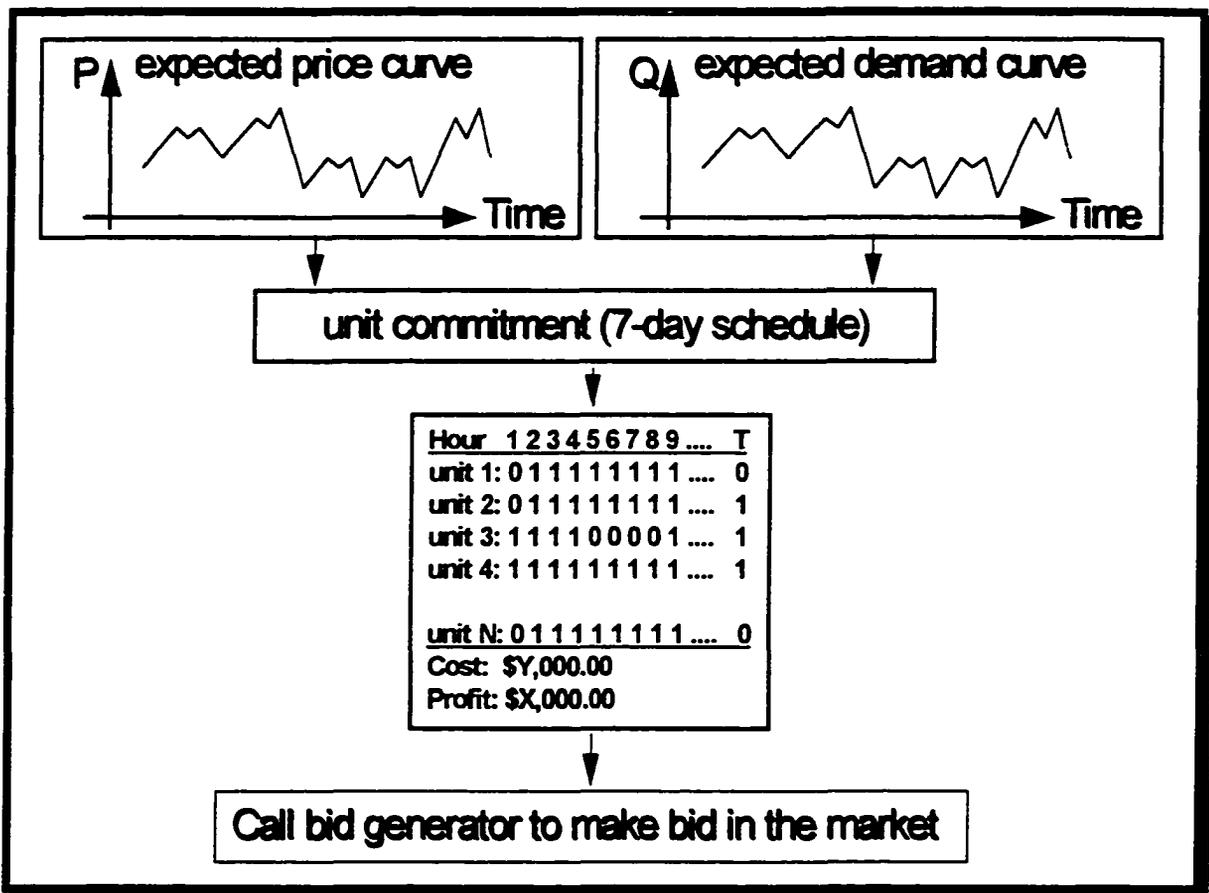


Figure 2.6 Using unit commitment as an input to bid generator.

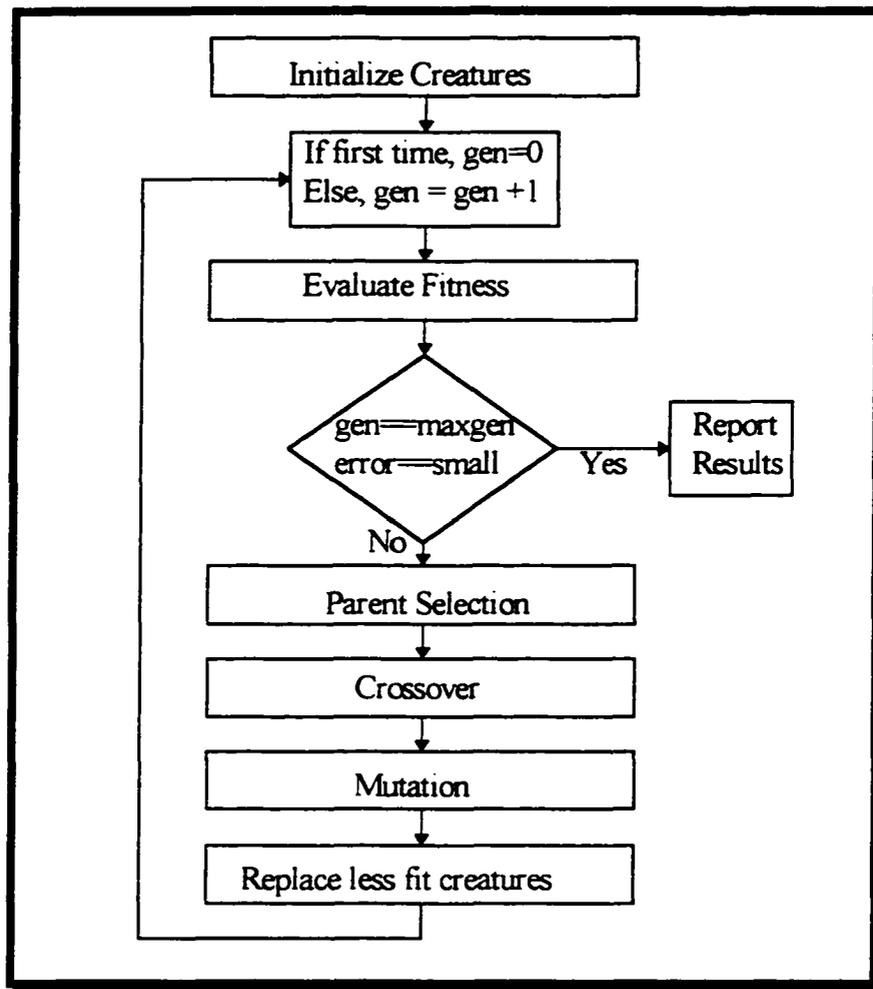


Figure 2.7 A simple genetic algorithm.

### 2.3.1 Parent selection

In biology, a polar bear that is white enough to blend into the snowy backdrop has a slightly greater chance of sneaking up on its prey. On average we might expect that a well-fed male polar bear would be more likely to win the dating game for the bachelorette of his choice than would be his hungry rival. The genes for producing a snowy-white polar bear live on through the bachelorette's bear cub and continue to propagate throughout the species as long as the trait of whiteness doesn't become a huge disadvantage. Similar to Darwin's theory of survival of the fittest for evolving biological species, members of the GA population are allowed to reproduce based on the fitness they exhibit. Creatures selected for reproduction are called parents. Parents are selected randomly from the population with a fitness bias which tends to select highly fit creatures for reproduction. There are several successful ways of selecting parents, and each method biases the random selection in a different way. Three commonly used methods are described below.

### 2.3.1.1 Roulette selection

With roulette selection, a probability wheel with values ranging from 0 to 1 is used to determine which creatures are selected to become parents. The fitnesses of the creatures are normalized and mapped onto the probability wheel. The probability of selecting a creature corresponds to its contribution to the sum of all creatures' fitnesses. See Figure 2.8. Roulette selection has a tendency to highly favor the fittest creature in the population. This tendency can be undesirable in small populations and may cause the population to converge early to a suboptimal solution. To illustrate this, consider a population where there are many zero fit creatures and only one creature with non-zero fitness, the single creature will be chosen as the parent for all new children. The only way the child can be different from the parent in this case will be due to mutation which will be explained later.

### 2.3.1.2 Rank selection

Similar to roulette selection, rank selection assigns selection probabilities to creatures based on the creature's rank when sorted by their fitness. By using the rank rather than the fitness to assign the selection probabilities we remove the problem with premature convergence to a suboptimal solution discussed in roulette selection. We say that this parent selection mechanism tends to preserve biodiversity better than roulette selection. See Figure 2.8.

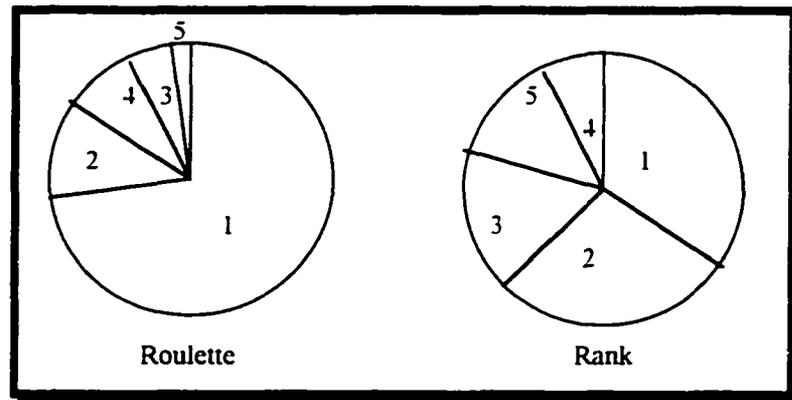


Figure 2.8 Roulette and rank probability wheels.

### 2.3.1.3 Tournament selection

Tournament selection is substantially different from the roulette and rank selection methods. It maintains biodiversity and works well for those problems which have a fitness landscape with tall narrow peaks. Each creature is assigned to a group of four (tournament) based on a number randomly selected from a uniform distribution. Within each group, the two creatures with the highest fitness are selected as parents, and their offspring replace the two creatures with the lowest fitness in their group. Using tournament selection

generally results in half of the population being replaced each generation, whereas the number of new creatures when using other parent selection techniques is an independent parameter that needs to be specified.

### 2.3.2 Crossover

New creatures, or children, are created in the reproduction process. Crossover is the first part of reproduction and is borrowed from the biological process where the DNA from a father and a mother are shared to create the genetic instructions for producing a child. As in biology, most genetic algorithms require two parents for reproduction, and the result is two children. In genetic algorithms, children are created by copying the contents of the data structure representing parent 1 into the data structure that will eventually represent child 1 and from the data structure of parent 2 into child 2 until a randomly selected crossover point is reached. At this point, the contents of parent 1 are copied into child 2, and the contents of parent 2 into child 1. This is commonly referred to as single point crossover and is presented graphically in Figure 2.9. Multiple point crossover operates on the same principle but includes more crossover locations, which can be helpful in GAs with long DNA strings.

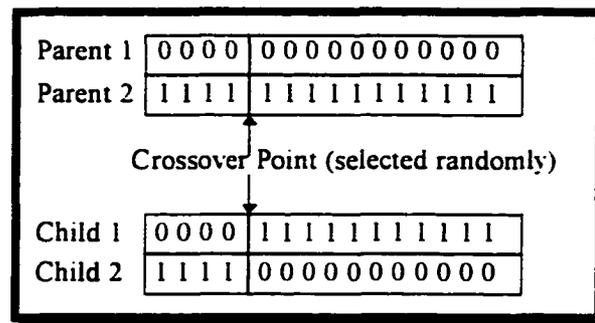


Figure 2.9 Single point crossover.

### 2.3.3 Mutation

Following crossover, we encounter mutation, which is the second part of the reproduction process. In biology, mutation consists of random copying errors during DNA replication of the billions of base pairs which contain the instructions for building a member of a species. In the short run, these random errors may or may not have a large effect on the resulting creature. In genetic algorithms, mutation can prevent the search from getting stuck in local optima. Mutation constantly introduces new genetic material into the gene at some low rate. If the gene to be mutated in the child is represented by a binary string, mutation involves flipping the bit (0 goes to 1, 1 goes to 0) at each location in the string with some probability. If the gene is represented by an integer, mutation might involve adding an integer that will result in a different valid integer occupying that gene location (loci). Depending on the problem, one might find it useful to change the mutation rate

depending on the maturity of the evolution. Initially when much biodiversity is already in the population, mutation may not be as important. Later, when crossover tends to make the population look fairly identical, the mutation rate could be increased. See Figure 2.10 for an example of a simple mutation.

#### 2.3.4 *Fitness evaluation*

In biology, the goal is to have one's genes live on. Anything that helps a member of a species (and its kin) propagate its genetic makeup to future generations adds to its fitness. In genetic algorithms, we generally have a more concrete goal in mind. Once that goal is known, there are many ways of creating fitness functions that will get the population to evolve the way the designer wants. The GA designer must use engineering judgment in determining how to award fitness. For example, if a GA was being used to solve the traditional unit commitment problem, a creature's fitness might be inversely proportional to the cost of the solution that it finds. Costs are first evaluated for the corresponding solution using our cost equations, and those creatures with higher associated costs are awarded a lower fitness. If the goal of the GA is to develop bidding strategies for auctions which maximize profit, creatures which find a solution resulting in a higher profit, are awarded higher fitness than those making a smaller profit. The designer can be creative in developing fitness evaluation criteria. While the primary fitness function might be profit or cost, secondary fitness criteria may be added (e.g., less risky strategies are better than risky ones for a given profit level).

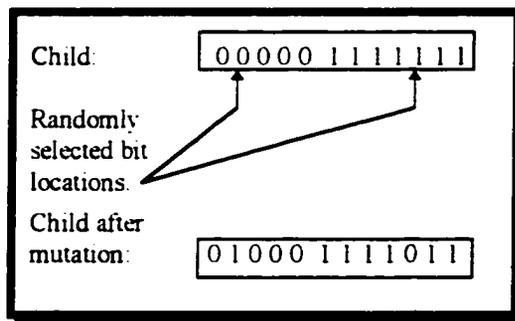


Figure 2.10 GA mutation.

#### 2.3.5 *A GA example: The string evolver*

A simple example of a genetic algorithm is the string evolver. In the string evolver, the goal is to have the GA find a target string of your choosing. First, a population of solutions is initialized randomly from an appropriate alphabet. Then, using only the parent selection, crossover, mutation, and replacement operators we've already discussed, the GA can find the answer. See Figure 2.11 for an example of the string evolver.

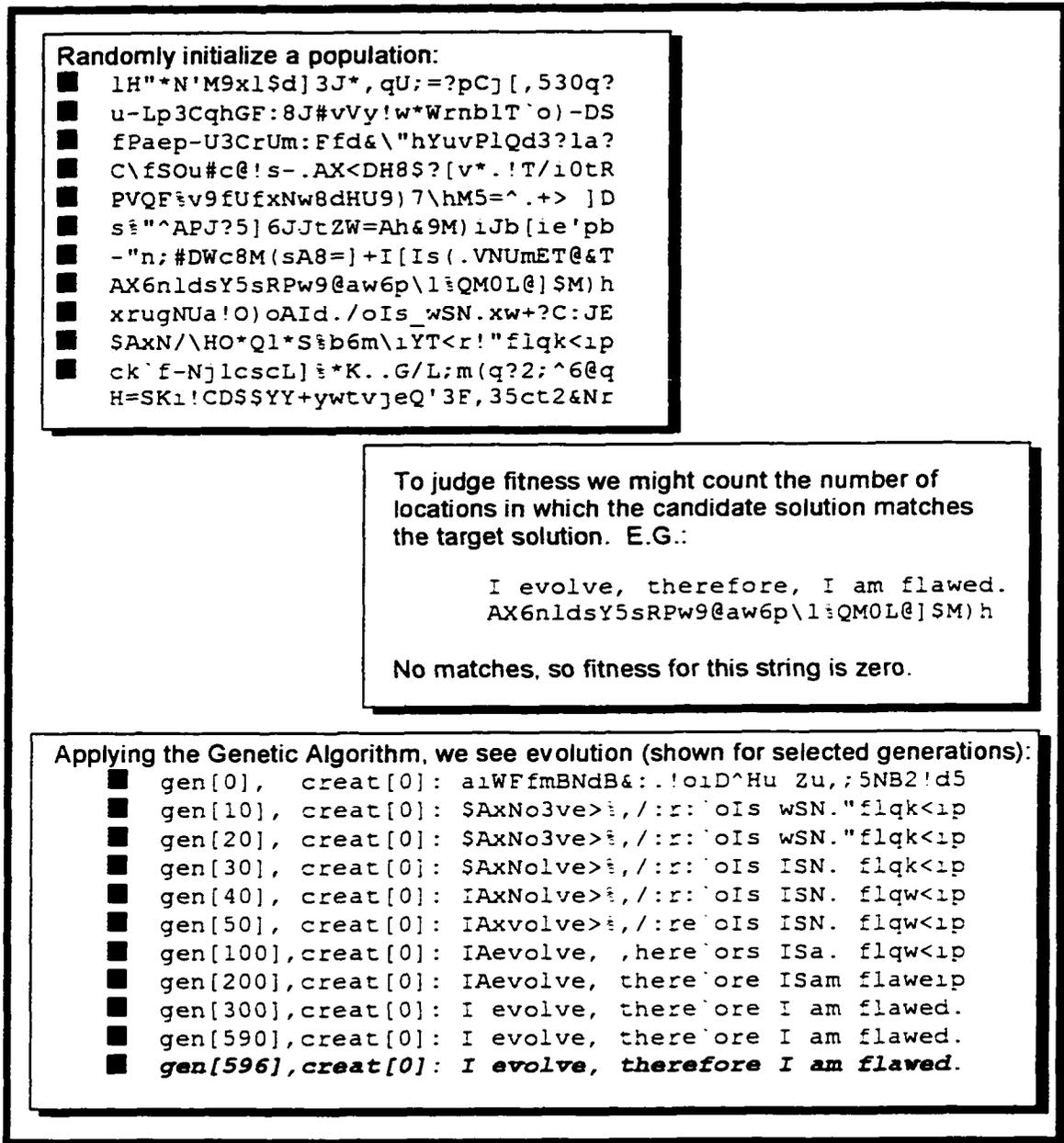


Figure 2.11 The string evolver example.

## 2.4 Market structure

The exact structure of the deregulated electricity markets emerging in various countries and control regions depends, to a large extent, on the regulators. To attempt to study each framework along the continuum from power pool to a power exchange like that in Australia in a single dissertation would be futile. Besides, things learned from an assumed market structure can often be used in another market framework. Therefore, this research predicts and

assumes that regional exchanges (i.e., electricity mercantile associations [EMAs]) will play an important role. Electricity trading of the future will be accomplished through bilateral contracts and through EMAs where traders bid for contracts via a double auction. The electric marketplace used in this chapter has been refined and described in various chapters. Recent research [Fahd and Sheblé, 1992] demonstrated an auction mechanism. Later [Sheblé, 1994b] the different types of commodity markets and their operation and outlined how each could be applied in the evolved electric energy marketplace were described. Others [Sheblé and McCalley, 1994] outlined how spot, forward, future, planning, and swap markets can handle real-time control of the system (e.g., automatic generation control) and risk management. Later work brought the above ideas together and demonstrated a power system auction game designed to be a training tool [Kumar and Sheblé, 1996b]. That game used the double auction mechanism in combination with classical optimization techniques.

A framework in which electric energy is only sold to energy service companies (ESCOs), is delivered on wires owned by distribution companies (DISTCOs) and transmission companies (TRANSCOs), and is only generated by generation companies (GENCOs) was described [Kumar and Sheblé, 1996a; Richter and Sheblé, 1997a; Richter et al., 1999; Richter et al., 1998] and is represented in Figure 2.12. The North American Electric Reliability Council (NERC) sets the reliability standards. ESCOs and GENCOs may interact with ancillary services companies (ANCILCOs). The contract prices are determined through a double auction. Buyers and sellers of electricity make bids and offers that are matched subject to approval of the independent contract administrator (ICA) who ensures that the contracts will result in a system operating safely within limits. The ICA submits information to an independent system operator (ISO) for implementation. The ISO is responsible for controlling the system to maintain its security and reliability.

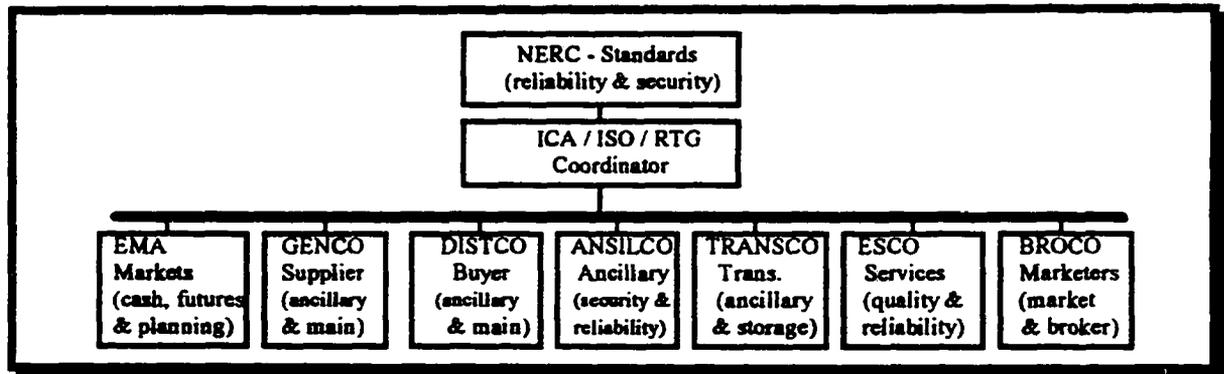
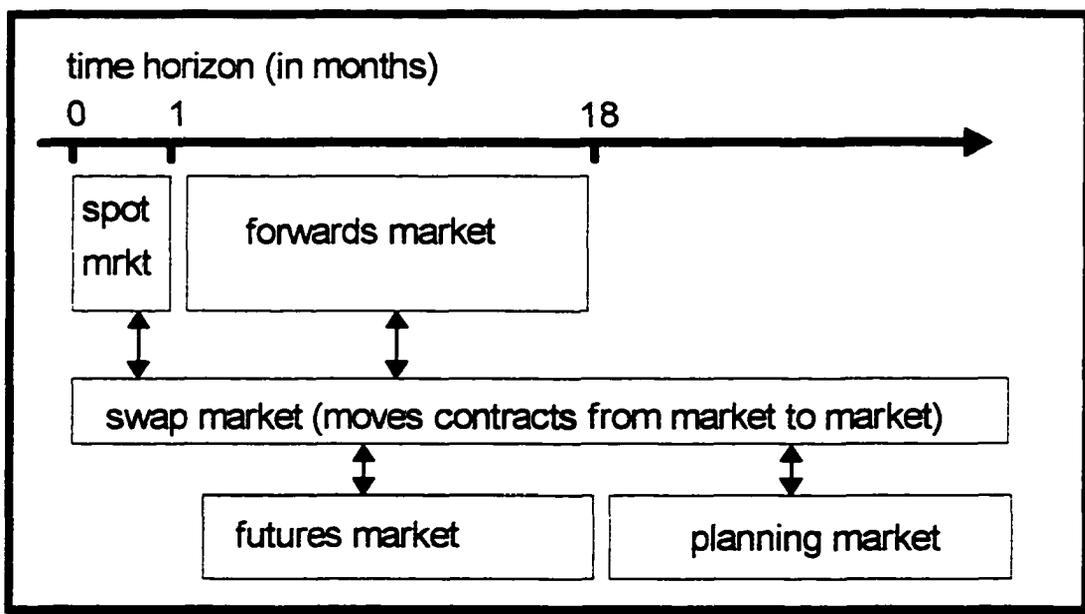


Figure 2.12 Brokerage system model.

The described framework [Sheblé, 1996] allows for cash (spot and forward), futures, and planning markets as shown in Figure 2.13. The spot market is what we are most familiar within the electric industry. The spot market is a market where sellers and buyers agree (either bilaterally or through an exchange) upon a price for a certain number of MWhr to be delivered sometime in the near future (e.g., 10 MWs from 1:00 p.m. to 4:00 p.m. tomorrow). The buyer needs the electricity, and the seller wants to sell. They arrange for the

electrons to flow through the electrical transmission system and they are happy. Also in the cash market, a forwards contract is a binding agreement in which the seller agrees to deliver an amount of a particular product with a specified quality at a specified time to the buyer. The forward contract is further into the future than is the spot market. In both forward and spot markets, the buyer and seller want to exchange the physical good (i.e., the electrical energy). On the other hand, a futures contract is primarily a financial instrument that allows traders to lock in a price for a commodity in some future month. Buying a futures contract is akin to purchasing insurance. It helps the traders to manage their risk by limiting potential losses or gains. Standardized futures contracts exist for commodities in which there is sufficient interest and in which the goods are generic enough that it is not possible to tell one unit of the good from another (e.g., 1 MW of electricity of a certain quality, voltage level, etc.). A futures options contract (see Figure 2.14) is a form of insurance that gives the option purchaser the right, but not the obligation, to buy (sell) a futures contract at a given price. The seller gets a certain and immediate cash payment at the time of sale of the options contract for agreeing to take the risk associated with price fluctuations in the future. Both futures and options are derivatives, meaning that their price is derived from the price of an underlying physical good.



**Figure 2.13 Interconnection between the markets.**

As mentioned above, both the options and the futures contracts are financial instruments designed to minimize risk. Although provisions for delivery exist, they are not convenient (e.g., the delivery point is not located where you want it to be located). The trader ultimately cancels his position in the futures market either with a gain or loss. The physical goods are then purchased on the spot market to meet demand with the profit or loss having been locked-in via the futures contract. Another market is the swap market. A swap contract is a customized agreement in which one firm agrees to trade its coupon payment for that of the other firm

involved in the swap. Finally, we have the planning market which exists to finance long-term projects like transmission lines and power plants.

There is some jargon used in the market and commodity trading area that might be new to people in the electric utility industry. For each options contract, there is someone "writing" the contract who, in return for a premium, is obligated to sell (buy) at the strike price. "Long" denotes ownership: "to go long" means to purchase the item in question. In the figure, "long" indicates that the trader has purchased the option and now has the right to buy (call) or the right to sell (put) the future. A trader who writes the option is "short": to go short is to sell the item in question. Let's assume that the item in question is a MWhr of electricity. In the long call diagram, the long trader has paid a premium (e.g., \$1) to the option writer for the call option. This call option gives the trader the right to buy a MWhr for the strike price (e.g., \$7). If the price goes above the strike price plus the premium (e.g., \$8), the trader has made a profit. The long trader has reduced risk by limiting his losses to the premium. In the diagrams labeled short, we see what happens from the option writer's point of view. He receives the premium for assuming the risk and is obligated to sell the MWhr at the strike price even though the market price is higher. The diagrams labeled "put" show how the put works. The long put trader pays a premium to lock in a maximum price (exercise price) that he will have to pay for the MWhr. The short put trader takes that premium in return for promising to sell the MWhr for that same exercise price. For people who are unaccustomed to looking at the above diagrams, they can be confusing at first glance. Table 2.1 gives the reader another way of looking at the long call example previously mentioned.

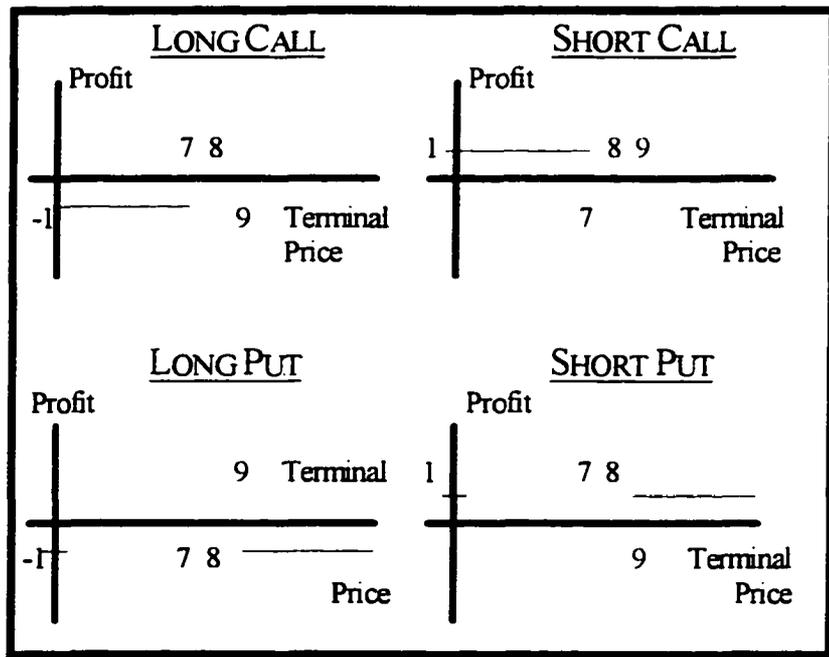


Figure 2.14 Methods of using options.

Table 2.1 Long call example

Strike Price	Spot Price at Maturity	Premium	Profit w/out Long Call	Profit w/ Long Call
\$7	\$4	\$1	-\$3	-\$1
\$7	\$5	\$1	-\$2	-\$1
\$7	\$6	\$1	-\$1	-\$1
\$7	\$7	\$1	\$0	-\$1
\$7	\$8	\$1	\$1	\$0
\$7	\$9	\$1	\$2	\$1
\$7	\$10	\$1	\$3	\$2
\$7	\$11	\$1	\$4	\$3
\$7	\$12	\$1	\$5	\$4

#### 2.4.1 Using the markets to manage risk

With neither an obligation to serve nor guaranteed rates of return the energy trader's objective becomes the maximization of profit for his shareholders. In a competitive environment, there may be times when ESCOs may be unable to purchase enough energy for their customers, or times when GENCOs may have excess generation. This uncertainty, combined with fluctuating prices and demand, makes profit difficult to predict in any particular scenario. We might then consider a distribution of bids and offers and develop strategies that maximize the trader's expected profit. If a trader uses the strategy long enough, he should get the expected profit associated with that strategy. In the short run, he might see gains or losses very different from the expected profit. This unpredictability means we consider the strategy risky. The term *risk* can be loosely defined as a measure of the lack of predictability of an outcome associated with a particular decision. Different strategies producing the same expected profits might well have different risks associated with each as we see in Figure 2.15. Since most traders cannot endure low or even negative profits for long periods, the trader would probably be willing to sacrifice some long-term expected profit in return for reduced risk. Economists use "utility functions" to describe and order preferences. Among other things, a trader's utility should vary directly with actual profit (should be similar to expected profit) and indirectly with risk.

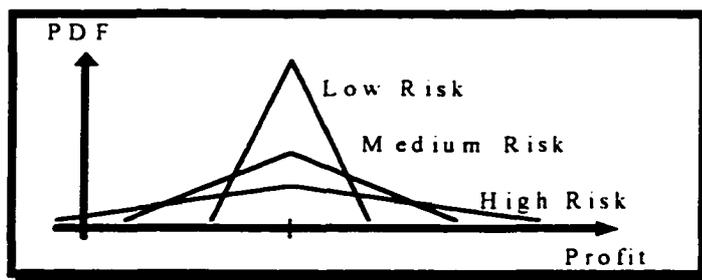


Figure 2.15 Different risks with the same expected profit.

### 2.4.1.1 Futures contracts

Futures contracts allow producers to hedge so that they can limit their losses. Other things being equal, a GENCO's profit varies with the price of electricity. Trying to predict the price months in advance so that profit can be known in advance is tricky. Suppose it is April, and because of some big decisions (unrelated to insider trading) the CEO wants to know what his or her GENCO's profit will be in July. Simply by considering our fuel contracts and using demand forecasts we can draw a profit curve based on the price as in Figure 2.16(a). In the figure, this profit curve is drawn as the line segment labeled "with no hedge." Not knowing the price means that we have the potential for large losses. The CEO doesn't want to see just a line on a graph—he or she wants a simple number. This is where the futures hedging comes into play. For the example in Figure 2.16(a), the GENCO can short (i.e., sell non-firm electricity they don't have yet) July electricity with futures contracts. When July arrives, if the spot price is low, they make money on their futures contract and lose on the electricity sold on the spot market. The gain on the futures market offsets the loss in the spot market. If the spot price in July is high, then the electricity sold on the spot market yields a profit while the futures contract will produce an offsetting loss. The result is that the net profit is much more predictable due to the hedge, and now we can give the board members that number they were looking for.

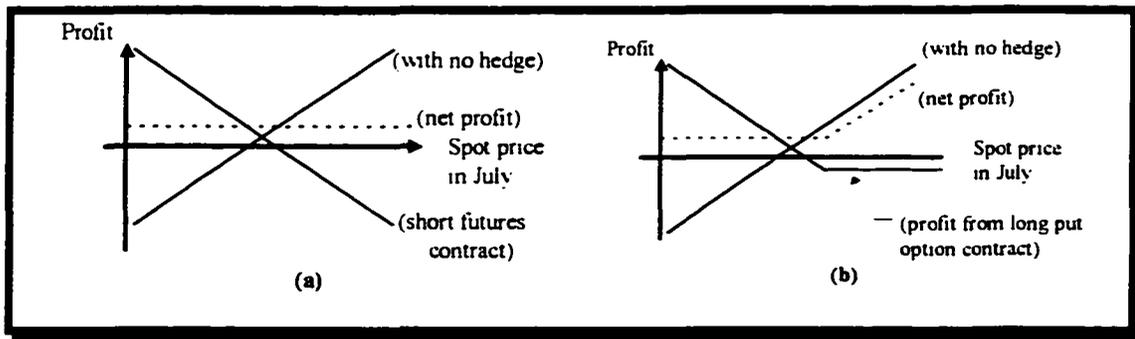


Figure 2.16 Hedging with futures and futures options contracts.

### 2.4.1.2 Futures options contracts

Futures options contracts give the holder the right, but not the obligation, to buy or sell a futures contract. They are not to be confused with option contracts on the physical good which are also available. Futures options contracts can be used to reduce risk. Consider the GENCO wanting to maximize its profit and reduce its risk. One alternative is that the GENCO pays a premium for an options contract that would give it the right to sell (short) electricity at the strike price. (If the price was higher than the strike price, the GENCO would let the option expire). Figure 2.16(b) shows how the option contract can be used to hedge profit. Notice that when the price is low, the GENCO can exercise the option and have a futures contract as in the previous

example to offset its losses in the spot market. When the price is high, the GENCO has no obligation to sell at the strike price; the net profit is the profit from the electricity produced by the GENCO and sold on the spot market less the premium paid for the options contract. The GENCO has limited the amount of money that it can lose, but can still reap the benefits of a high price in July. Another alternative using futures options would be a short call.

#### 2.4.2 Auctions

An auction is a method of matching buyer(s) and seller(s) through bids and offers. A double auction has both seller(s) and buyer(s) submitting bids and offers, while a single-sided auction only one of these is allowed to submit prices. The English electrical system uses a single sided auction in which electricity generators submit generator unit commitment schedules with their offer. The buyers do not submit any price bids. The sellers are in competition with each other. In addition to double-sided and single-sided multi-participant auctions, in the electric industry we should expect to see many bilateral agreements where two companies contract with each other outside of an organized exchange. It can easily be argued that bilateral agreements are special cases of double auctions in which there is only one buyer and one seller.

The double auction is the pricing mechanism for the markets assumed in our framework. In each of the markets, bids and offers are sorted into descending and ascending order respectively, similar to the Florida Coordination Group approach [Wood and Wollenberg, 1996]. If the buyer's bid is higher than the seller's offer to be matched, then this is a potentially valid match. The ICA must determine whether the transaction would compromise system security and whether sufficient transmission capacity exists. If the ICA approves, each potentially valid offer and bid helps to determine the final price, termed the equilibrium price. The average of the bid and offer of each pair of valid matches (weighted by the number of MWs) is used to determine one overall equilibrium price. Another pricing scheme we have used in some of our research is a discriminatory pricing in which the buy and sell bids and offers are matched and then the price for each transaction is the average of the bid and offer given by the partnered traders. An example of these two pricing schemes is given in Table 2.2.

**Table 2.2 Example of auction bid matching.**

Buy bids (\$/MW)	Sell offers (\$/MW)	MW amount	Contract?	Discriminatory Price (\$/MW)	One equilibrium price (\$/MW)
12.50	8.50	1	Yes	10.50	10.63
12.00	9.00	1	Yes	10.50	10.63
11.80	10.00	1	Yes	10.90	10.63
10.00	10.50	1	No	NA	NA
9.50	11.00	1	No	NA	NA

In the example shown in the table, there are three bids that are higher than the corresponding offers. If there is not a sufficient number of valid matches, then the true price has not been discovered. Price discovery occurs when there are a sufficient number of buy bids and sell offers to allow a predetermined portion of the total participants to be satisfied with the resulting contract. Submission of the bids marks the beginning of the bidding cycle. When the auctioneer reports the results of the auction to the market participants the cycle is complete. If, after the present cycle, the price has not been discovered, the auctioneer reports that price discovery did not occur, and asks for new bids and offers. The buyers may increase their bids and sellers may decrease their offers and another cycle of the auction is played. The cycles continue until price discovery occurs, or until the auctioneer decides to match whatever valid matches exist and continue to the next round or hour of bidding.

After price discovery, the auctioneer asks if another round of bidding is requested. If the market participants have more power to sell or buy, they request another round. Allowing multiple rounds of bidding each time period allows the participants the opportunity to use the latest pricing information in forming their present bid. This process is continued until no more requests are received or until the auctioneer decides that enough rounds have taken place. See Figure 2.17 for a block diagram of the auction process as we model it in our auction simulator.

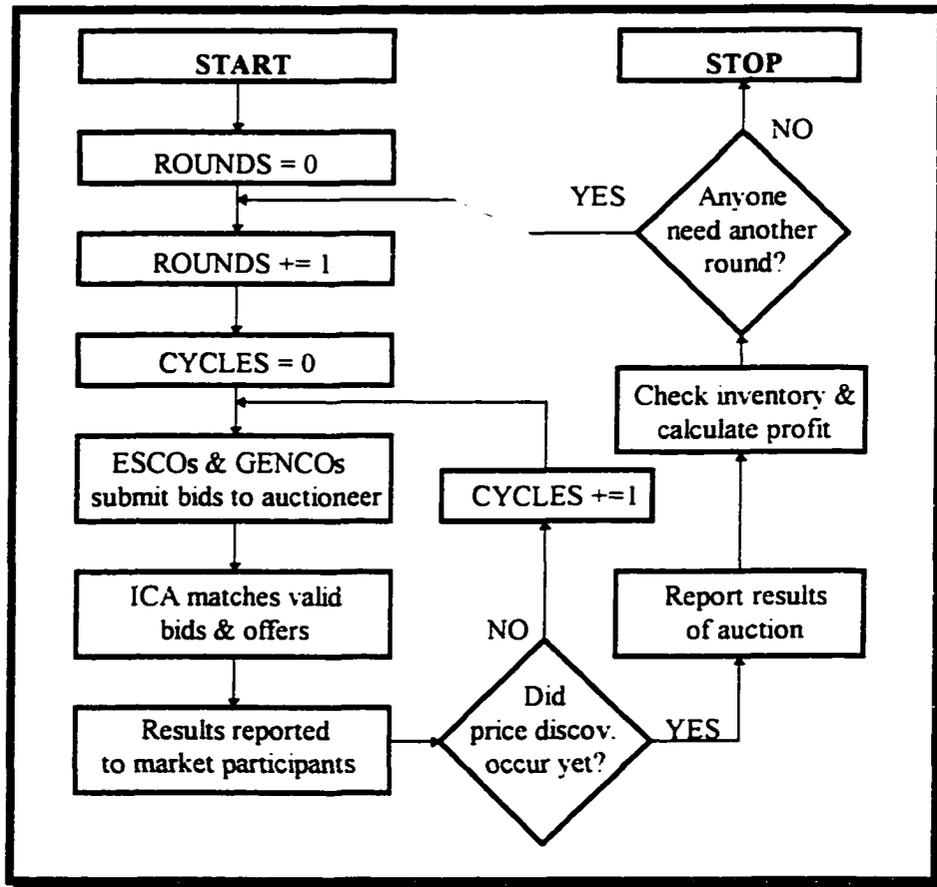


Figure 2.17 The auction process.

## 2.5 Limitations of the model

The model we assume is specified at a fairly high level and is meant to apply to a large set of possible deregulated scenarios. It allows the trader and energy producer to develop general strategies. Ideally the details of the particular deregulated framework in which one is interested can be incorporated into the model to refine the bidding strategies for actual use. Since we keep the model at a high level for most of the work described here, we avoid many of the complications that must be dealt with when using a more detailed model. For instance, all the financial contracts in the world won't change the fact that electrical current follows the laws of physics. The steady state models of the generation and transmission system which we use are a good start for deciding which transactions should be allowed or disallowed in the coming hours or days. However, the laws of physics dictate that supply and demand on the system must be in balance within fractions of a second. Considering the dynamic interactions of generation and load is more demanding than simply adding up the megawatt-hours on a particular line and will likely produce different control responses than a steady state model might suggest. Since predicting the exact size and nature of the load at any given point in time is difficult, the complex control responses to the changing load are also very difficult to specify meaningfully in advance. Without these control actions, the system can become unstable quite quickly and result in catastrophe. Therefore, the market framework that we describe here is built on the assumption that the power system is able to respond to sudden changes, and that accounting based on flexible ancillary service contracts can be done after the fact.

The goal of regulators espousing deregulation is not to offer generation companies the chance to make unlimited profits at the expense of the consumers and system integrity. Rather, the goal is to reduce consumer electricity rates through competition, eliminating the inefficiencies and dead weight losses that came with the monopolistic pricing of the past, while maintaining system reliability and integrity. To achieve that goal, the regulators of the power system must build in regulations that allows electrons to obey the laws of physics, and that will allow power system operators to mitigate the consequences of contingencies. Generator outages, transmission line failures, and transmission congestion are some of the challenges that power system operators will continue to face. Regulations must provide means of responding to these contingencies with minimal deviation from competitive pricing and no deviation from the laws of physics. Severe contingencies, e.g., blackouts, may require operating in a temporary state of 'martial law'.

## **3 EVOLVING BIDDING STRATEGIES WITH GENETIC ALGORITHMS AND GP-AUTOMATA**

### **3.1 Chapter overview**

The impending deregulation of the electrical industry in the USA and electric systems around the globe promises to open a multi-billion dollar industry to competition. Current research indicates that the double auction will be at the heart of several regional electrical commodity exchanges. The goal is to design comprehensive profitable bidding strategies for traders. This chapter reports on research using genetic algorithms to evolve bidding multipliers, bid amounts and selecting forecasting methods for trading agents. The second part of this chapter presents research that uses a technique that combines the use of genetic programming with finite state automata which we term GP-Automata. Adaptive strategies encoded by two populations of GP-Automata are tested in an auction simulator modeling distribution companies and generation companies buying and selling power via a double auction. In addition to evolving profitable bidding strategies, the resulting strategies can also be designed to imitate certain types of trading behaviors. These strategies can be used directly in on-line trading, or as realistic competitors in an off-line trading simulator. We report the results of specific experiments which test the effect of changing the size of the GP trees, and the effect of changing the number of states.

### **3.2 Introduction**

Regulations governing the electric utility industry in many parts of the world are being changed to promote competition. By increasing competition through deregulation of the US electrical transmission network, Congress and other regulators hope to increase power system efficiencies and to see benefits for electric consumers.

As mentioned before, for decades, electric consumers in the US had only their local, vertically integrated utility as a source of electricity. Electric utilities have always been guaranteed customers, and have not had to operate in a competitive environment. Both consumers and generators of electricity (or their representatives) will soon be faced with having to sell or purchase power through a commodity exchange. To be successful, these electricity traders will need to develop bidding strategies. This chapter focuses on developing such strategies.

The research presented in this chapter assumes an electric marketplace which is structured similar to commodities exchanges like the Chicago Mercantile Exchange, Chicago Board of Trade, and New York Mercantile Exchange (NYMEX) where commodities (other than electricity) have been traded for many years.

In the research described in this chapter, trading agents use a genetic algorithm (GA) to evolve bidding strategies for the electric energy market. In one experiment, these strategies are encoded in the form of combination integer and binary strings. In subsequent experiments reported later in the chapter, bidding strategies are coded in the form of finite automata coupled with genetic programming (i.e., GP-Automata) [Ashlock, 1995, Ashlock and Richter, 1997]. An optimal bidding strategy should be adaptive, able to properly react as the trading behavior of its competitors changes. Coding information in the form of GP-Automata, which evolve in a GA, allows complex adaptive strategies to develop. The results have been written up specifically for the electric energy market, but are directly applicable for other markets.

Section 3.2 surveys recently published work in this area, including research on evolving economic agents, genetic programming applied to auctions, and auction environments. Section 3.3 reports experiments using the genetic algorithm to evolve non-adaptive bidding strategies. Section 3.4 describes GP-Automata and the auction environments in which the strategies are designed to perform. It also describes an experiment performed to test the effect of modifying the tree size or the number of states in the GP-Automata, and presents the results of these experiments.

### 3.3 Review of recent work

Some research has been done in developing bidding strategies for electric systems in other countries. Researchers analyzed bidding strategies for the restructured Power Pool of England and Wales, and showed mathematically that there exists an optimal bidding strategy for its bidders [Finlay, 1995]. Finlay's work differs in that his objective was not to maximize the profit of the individual generation companies, and the system itself is different from those proposed in the USA. Hence it is not directly applicable to our scenario.

Research described the different types of commodity markets and their operation and outlined how each could be applied in the evolved electric energy marketplace [Sheblé, 1994b]. Under the described framework [Sheblé, 1996] companies presently having both generation and distribution facilities would, at a minimum, be divided into separate profit and loss centers. Power is generated by generation companies (GENCOs), transported via transmission companies (TRANSCO), and all power is sold to energy service companies (ESCOs) and subsequently delivered to consumers on wires owned by the distribution companies (DISTCOs).

The described framework [Sheblé, 1996] allows for a cash market, a futures market and a planning market. The cash market is for trading power for each 30 minute period in the next 30 days. The futures market allows electricity trading from 1 to 18 months into the future. Futures contracts are non-firm for a

specific month. This futures market provides a means for electricity traders to manage their risk. The other market is a planning market that can be used to develop capital to build new plants and would allow trading more than 18 months into the future. Figure 2.2 shows how these markets are interconnected. Others outlined how cash, future, planning and swap markets can handle real-time control of the system (e.g., automatic generation control) and risk management [Sheblé and McCalley, 1994].

Additional work brought the above ideas together and demonstrated a power system auction game designed to be a training tool [Kumar and Sheblé, 1996b]. That game used the double auction mechanism in combination with classical optimization techniques. Buyers and sellers interact through a central coordinator, an Independent Contract Administrator (ICA), who matches the bids subject to all operational constraints. The central coordinator is responsible for ensuring that the energy transactions resulting from the matched bids do not overload or render the electrical transmission system insecure. GENCOs and ESCOs coordinate only via the prices transmitted to a central auctioneer. The ICA monitors and responds to the power system limits and transmission capacities.

Developing bidding strategies with evolving trading agents for the deregulated electrical utility industry is a new field of research. Apart from the electrical utility industry, interest has grown in recent years for using evolving, or adaptive, agents to simulate trading behavior. Research with adaptive agents has proved to be a useful means of exploring trading markets outside of the electrical industry.

Evolving agents have learned to play financial markets [LeBaron, 1995]. Research in which trading agents decide who to trade with based on an expected payoff was conducted [Tsfatsion, 1995]. Genetic programming combined with a finite state automata was used to play a classic academic game called Divide the Dollar which involves bidding behavior and strategies [Ashlock, 1995]. Later the same game was used to study kinship effects and concluded buyers and sellers should come from separate populations to reduce the amount of collusion [Ashlock and Richter, 1997].

A game based on a double auction was used to verify that genetic search is useful [Andrews and Prager, 1994]. Researchers showed that GP-based agents actually do learn, and they compare the performance of the GP-based strategies to those developed using simulated annealing. In addition, they show that at the beginning of the genetic algorithm it is possible to use a less rigorous fitness test than is needed in later generations. While their findings may be useful to the genetic algorithm community, their experiments leave room for further improvements in strategy-building, including the use of a better auction model and additional actions for each of the GP-based strategies.

### **3.4 Evolving non-adaptive bidding strategies with a genetic algorithm**

#### ***3.4.1 The marketplace***

This research assumes the existence of regional commodity exchanges in which buyers and sellers participate in a double auction. This framework can be considered an extension of the framework being implemented in California. For the results presented in this chapter, TRANSCOs are considered to be exogenous to the market, while ESCOs and GENCOs are allowed to interact in an environment as described in the previous section. Although the framework covers the futures and options markets, the research described in this chapter is written up for only the cash/spot market. See chapter 2 for more details regarding the market framework.

#### ***3.4.2 The basics of genetic algorithms***

A genetic algorithm is a search algorithm often used in nonlinear discrete optimization problems. Data, initialized randomly in a data structure appropriate for the solution to the problem, evolves over time and becomes a suitable answer to the problem. GAs were inspired by the biological notion of evolution. See chapter 2 for additional details of genetic algorithms.

#### ***3.4.3 Using genetic algorithms for trading agent representation and evolution***

In the research described in this section, parameters used to develop GENCOs' bids are evolved using a GA. Each member of the GA population corresponds to a GENCO participating in an auction. There are three distinct evolving parts, or genes, for each of the GENCOs. First, the number of 1 MW contracts to offer at each round of bidding is evolving. This gene is filled with integer values. Valid integers are between 0 and a maximum value that corresponds to that GENCO's maximum capability divided by the number of rounds of bidding. Second, bid multipliers for each round of bidding are evolved. These bid multipliers are used in combination with the GENCOs' costs and their expected equilibrium price to develop a bid. This gene is represented by binary strings that are mapped to a value between the GENCO's cost and forecasted equilibrium price during the bidding process. Third, each GENCO has a gene that selects which prediction technique to use to forecast the equilibrium price. This is an integer valued gene with valid integers being from 0 to 4, since there are five prediction techniques (described later) from which each GENCO may choose. Additional forecasting methods can be incorporated easily. See Figure 3.1 for a representation of the trading agent's data structures.

Based on preliminary sensitivity testing, roulette selection was chosen to select parents in each generation of the genetic algorithm. Roulette selection is a parent selection method that chooses more highly fit creatures with a greater probability than the lesser fit creatures. This fitness bias is more pronounced (especially when

population sizes are small) than other parent selection methods like tournament selection or rank selection. A more thorough sensitivity testing could be done to determine which of these methods is the best for this problem.

Based on sensitivity tests, three point crossover was selected to create the children. Crossover is used on both the number of contracts desired and the bid multiplier. The bid multipliers for each GENCO are concatenated together into one string prior to crossover, and three crossover locations are selected randomly from a uniform distribution over the gene's entire length.

Agent 0	Rounds of bidding				
MWs each round	12	4	20	...	14
Mult. each round	01011	01101	10101	...	00101
Prediction Technique	0				
Agent 1	Rounds of bidding				
MWs each round	12	4	20	...	14
Mult. each round	01011	01101	10101	...	00101
Prediction Technique	2				
					o
					o
					o
Agent N	Rounds of bidding				
MWs each round	12	4	20	...	14
Mult. each round	01011	01101	10101	...	00101
Prediction Technique	3				

Figure 3.1 A population of agents/bidding strategies

The standard bit-flip mutation operator is used on the binary strings representing the bid multipliers. The number of contracts gene has the possibility of being mutated by two different mutation operators. The first mutation operator (mutation\_A) adds an integer to the existing integer. If the result is not a valid integer, the value is wrapped around, i.e. if the result is greater than the maximum, then the maximum is subtracted from it. (i.e., the mod operator was used). Further investigation might reveal better results if the gene is set to its maximum rather than wrapping it around. The second mutation operator (mutation\_B) shuffles the values among the different loci. This way if a good number is found in one locus, it can spread to other locations more quickly. Mutation on the prediction technique selection gene involves randomly selecting one of the valid predication techniques.

The fitness of each creature is exactly equal to its profit after participating in an auction. A generation level for each GENCO can be determined by the number of contracts that the GENCO was able to obtain during the auction process. Profit is calculated as the total cost to generate at that level, minus the total revenue. Total revenue is equal to the sum of the contract price multiplied by the number of contracts over all rounds of bidding.

At each generation, one half of the population is replaced with the children. Although the parents were not taken strictly from the top half of the population, it is always the creatures on the bottom of the population that are replaced each generation.

#### 3.4.4. Developing the GENCO offer

This subsection explains how to develop an offer from the previously shown data structures. The number of 1 MW contracts is taken directly from the “MWs each round” gene. In the research here, the bid multiplier binary strings can be used in the following two different methods. In the first of these methods, the string is mapped into a range where a string of all ‘1’s would set the offer price equal to the expected equilibrium price (EEP), and a string of all ‘0’s would set the offer price equal to the cost of the generator. In the second method the string is mapped to a multiplier that could range from slightly below one to slightly above one. This multiplier is then multiplied by the EEP such that the offer will be within some tolerance of that EEP (e.g., [number in the range of 0.75 to 1.25] x [expected equilibrium price] ). For the results shown in this section of the chapter the first method was used. Figure 3.2 shows how the data structures can be used to develop an offer. See Figure 3.3 for a block diagram of how the genetic algorithm, the price prediction and the auction processes fit together to evolve the GENCOs that are trading electricity.

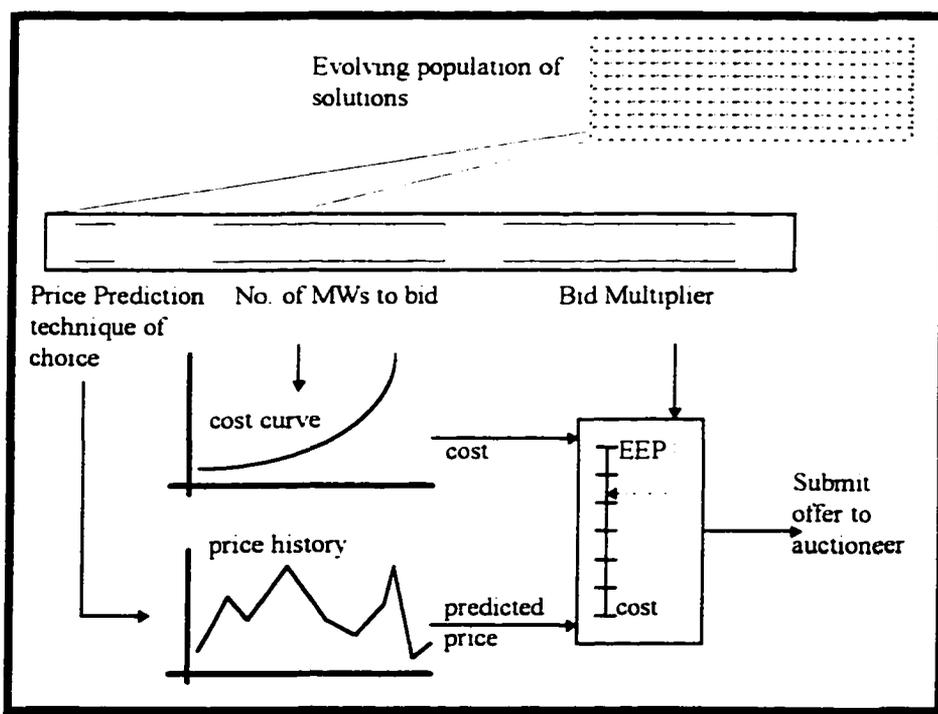


Figure 3.2 Developing the offer from the data structure.

There are two methods included for arriving at the expected equilibrium price. The first method uses no prediction techniques, and simply assumes that the current round's price will be the same as the previous round's equilibrium price. In the absence of additional information, this is a fair assumption given a stable market where the prices do not fluctuate very much. The other method of arriving at the EEP involves using a forecasting technique. A gene in the GENCO's data structure determines which forecasting method to use. The GENCOs may use of any forecasting method available. The techniques which we investigated were the following methods (but any other methods could also be included):

1. moving average (MVA)
2. weighted moving average (WMVA)
3. exponentially weighted moving average (EWMVA)
4. linear regression (LR)
5. Multi-layer perceptron neural network with back-propagation learning

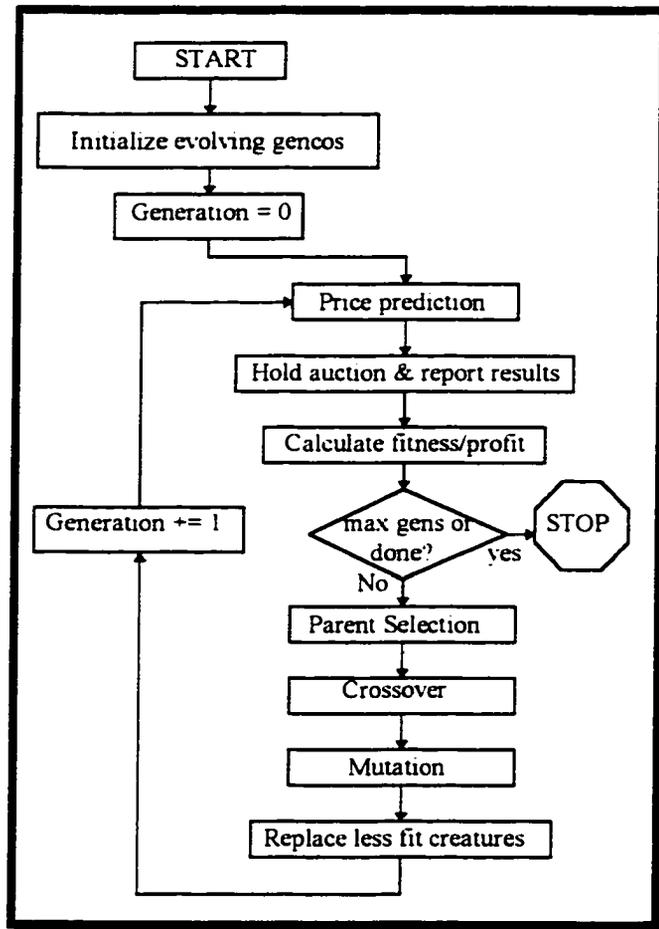


Figure 3.3 The GA agent evolution process.

The MVA, WMVA, EWMVA and LR are standard and can be found in any statistics text book. The inputs to be considered can be adjusted until the best forecasts are observed (minimum mean squared error). It was originally intended that each GENCO would have a unique set of parameters for its own forecasting technique, but tuning these parameters independently was found to be too computationally expensive for the research we were doing. Therefore, each GENCO that uses the same forecasting technique gets the same forecasted equilibrium price. Since performing this research, many new neural network techniques have emerged with reduced training times that would make them more practical for this application.

### 3.4.5 Results

The genetic algorithm was initialized with a rather small population size of 24, i.e., 24 GENCOs. Each generation, 12 new creatures replaced the 12 worst fit of the population. Roulette selection was used to select the parents. Based on engineering judgment, a mutation rate of twenty percent was used at each loci of the new creatures for the standard mutation operation. Mutation\_B was used 50% of the time, and the Mutation\_A was used the other 50%. Three point crossover was used during reproduction. Fitness was taken as the raw profit each GENCO received.

For simplicity, one ESCO was bidding against all of the GENCOs. The buy bid was a constant amount each round. Neither transmission constraints nor system stability violations were considered. The minimum generation of each GENCO is 200 MW; the maximum is 480 MW. Trading occurs for 24 rounds each generation. A maximum of 20 cycles is allowed for price discovery, at which time the round number is incremented. Each of the GENCOs has the same generation cost curve, represented by the following equation (where MW is the generation level):

$$\text{Cost of generation} = [200 + (8.0)(\text{MW}) + (0.00251)(\text{MW})^2] \text{ [Fuel cost]} \quad (3.1)$$

For the case shown in the following figures, the expected equilibrium price was taken to be the previous round's equilibrium price. The bid multiplier was used to bid between the GENCO's cost and the expected equilibrium price. The case shown is a seller's market, where the electricity demand is twice the supply. The ESCO wants 960 MW each round, but the maximum total supply is 480 MW/round. The ESCO bids \$20/MWhr each round. The GENCOs are allowed to evolve for 400 generations.

The graphical results in Figures 3.4-3.6 show that the auction simulator and GA bidding strategy evolver are working in an explicable manner. The average, maximum, and minimum fitness of the GENCO population is plotted on the top graph in Figure 3.6 for 400 generations. The number of MWs actually purchased by the ESCO is plotted in the second graph of Figure 3.6. In the remaining graphs in Figures 3.6, the average, minimum, and maximum offer prices of the GENCOs are plotted for each of the rounds of bidding at each generation. In the bottom graph of Figure 3.6, the resulting equilibrium price is plotted. In

Figure 3.4. the best GENCO's bid multipliers are plotted for the first generation and the last generation. The number of MWhr sold by the best GENCO each generation is plotted in Figure 3.5.

The fitness of the best GENCO/agent generally increases over the generations. The best GENCO is taking advantage of the high demand by increasing the number of MWs it is offering for sale. However, many of the lesser fit GENCOs are not learning that it is beneficial to sell as much as possible. Consequently the ESCO is purchasing less than half of the electricity it would like. Since the buy bid and sell offer are averaged to arrive at the price reported by the auctioneer, the agents exhibiting less savvy bidding behavior tend to keep the equilibrium price around the \$17-\$18/MWhr range. If they were all using the strategy of the best GENCO, in this particular case, the price would be \$20/MWhr. Price fluctuations are fairly small over the generations, but tend to vary a great deal over the rounds during each generation. The average offer is about \$15/MWhr for this case. The bid multipliers of the best GENCO evolved so that the offers were closer to the expected equilibrium price as opposed to the GENCO's cost.

#### *3.4.6 Conclusions of non-adaptive strategy evolution*

The auction simulation is working as expected. The evolution of the GENCOs is also proceeding as expected. The framework used and developed for this research should be a helpful tool for those who will be participating in the competitive electric marketplace of the future. It would also be helpful for those wishing to develop bidding strategies for other types of markets. Several other cases were run that have not been included in this dissertation. Multiple runs verified that the bid multipliers were functioning. The bid multiplier gene evolved to increase fitness while the other genes were held constant. The forecasting techniques must be properly tuned to get good forecasts. If the market is such that the equilibrium price does not fluctuate much, then using the previous equilibrium price works well. Market specific information including knowledge about the volatility can help to fine-tune the parameters for better performance. The GA is able to make use of a more complex data structure. Separate multipliers for each round of bidding evolve to result in a better solution than that derived from a single multiplier for all rounds.

The better agents of a particular generation have a strategy which obtains a higher profit. Since all agents are adapting their strategies at each generation, a particular strategy may prove to work well during one generation, but not so well during a later generation. It is difficult (but of great interest) to determine what makes a particular strategy a good one. Identifying the common features of the best agents at each generation and building an expert system rule base is the subject of continued research. Sensitivity analysis and testing will provide valuable information on the robustness of the best strategies.

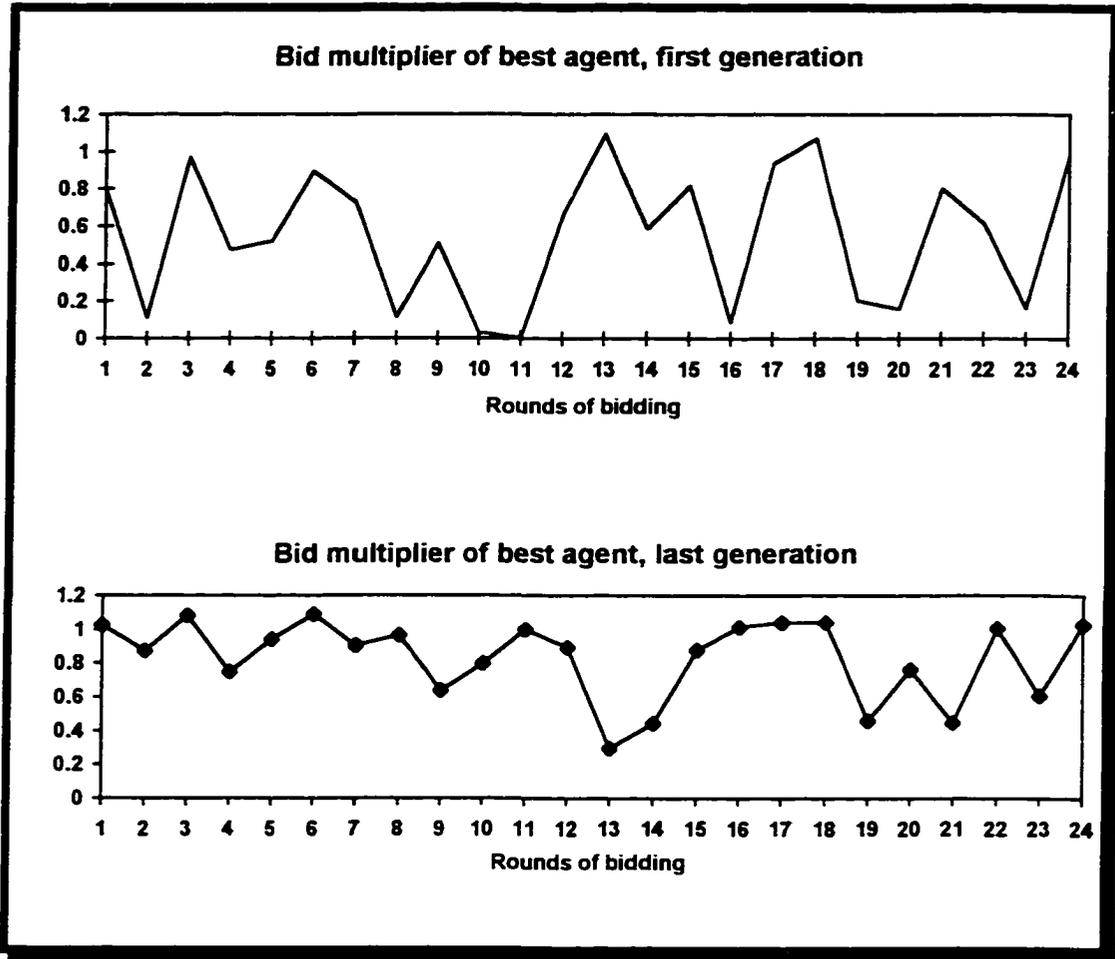


Figure 3.4 Bid multipliers of the best agent at the first and last generation.

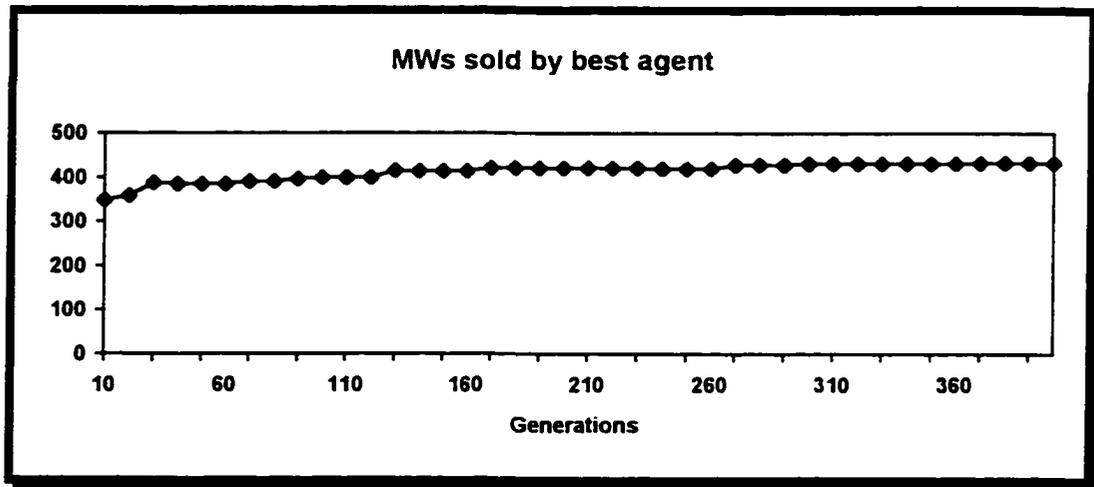


Figure 3.5 MWs sold by best agent each generation.

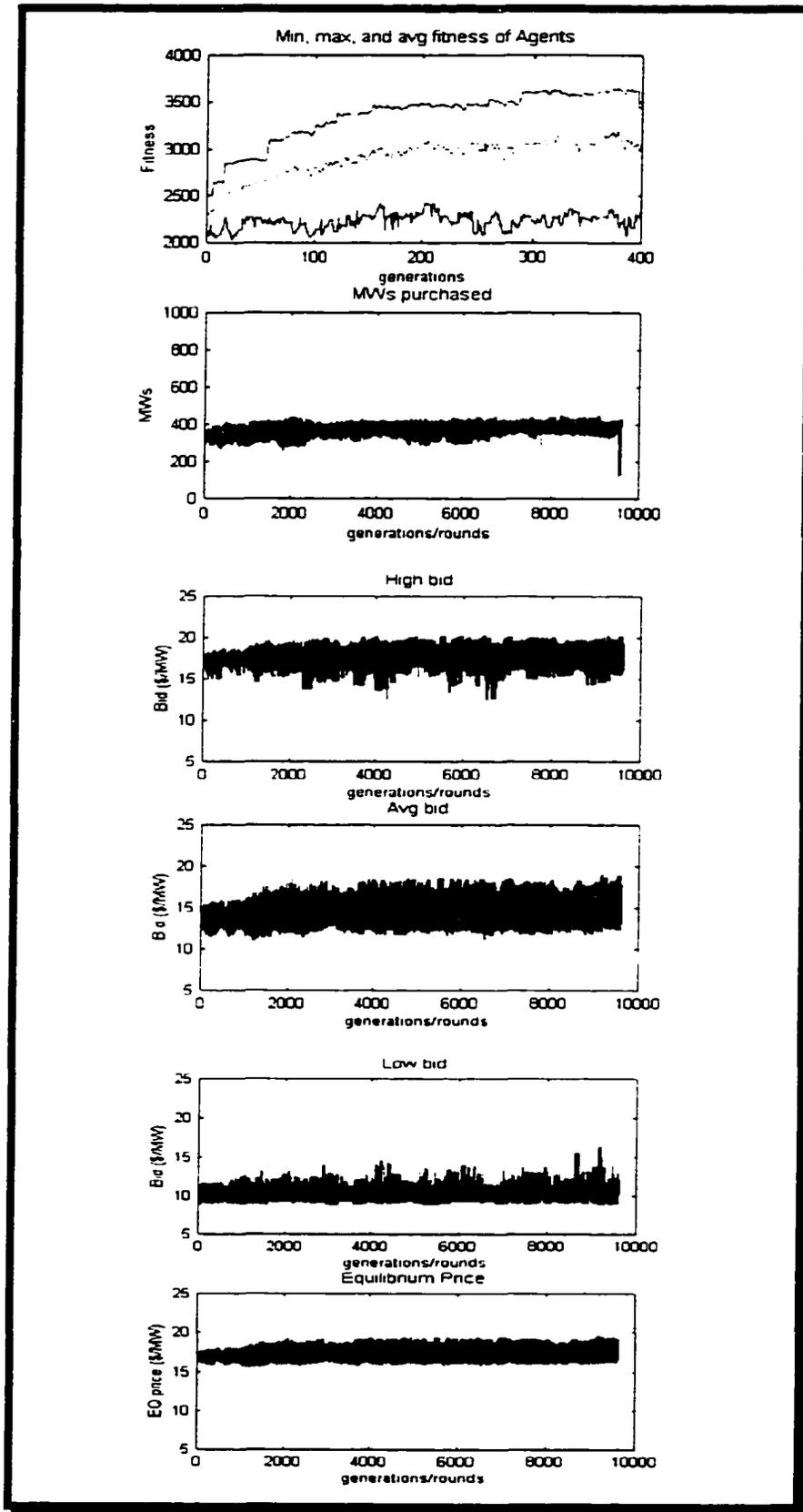


Figure 3.6 Fitness and trading results.

The research contained in this section was a first attempt at evolving bidding strategies. We have built on this research in an effort to build bidding strategies that are adaptive, that use more of the input information, and that are easier to develop. See the chapter on unit commitment for more recent research that uses a more detailed model that considers startup and shutdown costs, ramp constraints, minimum-up and minimum-down times for the generators. The unit commitment work includes separate cost curves for each generating unit. See the chapter on fuzzy logic for fuzzy bidding strategies. The next section of this chapter describes a continuation of this research which includes power marketers that both buy and sell power. It also includes the use of dual populations (GENCOs and ESCOs) that co-evolve. The population of ESCOs has its own bidding strategies that react to the GENCO's offers.

### 3.5 Adaptive bidding strategies with genetic programming and finite state automata

This section describes research on evolving bidding strategies that are adaptive. First popularized by John Koza in his book, *Genetic Programming*, genetic programs allow the designer to get away from the sometimes restrictive notion of a fixed data structure offered by traditional genetic algorithms. The GP field is still burgeoning and new and clever ways of combining genetic programming with other techniques are being developed. One such technique is known as GP-Automata and was developed by a researcher at Iowa State University [Ashlock, 1995]. GP-Automata combine finite state automata with genetic programs. This section of the chapter reports on bidding strategy development with GP-Automata. It builds on the research reported earlier and uses the market structure, assumptions, and genetic algorithm techniques discussed in Chapter 2. We first describe genetic programming and GP-Automata, and then report on several experiments using the GP-Automata technology.

#### 3.5.1 The basics of genetic programming

The process of genetic programming has been called automatic programming and is a sub-class of the genetic algorithm field. Genetic programming is a fairly new discipline and is attributed to Koza [1992]. Typically shown in either parse tree (see Figure 3.7), or S-expression form [e.g.:  $\text{avg}(\text{ite}(\text{sub}(\text{hbb}, \text{cost}), 20, \text{asb}), 10))$ ], genetic programs (GPs) are evolvable programs. Each parse tree contains some number of nodes and branches. The branches connect the various nodes which can be either an *operational node* which has arguments and performs some operation involving those arguments, or a *terminal node* which returns a constant value.

The designer specifies the set of valid operators and terminals suitable to the problem being investigated. For instance, in developing bidding strategies, suitable operators and terminals might be those described in Table 3.1. In designing GPs for the GP-Automata, it is desirable to give the trees an opportunity to return numbers in the range of competitive bids.

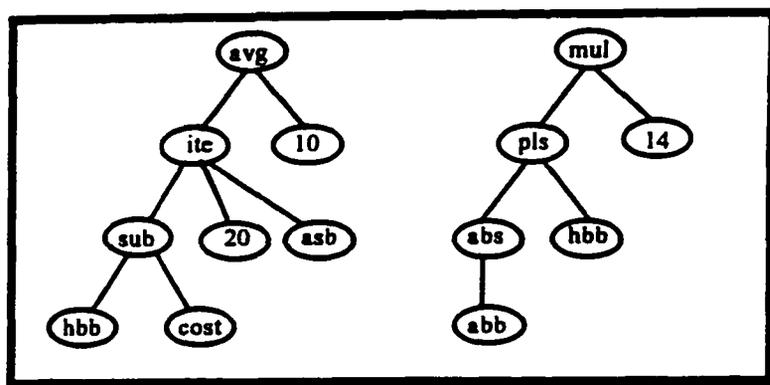


Figure 3.7 Sample GPs.

The GPs are traditionally evolved in a standard genetic algorithm (as described in the previous section) with the following modifications. Crossing over two parents involves randomly selecting a node from each parent and swapping the sub-trees rooted at those nodes. *Mutation* involves randomly selecting a node in the candidate child and throwing away its sub-tree. In its place a new sub-tree is generated randomly. See Koza [1992, 1994] for a more detailed look at GP.

Table 3.1 Valid operators and terminals for the GP.

Name	Type	Args	Description
gte	oper	2	Return 1 if 1st arg is $\geq$ 2nd arg; otherwise return 0.
ltn	oper	2	Return 1 if 1st arg is $\leq$ 2nd arg; otherwise return 0.
ite	oper	3	If 1st arg is even after truncation, return 2nd arg; else, 3rd arg.
abs	oper	1	Returns absolute value of arg.
mid	oper	2	Returns average of the two args.
mul	oper	2	Returns multiplication of 2 args
pls	oper	2	Returns addition of 2 args
sub	oper	2	Subtracts 2nd arg from 1st arg
max	oper	2	Returns the larger of the two args
min	oper	2	Returns the smaller of the 2 args
cyc	term	0	Returns current cycle number
cst	term	0	Returns the cost of gen. for the bid
asb	term	0	Returns the average sell bid
hsb	term	0	Returns the max sell bid
lsb	term	0	Returns the min sell bid
abb	term	0	Returns the average buy bid
hbb	term	0	Returns the max buy bid
lbb	term	0	Returns the min buy bid

### 3.5.2 The basics of GP-Automata

GP-Automata combine finite state automata with GPs. They were first described as such by a researcher at Iowa State [Ashlock, 1995] and were later used in experiments [Ashlock and Richter, 1997]. The typical finite state automaton specifies an action and “next state” transition for a given input or inputs. With only one or two binary inputs to work with, it can be fairly simple to develop a finite state diagram to cover the possible input/output relations. When the number of inputs is large, the task is much harder. The number of transitions needed to cover all possible combinations of inputs grows exponentially (e.g., 10 inputs each having 5 possible values would require  $5^{10}$  transitions). This is where genetic programming comes in. The GP-trees are bandwidth compressors. They are used by the GP-Automata for selecting which inputs to consider and for performing computations involving these inputs. See Figure 3.8 for an example of a GP-Automaton.

State	IF ODD		IF EVEN		GP (Decider)
	Action	Next State	Action	Next State	
1	14.5	1	U	1	lte (mul(10, abs ( hbb ))
2	*	1	37	3	ite(max(10, asb), hbb, lbb)
3	12	2	5	1	split (5, abb)
4	U	3	*	2	47
Initial Action		24	Initial State		2

Figure 3.8 A four state GP-Automaton.

Reading the rule encoded in the GP-Automaton in the figure is fairly simple. We see that this automaton begins by bidding the number in the ‘initial action’ field. Following the initial action, the ‘initial state’ tells us which state we would use next (in this case, 2). The GP-Automaton in the figure has four states. Coupled with each of these states is a GP-tree termed a decider. When executed, the decider returns a value between 0 and 100. Based on that returned value, one of the following two things will happen: (a) if that value is *even* after truncation, the action listed under ‘IF EVEN’ is taken and we move to the next state listed under ‘IF EVEN’; (b) if the returned value is *odd* after truncation, then we use the action and next state listed under ‘IF ODD’. The ‘action’ is the number listed in the action field of the automaton, with two exceptions. The first exception is the ‘U’ which indicates that the value returned by the decider should be taken directly as the action. The second exception is a ‘\*’ which indicates that further computation is necessary and hence the GP-Automata refrains from acting immediately. Instead, it immediately moves to the next state. This gives rise to the possibility of complex (multi-state) computation as well as infinite loops. To prevent infinite loops, after an externally specified maximum number of ‘\*’s, an action is selected at random from actions uniformly distributed over the valid range.

In this chapter we evolve a population of GP-Automata in a GA. After selecting parents as described later, offspring are produced using crossover and mutation. Crossover for the GP-Automata involves selecting (with a uniform probability) a crossover point ranging from zero to the number of states. We then copy first parent's states from zero to the crossover point to the first child and the second parent's states to the second child. Following the crossover point, the first child gets the second parent's state information and the second child gets the first parent's state information (including the associated decider). Before replacing less fit members of the population, each child is subjected to one of the following four types of mutation:

- MutationA replaces a randomly selected state or action with another valid entry.
- MutationB swaps the intact deciders of two randomly selected states within a candidate child.
- MutationC performs the GP crossover on two randomly selected states from the candidate child.
- MutationD generates an entirely new decider for a randomly selected state within the candidate child.

### *3.5.3 Using the strategies in an auction*

In each generation, the performance of each GP-Automata in the population is tested by using the strategy in an auction. GP-Automata representing seller strategies from the first population attempt to contract with GP-Automata representing buyer strategies from the second population. The price and the resulting profit is a function of all bids and offers, not just the bids produced by an individual GP-Automaton. Randomly assigning members of the population so that they aren't always paired with the same group of buyers and sellers helps to ensure that the resulting GP-Automata strategies will be robust. At the end of the competition, some fitness measure (e.g., profit from the contracts) is assigned to each GP-Automaton.

During the first cycle of bidding in an auction, the strategy defined by the GP-Automata uses the bid specified in the 'Initial Action' cell and goes to the state listed in 'initial state'. For each subsequent cycle of bidding the results of the previous cycle of bidding are available as inputs to the GPs. These inputs from the environment are supplied to the GP-Automata through the terminals listed in Table 3.1. The GP-trees use the statistics on the previous round's bidding stored in the terminals as well as numerical terminals in the range 0-100. For the experiments described in this chapter, the continuum from 0 to 100 was evenly divided into 10,000 discrete values. When these values were required to be an integer value, the number was truncated after the decimal point. Bids are taken from the action cell of the automata, except in the cases where the action is listed as a '\*' or a 'U', as described previously. The bids are submitted, along with the bids from the competing sellers and buyers, to the auctioneer for evaluation. The bids and offers are matched and a would-be price is reported, completing one cycle of the auction. The cycles continue until price discovery occurs or until some maximum number of cycles (maxcycles) has passed. There is a maxcycles parameter, which is selected uniformly over a range to prevent the strategies from falling into a local optima in which the strategies work well when the number of cycles is identical over the trials in a given generation.

#### *3.5.4 Evolving and testing the strategies: A first attempt*

The following parameters were used for the results presented in this section. The population size (i.e. number of GP-Automata in the population) was selected to be twenty-four. We used the roulette selection, replacing eight members of the population each generation. During every generation of the GA, each GP-Automaton's fitness was calculated by having it participate in an auction fifty times as a seller (GENCO) with five other sellers offering to supply 10 MWhr, and five buyers demanding 15 MWhr each. Each automaton in the population is attempting to sell 10 MWhr of electrical energy. The auction used for this experiment uses non-discriminating pricing, which means that each of the contracts awarded in a particular auction have the same equilibrium price. The \$5/MWhr cost associated with generating the 10 MWhr is the same for each automaton being tested in our population. Each automaton being tested faces the same 50 sets of bids and offers encountered by the other automata in the population. The GA was evolved for 35 generations. Each experiment/case was repeated twenty times, and the results shown in this section are averaged across those twenty runs.

In case 1, the buyers' bids are all set to a constant \$15/MWhr, and the sellers' offers are set to a constant \$5/MWhr. The top graph in Figure 3.9 shows the min., max., and average fitness of the automata in each generation of the GA. Because (as described previously) the maximum number of cycles is not always the same, the max. fitness will sometimes decrease from one generation to the next. The bottom graph in Figure 3.9 shows a histogram of the bids placed by our GP-Automata as they participate in the auction. We can see that the GP-Automata are learning to bid close to their cost of generation, which is \$5/MWhr. (Note: In Figures 3.9, 3.10, and 3.11 the frequency of bids within a certain tolerance have been lumped together and shown with a single row of bars since the data are hard to read when including histograms for all possible bids. Bids shown on the far left of the histograms correspond to \$0/MWhr, and bids shown on the far right correspond to \$20/MWhr, while the middle of the graph corresponds to \$10/MWhr, and so forth.)

In case 2 (shown in Figure 3.10), the offers of the competing sellers are distributed about \$5/MWhr and the buyers' bids are distributed around \$15/MWhr as shown in the lower graph of Figure 3.10. This makes it harder for the GP-Automata to develop a general strategy that works against all competitors, and the result is that their maximum fitness is lower than that of case 1. Their bids tend to be at or below cost which, when matched with buy bids as shown in the lower graph of Figure 3.10, results in some amount of profit. As we saw in case 1 and as we'll see in case 3, in case 2, they learn very quickly that they must not be underbid by the other generators, hence we see no successful strategies with bidding above \$5.

In case 3, which is shown in Figure 3.11, we initialize the competitors' bids for the first cycle of each auction with the same distributions as before, but now the distributions move toward each other over time. As the cycles pass, bids decrease and the offers increase, becoming a bit more competitive. The histogram for case 3 looks much like that of case 2, but the portion of the bids that are below the cost tend to be a bit more uniformly distributed, whereas in case 2, the closer the bids got to the cost from the left, the more common was

that bid. One explanation for this could be that since the ESCO bids are decreasing over the cycles, and since the equilibrium price will be determined by these low ESCO bids as well as our own bid, that any successful GP-Automaton-based players must learn to be near their cost on the first cycle. Contracts forming in subsequent cycles invariably result in a loss, and these members of the population are replaced. Another explanation may be that when the competitors' bids are moving, learning what to bid becomes too difficult and the GP-Automata are unable to make any headway. Comparison of the top graph in Figure 3.9 with those of Figure 3.10 and Figure 3.11 shows that the profit obtained by the GP-Automata decreases as the bids and offers of their competitors become more noisy. The profit level decreases further in case 3 where the competitors' bidding becomes more aggressive.

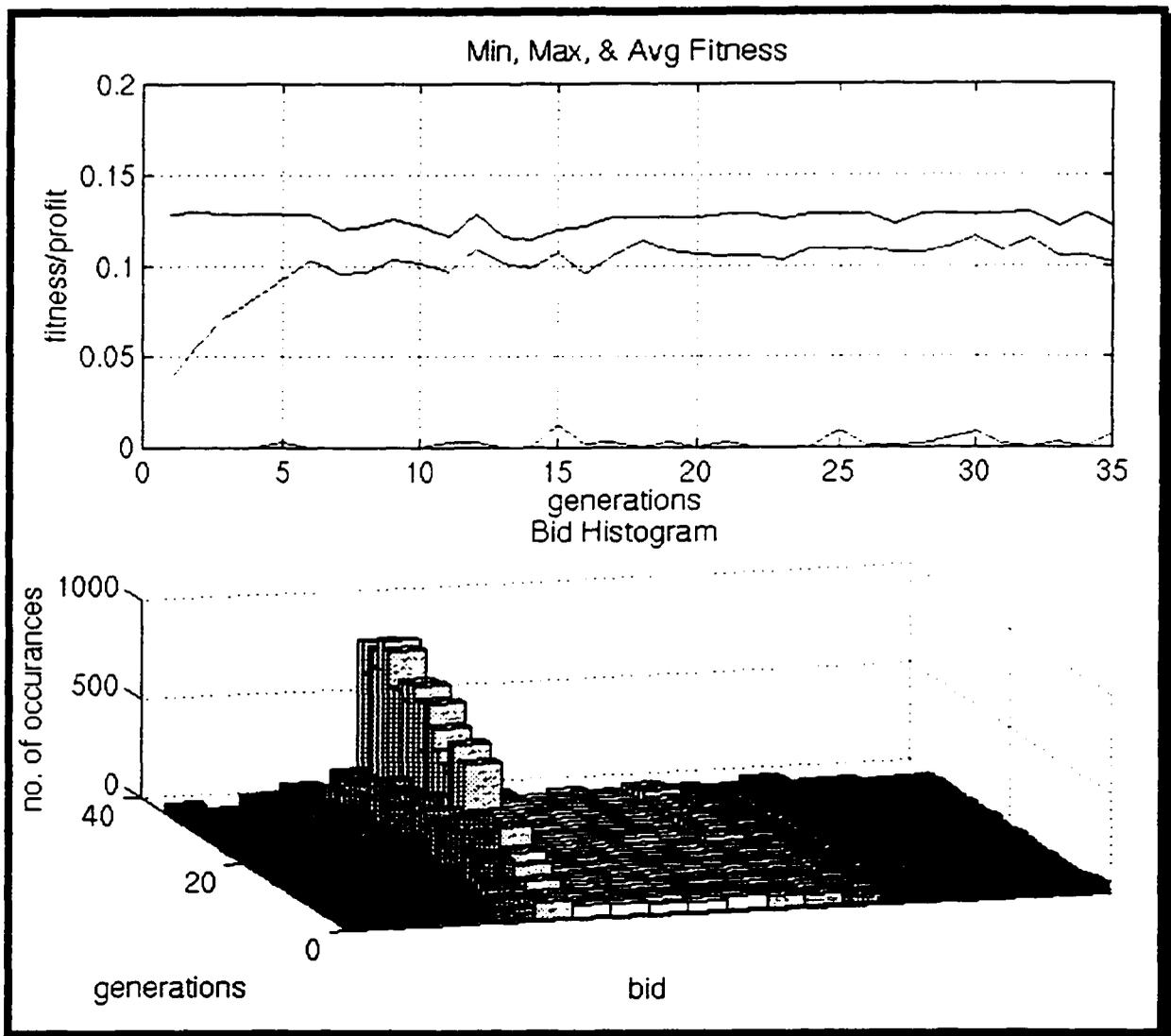


Figure 3.9 Case 1: Competitor buy bids at \$15/MWhr and sell offers at \$5/MWhr.

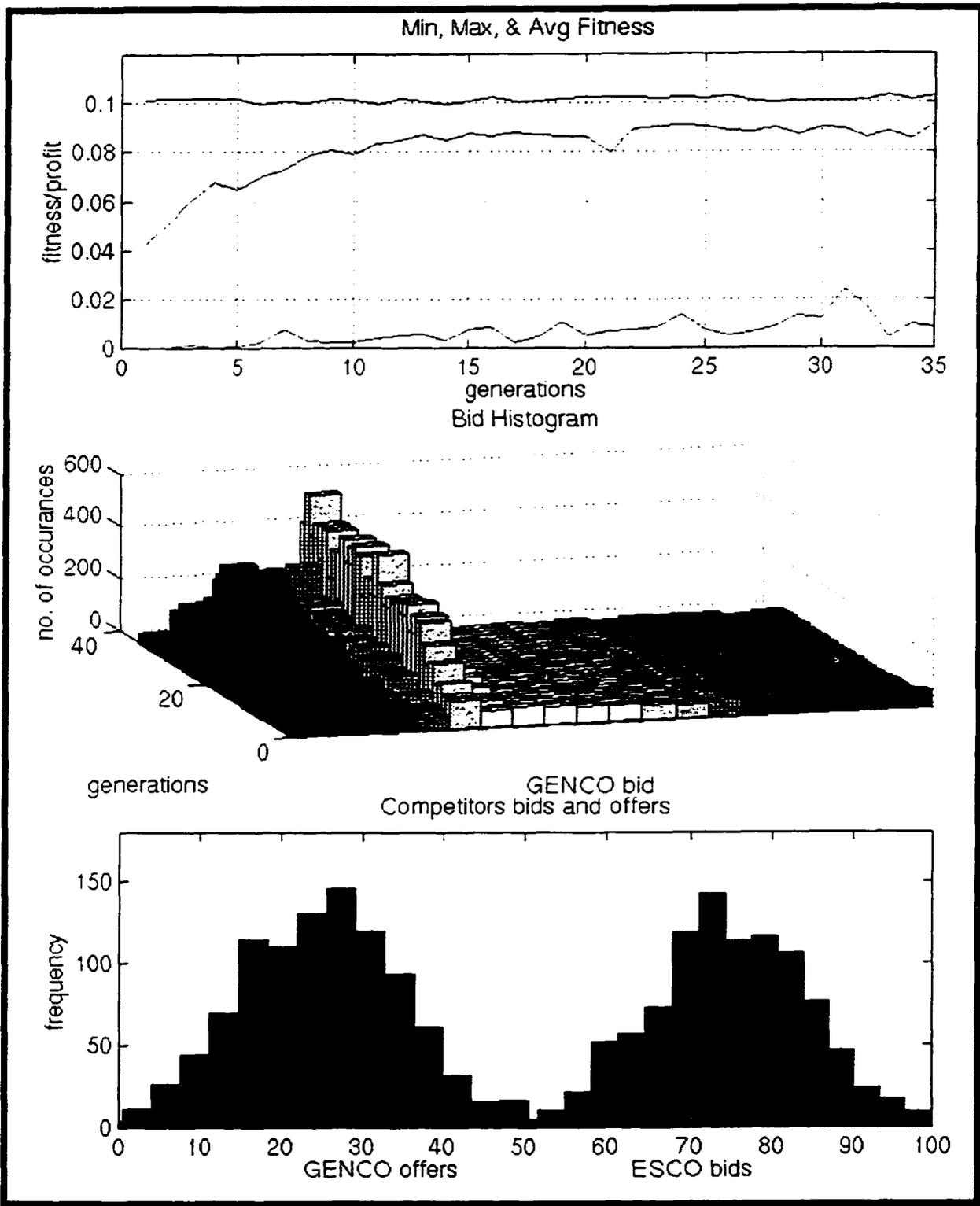
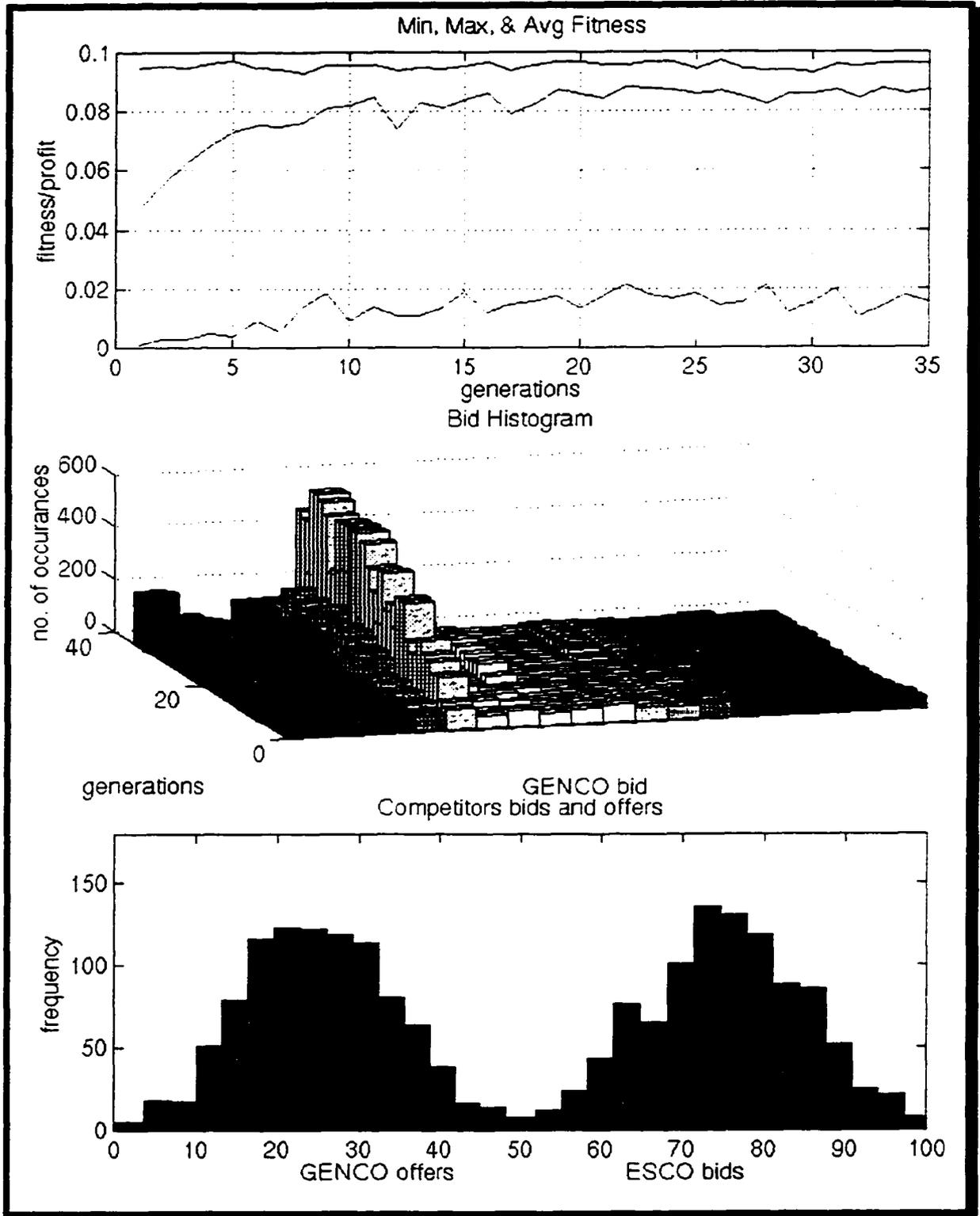


Figure 3.10 Case 2: Competitor bids distributed as shown.



**Figure 3.11 Case 3. Competitors' bids/offers moving toward together over cycles of the auction.**

In this experiment the results demonstrate that GP-Automata learn to bid in a sensible and explicable manner. The GP-Automata lend themselves well to scenarios where there are vast amounts of data available, and identification of crucial data is important. The company models used in the simulations described in this section of the chapter were fairly straight forward. A sensitivity analysis should be performed to see how the various parameters affect the performance of the strategies that are constructed. Among these parameters are the parent selection methodology, the population size, modifications on the auction, and testing the effect of tree size and reduction of the number of states.

### *3.5.5 An experiment to test the effect of GP-Automata tree size and state number*

Initially engineering judgment was used to arrive at the number of states and the size of the deciders in the GP-Automata. This section describes a first attempt to perform a test on the sensitivity of these two parameters. As in the last section, many of the parameters used here are set fairly closely to parameters used in the original Divide the Dollar experiment. Except for the parameters being tested, the GP-Automata were set up identically as in the previous section. However, in this experiment, two populations were used; one population was for the GENCOs and the other for the ESCOs. For the control case, each of the GP-Automata was allowed six states/deciders. They were allowed to have GP-trees with no more than 16 nodes. We used tournament selection with tournament size four and a population size of 24. The deciders were allowed to use those terminals and operators described in Table 3.1.

The double auction in which the strategies were tested matches 5 buyers and 5 sellers. While the GP-Automata are being tested for fitness, they compete in an auction: 5 members of their own population, against 5 members of the other population (not necessarily distinct). Since each strategy's performance depends on who it is competing with, buyers and sellers are arranged so that in each column, all buyers or sellers will appear exactly once. An example is given in Table 3.2. This is not an exhaustive testing of all groupings, but rather an indicative sample. After selecting a grouping (row) of sellers, the buyers are assigned as shown on the right side of the table. That row of sellers competes against each grouping of buyers once. Each GP-Automaton is assigned fitness in proportion to the profit it made from resulting contracts. At the next grouping of sellers, the buyers are reassigned in the same manner, and the process continues.

To put the GP-Automata on equal footing, the problem is constructed to be symmetric for the buyers and sellers. Each buyer will receive the same \$15/MWhr rate from his customers (\$15/MWhr is about 3/4 of the way from left to right since \$0/MWhr is shown on the left and \$20/MWhr is shown on the right) for the electricity it buys in the auction, and each seller is able to produce electricity at the same \$5/MWhr cost. Profit is calculated in the usual way. For the sellers, profit is the equilibrium price minus the cost of production, multiplied by the contract amount. For the buyers, profit is the revenue received from their customers minus the equilibrium price, multiplied by the contract amount.

**Table 3.2 Assignment of buyers and sellers to an auction.**

Sellers					Buyers				
21	19	4	3	21	23	14	20	5	9
4	11	2	7	16	7	24	13	1	3
3	22	16	24	17	10	11	8	6	23
...	...	...	...	...	...	...	...	...	...
12	9	15	8	5	19	1	3	17	1

Although more cases were actually run, three cases are presented in the results section of this chapter. The first is a control case, the results of which are shown in Figures 3.12 and 3.13. The control case has a population size of 24 GP-Automata for both the buyers and the sellers. Each GP-Automaton has six states. Each decider in the GP-Automata is limited to a maximum of 16 nodes. The max., min., and avg. fitness and a histogram of bids averaged across all 40 runs is shown in Figure 3.12. The histogram shows the number of bids that occurred within a particular range (the \$0 - \$20/MWhr range is divided into 20 parts) at each generation. To demonstrate the type and quality of information that goes into getting these averages, an individual run selected at random from the control case is shown in Figure 3.13. We can see from the fitness/profit plot on top that the ESCOs were making more profit than the GENCOs early on, but this switches about generation 10 when the GENCOs learn to bid better. Note from the histograms that both buyers and sellers favored a small number of the total possible numbers for their bids.

In the second case we limited the sellers to a maximum of only one node in their deciders. The buyers remain as described in the control case. Figure 3.14 shows the results averaged across 40 runs for case 2. Averaged across the 40 runs, the maximum, minimum and average fitness are very similar to the control case. The histogram shows that the GENCO bids seem somewhat less normally distributed than are the bids in the control case, with what seems to be a tendency to bid around \$10/MWhr. One explanation for the strong concentration of bids around \$10/MWhr is that the GP-Automata cannot build complex formulas from which to develop sophisticated bids. If the bids are to be taken to be the output of the decider, they must settle on either a fixed number, or one of the other terminals described in the operator and terminal table. Figure 3.15 shows an individual run selected at random from all of the runs in case 2. During the first 10 generations or so, the ESCOs are taking the bulk of the profit, when at that time the GENCOs learn that they can bid around \$15/MWhr since this is a market which favors the seller.

The third case presented in this chapter limits the number of states that the sellers can have to one. The deciders for the sellers are limited to a maximum of 16 nodes. The buyers remain as described in the control case. The three graphs in Figure 3.16 show the results averaged across 40 runs for case 3. Again the fitnesses were very similar to the control case. The histogram shows that the bids were distributed in much the same way when averaged across all runs, but there is a tendency for them to be less exploratory within a particular run (See Figure 3.17.)

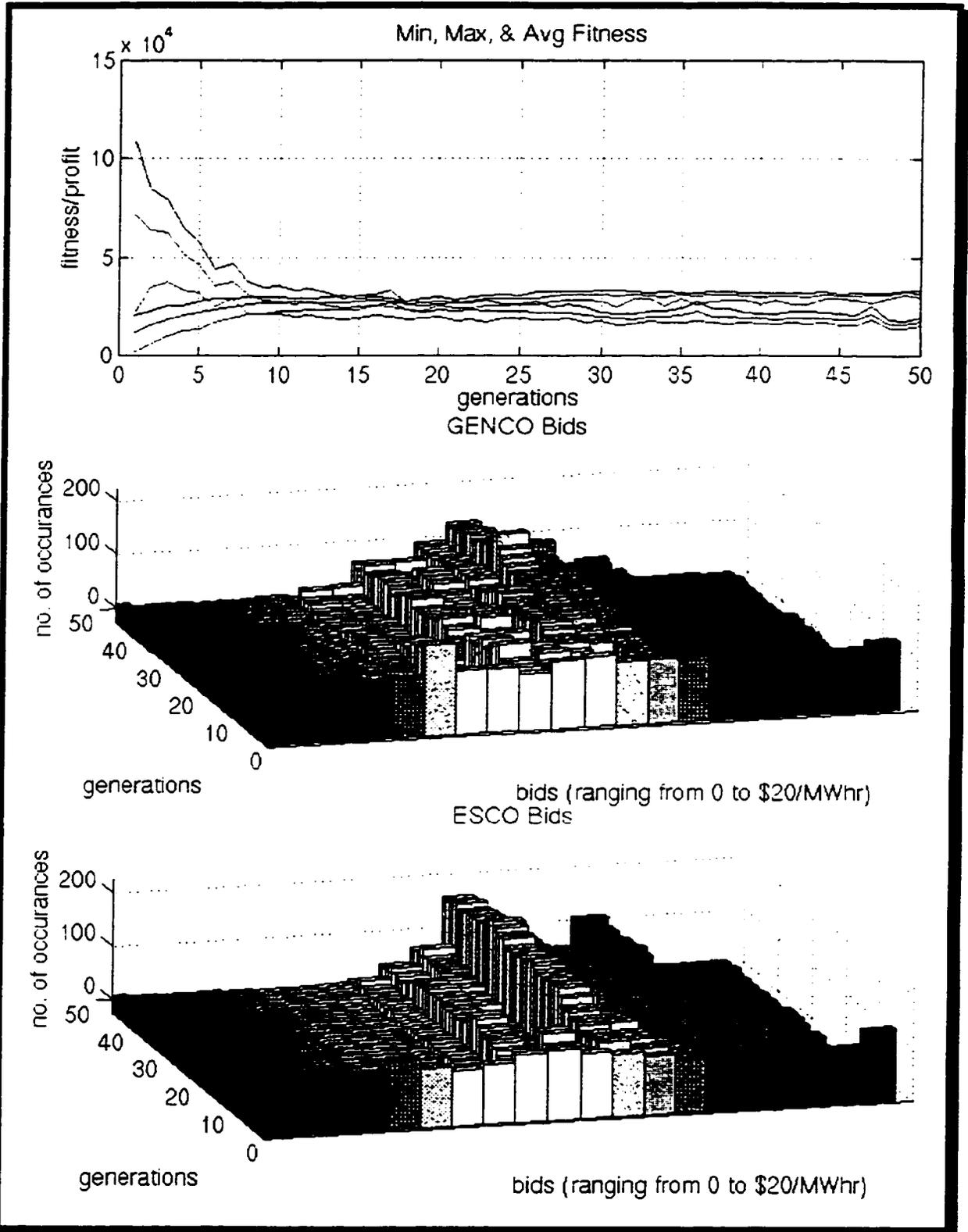


Figure 3.12 Results for the control case.

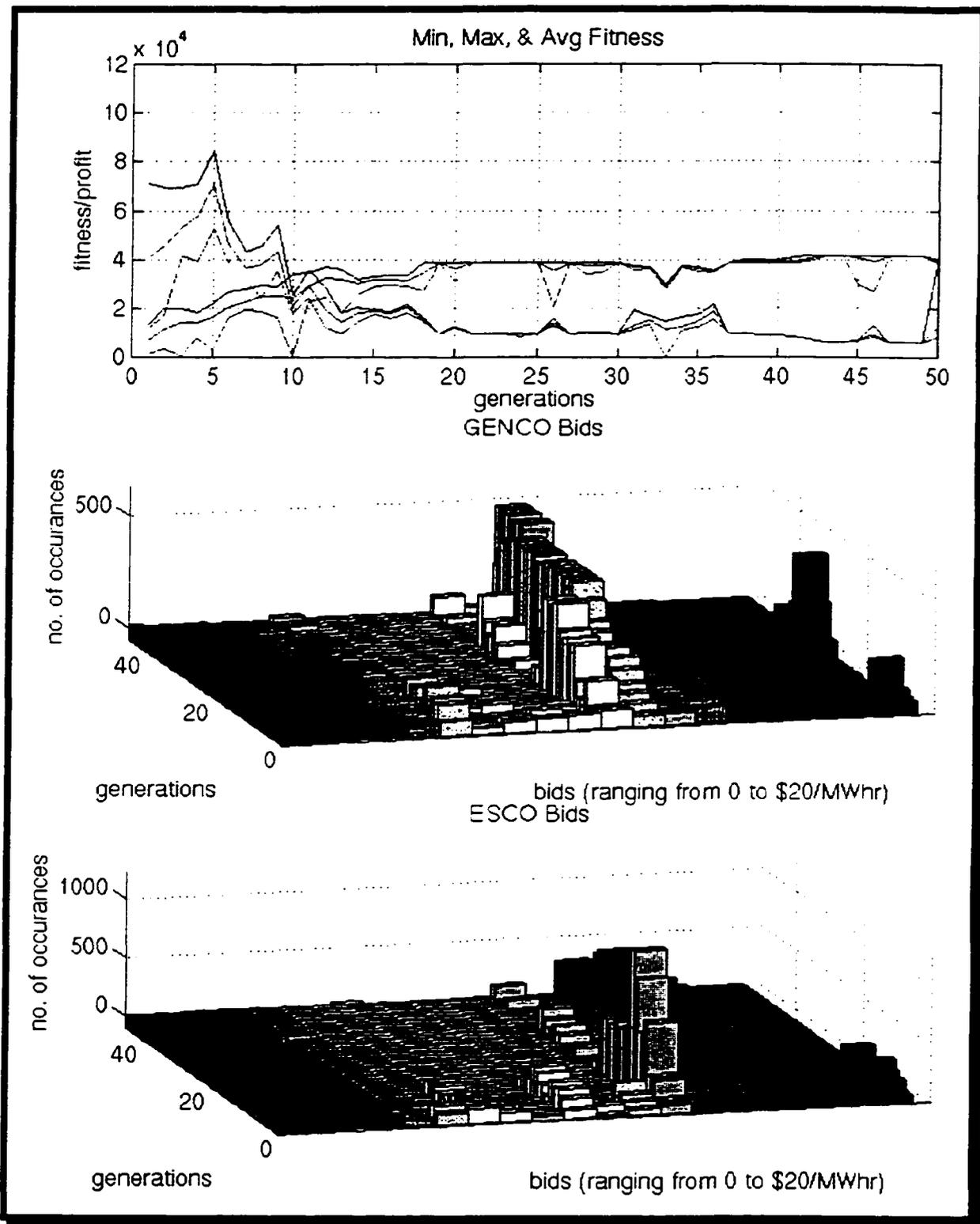


Figure 3.13 An individual run selected at random from the control case.

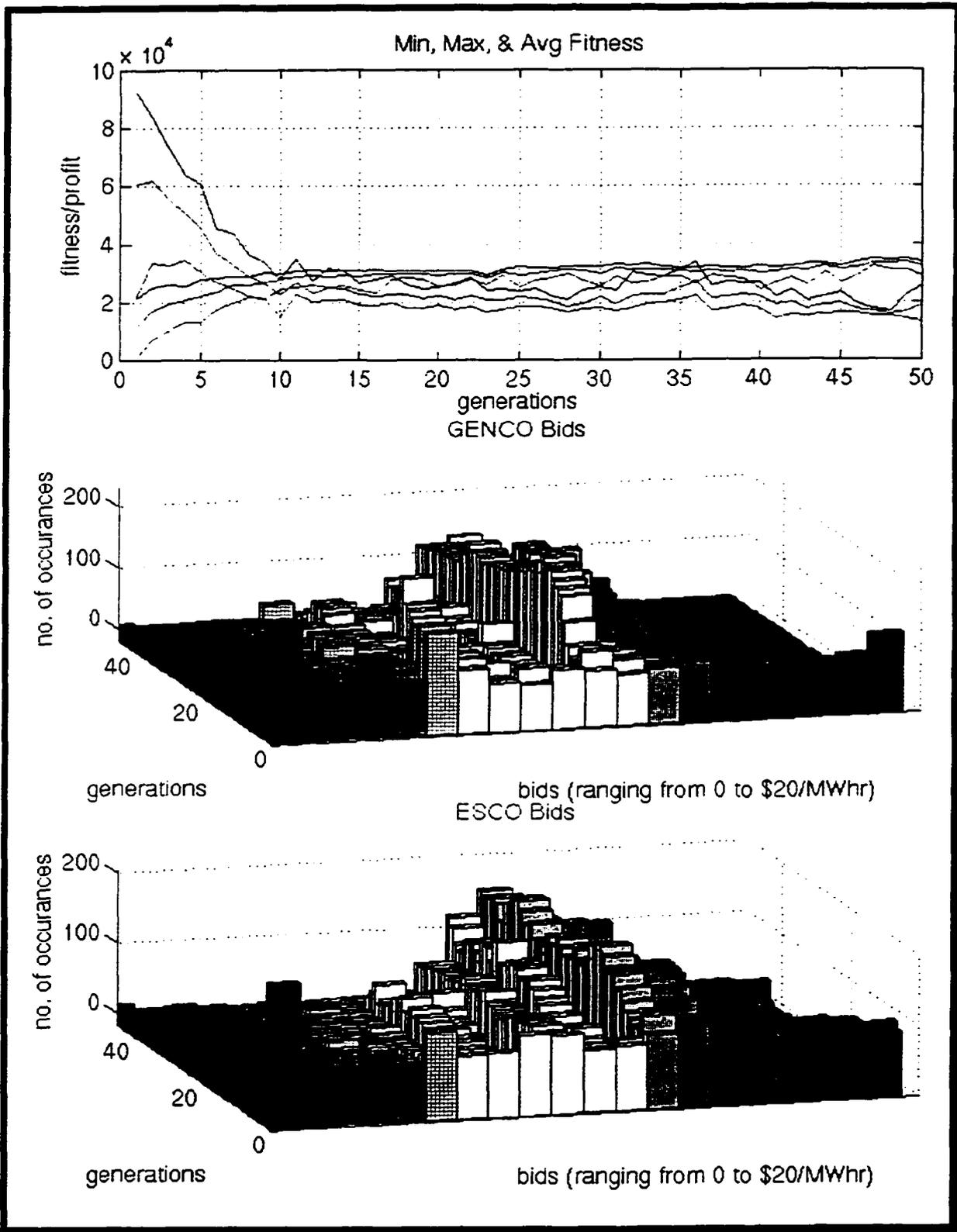


Figure 3.14 Case 2, averaged across 40 runs. GENCOs are limited to one node GPs.

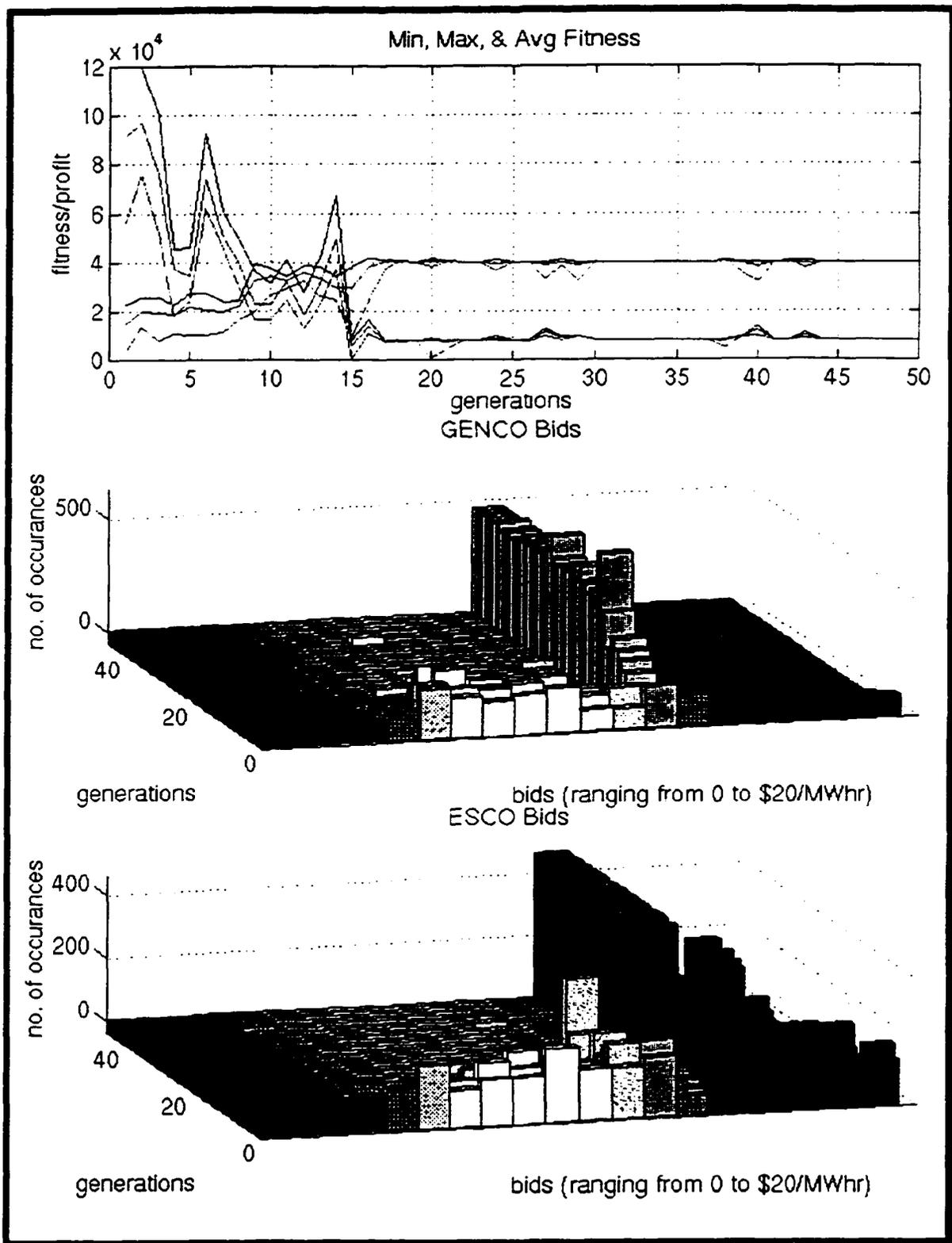


Figure 3.15 An individual run selected at random from the one node GP GENCO case (case 2).

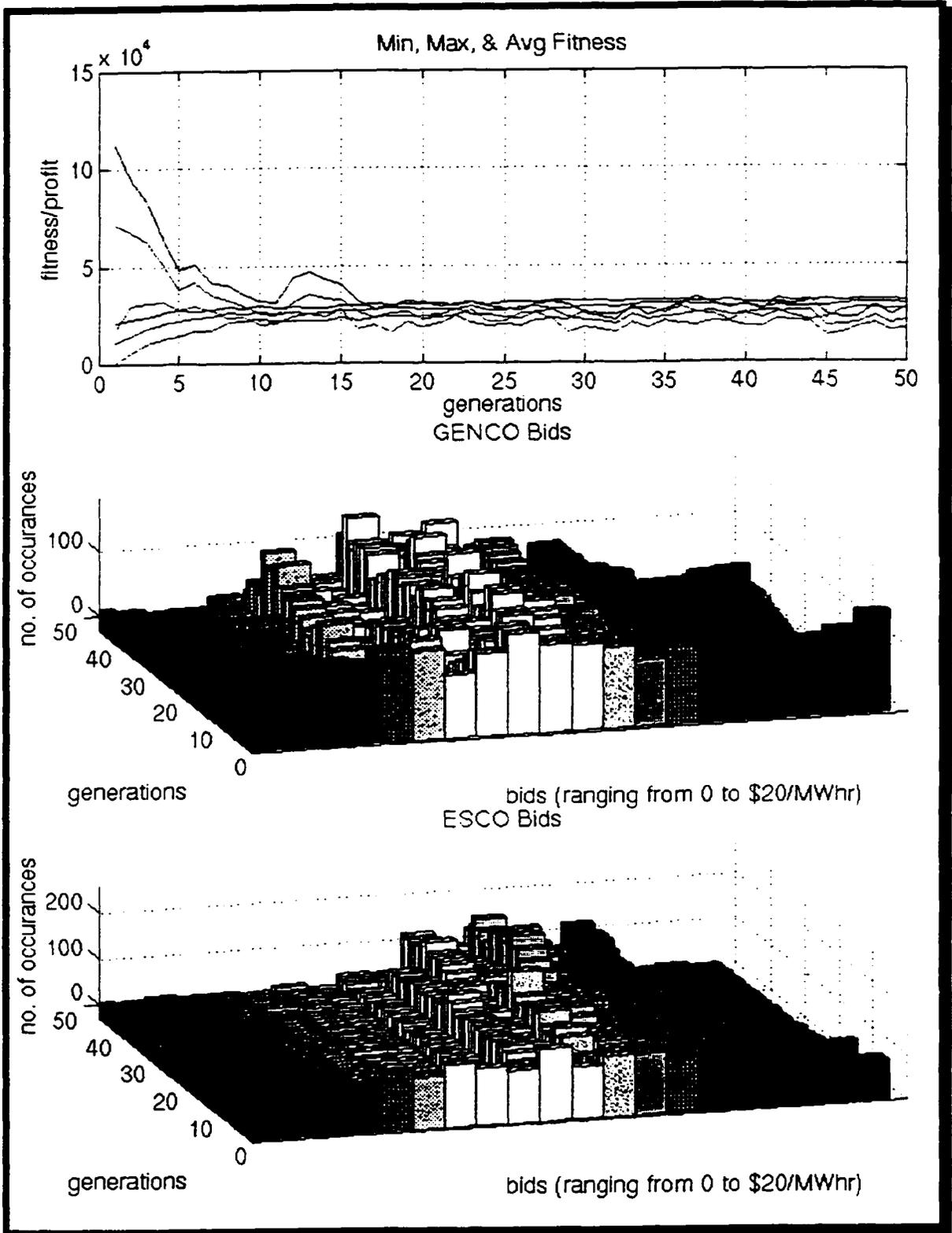


Figure 3.16 Case 3 averaged across 40 runs. GENCOs are limited to having only one state.

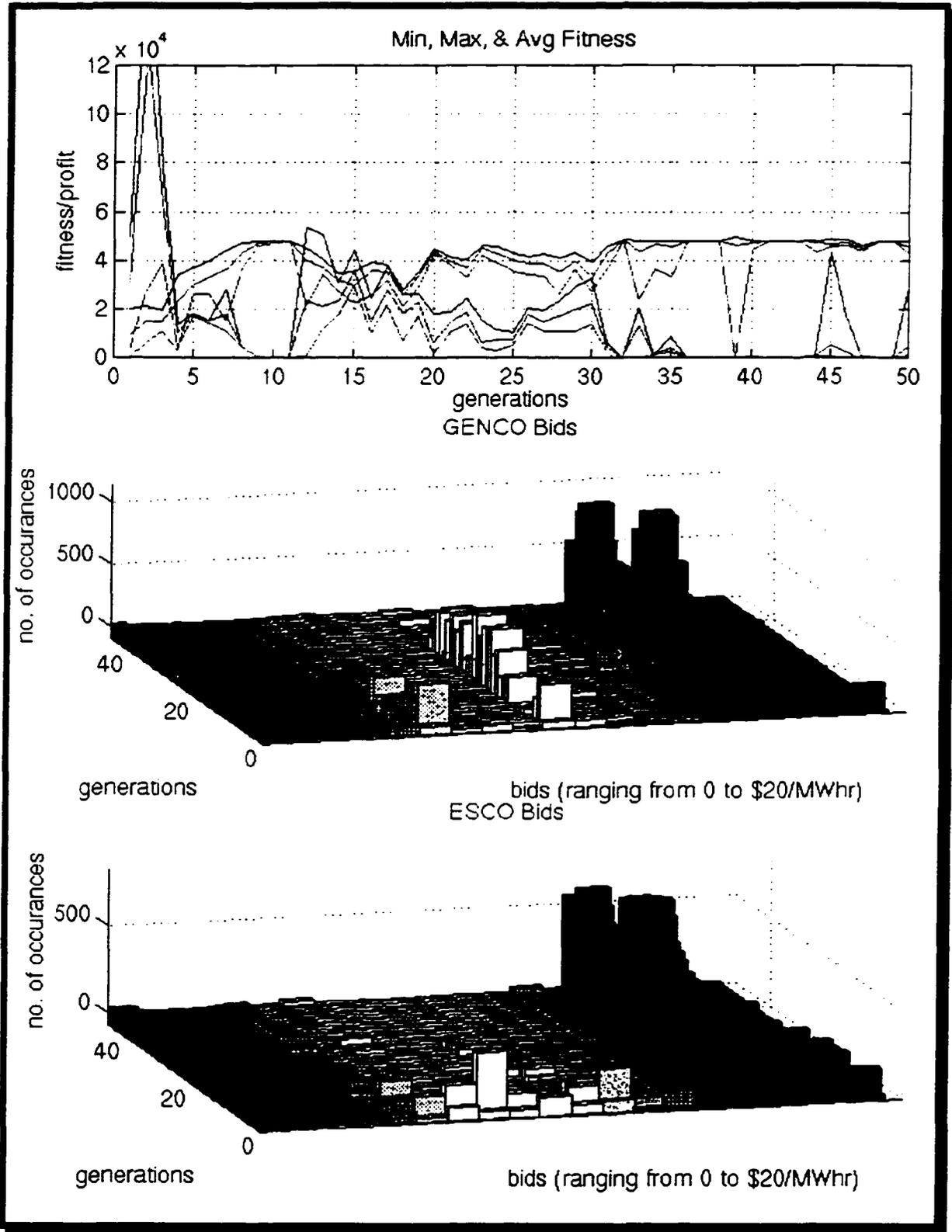


Figure 3.17 An individual run selected at random from the single state GENCO case (case 3).

### **3.6 Chapter conclusions and future research**

As stated earlier in the chapter, the results demonstrate that when GP-Automata bid in a multi-participant auction, they learn to bid in a sensible and explicable manner. The GP-Automata lend themselves well to scenarios where there are vast amounts of data available, and identification of crucial data is important. The company models used in the simulations described in this chapter were fairly straight forward. Adding more detail (e.g. available transfer capabilities, forecasted prices, unit commitment schedules) will increase the volume of information that the bidder needs to consider in making a bid. The fact that GP-Automata suffer very little degradation in performance when they are limited by number of states or by tree size gives some indication of how powerful the method is.

The auction that we have used is more realistic for the coming deregulated electricity marketplace than in previous research [Andrews and Prager, 1994; Ashlock, 1995; Ashlock and Richter, 1997], but more details remain to be added which will complicate matters. We plan to test in further experiments that the GP-Automata are able to make use of this additional data to increase the performance of the strategies. As Figure 2.7 shows, traders have to deal in more than one market. Extending the strategies to cover multiple markets is another area of future research.

In addition to adding the above details, a more complete sensitivity analysis should be performed to determine how the various parameters affect performance of the strategies that are constructed. Among these parameters are the parent selection methodology, the population size, and modifications on the auction.

## **4 FUZZY LOGIC IN THE COMPETITIVE ENVIRONMENT**

### **4.1 Chapter overview**

In the first part of this chapter, previous research done in the area of building bidding strategies for electric utilities in the competitive environment is extended. The reader is given the basics of fuzzy logic. A description of how fuzzy logic can be utilized to make bidding strategies adaptive is presented. Four methods for building bidding strategies that use fuzzy logic and/or genetic algorithms are discussed and outlined. This research discusses how economical inputs may be fuzzified for use in determining a generator's bid. Methods of tuning and searching for the optimal rule are discussed. A discussion of how an agent using the bidding strategies can compare them based on profitability is presented.

Automatic generation control is an important service provided by electric utilities in today's marketplace. Responding to mismatches between the scheduled interchanges between areas, it allows for load and generation uncertainties. Careful control of this service is likely to become more important than ever in the future, and may be delivered and priced as an ancillary service separate from electric power. In the second part of this chapter, fuzzy rules for automatic generation control of an electric power system are developed and presented. Interchange error, time error, and area control error are used in conjunction with fuzzy rules to control generation levels in a two-area system. The fuzzy controller described here outperforms a traditional controller connected to a system with the same generator parameters and same load changes and also performs better than a fuzzy automatic generation controller that uses only one input.

### **4.2 Using fuzzy logic to build bidding strategies that handle uncertainty**

#### ***4.2.1 Introduction***

Due to recent deregulation intended to bring about competition, the US electrical industry is in the midst of some major operational changes. Although the details of the deregulated marketplace for each region of the country are not yet fully defined, they are being more clearly defined as time passes. Many legislators, researchers, and electric customers and suppliers are convinced that electricity will be traded in a manner similar to other commodities at exchanges around the country.

Configuration of the transmission system and the fact that electricity flow is subject to the laws of physics, have some speculating that we will see the formation of regional commodity exchanges that would be oligopolistic in nature (having a limited numbers of sellers). Others postulate that the number of sellers will

be sufficient to have near perfect competition. Regardless of the actual level of the resulting competition, companies wishing to survive in the deregulated marketplace must change the way they do business and will need to develop bidding strategies for trading electricity via an exchange.

Economists have developed theoretical results of how markets are supposed to behave under varying numbers of sellers or buyers with varying degrees of competition. Often the economical results pertain only when aggregating across an entire industry and require assumptions that may not be realistic. These results, while considered sound in a macroscopic sense, may be less than helpful to a particular company not fitting the industry profile that is trying to develop a strategy that will allow it to remain competitive.

Generation companies (GENCOs), energy service companies and distribution companies (DISTCOs) that participate in an energy commodity exchange must learn to place effective bids in order to win energy contracts. Microeconomic theory states that in the long term, a hypothetical firm selling in a competitive market should price their product at its marginal cost of production. The theory is based on several assumptions (e.g., market players will behave rationally, market players have perfect information). These assumptions may be true industry-wide, but might not be true for a particular region or a particular firm.

Section 4.1.2 of this chapter describes the deregulated market-place to be considered during this research. Section 4.1.3, describes previous research on evolving bidding strategies for generation companies using genetic algorithms. We build on the idea and begin to discuss some of the advantages of strategies that are adaptive. Section 4.1.4 provides the reader with the basics of fuzzy logic, and looks at how the economic inputs of DISTCOs and GENCOs might be fuzzified in order to build better bidding strategies. Section 4.1.5 outlines the models we are using, and the research we are currently pursuing to build better bidding strategies. Finally, section 4.1.6 discusses how we can fuzzify decision analysis to decide which action to take for a given scenario. Conclusions and future research efforts for the entire chapter are presented in section 4.3.

#### ***4.2.2 The marketplace***

As in previous chapters, this research again assumes the existence of regional commodity exchanges in which buyers and sellers participate in a double auction. This framework has been adopted from Sheblé [Sheblé, 1996], and closely resembles the market framework being used in the Australian power system. See chapter 2 for more details regarding the market framework.

#### ***4.2.3 Evolving bidding strategies with genetic algorithms***

Briefly, a genetic algorithm (GA) is an algorithm which allows evolution of the contents of a data structure. GAs were developed by John Holland and are loosely based on the biological notion of evolution. The data structure being evolved contains a solution to the problem being studied. A population of syntactically valid solutions are initialized randomly during the first step of the algorithm. Each of the

solutions is assigned a fitness based on its suitability for solving the particular problem being studied. If these solutions are initialized randomly, the chances of them being highly fit during the first generation, is not very high. At each generation, the GA will randomly chose members of the population to be “parents” favoring the highly-fit members. The parents will then produce offspring via the crossover and mutation processes. Crossover is the means by which two parents produces two offspring and involves combining parts of each parent to produce each child. Mutation can be thought of as copying errors introduced into the children due to background noise. The newly produced offspring replace the members of the population that have a low fitness. As the generations progress, there is a tendency for the contents of the data structures to adapt such that they become more suited to solving the problem. See chapter 2 for a more complete description of genetic algorithms.

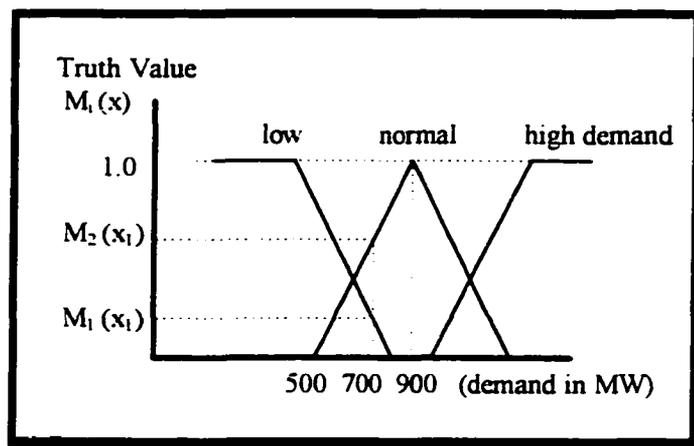
Researchers previously used a GA to evolve a structure containing bid multipliers [Richter and Sheblé, 1997a]. Others have used GAs for computational economics [Ioannides, 1995, Tesfatsion, 1995]. The bidding strategies that come from the evolved structures (shown in Figure 3.1) are fairly simple. The expected price of the electricity (obtained via some forecasting scheme) is multiplied by the bid multipliers and the result is used as the bid for that particular round of bidding. In addition to the bid multipliers, the number of MWs to offer for sale at each round of bidding and the choice of price prediction techniques are also evolved. The results of previous research [Richter and Sheblé, 1997a] are promising. As the GA progresses, the bidding strategies become better and yield more profit, indicating that “intelligent agents” are learning. However, the strategies are somewhat limited because they do not make use of inputs that are available during a particular round of bidding. Evolving fixed string bidding strategies [Richter and Sheblé, 1997a] is like learning the steps of a dance or memorizing a list of things to do mechanically in order to make a successful bid for a particular set of circumstances. Using that approach, means that the evolved rules will not be very adaptive, i.e., they don’t react to the environment. Each set of rules is evolved to be used only for a specific set of circumstances. If the circumstances vary from that, the set of rules may yield disappointing results. We could attempt to create scenarios in which we are interested, but we would find that the number of credible scenarios is so large that we could not possibly hope to cover them all. So the question becomes: How can we develop adaptive bidding strategies that will take advantage of currently available information?

#### ***4.2.4 Fuzziness and uncertainty***

When attempting to maximize the profit of a GENCO, other things being equal, the profit varies with the price of electricity. In reality, those other things (e.g., demand forecasts, production costs, etc.) have uncertainty associated with them. One natural way to deal with risky (uncertain) situations is to use decision analysis with big trees and we’ll talk about that later. Another way, which we’ll discuss now, is to use “fuzzy logic”, made popular by Lotfi Zadeh during the 1960s. Fuzzy logic provides a methodical means of dealing with uncertainty and ambiguity. It allows its users to code problem solutions with a natural language syntax

with which people are comfortable. Many of us regularly use fuzzy terms to describe things or events. For instance, if we were asked to describe a person, we might use terms like “pretty tall”, with a “big nose” and “somewhat overweight”. These terms can be defined differently by different people. There is a certain amount of ambiguity or uncertainty associated with any description involving natural language terms such as these. Most of the things we deal with daily in this universe are ambiguous and uncertain. “The only subsets of the universe that are not in principle fuzzy are the constructs of classical mathematics [Kosko, 1992].”

Fuzzy logic allows us to represent ambiguous or uncertain quantities with membership functions. The membership functions map the natural language descriptions onto a numerical value. Membership to a particular description or class is then a matter of degree. Using fuzzy logic, we might say that electrical demand is high in a region if it is 1200 MW or so, and normal if it is more or less 700 MW. What if the demand is 1000 MW? Using traditional logic we would classify it neither high nor normal. However using fuzzy logic, we might find that this demand is actually both high and normal, each to a certain degree (based on its membership function). Similarly we could have fuzzy membership functions for other inputs like fuel costs, risk aversion, level of competition, etc. Figure 4.1 shows how these classifications (membership functions) in graphical form.



**Figure 4.1 Fuzzy membership functions.**

Once defined, these inputs can then be used in a set of fuzzy rules. Multiple input conditions can be considered by combining rules with the “and” and “or” functions. For instance, some simple rules might be as follows:

- IF demand is HIGH, THEN bid should be HIGH
- IF (demand is LOW) AND (risk aversion is HIGH) THEN (bid should be LOW)
- IF (position is RISKY), THEN hedge with option contracts

where “high” and “low” bid would be defined using another membership function.

Although it may not be necessary, we could have an output for all combinations of inputs. A three input fuzzy rule system where each input is broken into five classifications might be represented as in Figure 4.2. The small squares each contain the output of a rule on how to bid relative to cost. Since some conditions might be very unlikely to occur each of these squares need not have an output. In addition, a particular input maybe classified in more than one square at a given instant. In the figure, the letters V, L, H, C, and N stand for very, low, high, cost, and normal respectively.

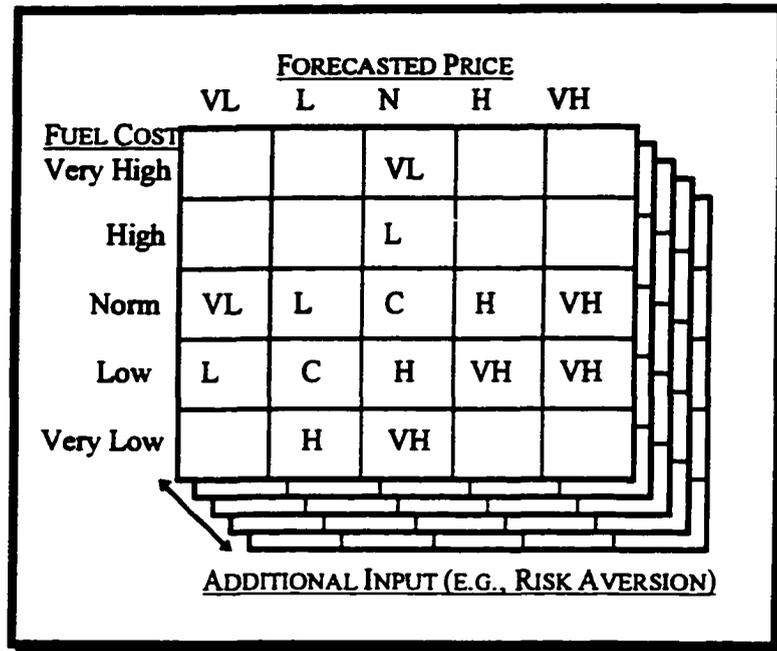


Figure 4.2 Three input fuzzy rule set to determine bid.

Figure 4.3 shows the fuzzy system architecture. The inputs are fed into the rule base. The output (i.e., the bid values in the example) of each rule can be classified by a fuzzy membership function in the same manner as the inputs. The output of each rule may be assigned a certain weight depending on how important we determine that rule or corresponding inputs to be. We can then sum the weighted output of the rules and determine an overall fuzzy output. However, when the time comes to place the bid we can't just say, "bid high". We need a way to convert the fuzzy output to a single number. This is called the defuzzification process.

Defuzzification formally means to round off a fuzzy set from some point in a unit hypercube to the nearest bit-vector vertex [Kosko, 1992]. Practically, defuzzification has been done by using the mode of the distribution of outputs as the crisp output, or by the more popular method of calculating the centroid or center of mass of the outputs and using that as the crisp output. The fuzzy centroid,  $\bar{B}$ , can be calculated as follows:

$$\bar{B} = \frac{\sum_{j=1}^p y_j m_B(y_j)}{\sum_{j=1}^p m_B(y_j)}, \quad \text{where } B = \sum_{k=1}^m w_k B_k'$$

and where  $B$  is the output distribution that contains all information, and  $m_B(y_j)$  is the membership value of  $y_j$  in the output fuzzy set  $B$ .

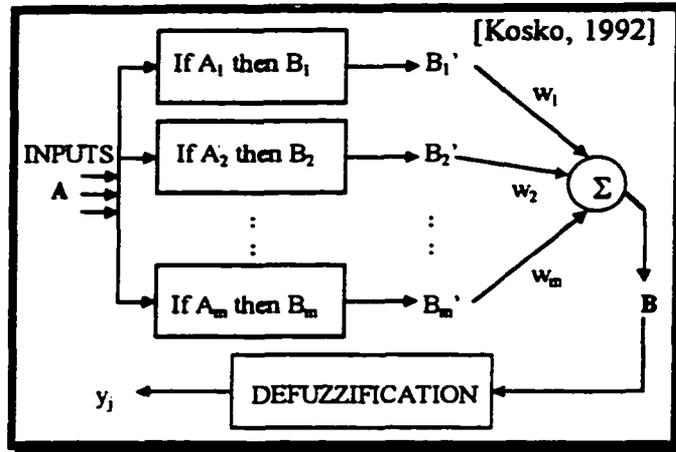


Figure 4.3 Fuzzy system architecture.

#### 4.2.5 Building bidding strategies

This section provides a comparison of approaches that we are taking in developing bidding strategies. We will be generating fuzzy bidding rules manually using expert knowledge. Following that, we shall search for good rule-sets from a limited search space. With a small number of inputs and a limited number of weighting, we can do an exhaustive search of all rules and determine the best possible rule. (The best rule is the one whose use results in the largest amount of profit for its user.) Next, we note that if we increase the number of fuzzy inputs, increase the number of membership functions describing the inputs, and allow more flexibility with the weighting, that perhaps it becomes desirable to use a genetic algorithm to search for the "optimal" rule, rather than do an exhaustive search. Finally, we will attempt the use of a developed technique [Richter et al., 1995] to extract, from a historical database containing the bidding details of an auction, the rules that were used by others in developing their bids.

The research described here builds on described techniques [Richter and Sheblé, 1997a]. To measure the performance of the bidding rules created in each of the methods described below, a group of GENCOs will compete to serve the electrical demands of the ESCOs. Electricity buyers will be aggregated into a single large ESCO. See Figure 4.4. TRANSCOs and transmission constraints will not be considered directly here, but can be accounted for after-the-fact if desired.

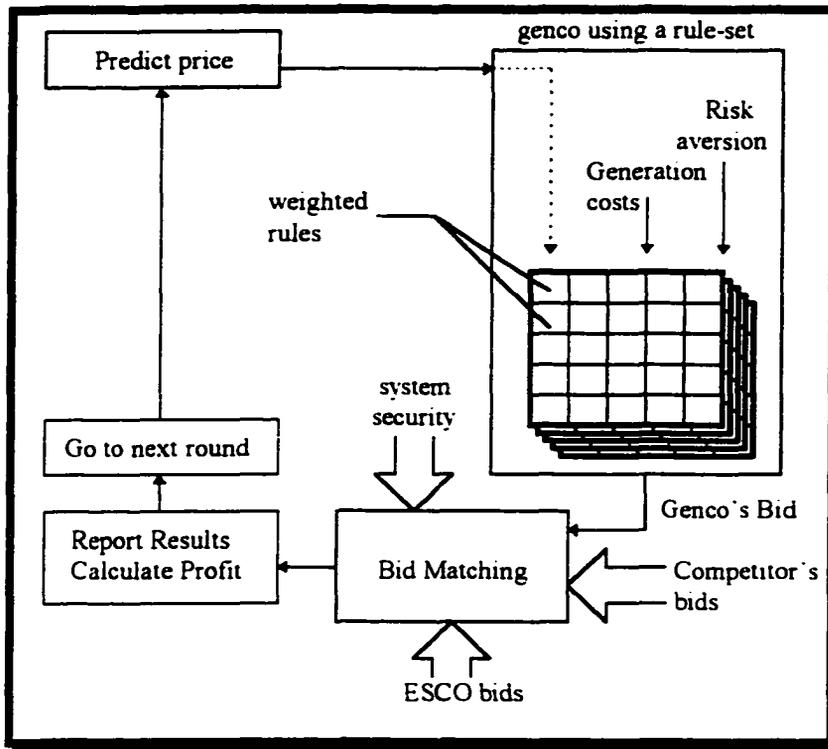


Figure 4.4 Using the rule-set.

#### 4.2.5.1 Generating the fuzzy rule-sets manually

If we consider only a limited number of fuzzy economical inputs, (e.g., expected price, risk aversion, and generating costs) then it is possible to generate rules manually with expert knowledge from power traders. We can transform the rules-of-thumb used by experienced power traders into a fuzzy rule base. We may also use theoretical economics to influence the rule-sets that we construct. If we have 3 fuzzy inputs, each divided into 5 classifications, then we could have need for as many as 125 rules in each rule-set (one for each little square in Figure 4.4). Each of the rules can be weighted according to its importance, if any weighting is allowed we have infinitely many possibilities.

#### 4.2.5.2 The search for the "optimal" fuzzy rule-set

To reduce the amount of time spent tuning the rule-sets, we can predefine a structure and allow a computer program to search through the possibilities to find the optimal rule-set. If we predefine each of the three inputs by five fixed ranges, and only allow discrete rule weightings (e.g., 0.0, 0.1, 0.2, ..., 1.0), then there are a finite number of permutations to investigate. A possible indication of optimality would be obtained by having an agent use each of the possible rule-sets while engaging in a fixed set of trial-auctions. In each of these trial-auctions, the agent would be competing with a set of agents that had evolved to play the market

described in other research [Richter and Sheblé, 1997a]. To ensure that the rules aren't market specific, the set of agents against which the rule will be competing can be taken from different populations and from various stages of evolution. This increases the certainty the tested rule will be profitable against a diverse set of agents and circumstances.

#### *4.2.5.3 Using a GA to evolve fuzzy rules for bidding*

If we relax the requirement that each rule must have a discrete weighting, we can see that the size of the search space becomes quite large. If we also increase the number of inputs to consider, the search space grows even larger. The exhaustive search no longer remains feasible. In addition, if we do not wisely choose the set of agents against which our rules will be tested, then we would be left with rules that are not extremely robust. Therefore, the plan is to use a GA to evolve rule-sets in a similar fashion as in previous research [Richter and Sheblé, 1997a], but with slightly modified data structures.

Each of the GENCOs will have its own evolving data structure consisting of a fuzzy set of rules, and weights associated with each of those rules. The weights will allow some rules to have more importance than others. Previous work allowed each of the individual GENCOs to have their choice of price forecasting techniques. This created a lot of overhead, and for simplicity, current research has each GENCO receiving globally forecasted data. In addition, the contract size (i.e. number of MWs to offer) at each round of bidding would be fixed rather than evolvable to reduce the search space.

#### *4.2.5.4 Using a GA to extract expert-system bidding rules from a historical database*

An investigation of GAs and other so-called artificial intelligence techniques revealed their ability to search through large databases in order to learn the expert system rules that can be used to reproduce decisions made to build the database. Presently this technique is being used to develop standardized treatment methods for hospital patients receiving medical care. Based on extensive records, the software is able to determine what the doctor did based on patient conditions. Similarly, a database of trading data could be fed into the software (which would require tuning and some restructurization) to figure out what bidding rules were being used by the traders. Determining the rules that other electricity traders and brokers are using could be of great benefit to those who wish to gain a competitive edge when participating in the deregulated market. See chapter 6 on improving bidding strategies through intelligent data mining for more detail.

#### *4.2.6 Fuzzifying decision analysis*

Earlier in the chapter, we mentioned that decision analysis with really big trees offers a method to compare different scenarios when inputs are uncertain. Traditional decision analysis involves drawing a decision tree, where trees consist of branches and nodes. Decision nodes are drawn with a square and mean

that there is a decision to be made. Each choice at a decision node will lead to a different possible set of circumstances/outcomes. Circular nodes are chance nodes where analysis has indicated that many possible futures exist each with a certain anticipated probability and each with a different effect on the final outcome. Each branch must be assigned a specific probability value.

Consider the tree in Figure 4.5. The objective is to maximize expected profit subject to our risk tolerance. The choice is to bid high, or to bid low. In the traditional case, our analysts would have to give us specific numbers to describe the possibilities. For instance, demand will be 1000 MW with a probability of 0.25, and it will be 500 MW with a probability of 0.75. The costs will be \$10/MW with a 0.25 probability and \$5.0/MW with a 0.75 probability. We can then calculate an expected profit for each of the decision node branches. In the traditional case, we would have actual bids (e.g., high = \$15.0/MW and low = \$10.0/MW). We would then make the bid that maximized our expected profit. The crisp probabilities are taken as certain even though our planners might have liked to use fuzzy terms like "high" or "low" defined by a membership function. We are basically ignoring the part about "subject to our risk tolerance" when making the initial decision. Risk can then be accounted for after the fact using the traditional means (futures and options), but the measures taken to reduce risk could change which decision produces the highest expected profit.

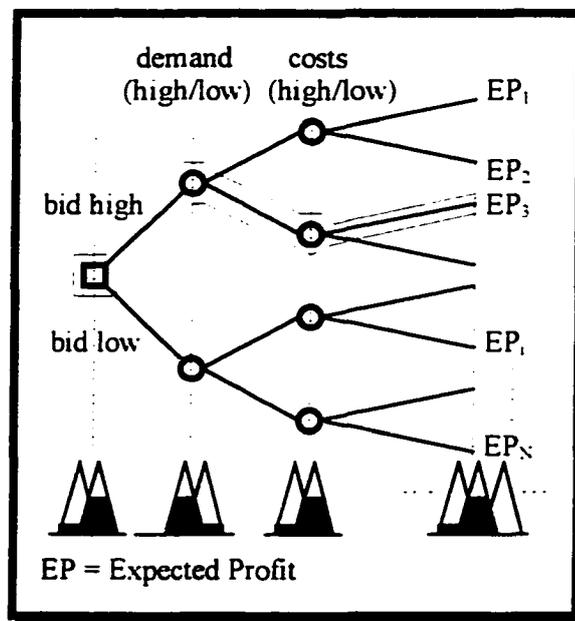


Figure 4.5 Evaluating the alternatives with decision analysis.

Fuzzy logic provides a means of incorporating the uncertainty into the decision tree. We propose using fuzzy terms to describe the conditions that the chance nodes may take on and use membership functions to describe how much emphasis to place on the various outcomes. In Figure 4.5, we may be fairly certain that the demand is high, but we think there is a small chance that it could be low (where high and low are described by

membership functions). Rather than choosing any one particular path through the tree, the fuzzy method is influenced by all possibilities. Likewise the expected value is expressed in a membership function which basically has a built in description of the risk associated with that scenario. In addition to the inputs, the output is described by a fuzzy membership function.

### **4.3 Automatic generation control with a fuzzy logic controller**

#### **4.3.1 Introduction**

The interconnected electrical systems of the United States and of other countries are often complex networks coordinated by agreement amongst individual utilities and regional power pools. Generation is predominantly produced by electrical utilities with their own generation. However, the number of independent power producers and other non-utility generation facilities are increasing in the new business environment [Sheblé, 1996]. The utilities frequently buy and sell power between themselves and others via the transmission lines. The power is transmitted to loads within control areas. Each control area normally has at least one generator that is controlled to provide regulation of area frequency, and tie line flow. This coordinated action assists in maintaining the secure and reliable operation of the power system. This controlled generator may or may not be located in the same area as the load to which it is responding. It might even be a jointly owned unit or be a unit operating to fulfill a contract specifically written to supply regulation capability. Automatic generation control (AGC) is responsible for varying the output of the selected area generators to maintain the system frequency and tie line flows, and to minimize accumulated time error and interchange error.

Fuzzy control systems have been extensively used in Japan and have recently gained popularity for solving control problems in the United States. Fuzzy Logic Control (FLC) systems allow the use of fuzzy or “non-crisp” inputs in a control process. They allow precise control of plant or process using easy to understand “English” rules that can be developed without knowledge of mathematical control theory. FLC has been successfully applied to problems in image processing, control of robotics, decision trees and expert systems [Kristjánsson et al., 1993].

The application of a fuzzy logic controller to AGC has been described, but only used one input to control the plant [Indulkar and Raj, 1995]. The research described in this chapter uses two additional inputs to control the plant. By including the accumulated time error and the interchange error as separate inputs to the FLC described here, the area control error, and the observed time and interchange errors should be reduced for the system. Because of incomplete information, we were not able to duplicate the results or perform a direct comparison with the results of previous researchers [Indulkar and Raj, 1995]. However, the results found for the classical system presented in this chapter are a significant improvement.

This section of the chapter presents the use of fuzzy logic for automatic generation control. We introduce the reader to AGC in section 4.2.2. In section 4.2.3, a brief overview of fuzzy logic is presented. Section 4.2.4 describes our development of a fuzzy logic controller for AGC. Section 4.2.5 presents the results obtained using a three input FLC. Section 4.2.6 describes several extensions to AGC using FLC systems for other applications. Conclusions and future research efforts for the entire chapter are presented in section 4.3.

### *4.3.2 Automatic generation control*

As stated above, utilities and other power producers cooperate to operate in a coordinated fashion. These entities have agreed that it is beneficial for them to work together in the planning, control, and operation of the national electrical system. They purchase power over the transmission lines which connect generation with load in neighboring areas. To enforce the contracts, transmission lines are then monitored by equipment that measures the amount of power flowing on the transmission lines and the frequency in each area.

Electric utilities presently generate, or purchase, exactly the amount of power needed to supply their own distribution customers (native load), to sell to neighboring utilities, and to cover their losses on the system. As an agreed upon standard, the systems have been designed and built to operate at 60 hertz. They are able to produce their power very near the agreed frequency by careful control of their generators. In addition to the changes in generation, changes in load can also cause the frequency of the system to increase or decrease.

When two isolated electrical utilities are interconnected, their respective frequencies are synchronized. These two utilities may operate as a single control area or as two or more control areas. Assume that the utilities have agreed to operate as two control areas. If these two utilities have agreed to transmit power from one area to the other, the selling utility increases its generation and the buying utility decreases its generation. This maintains the system frequency but causes the voltage angle in the selling area to lead the receiving area's voltage angle. Power then flows from the leading system to the lagging system. If either control area improperly changes generation, then the system frequency will not remain constant.

In an interconnected system, as generation or loads change unexpectedly, the frequency and the interchange power change. This unscheduled change to the interchange power flow and to the frequency is undesirable. Furthermore, these unscheduled changes may mean that electric suppliers which are not party to the contract could be supplying the energy which should be supplied by another utility. System frequency may increase or decrease, and errors may occur in electronic devices which require regulated power. One problem with off nominal frequency operation is that synchronous wall clocks which will be slow or fast.

Responding to the errors in frequency and power flow, adjustments can be made which minimize the disturbance to the electrical system. The faster the error can be eliminated, the interchange and time errors are reduced and the payments for the inadvertent energy flows are minimized. These adjustments may be made manually, or automatically by computer. Automatic generation control has been implemented at most electric

utilities, but some operators feel that they are better able to control the system using heuristics or their expert knowledge gained from years of experience controlling the system.

The generator and the system load have a certain inertia associated with their respective rotating masses which tends to dampen the effects of load changes. In addition, generators are installed with a controller, called a governor, which responds quickly to any changes in frequency. However, the amount of frequency sensitive load is not known. Even with the system inertia and the governor action, additional control is required, which is provided by the human operator or a traditional AGC controller using area control error (ACE) as a performance indicator. In contrast to the fuzzy controller which will be explained later, traditional AGC controllers are linear. Figure 4.6 shows a traditional AGC control block diagram representation.

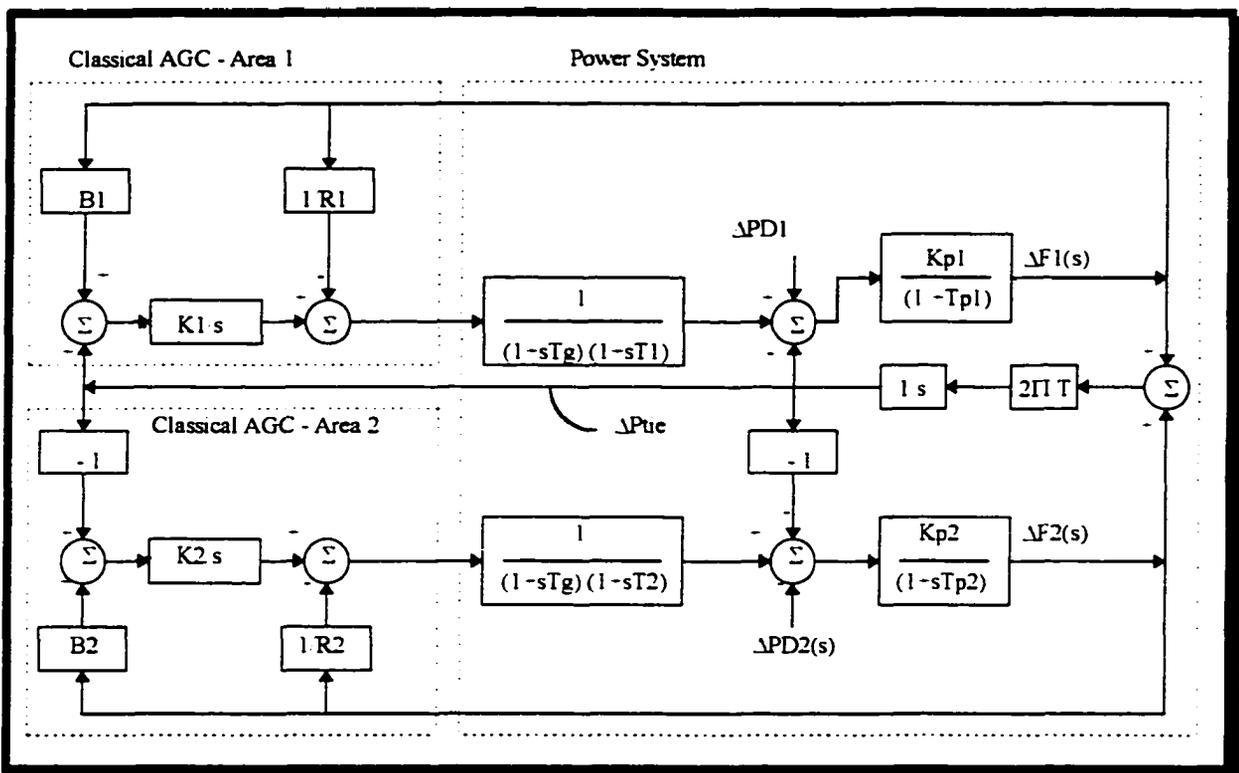


Figure 4.6 Traditional Automatic Generation Control Block Diagram [Wood and Wollenberg, 1996]

The ACE represents the shift in generation needed to restore the frequency and net interchange power flow to the scheduled values [Wood and Wollenberg, 1996]. The ACE in each area is traditionally calculated as  $ACE = -\Delta P_{tie} - B \Delta \omega$ , where  $\Delta P_{tie}$  is the change in the connected load, and  $\Delta \omega$  is the change in frequency. In a two area system, the  $\Delta P_{tie}$  for one area will be the negative of the other area.

Controllers may use integrals or derivatives of their inputs to achieve more accurate and precise control. For automatic generation control, the ACE is often augmented with time integrals of these interchange and

time errors. This ACE is then used as feedback into the linear plant controller and adjusts the plant output during the next control interval time step. This section of the chapter assumes that the control interval is 2/4. Two seconds for acquiring data and four seconds for communicating generation changes.

### 4.3.3 Fuzzy logic control (FLC)

FLC allows fairly simple control strategies, written in English, to be used to control a process. Inputs are assigned to ranges corresponding to some classification like large or small. However, an input can actually belong to more than one classification at the same time. We associate membership values with inputs indicating how well, or to what degree, the particular classifications represent the input.

For example, we might say that an object is heavy if it weighs ten pounds, and light if it weighs only five pounds. What if the object in question weighs seven or eight pounds? Using traditional logic this object would not be classified as heavy or light, but using fuzzy logic, we could say that this object is heavy to a certain degree (based on its membership function), and that it was also light to a certain degree. In fuzzy logic terminology, this degree is called its membership value, or truth value as depicted in Figure 4.7.

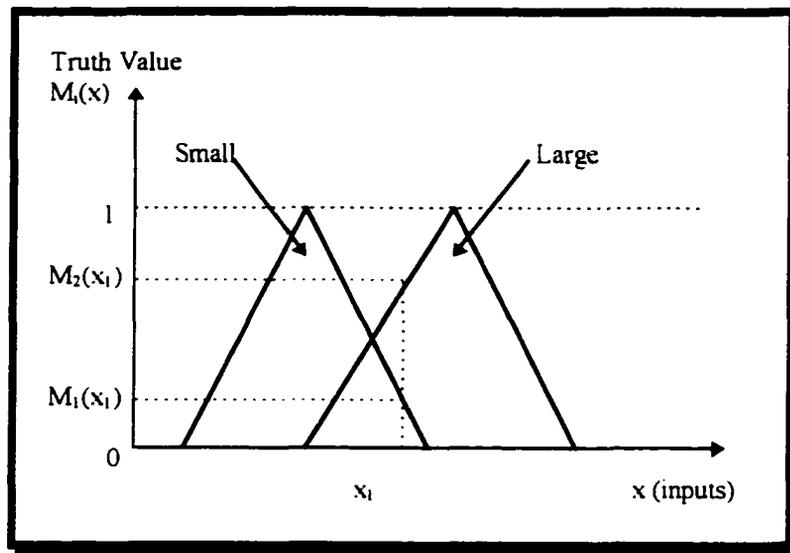


Figure 4.7 Fuzzy input membership functions.

The same way an object can be both heavy and light to a certain degree, an error value can be both large and small in value. By having a mathematical method that allows us to define control inputs and outputs less rigidly than a traditional classification, we can create controllers which are based on easy to define rules written in English. Multiple input conditions can be considered by combining rules with the "and" and "or" functions. The truth value of two input criteria when using an "and" condition, is the minimum of the considered truth values. Inputs combined using the "or" function will have the highest truth value of the inputs being considered.

FLC allows us to develop control strategies without development of optimal rules as found in mathematical control theory. We can obtain output values by correlating one input (or multiple inputs) to an output via one fuzzy rule (or more) of the following form as shown in Figure 4.8: *IF (lawn is green) AND (meteorologists predict no rain) THEN (water lawn a little)*. These output values can be divided into fuzzy ranges in the same manner as the inputs. With defuzzification, a fuzzy output can be changed to a crisp number which indicates the proper gallons of water to put on your lawn. The fuzzy output is “defuzzified” to a number, e.g. “little” might range from 20-30 gallons. Many books have been written on fuzzy systems and the details of FLC systems. The reader is referred to “*Essentials of Fuzzy Modeling and Control*” by Yager and Filev [1994] for additional information.

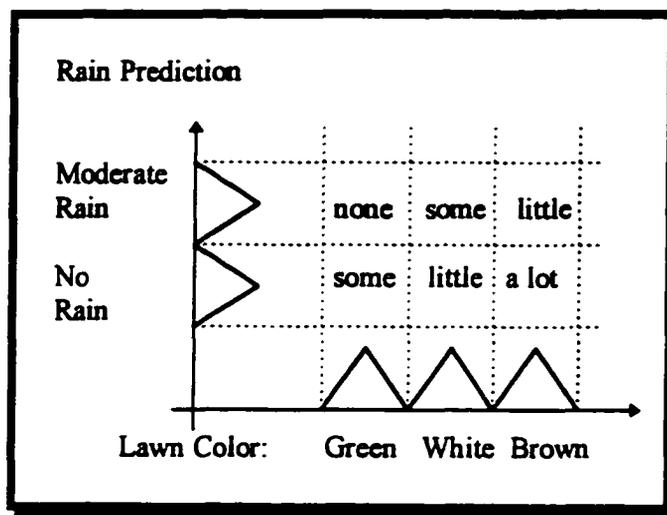


Figure 4.8 Using two inputs to determine a fuzzy output.

#### 4.3.4 Fuzzy AGC

For the results described in this section of the chapter, a two area system model was used. Area 1 was modeled with load and an equivalent single thermal generating unit, while area 2 was modeled with load and an equivalent hydroelectric generating unit. For the fuzzy results shown here, the classical AGC controllers were replaced with FLCs as shown in Figure 4.9. In the figure  $\Delta Pg1$  (change in plant 1 generation) and  $\Delta Pg2$  (change in plant 2 generation) are the requested changes to the AGC controlled generation in areas 1 and 2.

The fuzzy control for each region, has three inputs, classical ACE and the augmented inputs: the time-integral of the frequency error (time-error), and the time-integral of the tie-line flow (interchange error) as shown in Figure 4.10. The primary control is linked to the ACE, with the other two inputs being of secondary importance, especially in the short run.

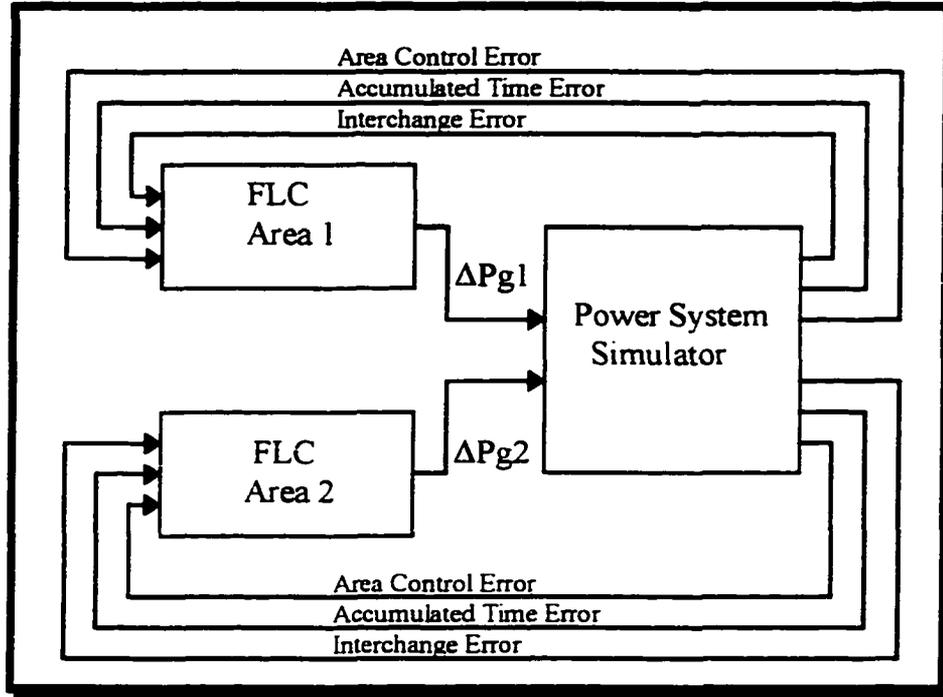


Figure 4.9 FLC block diagram.

The input signals were divided among ranges primarily by triangular membership functions. The different membership functions were labeled as negative large (NL), negative small (NS), zero (ZE), positive small (PS), and positive large (PL). The output was divided similarly, but also had a negative medium (NM) and a positive medium (PM) range. This choice of divisions is based on the classical ranges for ACE as shown in Figure 4.11. The rules for determining whether the plant output should increase or decrease during the next control time step, are based on the errors from the previous control time step.

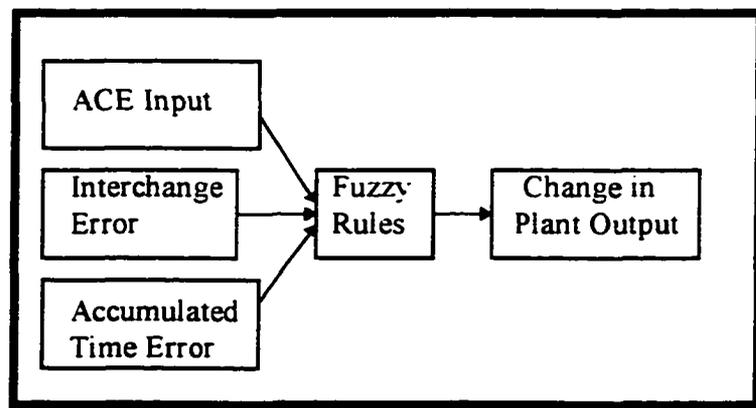


Figure 4.10 FLC Block Diagram.

	Condition Action	MW off nominal
Excess Generation	Trip AGC	5
	Emergency Assist	4
	Excessive ACE	3
	Normal Region	2
	Deadband	1
Deficient Generation	Target Generation Setpoint	0
	Deadband	-1
	Normal Region	-2
	Excessive ACE	-3
	Trip AGC	-5

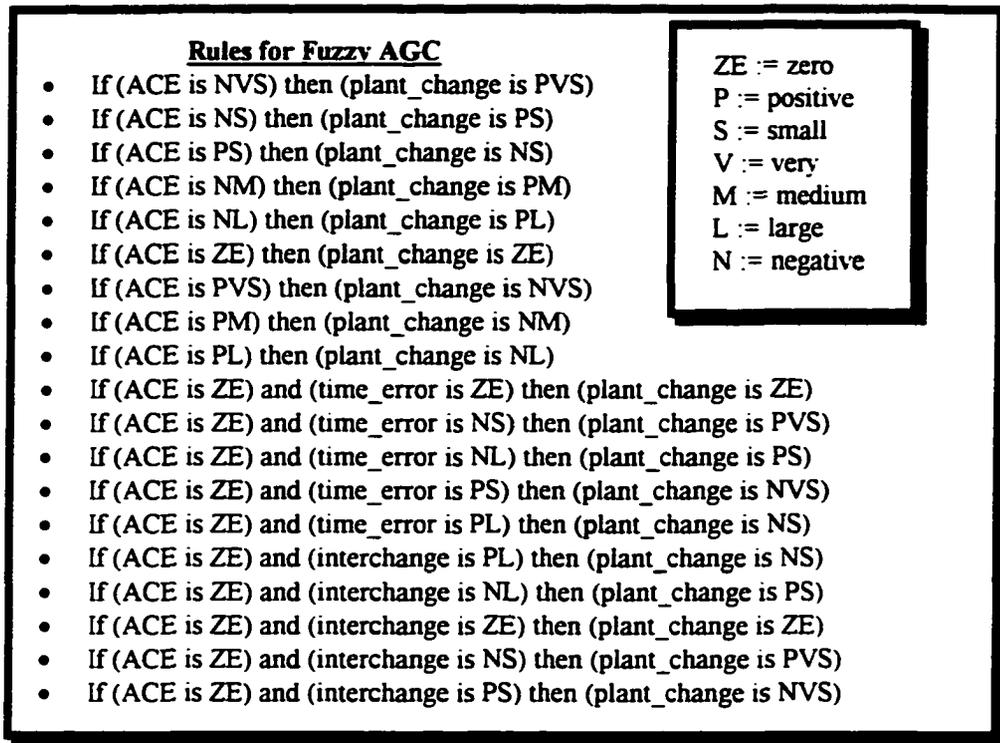
Figure 4.11 ACE ranges and associated actions and conditions.

#### 4.3.5 Results of the FLC AGC

The two area power system was modeled as shown in the right side of Figure 4.6. The left area shows the proportional integrative control system used to control the response of the plants under the traditional control scenario. In each of the two areas one generator has been selected for AGC. The control systems shown on the left of Figure 4.6 were lifted out and a fuzzy controller was inserted into each of these areas. The parameters shown in Table 4.1 were used in the system simulation for both the traditional control and for the fuzzy logic controller. Upon initiating the simulation, a steady state load deviation in area 1 of 0.05 per unit was input. The time step used in our simulation was 0.05 seconds, the fuzzy rules in Figure 4.12 were used in the fuzzy controller to achieve to the plant output shown in the results.

Table 4.1 Parameter settings for plant simulation.

Parameter	Setpoint	Parameter	Setpoint
kg1	1.0000	tg1	0.0800
kt1	1.0000	tt1	0.3000
kp1	120.0000	tp1	20.0000
r1	2.4000	t12	0.0100
b1	0.4250	alpha1	0.1000
a1	0.0400	k1	1.0000
k	0.0166	a12	-1.0000
kg2	1.0000	tg2	0.0800
kt2	1.0000	tt2	0.3000
kp2	120.0000	tp2	20.0000
r2	2.4000	d	0.5000
b2	0.4250	alpha2	0.1000
a2	0.0400	k2	1.0000



**Figure 4.12 Fuzzy rules for the AGC controller.**

Triangular and trapezoidal membership functions were used to define the various classification ranges. The ranges for small, medium and large were defined slightly different for the two areas. However, the same nineteen rules were used to control the plant in both areas of the simulation. The fuzzy logic toolbox available with MATLAB was used to develop the membership functions. The software produces a file containing the membership function definitions along with the ranges, this data is shown in Figures 4.13 and 4.14 which include all the data needed to define the membership functions that will allow the nineteen rules specified earlier to function as described in the results section of this part of the chapter. The data in the figures comes from two files that are developed automatically when using the MATLAB fuzzy logic toolbox graphical rule builder.

As mentioned above, the rules were evaluated by modifying an existing program designed for the traditional linear AGC controller tuned for a two area system. The two interconnected areas are operating at steady state with a scheduled interchange power flow and at required frequency. When the load is perturbed in one of the areas, the traditional control responds to eliminate the deviations. Different load disturbances result in slightly different error patterns, but the controller is able to drive the error to zero, which is better than having no controller.

The traditional controller was replaced with a fuzzy controller, using the rules listed in this section of the chapter. The load was perturbed with the same step change as was used for the traditional controller. Using the fuzzy controller, the absolute values of ACE, the time error, the interchange error, the unscheduled power

flow, and the frequency error were all decreased. Graphs showing the cases the traditional controller and the fuzzy controller are shown in Figures 4.15, 4.16, and 4.17. The graphs show the FLC as a thin line, and the response of the classical controller as heavier lines made of '+'s.

The results shown in these graphs show that the FLC controller can be tuned so that it responds very quickly. In fact, for this particular case, it may be responding a bit to fast as some of the error values are the negative of the traditional controller, but the errors are smaller than that achieved by the traditional controller.

```
[System]
Name='AREA1RUL'
Type='mamdani'
NumInputs=3
NumOutputs=1
NumRules=19
AndMethod='min'
OrMethod='max'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='ACE'
Range=[-0.04 0.04]
NumMFs=9
MF1='NL':'trapmf,[-5 -4.5 -0.032 -0.0176]
MF2='NVS':'trimf,[-0.008 -0.004 -0.0004]
MF3='ZE':'trimf,[-0.002 0 0.002]
MF4='PVS':'trimf,[0.0008 0.004 0.008]
MF5='PL':'trapmf,[0.0176 0.032 0.08 0.088]
MF6='NM':'trimf,[-0.028 -0.018 -0.008]
MF7='PM':'trimf,[0.008 0.018 0.028]
MF8='NS':'trimf,[-0.012 -0.008 -0.004]
MF9='PS':'trimf,[0.004 0.008 0.012]

[Input2]
Name='time_error'
Range=[-0.001 0.001]
NumMFs=5
MF1='NL':'trimf,[-0.0015 -0.001 -0.0005]
MF2='NS':'trimf,[-0.0006668 -0.0004 -6.668e-05]
MF3='ZE':'trimf,[-0.0003334 0 0.0003332]
MF4='PS':'trimf,[0.0001333 0.0004 0.0006668]
MF5='PL':'trimf,[0.0005 0.001 0.0015]

[Input3]
Name='interchange'
Range=[-0.05 0.05]
NumMFs=5
MF1='NL':'trimf,[-0.075 -0.05 -0.025]
MF2='NS':'trimf,[-0.035 -0.02 -0.005]
MF3='ZE':'trimf,[-0.0125 0 0.0125]
MF4='PS':'trimf,[0.005 0.02 0.035]
MF5='PL':'trimf,[0.025 0.05 0.075]

[Output1]
Name='plant_change'
Range=[-0.005 0.005]
NumMFs=9
MF1='NM':'trimf,[-0.003711 -0.002236 -0.0007615]
MF2='NS':'trimf,[-0.001501 -0.000875 -0.0002494]
MF3='ZE':'trimf,[-0.00012 0 0.00012]
MF4='PS':'trimf,[0.000241 0.0008665 0.001492]
MF5='PM':'trimf,[0.000958 0.002233 0.003509]
MF6='NL':'trapmf,[-0.009 -0.008 -0.004 -0.0025]
MF7='PL':'trapmf,[0.0025 0.004 0.008 0.009]
MF8='NVS':'trimf,[-0.00035 -0.0002 -8.168e-05]
MF9='PVS':'trimf,[8.335e-05 0.0002 0.00035]
```

Figure 4.13 Ranges defining membership functions of area 1 controller.

```

[System]
Name='AREA2RUL'
Type='mamdani'
NumInputs=3
NumOutputs=1
NumRules=19
AndMethod='prod'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='ACE'
Range=[-0.001 0.001]
NumMFs=9
MF1='NL':'trapmf,[-0.125,-0.1125,-0.0008,
0.00044]
MF2='NVS':'trimf,[-0.0002 -0.0001 -1e-05]
MF3='ZE':'trimf,[-5e-05 0 5e-05]
MF4='PVS':'trimf,[2e-05 0.0001 0.0002]
MF5='PL':'trapmf,[0.00044 0.0008 0.002 0.0022]
MF6='NM':'trimf,[-0.0007 -0.00045 -0.0002]
MF7='PM':'trimf,[0.0002 0.00045 0.0007]
MF8='NS':'trimf,[-0.0003 -0.0002 -0.0001]
MF9='PS':'trimf,[0.0001 0.0002 0.0003]

[Input2]
Name='time_error'
Range=[-0.001 0.001]

[Input3]
Name='interchange'
Range=[-0.05 0.05]
NumMFs=5
MF1='NL':'trimf,[-0.075 -0.05 -0.025]
MF2='NS':'trimf,[-0.035 -0.02 -0.005]
MF3='ZE':'trimf,[-0.0125 0 0.0125]
MF4='PS':'trimf,[0.005 0.02 0.035]
MF5='PL':'trimf,[0.025 0.05 0.075]

[Output1]
Name='plant_change'
Range=[-0.0005 0.0005]
NumMFs=9
MF1='NM':'trimf,[-0.0003711 -0.0002236 -7.613e-05]
MF2='NS':'trimf,[-0.0001501 -8.747e-05 -2.494e-05]
MF3='ZE':'trimf,[-1.2e-05 0 1.2e-05]
MF4='PS':'trimf,[2.41e-05 8.667e-05 0.0001492]
MF5='PM':'trimf,[9.58e-05 0.0002233 0.0003509]
MF6='NL':'trapmf,[-0.0009 -0.0008 -0.0004 -0.00025]
MF7='PL':'trapmf,[0.00025 0.0004 0.0008 0.0009]
MF8='NVS':'trimf,[-3.5e-05 -2e-05 -8.167e-06]
MF9='PVS':'trimf,[8.333e-06 2e-05 3.5e-05]

```

Figure 4.14 Ranges defining membership functions of area 2 controller.

#### 4.3.6 Extensions

Many enhancements are being developed and have been suggested to improve traditional AGC controllers, especially for operation in the competitive environment. Future extensions to this research will propose that fuzzy logic be used to extend AGC in the following ways:

- Fuzzy filtering of ACE, determining which signals are just noise, and which signals contain proper control information.
- Fuzzy prediction of future load and ACE values based on historical load patterns to reduce the amount of control.
- Fuzzy determination of when to minimize the ACE based on the different area control regions and loading of economic generators

- Fuzzy learning of the system model, which will allow the rules to adapt to correct for tolerances and errors in the rule ranges

If a predicted ACE and load can be obtained, then this information can be used to determine when it is the best time to reduce the error. Some AGC responses are mandatory, but others are recommended, or permitted. The predictions and filtered signals, along with the allowed responses may be used to determine the correct signal to send to the governor. Adaptive fuzzy rules will allow the system to learn to change its response to different situations based on what happened during the previous time period or periods.

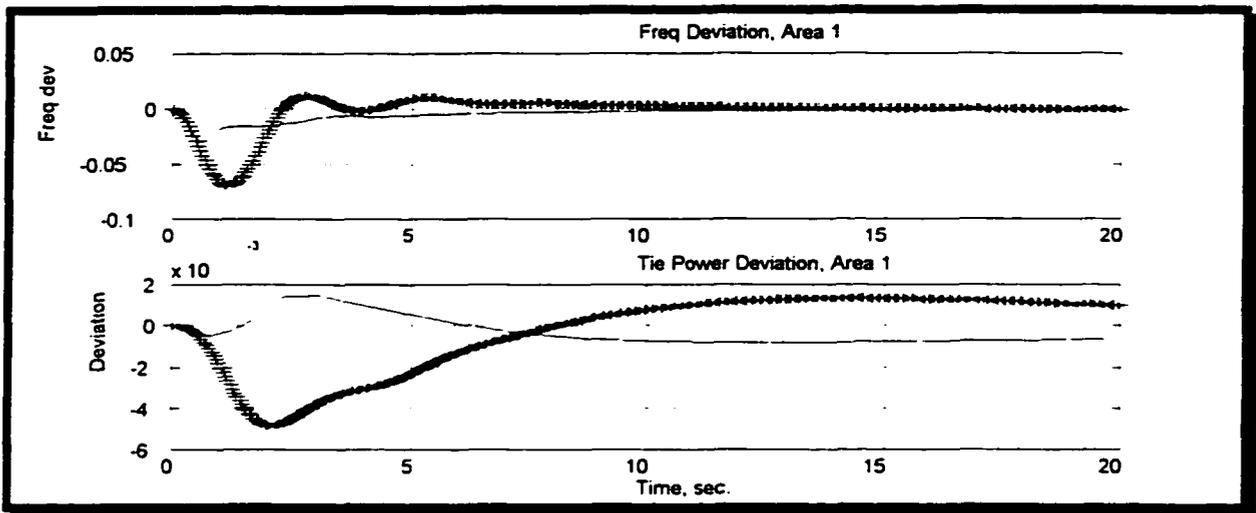


Figure 4.15 Area 1 Frequency and Ptie Deviation.

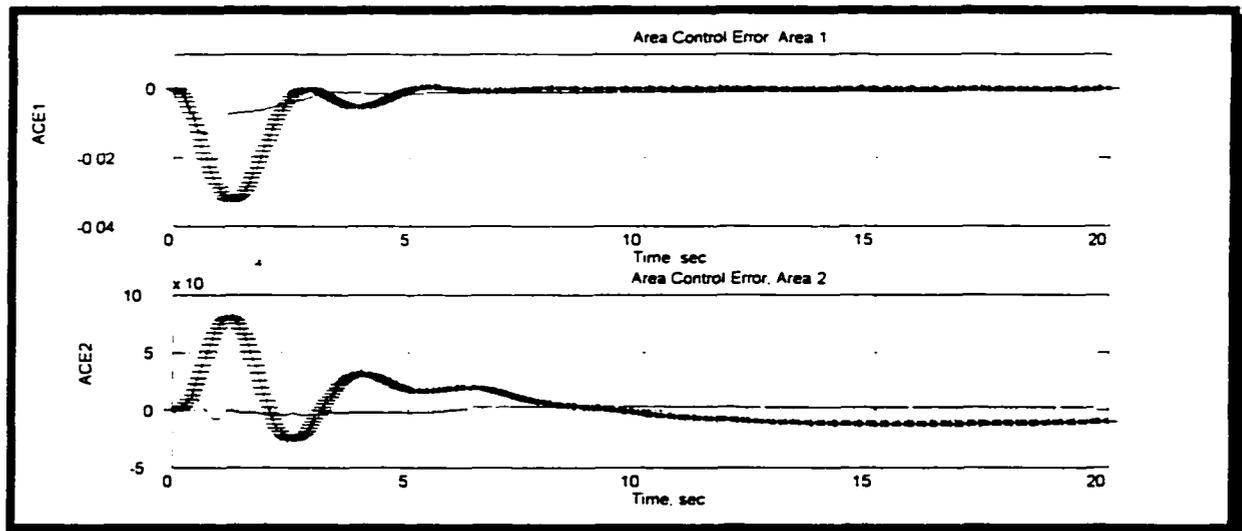


Figure 4.16 Area 1 and Area 2 Control Error.

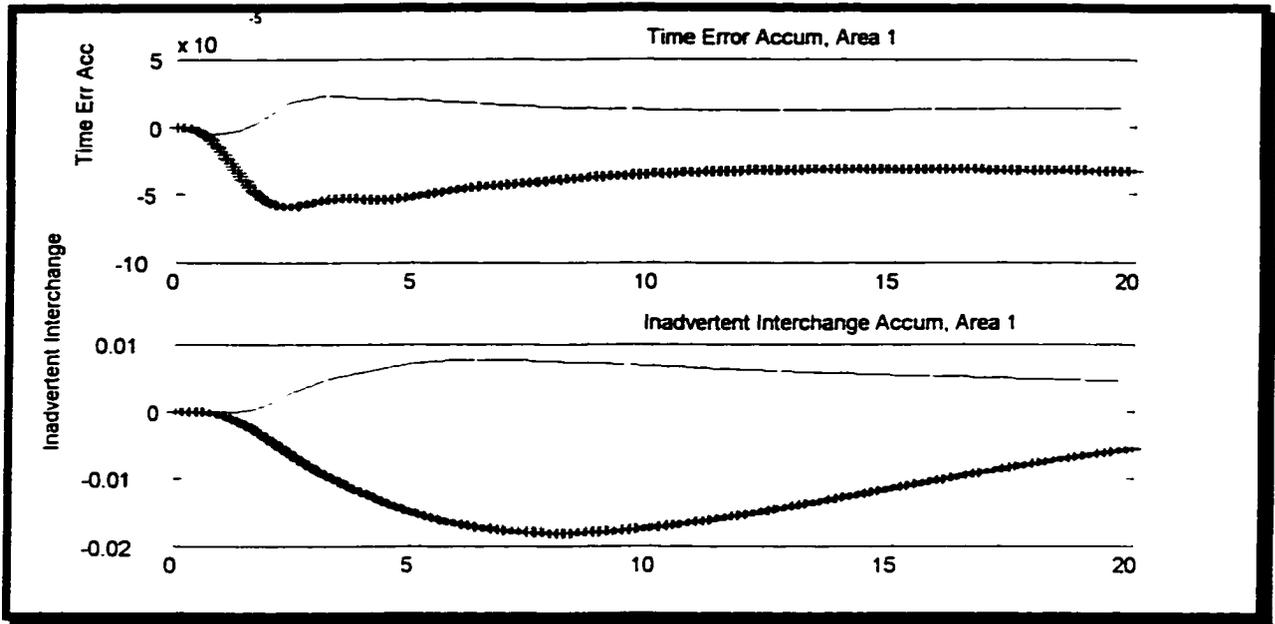


Figure 4.17 Accumulated Time and Interchange Errors.

#### 4.4 Chapter summary and future research

Building good bidding strategies for electricity traders as they move into the deregulated marketplace will continue to be important for those companies wishing to remain profitable. This chapter described such research and points to new methods which are currently being investigated in order to build more robust adaptive bidding strategies. The deregulated market-place that this research assumes will become standard throughout the USA and is being incorporated into our auction simulator. The bidding rule-sets or strategies obtained from each method described in this chapter should be tested in auction simulations. Comparisons of profitability between the fuzzy bidding rules to the method which uses bidding multipliers as described in chapter 3 should be performed in future research.

In addition to maximizing profit, traders should consider the risk associated with a particular strategy. Futures and options contracts can be used to reduce the risk associated with an energy trader's position in the market. In addition to futures and options, we have proposed other methods of reducing risk including the use of fuzzy logic, enhancement of our genetic algorithm and GP-Automata bidding strategy evolvers so that fitness function incorporates risk.

The fuzzy logic controller lends itself well to the AGC problem. It responds rapidly, and can be tuned to outperform the traditional controller for the cases studied. The rules can be formulated in plain English, making it easy for the operator to see how they work, and how to easily modify them.

The FLC AGC described in this chapter uses rules with fixed ranges. Future research will make these rule adaptive so that manual re-tuning is not necessary when generator parameters are changed. The electric utility industry is becoming more competitive. This is changing the way transactions take place between companies. These changes will make it more important than ever to have tight control over the interchange contracts. Future research will incorporate fuzzy control into a multi-area AGC controller as described the companion papers by Kumar et al. [1996a, 1996b].

## **5 PROFIT-BASED UNIT COMMITMENT FOR THE COMPETITIVE ENVIRONMENT**

### **5.1 Chapter overview**

As the electrical industry restructures, many of the traditional algorithms for controlling generating units are in need of modification or replacement. Previously utilized to schedule generation units in a manner that minimizes costs while meeting all demand, the unit commitment (UC) algorithm is one of these tools that must be updated. A UC algorithm that maximizes expected profit will play an essential role in developing successful bidding strategies for the competitive generator. Simply bidding to win contracts is insufficient; profitable companies require bidding that results in contracts that, on average, cover the total generation costs. No longer guaranteed to be the only electricity supplier, a generation company's share of the demand will be more difficult to predict than in the past. Removing the obligation to serve softens the demand constraint. A price/profit-based UC formulation that considers the softer demand constraint and allocates fixed and transitional costs to the scheduled hours is presented in this chapter. In addition, a genetic algorithm solution to this new UC problem is described, and results for an illustrative example are presented. This chapter discusses using different allocations of the fixed and transition costs to increase profitability, and look at ways of handling uncertainty in price and load forecasts.

### **5.2 Introduction**

The US electric marketplace is in the midst of major changes designed to promote competition. No longer vertically integrated with guaranteed customers and suppliers, electric generators and distributors will have to compete to sell and buy electricity. The stable electric utilities of the past will find themselves in a highly competitive environment. Although some states (e.g., California) are already operating in a restructured environment, a standardized final market structure for the rest of the US has not yet been fully defined. The research is performed with the belief that regional commodity exchanges, in which electricity contracts are traded, should play an important role.

Previous chapters and research [Kumar and Sheblé, 1996a; Kumar and Sheblé, 1996b; Sheblé, 1994b; Sheblé, 1996] have described a framework in which distribution companies (DISTCOs), generation companies (GENCOs), energy services companies (ESCOs) and transmission companies (TRANSCO) interact via

contracts. The contract prices are determined through an auction. Electricity traders make bids and offers that are matched subject to the approval of an independent contract administrator (ICA) who ensures that the system is operating safely within limits.

Operating within such a framework, traders will create and implement bidding strategies to make their bids and offers. These bidding strategies might be designed to limit the traders' risk, to maximize profit, or some combination of both. Chapter 3 reported research that uses genetic algorithms and genetic programming to evolve bidding strategies that maximize profit for the spot market. An investigation of managing an energy trader's risk and profitability by combining spot market contracts with options and futures was performed [Richter and Sheblé, 1998]. For simplification, the work described in chapter four avoided the UC problem by ignoring start-up and shut-down costs, minimum up-times and down-times, ramp rates etc. In this chapter a profit-based UC is considered and its implication for bidding strategies is discussed.

A genetic based UC algorithm was developed and implemented [Maifeld and Sheblé, 1996]. The algorithm was able to consistently find multiple good unit commitment schedules in a reasonable amount of time. Another advantage it held over other UC solution techniques is that it could penalize solutions which violate constraints with additional non-linear costs. Rather than saying a solution is simply unacceptable, a cost associated with making it acceptable can be assigned to the solution. This is akin to paying a fine, or paying for damage associated with exceeding limits. These attributes remain desirable qualities. We have updated this algorithm for the price/profit based competitive environment and provide results of its use on some illustrative examples. Recent research describes a genetic algorithm unit commitment program for the less regulated environment [Kondragunta, 1997]. Although there is some overlap between his work and the work described in this chapter, the market framework assumptions described in chapter 2 lead to different conclusions primarily regarding the way EDC is done and the obligation to serve.

The remainder of this chapter is organized as follows. Part 5.2 provides a brief description of the UC problem and formulation and highlights modifications needed for the competitive environment. Part 5.3 describes a genetic algorithm solution to the updated UC problem. Part 5.4 presents the results of some illustrative examples. Part 5.5 discusses implications of the updated UC on bidding strategies. Part 5.6 outlines a method for making the strategies more robust under conditions of uncertain demand and prices. Finally, part 5.7 provides some conclusions and identifies areas of future work.

### **5.3 Updating unit commitment**

For the vertically integrated monopolistic environment, UC is loosely defined as the scheduling of generating units to be on, off, or in stand-by/banking mode, and is done in such a manner that costs are minimized and constraints like demand and reserves are met. Considering inputs like variable fuel costs,

start-up and shut-down parameters/constraints of each power plant, and crew constraints adds to the complexity of the problem. The schedules are valued based on their costs. In order to determine the cost associated with a given schedule, an economic dispatch calculation (EDC), where each of the non limit-constrained operating units is set so that their marginal costs are equal, must be performed for each hour under consideration. One possible way to determine the optimal schedule is to do an exhaustive search. Exhaustively considering all possible ways that units can be switched on or off for a small system can be done, but for a reasonably sized system the amount of time it would take is too long. Solving the UC problem for a realistic system generally involves using methods like Lagrangian relaxation, dynamic programming, genetic algorithms or other heuristic search techniques. Many references for the traditional UC are available [Sheblé and Fahd, 1994; Wood and Wollenberg, 1996].

In the past, demand forecasts advised power system operators of the amount of power that needed to be generated. In the future, spot and forward bilateral contracts will make part of the total demand known *a priori*. The remaining part of the demand will be predicted as in the past. However, the GENCO's share of the remaining demand may be difficult to predict since that share will depend on how its price compares to that of other suppliers. The GENCO's offer price will depend on its prediction of its share of this remaining demand since that will determine how many units they have switched on or have in banking mode. The UC schedule directly affects the average cost and indirectly the offering price, making it an essential input to any successful bidding strategy.

In the past, utilities had an obligation to serve their customers. This was translated into a demand constraint that ensured all demand would be met. For the UC problem, this might have meant switching on an additional unit just to meet a remaining MW or two. With the obligation to serve gone, the GENCO can now consider a schedule that produces less than the forecasted demand. They can allow others to provide that 1 or 2 MWs that would have increased their average costs rather than altering their schedule to compete for a contract which they may be unlikely to secure.

Demand forecasts and expected market prices are important inputs to the profit-based UC algorithm; they are used to determine the expected revenue which affects the expected profit. If a GENCO comes up with two potential UC schedules each having different expected costs and different expected profits, it should take the one that provides for the largest profit, which will not necessarily be the one that costs least. Since prices and demand are so important in determining the optimal UC schedule, price prediction and demand forecasts become crucial. A good description of the UC problem and a stochastic solution that considers spot markets has been presented [Takriti et al., 1997]. Another one of the main differences of their work with the work described in this chapter is that they chose to minimize costs rather than maximize profits.

The existence of liquid markets gives energy trading companies an additional source from which to supply power. To the GENCO, the market supply curve can be thought of as a pseudo-unit to be dispatched. The supply curve for this pseudo-unit represents an aggregate supply of all of the units participating in the market

at the time in question. The price forecast essentially sets the parameters of the unit. This pseudo-unit has no minimum up-time, minimum down-time, nor ramp constraints; there are no direct start-up and shutdown costs associated with dispatching the unit. As described later, each GENCO is responsible for its own unit's transitional costs which must be recovered through adjustments in its offering price. The offer should roughly be equivalent to the marginal cost of the unit at the hour considered shifted by some amount to account for profit and for transitional costs.

The liquid markets that allow the GENCO to schedule an additional pseudo unit, also act as a load to be supplied. The total energy supplied should consist of previously arranged bilateral contracts and bilateral or multilateral contracts arranged through the markets (and their associated reserves and losses). While the GENCO is determining the optimal unit commitment schedule, the energy demanded by the market (i.e., market demand) can be represented as another DISTCO or ESCO buying electricity. Each entity buying electricity should have its own demand curve. The market demand curve should reflect the aggregate of the demand of all the buying agents participating in the market.

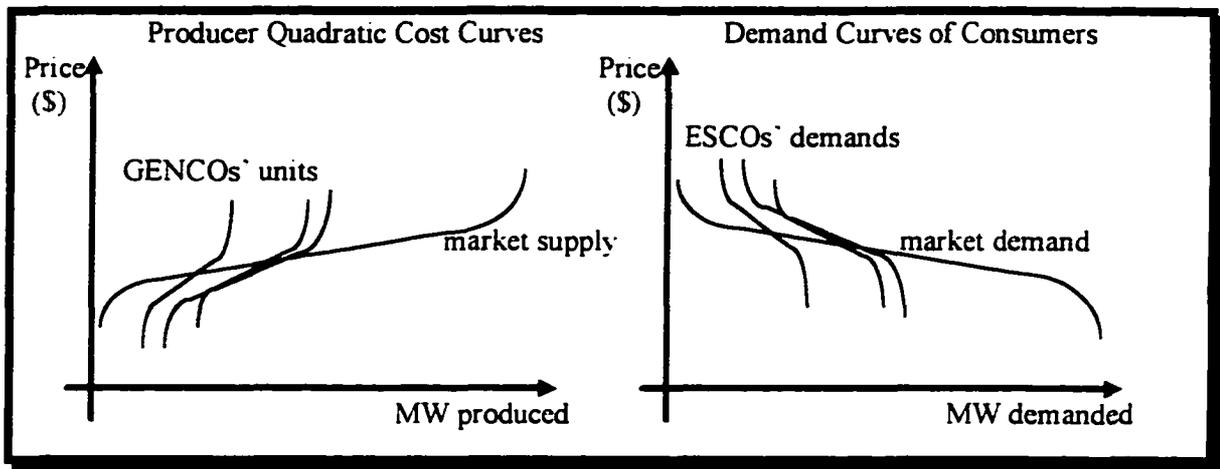


Figure 5.1 The market as an additional generator and an additional load.

Mathematically the traditional cost-based UC problem has been formulated as follows [Sheblé, 1985]:

$$\text{Minimize } F = \sum_n \sum_t^T [(C_{nt} + MAINT_{nt}) \cdot U_{nt} + SUP_{nt} \cdot U_{nt}(1 - U_{nt}) + SDOWN_{nt} \cdot (1 - U_{nt}) \cdot U_{nt-1}]$$

subject to the following constraints:

$$\sum_n (U_{nt} \cdot P_{nt}) = D_t \quad (\text{demand constraint})$$

$$\sum_n^N (U_{nt} \cdot P_{\max_n}) \geq D_t + R_t \quad (\text{capacity constraint})$$

$$\sum_n^N (U_{nt} \cdot Rs_{\max_n}) \geq R_t \quad (\text{system reserve constraint})$$

As we redefine the UC problem for the competitive environment, the demand constraint changes from an equality to less than or equal (we assume that the buyers purchase reserves per contract) relationship, and the objective function shifts from cost minimization to profit maximization as shown in the formula below. The updated UC process is shown in block diagram form in Figure 5.2.

$$\text{Max } \Pi = \sum_n^N \sum_t^T (P_{nt} \cdot fp_t) \cdot U_{nt} - F \quad (\text{expected revenue} - \text{expected costs})$$

subject to:

$$\sum_n^N (U_{nt} \cdot P_{nt}) \leq D'_t \quad (\text{new demand constraint})$$

$$P_{\min_n} \leq P_{nt} \leq P_{\max_n} \quad (\text{Capacity limits})$$

$$|P_{nt} - P_{n,t-1}| \leq \text{Ramp}_n \quad (\text{Ramp rate limits})$$

where individual terms are defined as follows:

$U_{nt}$	= up/down time status of unit n at time period t ( $U_{nt} = 1$ unit on, $U_{nt} = 0$ unit off)
$P_{nt}$	= power generation of unit n during time period t
$D_t$	= load level in time period t
$D'_t$	= forecasted demand w/ reserves for period t
$fp_t$	= forecasted price/MWhr for period t
$R_t$	= system reserve requirements in time period t
$C_{nt}$	= production cost of unit n in time period t
$SUP_{nt}$	= start-up cost for unit n, time period t
$SDOWN_{nt}$	= shut-down cost for unit n, time period t
$MAINT_{nt}$	= maintenance cost for unit n, time period t
$N$	= number of units
$T$	= number of time periods
$P_{\min_n}$	= generation low limit of unit n
$P_{\max_n}$	= generation high limit of unit n
$Rs_{\max_n}$	= maximum contribution to reserve for unit n

There may be a tendency to think that maximizing the profit is essentially the same as minimizing the cost. This is not necessarily the case. We have to remember that since we no longer have the obligation to serve, the GENCO may choose to generate less than the demand. This allows a little more flexibility in the UC schedules. In addition, our formulation assumes that prices fluctuate according to supply and demand. In the past engineers assumed that if they could level the load curve, they would be minimizing the cost. When maximizing profit, the GENCO may find that under certain conditions it may profit more under a non-level load curve. The profit depends not only on cost, but on revenue. If revenue increases more than the cost does, the profit will increase.

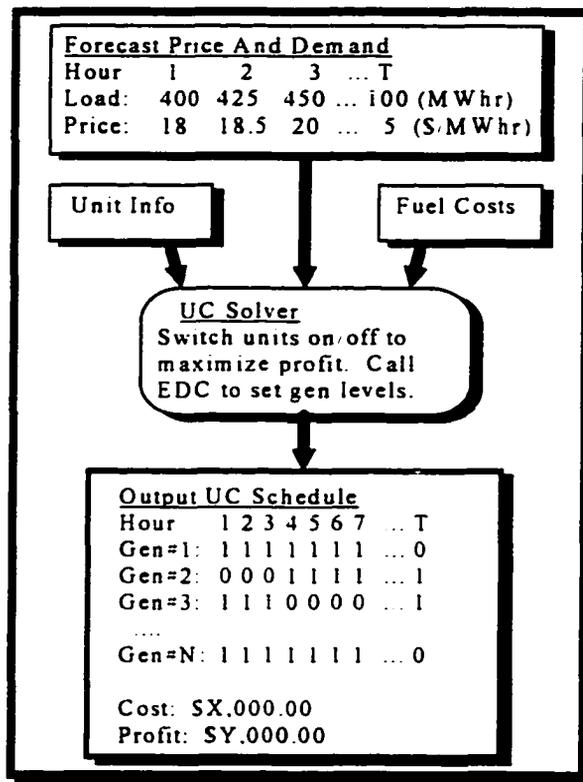


Figure 5.2 The UC problem.

The economic dispatch calculation (EDC) is an important part of UC. Formerly used to set generation so that costs were minimized subject to meeting a demand constraint (Figure 5.3 (a)), for the price-based UC that we present in this chapter, it was necessary to redefine EDC. Where the old EDC ignored transition and fixed costs to adjust the power level of the units until they each had the same incremental cost (i.e.,  $\lambda_1 = \lambda_2 = \dots = \lambda_t = \dots = \lambda_T$ ), our new EDC attempts to set production so that a pseudo  $\lambda$  equals the competitive price (i.e., produce until the marginal cost equal the price). The pseudo  $\lambda$  is the incremental cost modified to account for

transition and fixed costs and is shown in Figure 5.3 (b). A simple way to allocate the fixed and transitional costs which results in a \$/MWhr figure is shown in the following formula:

$$\lambda_i = fp_i - \frac{\sum_t \sum_n (\text{transition costs}) + \sum_t \sum_n (\text{fixed costs})}{\sum_t \sum_n P_{nt}}$$

Other allocation schemes that adjust the marginal cost/price according to the time of day or price of power would be just as easy to implement and should be considered in building bidding strategies. Other allocation schemes are discussed later in this chapter. Transition costs include start-up, shut-down and banking costs, and fixed costs (present for each hour that the unit is on), which would be represented by the constant term in the typical quadratic cost curve approximation. For the results presented later in this chapter, we approximate the summation of the power generated by the forecasted demand.

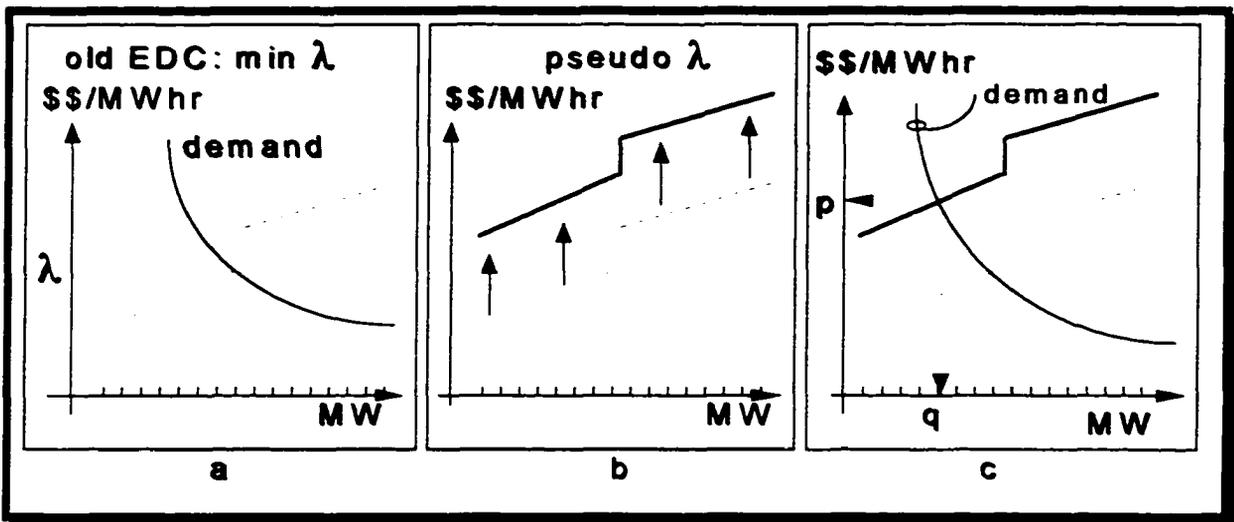


Figure 5.3 Defining a pseudo  $\lambda$ .

The competitive price is assumed to be equal to the forecasted price. If the GENCO's supply curve is indicative of the system supply curve, then the competitive price will correspond to the point where the demand and supply curves cross in Figure 5.3 (c). The EDC sets the generation level corresponding to the point where our GENCO's supply curve crosses the demand curve, or to the point where the forecasted price is equal to the supply curve, whichever is lower.

Figure 5.4 attempts to show how the fixed costs, the start-up costs, and the shut-down costs are allocated for a simple example. For this example, the load is assumed to be described by a simple triangular function as

shown in the top graph of Figure 5.4. This example assumes that 2 units are required to cover the peak load and that one unit can handle the load on the off-peak hours. For simplicity, the variable fuel costs are roughly proportional to the amount of electricity being generated. In the bottom graph of Figure 5.4 it can be seen that an average cost is added to the variable fuel cost resulting in a cost that looks like the variable costs, but shifted up.

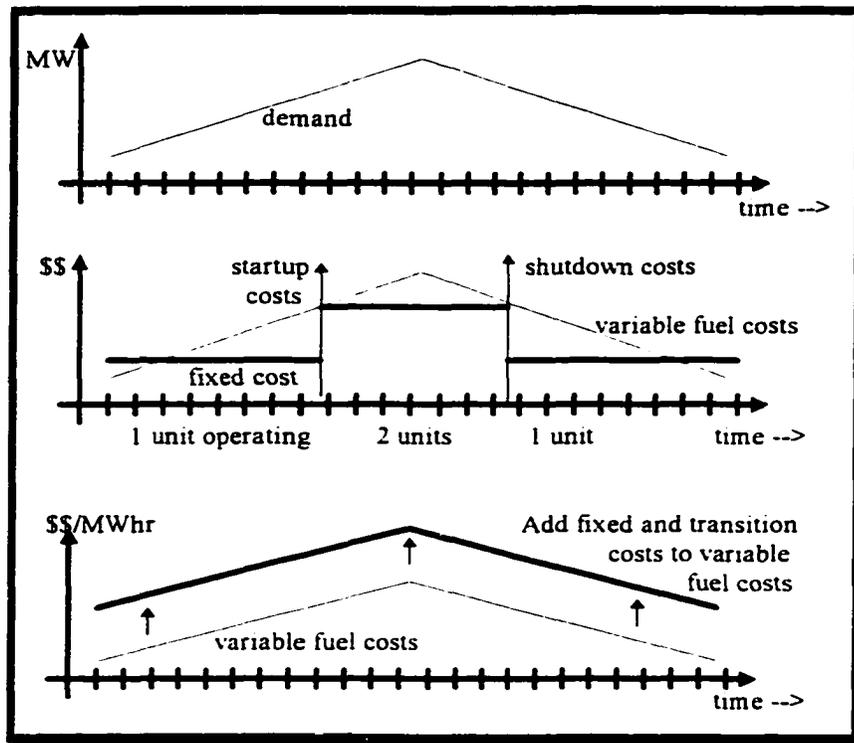


Figure 5.4 Example of fixed and transition costs allocation.

## 5.4 Genetic-Based UC algorithm

### 5.4.1 The basics of genetic algorithms

A genetic algorithm is a search algorithm often used in nonlinear discrete optimization problems. Data, initialized randomly in a data structure appropriate for the solution to the problem, evolves over time and becomes a suitable answer to the problem. GAs were inspired by the biological notion of evolution. See chapter 2 for additional details of genetic algorithms.

### 5.4.2 GA for price-based UC

The algorithm presented in this chapter for solving the new UC problem is a modification of the genetic-based UC algorithm described by [Maifeld and Sheblé, 1996]. Most of the modifications are to the fitness function which no longer minimizes cost, but maximizes profit. In addition, more user friendly I/O routines were added to make it easier to load input data and export the results. The intelligent mutation operators are preserved in their original form. The form of the schedules are the same. The updated algorithm is as shown in block diagram format in Figure 5.5.

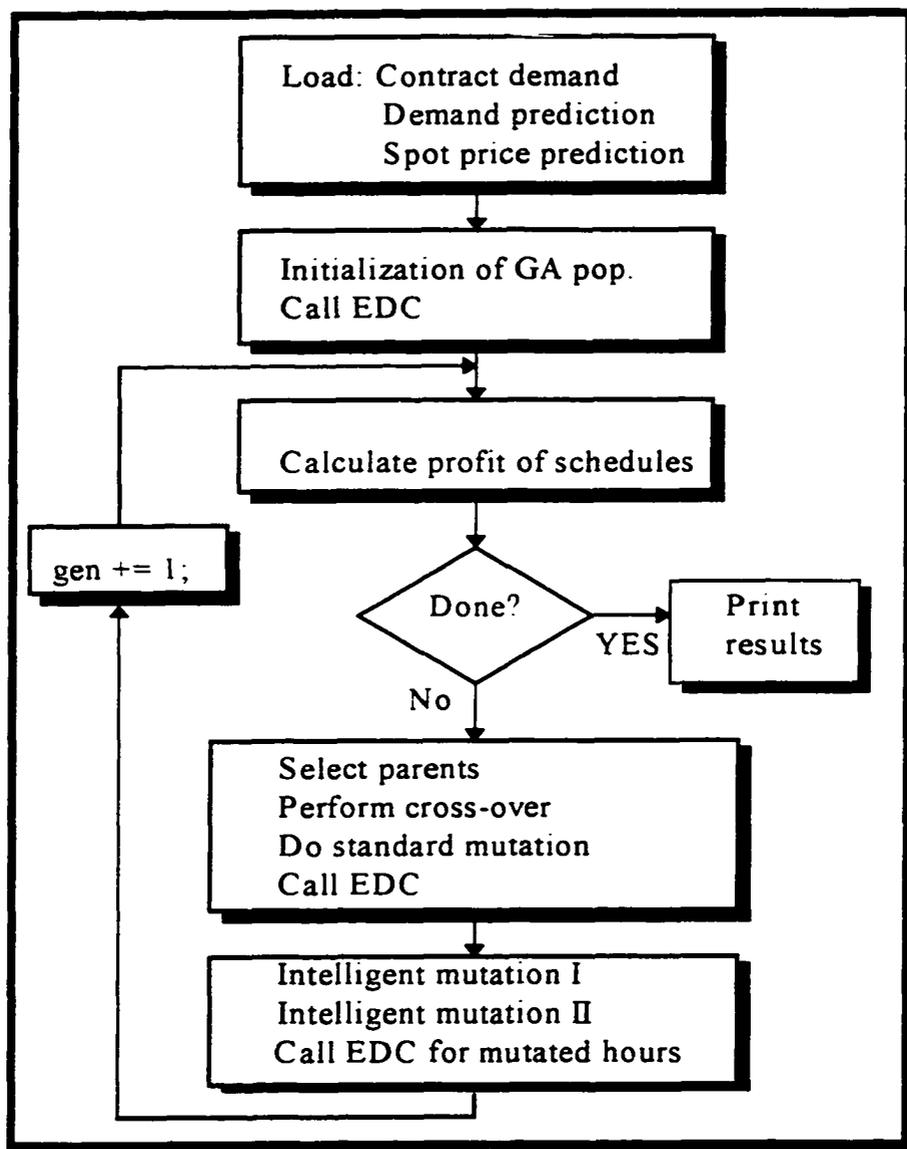


Figure 5.5 GA UC block diagram.

Another modification which makes the algorithm described here different than the previous algorithm is that a new EDC routine was written. This new EDC sets each unit's generation level such that its marginal cost (modified to account for fixed and transition costs) is equal to a hourly price as described earlier in this chapter. Any power generated in excess of the demand will not generate any revenue, but will add to the cost. This will be reflected in the fitness of the schedule which is equal to the profit and these schedules should die out quickly.

The algorithm first reads in the contract demand and prices, the forecast of remaining demand and forecasted spot prices (which are calculated for each hour by another routine not described here). During the initialization step, a population of UC schedules is randomly initialized. See Figure 5.6. For each member of the population, EDC is called to set the level of generation of each unit. The cost of each schedule is calculated from the generator and data read in at the beginning of the program. Next, the fitness/profit is calculated. "Done?" checks to see whether we have reached the maximum generations allowed, or whether we have met other stopping criteria (at this time we are simply using the number of generations). If done, then the results are written to a file which will be used as an input for our bidding strategy builder described in other chapters. If not done, the algorithm goes to the reproduction process.

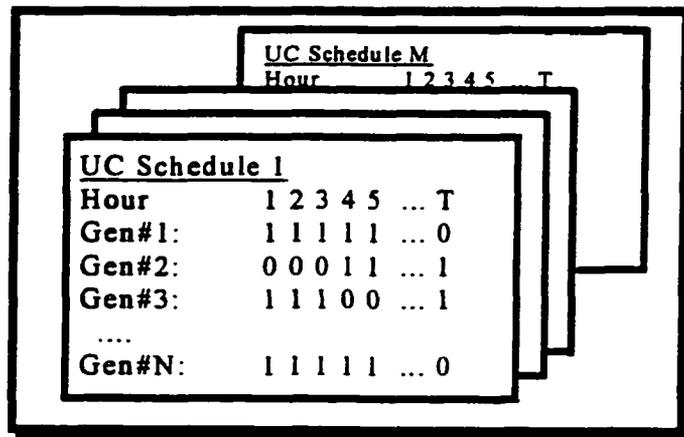


Figure 5.6 A population of UC schedules.

During reproduction, new schedules are created. The first step of reproduction is to select parents from the population. After selecting parents, candidate children are created using two point crossover as shown in Figure 5.7. Following crossover, standard mutation is applied. Standard mutation involves turning a randomly selected unit on or off within a given schedule.

An important feature of the previously developed UC-GA [Maifeld and Sheblé, 1996] is that it spends as little time as possible doing EDC. After standard mutation, EDC is called to update the profit for the mutated hour(s). An hourly profit number is maintained and stored during the reproduction process which dramatically reduces the amount of time required to calculate the profit over what it would be if EDC had to

work from scratch at each fitness evaluation. In addition to the standard mutation, the algorithm uses two “intelligent” mutation operators that work by recognizing that, because of transition costs and minimum up and down times, 101 or 010 combinations are undesirable. The first of these operators would purge this undesirable combination by randomly changing 1s to 0s or vice versa. The second of these intelligent mutation operators purges the undesirable combination by changing 1 to 0 or 0 to 1 based on which of these is more helpful to the profit objective.

UC Schedule Parent 1						
Hour	1	2	3	4	5	... T
Gen#1:	1	1	1	1	1	... 0
Gen#2:	0	0	0	1	1	... 1
Gen#3:	1	1	1	0	0	... 1
Gen#4:	1	1	1	1	1	... 0
Gen#5:	0	0	0	1	1	... 1
Gen#6:	1	1	1	0	0	... 1

UC Schedule Parent 2						
Hour	1	2	3	4	5	... T
Gen#1:	1	1	1	1	1	... 0
Gen#2:	1	1	1	1	1	... 0
Gen#3:	1	1	1	1	1	... 0
Gen#4:	1	1	1	1	1	... 0
Gen#5:	1	1	1	1	1	... 0
Gen#6:	1	1	1	1	1	... 0

UC Schedule Child 1						
Hour	1	2	3	4	5	... T
Gen#1:	1	1	1	1	1	... 0
Gen#2:	0	0	1	1	1	... 1
Gen#3:	1	1	1	1	1	... 1
Gen#4:	1	1	1	1	1	... 0
Gen#5:	0	0	1	1	1	... 1
Gen#6:	1	1	1	1	1	... 1

UC Schedule Child 2						
Hour	1	2	3	4	5	... T
Gen#1:	1	1	1	1	1	... 0
Gen#2:	1	1	0	1	1	... 0
Gen#3:	1	1	1	0	0	... 0
Gen#4:	1	1	1	1	1	... 0
Gen#5:	1	1	0	1	1	... 0
Gen#6:	1	1	1	0	0	... 0

Figure 5.7 Two point crossover on UC schedules.

### 5.5 Price-Based UC-GA results

The UC-GA was run on a small system so that its solution could be easily compared to a solution by exhaustive search. Before running the UC-GA, the GENCO needs to first get an accurate hourly demand and price forecast for the period in question. Developing the forecasted data is an important topic, but beyond the scope of our analysis. For the results presented in this section, the forecasted load and prices are taken to be those shown in Table 5.1. In addition to loading the forecasted hourly price and demand, the UC-GA program needs to load the parameters of each generator to be considered. We are modeling the generators with a quadratic cost curve (e.g.,  $A + B(P) + C(P)^2$ ), where  $P$  is the power level of the unit. The data for the 2 unit case is shown in Table 5.2.

In addition to the 2 unit cases, a 10 unit, 48 hour case is included in this chapter to show that the GA works well on larger problems. While dynamic programming quickly becomes too computationally expensive

to solve, the GA scales up linearly with number of hours and units. Figure 5.8 shows the costs and average costs (without transition costs) of the 10 generators, as well as the hourly price and load forecasts for the 48 hours. The data was chosen so that the optimal solution was known a priori. The dashed line in the load forecast represents the maximum output of the 10 units.

**Table 5.1 Forecasted demand and prices (2 generator case).**

Hour	Load forecast (MWhr)	Price forecast (\$/MWhr)	Hour	Load forecast (MWhr)	Price forecast (\$/MWhr)
1	285	25.87	8	328	8.88
2	293	23.06	9	326	9.12
3	267	19.47	10	298	8.88
4	247	18.66	11	267	25.23
5	295	21.38	12	293	26.45
6	292	12.46	13	350	25.00
7	299	9.12	14	350	24.00

**Table 5.2 Unit data for 2 generator case.**

	Generator 0	Generator 1
Pmin (MW)	40	40
Pmax (MW)	180	180
A (constant)	58.25	138.51
B (linear)	8.287	7.955
C (quadratic)	7.62e-06	3.05e-05
bank cost (\$)	192	223
start-up cost(\$)	443	441
shut-down cost(\$)	750	750
min-up time (hr)	4	4
min-down time (hr)	4	4

Before running the UC-GA, the user needs to specify the control parameters shown in Table 5.3, including the number of generating units and number of hours to be considered in the study. The 'popsize' is the size of the GA population. The execution time varies approximately linearly with the popsize. The number of generations indicates how many times the GA will go through the reproduction phase. System reserve is the percentage of reserves that the buyer must maintain for each contract. Children per generation tells us how much of the population will be replaced each generation. Changing this can affect the convergence rate. If there are multiple optima, faster convergence can trap the GA in local sub-optimal solution. 'UC schedules to keep' indicates the number of schedules to write to file when finished. There is also a random number seed that is set between 0 and 1.

In the 2 generator test cases, the UC-GA was run for the units listed in Table 5.2, and for the forecasted loads and prices listed in Table 5.1. The parameters listed in Table 5.3 were adjusted accordingly. To ensure that the UC-GA is finding optimal solutions, an exhaustive search was performed on some of the smaller cases. Table 5.4 shows the time to solution in seconds for the UC-GA and the exhaustive search methods. For small cases the exhaustive search was performed and solution time compared to that of the UC-GA. Since the exhaustive search solution times were estimated to be prohibitively lengthy, the latter cases were not compared against exhaustive search solutions. Cases with known optimal solutions were used to verify that the UC-GA was, in fact, working for the large cases.

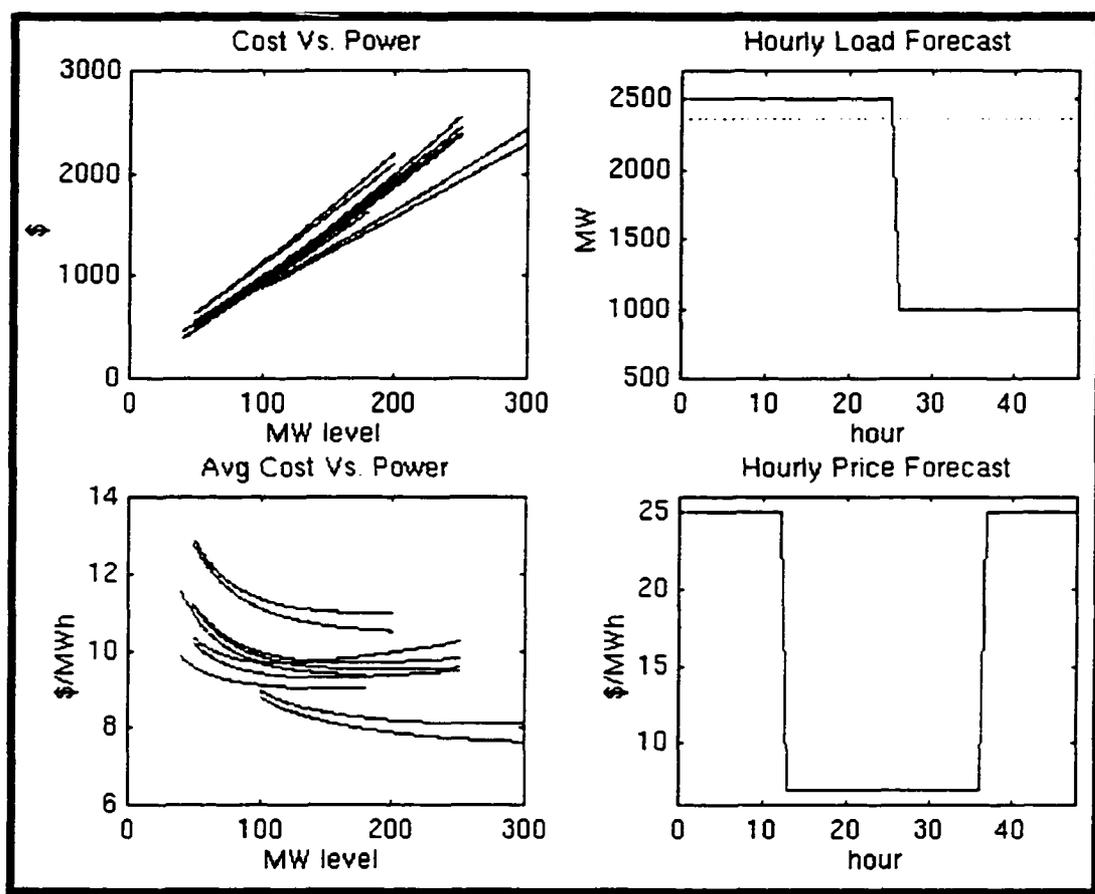


Figure 5.8 Data for 10 unit, 48 hour case.

Table 5.3 GA control parameters.

Parameter	Setpoint	Parameter	Setpoint
# of Units	2	System Reserve (%)	10
# of Hours	10	Children per generation	10
Popsize	20	UC Schedules to keep	1
Generations	50	Random number seed	.20



In the schedules shown in Table 5.5, it may appear as though minimum up and down times are being violated. When calculating the cost of such a schedule, the algorithm ensures that the profit is based on a valid schedule by considering a zero surrounded by ones to be a banked unit, and so forth. In addition, note that we only show the best solution of the population for each of the cases. To show more would take too much space. The additional valid solutions, which may have nearly as much profit, are one of the main advantages of using the GA. This gives the system operator the flexibility to choose among a group of schedules to accommodate things like forced maintenance.

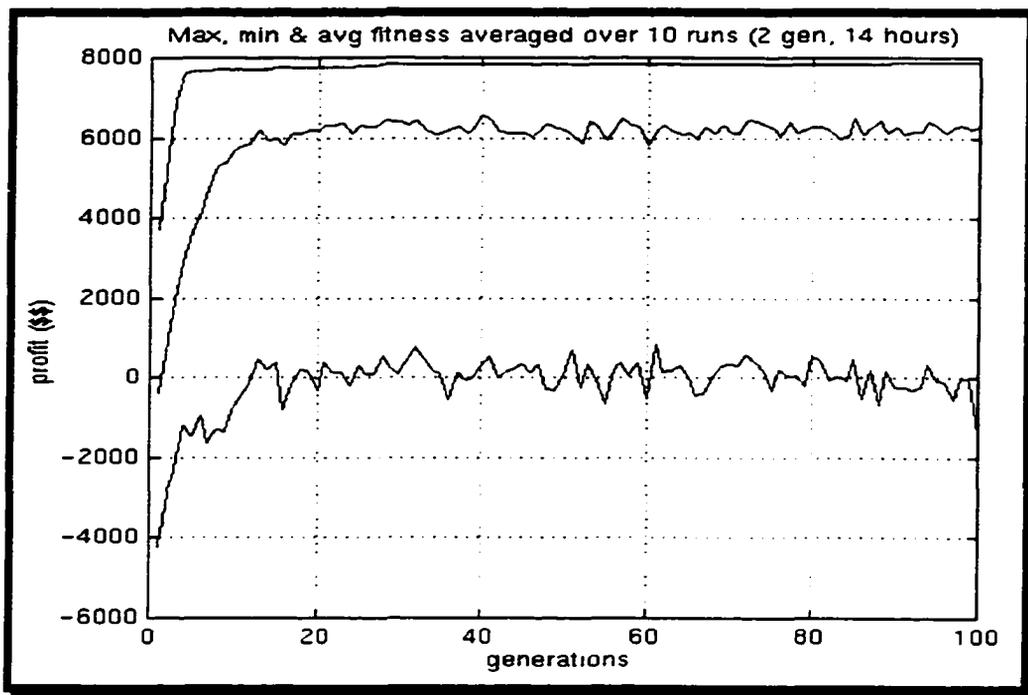


Figure 5.9 Max., min., and avg. fitness vs. GA generations for the 2 generator, 14 hour case.

## 5.6 UC and bidding strategies

UC will remain an important tool in the new environment. Although customers are no longer guaranteed, bilateral contracts will ensure that the GENCO knows the majority of its load ahead of time. An accurate forecast of the remaining demand and hourly prices will be important inputs for solving the UC problem. Once the UC schedules are generated, they will be of little use to the GENCO unless it can actually win customers from competitors at the price that it assumed in determining the UC schedule. For this reason, the UC schedule becomes an important input to the hourly bidding strategy builder.

When using a GA to solve the UC problem, solutions with more profit are valued more highly than those with less profit. GAs evolve because of selection pressure that works well when a fitness gradient exists. The

population of solutions that the GA ultimately finds, tends to be filled with solutions that have roughly the same profit levels. If all of the solutions have the same profit level, evolution may come to a standstill. It is quite common in the GA field to add a secondary fitness measure that takes over when the fitness gradient of the primary fitness function becomes small. A natural application of that procedure for this problem would be to measure the solution's flexibility. Since forecasts of load and prices may vary widely from the actual prices and loads, we may wish to reduce or increase the amount of power that we are generating as we get closer to the time of production. A schedule that allows a wider range of possible power generation levels without switching additional units on or off, is more valuable than a schedule that is very rigid. So the primary fitness measure becomes the profit level, but the flexibility of the schedule also plays a part in judging which schedule is better. When a GA has a second (or more) fitness measure, it is said to have a lexical fitness function.

One can begin to think of an entire UC schedule as being a bidding strategy, or as having a very close connection to hourly bidding strategies. Another area in which this is evident is the method of assigning the transition and fixed costs associated with the schedule. Above we described a scheme that allocated these costs in proportion with the number of MWs being produced at any given period. Allowing other factors (e.g., time-of-day, day-of-week, time of increased competition) to influence this allocation could provide additional profits. In a market in which one is submitting an entire schedule this can be very important. One might shift peak period costs to off peak hours to win the bid which may be decided on the price of a particular time of day. However, in a market framework as described earlier in this thesis, this is less important. One can argue that allowing utilities to shift costs to off-peak hours to win a bid will decrease the total social welfare, since their total cost of generation is higher. Furthermore a market framework that awards utilities contracts based on the price or cost they quote for a certain hour of the day or week and then guarantees that the generation companies will recover their start-up and shut-down costs is sub-optimal.

### 5.7 Handling uncertainty in price and load forecasts

After the solution is found, one might be curious as to what would happen if the forecasts were different. Suppose the price at hour 100 was 10% higher than anticipated. At present, the UC-GA user can modify the forecasts and re-run the algorithm. However, the possible set of cases that might be of interest could be large. Re-running the algorithm for each case could be quite time consuming, even if the algorithm starts from the near optimal solution found before making the change to the forecast.

When forecasting loads and prices, the further into the future is the time period, the more likely it is that the forecast will deviate from the expected value. So, along with the expected price and load for each time period, our industrious engineer in the forecasting department decides to give us these expected values along with error bars which indicate how confident he is of their precision (see Figure 5.10). How can we use this additional information?

We intend to use this additional information as follows. Rather than use the profit calculated directly by using the expected price and load curves, we assume that there is a distribution (probably gaussian) of values about that data point. We use Monte Carlo sampling to determine an expected profit. We draw a number of data points from the distribution of profits at that hour and a corresponding number of data points from the distribution of load forecasts at that hour. These will be used in determining the expected profit of that particular unit commitment schedule. During each generation of the genetic algorithm we monitor the fitness of each member of the population. Early on in the evolution of the genetic algorithm, we will use a small number of data points to speed up the evolution. As the UC schedule solutions in the population come closer to the optimal, we increase the number of data points. With no proof that we are ever going to get to the optimal, how do we know when to begin increasing the number of Monte Carlo sample points? The variance of the solutions is a good indicator of when to increase the number of data points.

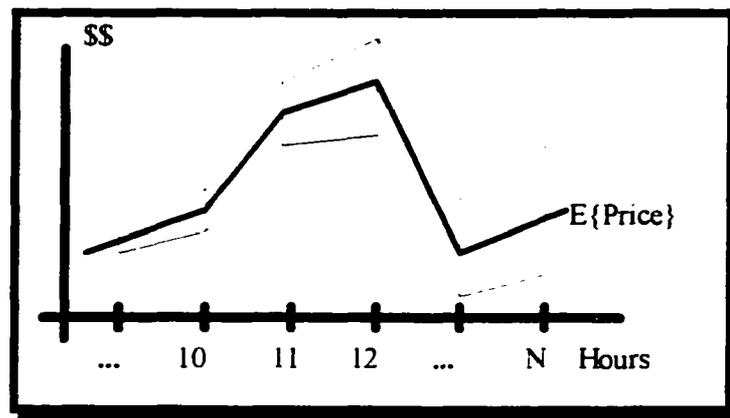


Figure 5.10 Expected prices with error bars.

The result of the above should be a schedule which would provide the largest expected profit. The method described above, is expected to increase the computational requirements of the GA, but we still expect reasonable solution times as the UC-GA performs quite rapidly.

### 5.8 Conclusions and future research

The UC-GA has been rewritten for price-based operation. Some might argue that UC schedulers are no longer needed—a GENCO can just go to the spot market to buy the electricity it needs. This can and should be considered a valuable option, but the GENCO's business is still one of generating electricity and they ultimately need to come up with a schedule by which they operate their generating units. The GA is a useful tool in searching large discrete solution spaces, and the space of solutions is quite large, making GA appropriate for the UC problem.

Ideally, the GENCO would run the UC-GA for the expected prices and demands that they consider most likely. These prices and demands may be uncertain. Running several cases would allow the user to know how sensitive the schedules are to variations in the inputs. Our UC-GA is presently being enhanced to provide the user with additional information that identifies which schedules allow the user more market flexibility for a given level of profit. We are essentially building in an on-line sensitivity analyzer.

## **6 IMPROVING BIDDING STRATEGIES THROUGH INTELLIGENT DATA MINING**

### **6.1 Chapter overview**

This research reports that an investigation of GAs and other so-called artificial intelligence techniques revealed their ability to search through large databases in order to learn the expert system rules that were used to develop that database. Commercial grade software using this technique which is presently being used to develop standardized treatment methods for hospital patients receiving medical care has been developed. Based on extensive records, software (which has since been sold to hospitals) is able to determine what the doctor did based on patient conditions. Although some modifications may be required to make this marketable for the trading industry, this GA-based software can be used on a database of trading data to infer the bidding rules that traders used to generate the bids in the database. The GA evolves a population of bidding rules that achieve the same results as those being used by the traders who generated the bidding data. Determining the rules that other electricity traders and brokers are using should be of great benefit to those who wish to gain a competitive edge when participating in the deregulated market. Such a tool would also be welcomed by regulating agencies who wish to ensure that the markets are efficient and fair.

### **6.2 Introduction**

Many industries, (e.g., power systems, commodity trading, health care) have problems that require highly specialized knowledge to solve or diagnose. These problems have typically been addressed by human experts who have had much training and have domain specific knowledge. This knowledge is the fundamental ingredient of an expert's problem solving abilities. Systems equipped with the appropriate knowledge have been shown to demonstrate expert level performance in many applications. Despite the success, the current state of expert system technology suffers from some serious limitations. One of these limitations is that the development of an expert system, for the most part, remains an art. While tools and methodologies have emerged to provide considerable help, the process of representing and refining the knowledge utilized by the domain expert remains ill-defined and time consuming. Application of expert systems still remains restricted to fairly narrow, self contained problem domains, and performance typically degrades sharply as the system approaches the boundaries of human knowledge. With traditional computer based expert systems, there is little ability to adapt or reorganize knowledge as performance requirements change over time. The greatest potential for removing these limitations lies in the area of machine learning. Expert systems for problems

with a large number of inputs can be quite complex and the best solution may be difficult to identify. This is especially true when the function is a nonlinear problem. A group of expensive experts may disagree on the best method to solve the problem in question. This can be quite expensive if the experts are paid large salaries like hospital doctors. In this chapter we present an intelligent data-mining technique for searching through a database of historical information that infers the expert system rules used by the human experts, and compares competing sets of expert system rules based on the profit associated with using them. Our solution to the limitations of expert systems developed by the Delphi Method (where people sit at a table discussing each point of a problem until they agree on the best action for that aspect of the problem) involves an adaptive learning strategy based on genetic algorithms.

### 6.3 Applications of expert systems

Many industries have control applications that are appropriate for expert systems. For example, in the health care field, there are many areas that might be improved through the use of an expert system. Many doctors would welcome an expert system that could instantaneously suggest a diagnosis or prescribe tests based on symptoms and history of an admitted patient. Power system operators operating in an environment where many inputs are changing simultaneously could benefit from an expert control system. On occasion, differences in doctors' training, opinions and methodologies could lead them to diagnose the same patient's problem differently. Likewise, a commodity trader who is faced with a set of inputs (e.g., Dow Jones Index has fallen 10% in the last two hours, the fed raised interest rates yesterday, etc.) might decide on a different position than his peer at the same firm. If an expert system were to be implemented in these cases, it could reduce the number of these inconsistent diagnoses or reactions to market information. An expert system implemented at a hospital would be useful in diagnosing patients more quickly, which would be especially helpful in trauma centers where correct decisions must be made immediately. Implementation of an expert system can sometimes provide a solution or diagnosis for a problem that in the past was not solvable since computers are able to handle many more inputs than a trained human expert. A number of expert systems have already been implemented successfully in the healthcare field.

Electrical power system operators have applications that have also benefited from expert systems. In recent research, expert systems have been used to help the operator in control of electrical power systems to maintain safe operating conditions by controlling real time data acquisition and numerical algorithms (load flow, stability analysis) [Germond and Niebur, 1992; Huneault et al., 1994]. In fact, the use of expert systems has been studied for many areas within the electric utility industry and many papers have been published regarding the subject. Some of these areas include: distribution planning, system design, feeder configuration, substation considerations, planning, operations, transmission planning, operations, load

forecasting, generation commitment scheduling, load shedding, security assessment, and line overload alleviation.

If the limitations of the expert system development discussed in the introduction were removed, we might see a more widespread implementation of expert systems. In this chapter we report on research being conducted that explores the use of genetic algorithms to learn or infer the expert system rules used in treating patients in a hospital. We also describe how to use this research for inferring bidding strategies for the competitive electric marketplace. The techniques presented here were developed for the healthcare field, but the methods are directly applicable to other industries. Some minor modifications may be needed to tailor the software for inferring bidding rules from trading data, and these modifications will be pointed out if appropriate. For the most part, the method of using genetic algorithms to learn or infer expert system rules from a generic database remain the same as those used for the healthcare industry.

#### **6.4 The basics of machine learning**

Artificial intelligence is a term that is often heard when people are talking about systems that can do a job that would normally require a human. People will often have long debates defining terms like intelligence and what makes it real or artificial. We will leave the definitions to others for the time being and be a bit more practical. Among these so-called artificially intelligent systems are neural networks, complex lookup tables, fuzzy logic systems and genetic algorithms. It is quite common when using these techniques to have a learning phase during which the system's performance is tested. In the neural network field, researchers talk about supervised and unsupervised learning. During supervised learning, the system is presented with a set of inputs for which the correct response is known a priori. An output is generated and the system is rewarded or reinforced for producing what we think of as correct responses. In a system with unsupervised learning, a set of inputs is presented to the system and the system draws conclusions based on the spatial relationship of the data in a domain space. In general, machine learning can be thought of as the automatic improvement in the performance of a computer system over time, as a result of experience. Machine learning can be applied to almost any domain, but in practice the greatest successes have been related to classification tasks. Some of the advantages of an expert system based on a learning algorithm over the traditional type, are that the system with the learning algorithm:

1. Delivers more consistent solutions
2. May find inconsistencies in the data
3. Can find a rule for situations that don't occur very often
4. Obtains answers less expensively

In order for a machine or complex system to learn, one can argue that the following components are needed: system rules, a system model, a means of evaluating the rule's performance, a means of updating the rules or rule generating mechanism based on the evaluation of the rules performance, system inputs, and system outputs. See Figure 6.1. In such a setup, the rules are a set of information structures that encode the system's present level of expertise. The performer is a task algorithm that uses the rules to guide its activity. The critic is a feedback module that compares actual results with those desired, and the learner is a mechanism that uses feedback from the critic to amend the rules.

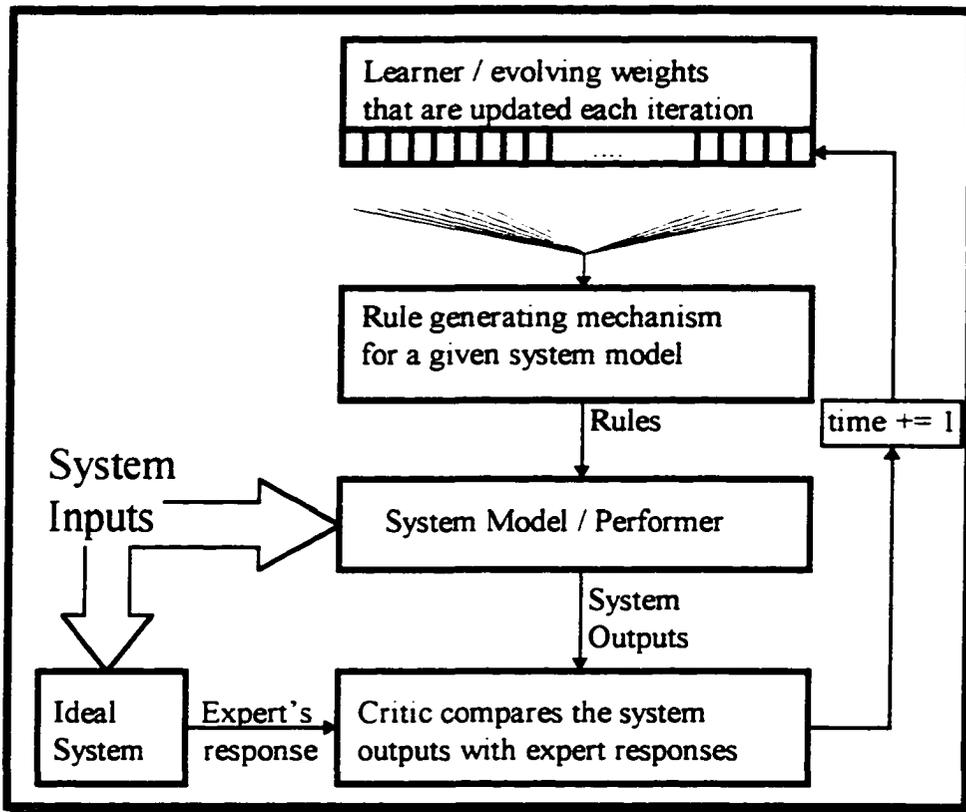


Figure 6.1 A general learning structure

Many problems are discrete in nature, with an input requiring a particular response. When such a problem has many inputs, it is often helpful to draw a decision tree that helps to separate and identify the best course of action for each combination of inputs. In any decision, and in any decision tree, we have natural nodes that represent inputs or conditions in the world, and decision nodes that represent a decision that must be made. In the decision tree, decision nodes are commonly drawn as squares, and natural nodes are often drawn as circles. See Figure 6.2 for the best path through a sample decision tree. For any set of inputs, there is an optimal combinations of decisions to be made. There is a specific cost and benefit associated with each path through the decision tree. If we are attempting to minimize the length of a hospital stay, then the

combination of decisions that results in discharging the healthy patient most quickly is the best. If we are attempting to find a bidding strategy that maximizes our profit, then the decision tree path representing the combination of decisions resulting in the largest profit for a given set of inputs is the optimal path. In most practical problems, there are many decision and natural nodes to be considered for a given problem. Decision trees with 40 or 50 levels, with each level having only a binary decision or input, approach the limits of what we can solve quickly with today's computers, assuming we search through all possible combinations to find the optimum.

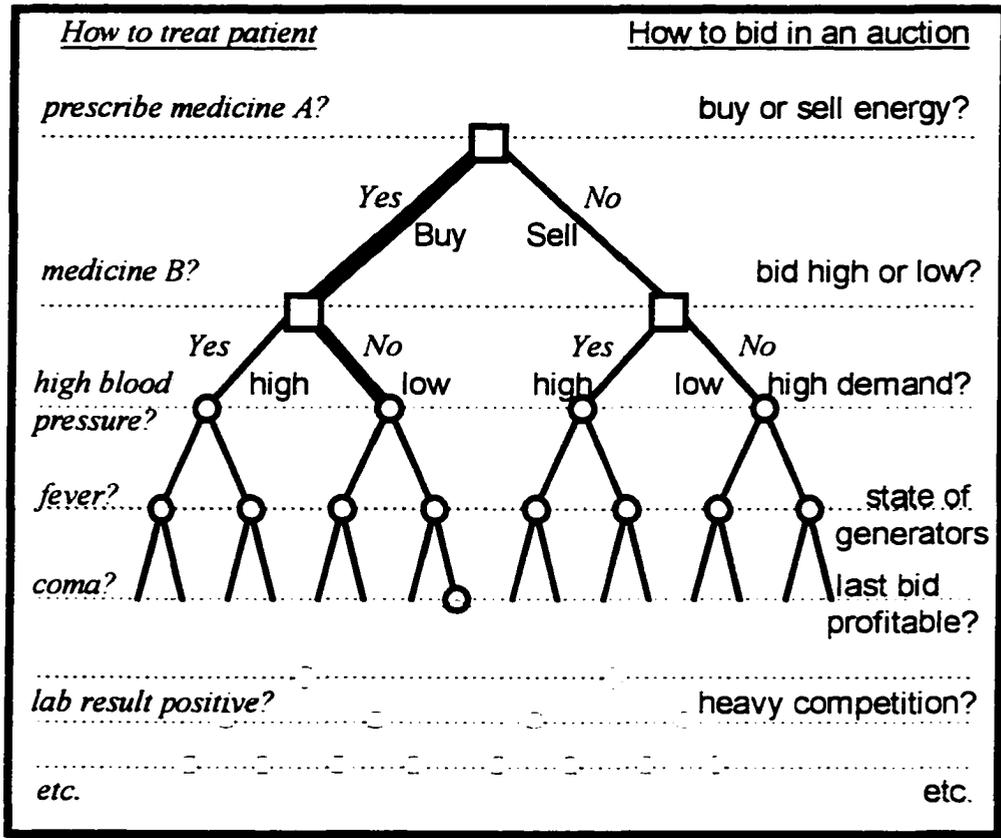


Figure 6.2 A sample decision tree for patient treatment or auction bidding.

In discrete problems, we can think of the learning process as a search through the space of possible decision tree paths to find the one with the best response to a given set of inputs. In anything other than trivial problems, the number of paths becomes enormous, and to use a completely enumerative search technique would be computationally infeasible with existing computer technology. When solving a problem, many of the available inputs may not have a large effect on the outcome of the decision, and can be ignored. The research we report on in this chapter uses a genetic algorithm to learn the expert system rules to correctly classify a system's inputs. The GA performs two functions while inferring the rules that experts have used to build the database. First, it learns which of the inputs are actually important, weeding out unnecessary inputs that

simply take up memory and slow down the decision process. Second, after deciding what input data is useful, it finds the appropriate output decisions to be made in light of the given inputs.

### **6.5 Genetic algorithms**

Developed by John Holland, genetic algorithms are a fairly robust optimization techniques that work well for many nonlinear discrete applications. They often will produce solutions to a problem when more traditional techniques fail. Inspired by the biological notion of evolution, populations of solutions to the problem being investigated evolve according to the principle of survival of the fittest. The research described in this chapter involves the use of genetic algorithms. The reader is encouraged to read more about the details of genetic algorithms in Chapter 2.

### **6.6 Expert systems**

The building of critical decision criteria can be accomplished by expert systems. In general terms, an expert system is a classification system which uses a production system of rules that embody the knowledge and experience of the expert. An expert system of rules can represent a knowledge base in a compact, computationally complete manner. The rules consist of an antecedent and a consequent. The antecedent identifies the condition to which you would like to respond. The consequent spells out the action to be taken for a given antecedent. Commonly the expert system rules are written in an if-then-else program construction (much like the decision trees described earlier). These statements allow the user to easily incorporate the problem specific criteria into the rule base.

The expert system rules are essentially set up in the form of a decision tree. Each node in the tree represents an if-then-else operator and each branch represents the path of the decision made at the previous node. For problems fitting this if-then-else construction, the number of possible outcomes doubles with each additional if-then-else node layer included. The expert system rules are evaluated beginning at the top node and then with each subsequent layer of nodes. Eventually, each node and branch of the tree will be tested.

Typical inputs in our patient treatment research might be patient demographic and dietary information, observed symptoms, preliminary test results, medications administered, and procedures prescribed. Among the typical inputs for our auction bidding strategy research might be fuel costs, transmission costs, forecasted prices, forecasted demands, unit outage information, and level of competition. Inputs like these are presented to the rules at the decision nodes. For example, if the patient has high blood pressure, then the corresponding node chooses to perform some action, if not, it will choose to perform some other action.

Generating the tree of structured if-then-else statements is a matter of deciding on the proper action for each of the possible inputs or combination of inputs. One traditional method of generating such a tree is

known as the Delphi method. The Delphi method is where a group of expert meet and then sit and argue about the best action for each input. The Delphi method can be expensive and time consuming.

### 6.7 Using GAs to evolve expert system rules

So if the Delphi method is expensive and time consuming, what can we do? Well, the method described in this research assumes that experts have been solving problems or responding to inputs and that both the inputs and corresponding responses have been stored in a database. If the database is large, it very well may contain sufficient information to determine most of the input-output relationships commonly observed in the system. A non-expert can 'simply' sort through this database and come up with these relationships. This is sometimes called data mining. When a database is large, data mining that performs an exhaustive search can be too computationally intensive. This is why we allow evolution (through genetic algorithms) to decide what part of the database should be searched. See Figure 6.3.

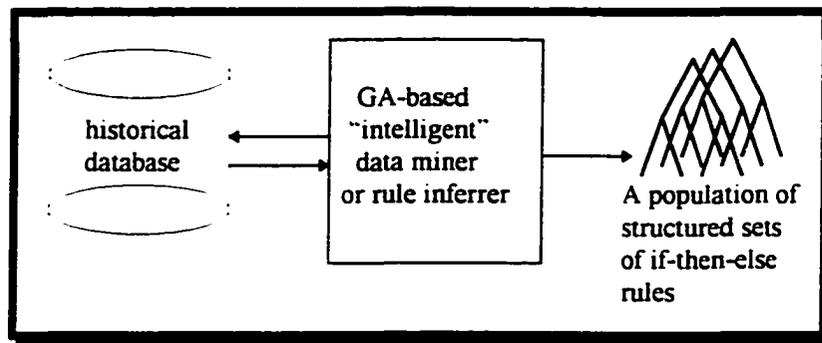


Figure 6.3 The GA “intelligently” searches through the database for rules.

#### 6.7.1 Data storage and preprocessing

The database stores information by patient or trader record, arranged also by the various inputs and either 8 hour shifts or rounds of bidding. Some of the inputs are active decisions that the expert made in the previous 8 hour shift or round of bidding (e.g., Tylenol was prescribed). Other inputs are more passive and may or may not be indirectly related to the experts decision (e.g., the interest rate decreased a half a point). As we increment time or the rounds of bidding, a previous hour’s decisions/outputs become the active inputs. See Figure 6.4. The way the GA reads from the database was developed specifically for the data we were getting from the healthcare industry. For each patient, at each 8 hour shift, for each input, there are either one or five bits that can be zero or one describing that input. For the single bit inputs, one signifies that the input was given, observed, or prescribed. For the five bit data, the first bit indicates whether that input was tested for, the second bit indicates whether the condition was normal or abnormal, the third indicates whether the

observed value was high, the fourth bit indicates whether the observed value was low, and the final bit indicates if the condition was extreme or not.

One could guess that the vast majority of these data points will be zeros. If the GA string is too long, there is a chance that evolution will take a long time to converge to the optimal. In order to focus the GA on those input-output relationships for which a rule is really necessary, a compression routine was developed. The compression routine removes inputs from the GA string if they were never observed, or if they were always observed. These inputs are stored and re-injected into the solution at the appropriate location, when the GA is done evolving.

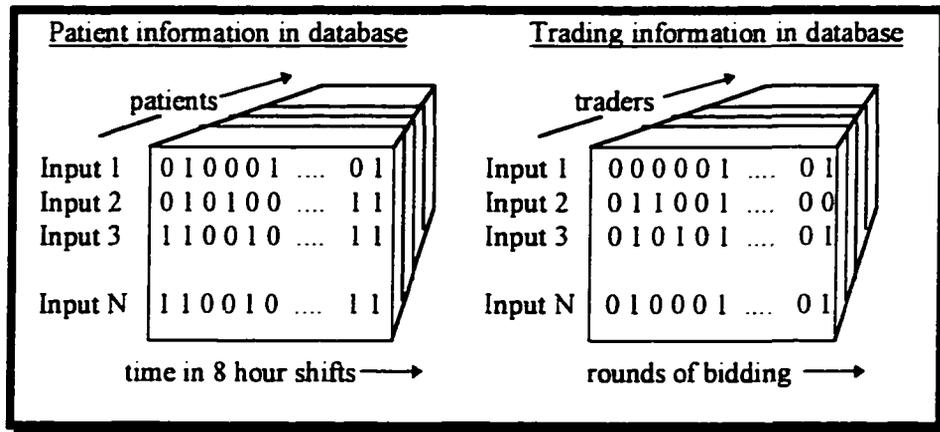


Figure 6.4 Database information.

### 6.7.2 A data structure to encode the solutions

The process begins with the initialization of a population of solutions each representing an expert system with its own set of rules. Initializing each member of the population involves filling a vector representing the inputs to consider and the output actions to be taken randomly with zeros and ones (where the inputs and outputs may be represented as zeros and ones). An assumption that we make is that the expert uses some subset of the total amount of input information available in making his decision. Part of the vector identifies which inputs should be considered important. The remainder of the randomly initialized data vector consists of the outputs to be performed when encountering the inputs identified in the first part of the gene. In our research, the first part of the gene is called the *in\_gene*, and the second part is called the *out\_gene*. See Figure 6.5 for a sample population. A gene is evaluated by first looking at the *in\_gene*. The *in\_gene* specifies which, of all possible inputs, should be considered when using this particular expert system rule. The *out\_gene* is a vector of ones and zeros where the ones correspond to possible actions to be taken and zeros are actions not requested or prescribed. For instance, in rule 1 from the figure, if inputs 1, 4, 5, 7, 8, and 12 are true, then the actions to be taken are identified by the ones in positions 1, 5, 6, 8, 9, 10, 11, and 14 of the *out\_gene*.

### 6.7.3 Calculating the fitness

To judge the fitness of the rules, each rule is used to develop bids or prescribe treatments for the traders and patients in the database. The actions taken by the GA generated rule are then compared with the actions originally taken by the expert. For each of the binary outputs there are four possible rewards that the rule may receive. The best scenario is to have the GA generated rule calling for an action which the human expert also called for. Second best is to have the GA generated rule calling for no action when none was taken by the expert. Third best is when the GA calls for action when the expert took no action. The remaining situation is when the GA does not take any action in a scenario where the expert did take action. The fitness is a summation of the rewards associated with the above four scenarios occurring for each output for each patient on which the rule is used. The weightings of these four situations are parameters of the program and can be changed based on the knowledge of the problem. We set them using our engineering judgment.

```

1 IF ( 1. 4. 5. 7. 8. 12) THEN (1 0 0 0 1 1 0 1 1 1 1 0 0 1)
2 IF ( 1. 3. 6. 9. 10. 12) THEN (1 1 0 0 1 1 0 1 0 1 1 0 0 1)
3 IF ( 1. 2. 5. 7. 9. 12) THEN (1 0 1 0 1 0 1 1 0 1 0 0 0 1)
4 IF ( 1. 4. 7. 8. 11. 12) THEN (1 0 0 1 1 1 0 1 0 1 1 0 0 1)
5 IF ( 2. 3. 4. 9. 10. 11) THEN (1 0 0 0 1 1 0 1 1 1 1 1 1 0 1)
...
N IF ( 4. 5. 6. 8. 9. 12) THEN (1 0 1 0 1 0 0 1 0 1 1 1 1 0 1)

```

Figure 6.5 Sample GA population.

When the rule's fitness is being calculated, a switch in the parameter file allows fitness to be determined either by its usage on only the best fitting patient or trader, or by the rules usage on as many traders or patients as there are which fit the `in_gene` part of the rule. The goal of the user should help to determine what fitness calculation method is preferred. Increasing the length of the `in_gene` reduces the number of patients or traders for which the rule will be suitable. Decreasing this `in_gene` length increases the number of patients or traders for which the rule will be suitable. If we require that all of the `in_gene` elements of a particular rule match a given trader's or patient's input data prior to its use, it will not be suitable for many scenarios. In order to increase the number of patients or traders for which the rule is applicable, a threshold value was included. The threshold is the minimum number of items within the `in_gene` that must be true for it to be suitable for treating a particular patient. The higher the threshold is set, the more specific the rule is (fitting only a small number of patients or bidding scenarios), and the lower the threshold is set, the less specific is the rule. The threshold is an easily adjustable parameter in the program.

Early prototypes of the software designed for health care assumed the doctor's decision for the next 8 hour shift ( $n+1$ ) was primarily dependent on inputs from the present 8 hour shift ( $n$ ). To include a limited

dependence on previous shifts, a feature (admittedly a hacked feature) that allowed inputs from the two shifts (n-1, n-2) prior to the current shift. While writing the software, preliminary discussions were held regarding future versions of the software based on genetic programming. A genetic programming version would allow the program to automatically determine dependencies on any input data prior to the current shift in the database. The developed software package has easily adjustable parameter settings which allow part of the input gene to specify inputs from shift n-1 and part from shift n-2. The adjustable parameters discussed in this section are listed with sample setpoints and descriptive remarks in Table 6.1.

**Table 6.1 Adjustable parameters used in the program.**

Parameter	Setpoint	Remarks
no of oneAs	29	Inputs coded w/ 1 bit that are taken as inputs and outputs
no of fivAs	15	Inputs coded w/ 5 bits that are taken as inputs and outputs
no of oneBs	10	Inputs coded w/ 1 bit that are taken only as inputs
no of fivBs	0	Inputs coded w/ 5 bits that are taken only as inputs
no of shifts	8	Number of 8 hour shifts or rounds of bidding in database
in gene len	10	How many inputs to consider in the in gene
no shift p2	1	No. of inputs in in gene that should come from 2 shifts ago
no shift p1	3	No. of inputs in in gene that should come from 1 shift back
rqstd threshold	8	Minimum no. of inputs in in gene which must be true
generations	5001	No. of generations that the GA should evolve
mut percent	20	Percent of mutation when creating child solutions
Xoverpts	3	No. of crossover points used in creating child solutions
best	50	Evolving rule took a positive action which doctor also took
secbest	10	Rule prescribed no action, and neither did the expert
thirdbest	-1	Rule prescribed an action, but expert did not
worst	-25	Expert took action, but rule did not
LOS bias	50	(1-100) rule bias to favor patients w/ short length of stay
method	1	(0/1) Base fitness on all fitting rule or only the best match
popsize	32	Population size of the GA
number new	16	How many children solutions to create each generation
no to save	10	How many evolved solutions to store to file when finished
no of patients	5	How many patient or trader records are available in database
select method	1	Parent selection method (0 rank, 1 roulette, 2 tournament)

#### **6.7.4 Creating new solutions through evolution**

After the fitness for each member of the population is calculated, parents are selected with some bias towards more highly fit solutions. The program has a switch in the parameter file that allows the user to select between three parent selection methods each having a different fitness bias. There are reasons which makes one more appropriate than the others depending on the situation we are in. For instance, roulette selection may have a tendency to prematurely converge to a local optima when the population size is small. After parents have been selected, the crossover and mutation operators are used to create children solutions. As

described earlier, crossover promotes the exploration of new regions in the search space. Crossover is a structured, yet randomized mechanism of exchanging information between strings. Mutation ensures that no string position will ever be permanently fixed at a certain value. Mutation in a binary alphabet operates by toggling the value in any given matrix position with some probability of mutation. The number of crossover points and mutation rate can be easily adjusted in the parameter file.

The best members of each generation are preserved for the next generation. The children replace the members of the parent generation with the lowest fitness. This is an elitist replacement strategy and ensures that the best individuals are never lost in moving from one generation to the next, meaning that if we were to graph the maximum fitness of the population at each generation, it would never decrease. Figure 6.6 shows the genetic algorithm used for this program in block diagram format.

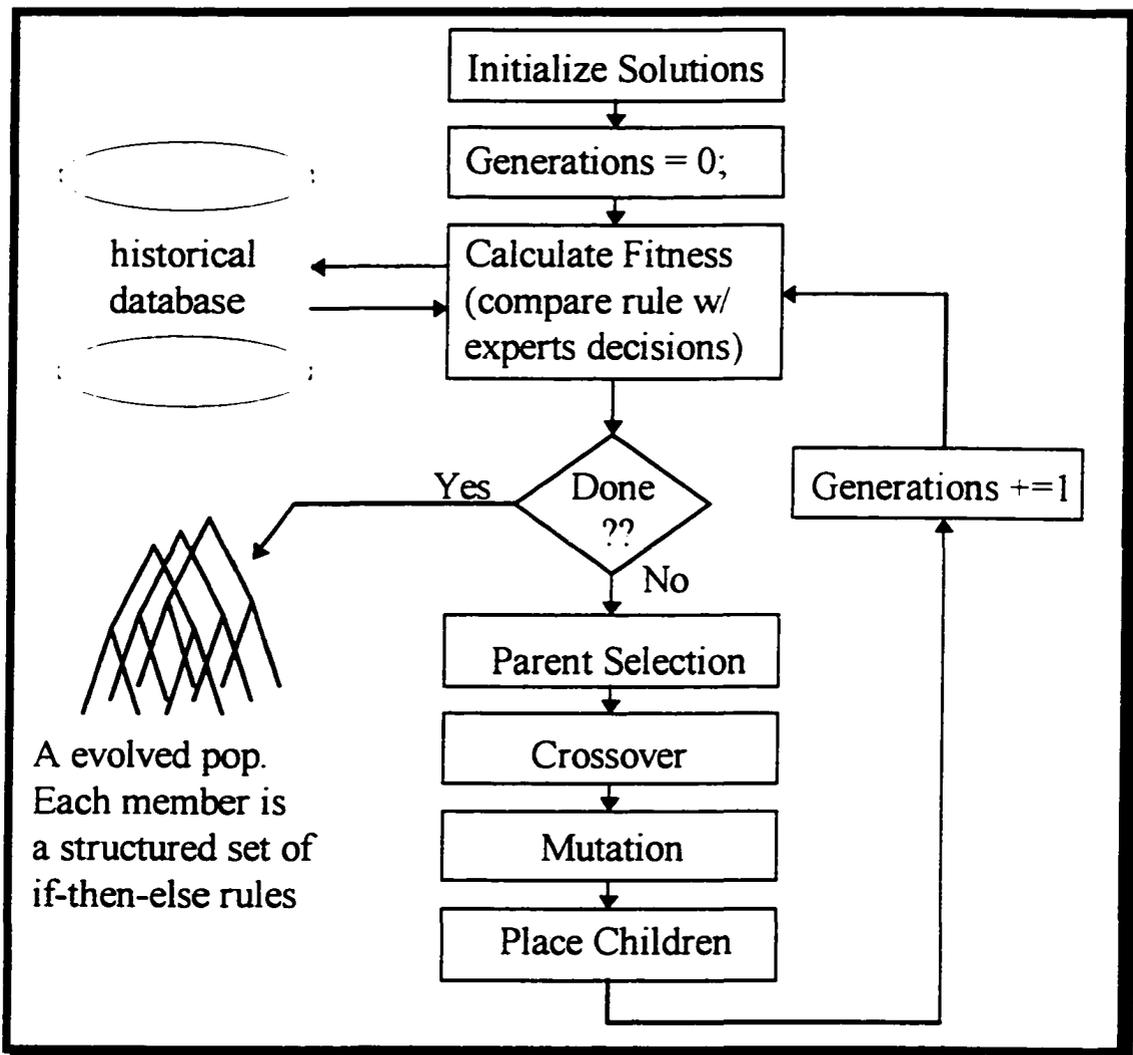


Figure 6.6 The GA searches through the database for good rules.

## 6.8 Results

The genetic algorithm expert system developer engine has been installed in several hospitals and clinics. It is presently being used to assist in the development of “critical pathways” for the healthcare industry. Because of the importance of ensuring the system of rules that make up a critical pathway will not endanger the patient, the intent is that qualified experts perform a comprehensive review process that identifies professional reasons for treating patients with the evolved rules. At present, the rules evolved by the GA only indicate the input to be used or prescribed and do not prescribe any levels or dosages. Even critical pathways developed by the Delphi method are reluctant to do this as it is best left to the doctor treating the patient.

For application to an auction bidding problem, we need to identify the parallelisms to the healthcare industry and set the parameters accordingly. For instance, the number of patients would be akin to a number of generation buses at which power would be sold (the idea being that a trader might have a separate set of trading rules for each bus); the number of shifts contained in the database becomes the number of bidding rounds contained in the database. The decisions are related to the bids which are to be submitted to an energy exchange. The set of rules that evolves may be specific for particular generation bus.

The inputs consisted of single bit data like which medications have been given, which procedures have been ordered, what symptoms have been observed, and five bit data which give the results of various lab tests that have been ordered. Often the results of lab tests are not available during the 8 hour shift that the test was performed. Outputs consisted of single bit data indicating tests that were prescribed, procedures ordered and medications prescribed by the doctors treating the patients. As described in the fitness calculation section, the expert systems constructed by the genetic algorithm were compared to the actual decisions made by the doctors. If needed, the five bit data trading data can indicate at what level an input is observed. The single bit data which serve as inputs only might indicate whether:

- system is (or is not) heavily loaded
- auction bidding is (or is not) currently highly competitive
- weather is (or is not) warm
- the time in question is (or is not) a weekday
- the time in question is (or is not) a holiday
- the time in question is (or is not) a peak period

The number of single bit data which are both input and output (depending on what point in time we view them from) should be equal to the number of binary decisions being made in any given problem. This could be correlated with the number of products that one is offering for sale, i.e., bid a high or low number of MWhr, make a high or low (\$/MWhr) offer for the real energy, likewise for reactive power, and the various components of the bids for each of the ancillary service desired or offered.

### 6.8.1 An example

We provide a small example to demonstrate the type of output generated by the genetic algorithm. The parameters used in this example are those listed in Table 6.1. The data was taken from patients who were admitted to an Iowa hospital for total knee replacement. The input data encoded with five bit which consist of things that the doctor might prescribe (active) is shown in Figure 6.7. For this particular example, there is no five bit passive data. The single bit data is shown in Figure 6.8. The goal is to build a rule that indicates what medicines to prescribe or procedures to order for the subsequent 8 hour shift. The choice of rewards for matching the doctors decisions favors a positive action on the part of the GA. Because the raw data shown in the figures is largely zero, equal rewards for positive and negative outputs would tend to produce rules that do nothing.

```

Shift 1
pat 1: 11010 10000 10000 10000 10000 10000 10000 11100 10000 00000 11100 11100 11010 10000 10000
pat 2: 11010 00000 11000 00000 10000 11000 10000 11100 10000 00000 00000 10000 11010 10000 10000
pat 3: 10000 10000 11000 10000 10000 11000 11010 10000 10000 00000 10000 10000 11010 10000 10000
pat 4: 00000 00000 00000 00000 00000 00000 00000 10000 10000 00000 00000 00000 00000 00000 10000
pat 5: 11010 10000 10000 10000 10000 10000 10000 11100 11100 00000 10000 10000 11010 10000 10000
Shift 2
pat 1: 00000 10000 00000 11010 00000 00000 00000 00000 00000 00000 00000 10000 00000 10000 00000
pat 2: 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000
pat 3: 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 11000 00000 00000 00000 00000
pat 4: 00000 10000 11000 10000 00000 11000 00000 00000 00000 00000 11100 00000 11010 10000 00000
pat 5: 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000
Shift 3
pat 1: 00000 00000 00000 00000 00000 00000 00000 00000 00000 10000 00000 00000 00000 00000 00000
pat 2: 00000 00000 00000 00000 00000 00000 00000 00000 00000 10000 00000 00000 00000 00000 00000
pat 3: 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000
pat 4: 00000 00000 00000 00000 00000 00000 00000 00000 00000 10000 00000 00000 00000 00000 00000
pat 5: 00000 00000 00000 00000 00000 00000 00000 00000 00000 10000 00000 00000 00000 00000 00000
Shift 4
pat 1: 00000 10000 00000 10000 00000 00000 00000 11100 11100 00000 00000 11100 11010 10000 00000
pat 2: 00000 10000 00000 10000 00000 00000 00000 11100 10000 00000 00000 10000 11010 10000 00000
pat 3: 00000 10000 00000 11100 00000 00000 00000 11100 11100 00000 00000 11100 11010 11100 00000
pat 4: 00000 10000 00000 10000 00000 00000 00000 11100 11100 00000 00000 10000 11010 10000 00000
pat 5: 00000 10000 00000 11010 00000 00000 00000 11100 11100 00000 00000 10000 11010 10000 00000

```

Figure 6.7 Five bit "active" input data.

The GA was run for 5000 generations with the parameters shown in Table 6.1 which took about four minutes. After the GA evolves, the information is re-injected into the rule. A sample output rule taken from the evolved population is shown in Figure 6.9. Part of the data preprocessing is to remove input data that is non-changing. For example, if aspirin is always prescribed, then this is written to file and removed from the list of things for which the GA is searching. Figure 6.10 shows the data which is removed through the compression process for the example shown here. Per request of the company for which this software was developed, separate rules for each shift of the patient stay are evolved (vs. a general set of rules that would work at any given shift). The data shown in this example is meant to provide the reader with an example of the types of rules that the GA produces. The rule has not been reviewed by competent medical experts for validity.

<u>Single bit active decision input data</u>	<u>Single bit data (input only)</u>
Shift 1	Shift 1
pat 1: 000000000000000000000000000000	pat 1: 11100100101
pat 2: 000000000000000000000000000000	pat 2: 10010011100
pat 3: 000000000000000000000000000000	pat 3: 00011100111
pat 4: 000000000000000000000000000000	pat 4: 10011110010
pat 5: 000000000000000000000000000000	pat 5: 01101101011
Shift 2	Shift 1
pat 1: 0000000000000000000000001111111	pat 1: 01111111000
pat 2: 000000001000000000000000110011	pat 2: 01110000111
pat 3: 000000001000000000000000110011	pat 3: 00100000001
pat 4: 000000000000000000000000110011	pat 4: 11000100101
pat 5: 00000000100000000000000010011	pat 5: 10001011101
Shift 3	Shift 1
pat 1: 000100000000011110000000000001	pat 1: 01100101000
pat 2: 000100000100011010000000000001	pat 2: 01100111101
pat 3: 011100000000011010000000000001	pat 3: 10110110110
pat 4: 000101000000011010000000000001	pat 4: 10000110010
pat 5: 000100000000011110000000000001	pat 5: 11111000101
Shift 4	Shift 1
pat 1: 111011000000011010000000000001	pat 1: 10100100111
pat 2: 100010000000011010000000000001	pat 2: 10010100110
pat 3: 000000000000011010000000000001	pat 3: 00000100101
pat 4: 011000000000011010000000000001	pat 4: 01000001101
pat 5: 011000000000011010010000000001	pat 5: 10011110100

Figure 6.8 Single bit input data.

(ALL patients that meet in \_gene contribute to rule fitness)  
 This rule's fitness: 827  
 Fitness of best rule: 827  
 Patient 0 is best treated by this rule.  
 4 patients treated by this rule, including patient(s):  
 0 1 2 4  
 7 of 10 must be true.  
 If during shift 0: FIVA 0 was abnormal  
 If during shift 0: FIVA 1 was given  
 If during shift 0: FIVA 5 was given  
 If during shift 0: FIVA 5 was abnormal  
 If during shift 0: FIVA 7 was low  
 If during shift 0: FIVA 11 was abnormal  
 If during shift 0: FIVA 12 was abnormal  
 If during shift 0: FIVA 12 was high  
 If during shift 0: FIVA 13 was given  
 If during shift 0: ONEB 5 was given  
 THEN do the following in shift 1:  
     Prescribe oneA(s): 22 23 24 25 26 27 28  
     and order fivA(s): 1 2 5 10 12

Figure 6.9 Sample output data.

```

Shift(0)
oneAs: 0,0 1,0 2,0 3,0 4,0 5,0 6,0 7,0 8,0 9,0 10,0 11,0 12,0 13,0 14,0 15,0 16,0 17,0 18,0 19,0 20,0
21,0 22,0 23,0 24,0 25,0 26,0 27,0 28,0
fivAs: 0,X X 0 X 0 1,X 0 0 0 0 2,X X 0 0 0 3,X X 0 0 0 4,X 0 0 0 0 5,X X 0 0 0 6,X X 0 X 0 7,1 X X 0 0
8,1 X X 0 0 9,0 0 0 0 0 10,X X X 0 0 11,X X X 0 0 12,X X 0 X 0 13,X 0 0 0 0 14,1 0 0 0 0
oneBs:
fivBs:
output: 0,0 1,0 2,0 3,0 4,0 5,0 6,0 7,0 9,0 10,0 11,0 12,0 13,0 14,0 15,0 16,0 17,0 18,0 19,0 20,0 21,0
24,1 27,1 28,1 29,0 33,0 35,0 36,0 37,0 43,0

Shift(1)
oneAs: 0,0 1,0 2,0 3,0 4,0 5,0 6,0 7,0 9,0 10,0 11,0 12,0 13,0 14,0 15,0 16,0 17,0 18,0 19,0 20,0 21,0
0,X X 0 X 0 1,X 0 0 0 0 2,X X 0 0 0 3,X X 0 X 0 4,X 0 0 0 0 5,X X 0 0 0 6,X X 0 X 0 7,X X X 0 0
8,X X X 0 0 9,X X 0 0 0 10,X X X 0 0 11,X X X 0 0 12,X X 0 X 0 13,X 0 0 0 0 14,X 0 0 0 0
oneBs:
fivBs:
output: 0,0 3,1 4,0 6,0 7,0 8,0 10,0 11,0 12,0 13,1 14,1 16,1 17,0 18,0 19,0 20,0 21,0 22,0 23,0 24,0
25,0 26,0 27,0 28,1 29,0 30,0 31,0 32,0 33,0 34,0 35,0 36,0 37,0 39,0 40,0 41,0 42,0 43,0

Shift(2)
oneAs: 0,0 4,0 6,0 7,0 10,0 11,0 12,0 17,0 18,0 19,0 20,0 21,0
0,X X 0 X 0 1,X 0 0 0 0 2,X X 0 0 0 3,X X 0 X 0 4,X 0 0 0 0 5,X X 0 0 0 6,X X 0 X 0 7,X X X 0 0
8,X X X 0 0 9,X X 0 0 0 10,X X X 0 0 11,X X X 0 0 12,X X 0 X 0 13,X 0 0 0 0 14,X 0 0 0 0
oneBs:
fivBs:
output: 3,0 6,0 7,0 8,0 9,0 10,0 11,0 12,0 13,1 14,1 15,0 16,1 17,0 18,0 20,0 21,0 22,0 23,0 24,0 25,0
26,0 27,0 28,1 29,0 30,1 31,0 32,1 33,0 34,0 35,0 36,1 37,1 38,0 39,0 40,1 41,1 42,1 43,0

Shift(3)
oneAs: 6,0 7,0 10,0 11,0 12,0 17,0 18,0 19,0 20,0 21,0
0,0 0 0 0 0 1,X 0 0 0 0 2,X X 0 0 0 3,X X X X 0 4,0 0 0 0 0 5,X X 0 0 0 6,0 0 0 0 0 7,X X X 0 0
8,X X X 0 0 9,X X 0 0 0 10,X X X 0 0 11,X X X 0 0 12,X X 0 X 0 13,X X X 0 0 14,0 0 0 0
oneBs:
fivBs:
output: 3,0 4,0 5,0 7,0 8,0 9,0 10,0 11,0 13,1 14,1 15,0 16,1 17,0 18,0 20,0 21,0 22,0 23,0 24,0 25,0
26,0 27,0 28,1 29,0 30,0 31,0 32,0 33,0 34,0 35,0 36,1 37,1 38,0 39,0 40,0 41,1 42,0 43,0

Shift(4)
oneAs: 6,0 7,0 10,0 11,0 12,0 17,0 18,0 20,0 21,0 28,1
0,0 0 0 0 0 1,X 0 0 0 0 2,0 0 0 0 0 3,X X X X 0 4,0 0 0 0 0 5,0 0 0 0 0 6,0 0 0 0 0 7,X X X 0 0
8,X X X 0 0 9,X 0 0 0 0 10,0 0 0 0 0 11,X X X 0 0 12,X X 0 X 0 13,X X X 0 0 14,0 0 0 0
oneBs:
fivBs:
output: 1,0 2,0 3,0 5,0 7,0 8,0 9,0 10,0 11,0 13,1 14,1 15,0 16,1 17,0 18,0 19,0 20,0 21,0 22,0 23,0
24,0 25,0 26,0 27,0 28,1 29,0 31,0 33,0 34,0 35,0 36,1 37,1 38,0 39,0 41,1 43,0

```

Figure 6.10 Data removed by the compression function.

## 6.9 Conclusion and future research

In test runs, the genetic algorithm has demonstrated its ability to learn from a system's inputs and outputs. The genetic software engine described in this chapter has been integrated into a complete healthcare software package which has been installed at several sites around the country. Preliminary reports from the company for which it was developed indicate that the GA-based software engine finds 70% of the rules that should be contained in a standardized critical path. There are some inherent problems with the software in its present form. The form of the solution that we are forcing the GA to use assumes dependence on only the three shifts prior to the decision to be made. This is an obvious problem. Another problem is that the form of the solution assumes we can convert all the input data points and output decisions to a binary format which is often

cumbersome. A possible solution to this problem might be obtained by using genetic programming [Koza, 1992]. For inferring rules, genetic programming is similar in principle to genetic algorithms, but it allows the number of facts and rules included in a rule to be a variable of the algorithm.

Discussions were held with the company funding the research regarding the use of genetic programming for the next phase of this research. The consensus was to continue the research, but due to problems with the funding, work was delayed. Since that time the company has had some difficult financial problems and is currently being purchased by the Minnesota Mining and Manufacturing (3M) company. Future versions are still under consideration for development, but will most likely be geared toward determining the bidding rules that were used to develop bids contained in a database. Traders who participate in markets where all information is disclosed (such as the Australian deregulated electric market) should be very interested in an enhanced version of this software. Regulating agencies that oversee markets should be interested in this software for detecting collusion, and ultimately for verifying market efficiency.

## **7 CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH**

### **7.1 Commentary**

Ready or not, deregulation around the world is changing the way electrical system is operated. In our new and more competitive environment, we can expect to see the power system moving closer and closer to its limits with profit motivating every operating decision. Now a profit driven industry, we can expect to see vendors developing and marketing new operating tools designed to help the participants eke out every dollar they can. Many of those tools will be borrowed from other industries, and some will be modifications of existing electrical industry tools. This dissertation has reported on several research projects that, directly or indirectly, have significance to the competitive electric generation company or energy trader participating in a competitive environment. The financial tools reported in this dissertation were not meant to provide easy answers to the problems that participants will face. Rather, in much of the work we are attempting to develop methods of studying complex competitive behavior and learning from these behaviors. Other tools (e.g., the unit commitment and fuzzy AGC) are modifications of existing algorithms that needed updating to maintain their usefulness to participants of the competitive environment.

### **7.2 Future market framework and assumptions**

Chapter 2 provided the reader with an introduction to the framework of competitive marketplace we assume. The actual framework of any given market is determined by regulators and may likely differ from our market framework. We also presented many of the assumptions we used for the research reported in later chapters. The basics of genetic algorithms were highlighted for the reader.

In the competitive deregulated electric energy marketplace bidding strategies will be important to profitability. In addition to maximizing profit, traders should consider the risk involved with a particular strategy. Keeping that in mind, we presented an introduction to using options and futures contracts to reduce the risk associated with an energy trader's position in the market.

In the vein of building less risky and more comprehensive bidding strategies, future research should investigate allowing evolving economic agents access to options and futures contracts. The shape of an options position could be defined by four numbers. One needs only specify the type of option (long call, long put, short call, or short put), the strike price, the contract amount, and the premium. The agents which learn to use the options properly will have an advantage in reducing the risk of their strategies. The agent's fitness

will not consist only of short term profit in the spot market, but will depend on the trader's performance over the long term as they participate in an auction simulator which must be extended to include multiple markets.

### 7.3 Evolving bidding strategies

The results provided in the chapter demonstrated that GA and GP-Automata based agents learn to bid in an explicable manner in a multi-participant auction. The fact that GP-Automata suffer very little degradation in performance when they are limited by number of states or by tree size gives some indication of how powerful the method is. Since the GP-Automata lend themselves well to scenarios where there are vast amounts of data available, adding more detail (e.g. available transfer capability information, forecasted prices, unit commitment schedules) to the agent models is an easy suggestion to make. Ideally this larger volume of information would help the bidder in making a bid. As Figure 2.7 shows, traders have to deal in more than one market. Extending the strategies to cover multiple markets is another area of current investigation. In addition to adding the above details, there is a need to perform a more complete sensitivity analysis to see how the various parameters affect performance of the strategies that are constructed. Among these parameters are the parent selection methodology and the population size.

### 7.4 Fuzziness in the competitive environment

Building good bidding strategies for electricity traders as they move into the deregulated marketplace will continue to be important for those companies wishing to remain profitable. Chapter 4 described research performed in this area, and points to directions of current investigation designed to build more robust adaptive bidding strategies. Future research should actually utilize these techniques to implement strategies in an auction simulator. The fuzzy membership functions for inputs like demand and costs can be defined through the use of forecasting methods. Functions describing others bidding behavior can be defined from historical data. The GP-Automata adaptive agent work can be coupled with this research to evolve the fuzzy rules. To provide a more realistic auction situation, a user friendly means of inputting different types of bidding strategies should be developed so that a heterogeneous set of strategies may be pitted against one another.

In the second half of the chapter we saw that fuzzy logic controller lends itself well to the AGC problem. It responds rapidly, and can be tuned to outperform the traditional controller for the cases studied. The rules can be formulated in plain English making it easy for the operator to see how they work, and how to easily modify them.

The FLC AGC described in this chapter used rules with fixed ranges. Future research could make these rule adaptive so that manual re-tuning is not necessary when generator parameters are changed. The electric

utility industry is becoming more competitive and more modernized. The proliferation of microprocessor based control equipment distributed or networked throughout the power system makes using fuzzy logic a very feasible possibility. The introduction of increased competition in the power systems of the world will make it more important than ever to have tight control over the interchange contracts. Another area of future research is to incorporate fuzzy control into a multi-area AGC controller as described in the companion papers by Kumar et al. [1996a, 1996b].

### **7.5 Unit commitment**

The UC-GA has been rewritten for price-based operation to help the GENCO come up with a schedule with which they can operate their generating units in a profitable manner. The GA is a useful tool in searching large discrete solution spaces, and since the space of unit commitment solutions is quite large, the GA is quite appropriate for the UC problem. Ideally, the GENCO would run the UC-GA for the expected prices and demands that they consider most likely. Since these prices and demands may be uncertain it may be necessary to run several cases would allow the user to know how sensitive the schedules are to variations in the inputs. Future research should augment the algorithm by implementing the Monte Carlo sampling in the fitness function discussed in the chapter. Other methods of looking at uncertain prices (such as ways of inputting a deterministic price versus demand curve at each hour) should be investigated. The UC-GA formulation described here is presently being enhanced to provide the user with additional information that identifies which schedules allow the user more market flexibility for a given level of profit. Future research should modify the code to implement the changes described above which would akin to building in an on-line sensitivity analyzer.

### **7.6 Inferring rules by intelligently mining a database**

In test runs, the genetic algorithm demonstrated its ability to learn from a system's inputs and outputs. As mentioned in the chapter, the genetic software engine described in this chapter was designed to be part of a complete healthcare software package. It was successfully installed at several sites around the country. However, there are some inherent problems with the software in its present form. The form of the solution that we are forcing the GA to use assumes dependence on only the three shifts prior to the decision to be made. This is an obvious problem. Another problem is that the form of the solution assumes we can convert all the input data points and output decisions to a binary format that is often cumbersome. A possible solution and suggestion for future research is to use genetic programming [Koza, 1992], or GP-Automata. For inferring rules, genetic programming is similar in principle to genetic algorithms, but it allows the number of facts and rules included in a rule to be a variable of the algorithm.

As mentioned in the chapter, discussions were held with the company funding the research regarding additional research, but due to difficult financial problems it is unlikely that this will happen soon. Future research should be geared toward tuning the method for inferring bidding rules from a database. Traders who participate in markets where all information is disclosed (such as the Australian deregulated electric market) should be very interested in an enhanced version of this software. Regulating agencies that oversee markets should be interested in this software for detecting collusion, and ultimately for verifying market efficiency. Therefore, future research should focus on the trading arena, and should use either GP-Automata or GP making the data structures used in evolution more generic, and less likely to be industry specific.

## **7.7 Additional suggestions for future research**

### ***7.7.1 Evolving multi-market electric commodity portfolios and bidding strategies***

This research involves extending the auction simulator to allow agents to participate in multiple markets. Fitness will not consist only of short-term profit in the spot market, but will depend on the trader's performance over the long term. Among the possible ideas for developing good strategies is to allow a GA to determine an options position. An evolving data structure of four numbers could specify the type of option (long call, long put, short call, or short put), the strike price, the contract amount, and the premium. As mentioned above, the agents which learn to use the options properly will have an advantage in reducing the risk of their strategies.

### ***7.7.2 Forecasting electric prices and demand with uncertainty***

Given historical data on electricity, we would like to evolve GP-Automata to do forecasting. Electricity prices and demand may fluctuate based on the weather, time of the day, time of the week, time of the year, or for various other reasons. These forecasts may depend on input data from various time periods. The GP-Automata will be given the task of selecting which inputs are important in predicting price or demand. Given a matrix of data, the GP-Automata evolves footprints (in a manner similar to texture classification work [Ashlock and Davidson, 1997]) that indicate which data are important for predicting the price. The inputs identified by the foot patterns are then used in a second process where a GP or GP-Automata learns the regression coefficients to produce the smallest mean squared error in the predictions.

## REFERENCES

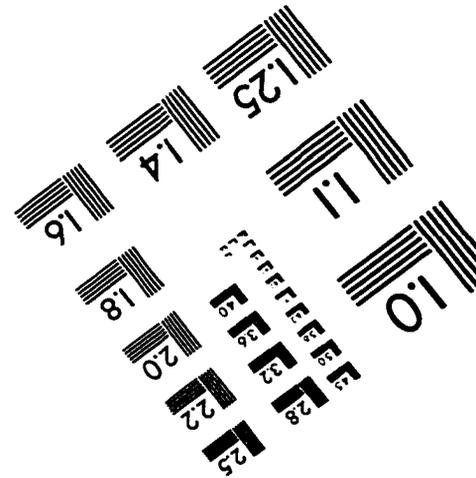
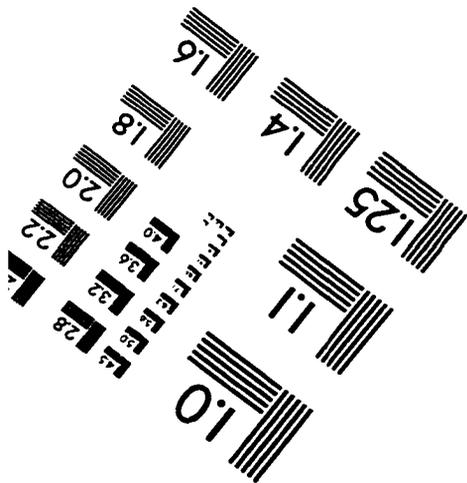
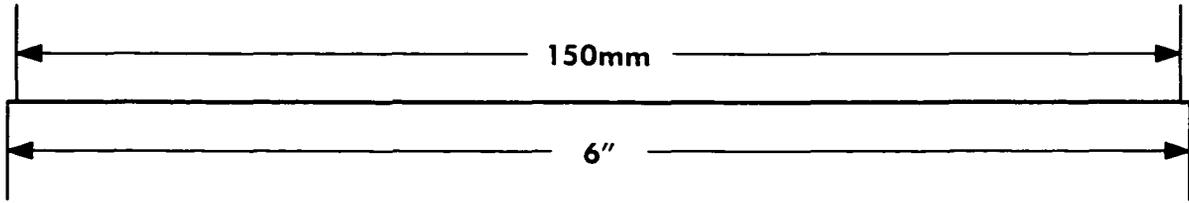
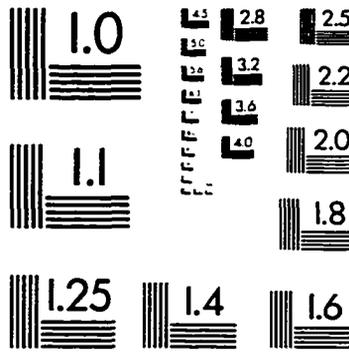
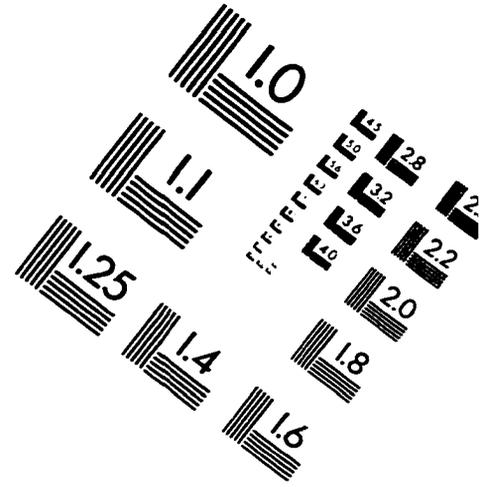
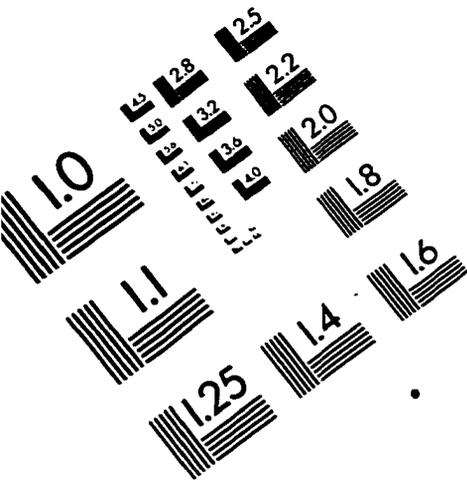
- M. Andrews and R. Prager, "Genetic programming for the acquisition of double auction market strategies." in *Advances in Genetic Programming*, Edited by: K. Kinnear, Jr., Cambridge, MA: The MIT Press, 1994.
- D. Ashlock and J. Davidson. "Genetic algorithms for automatic texture classification." *Proceedings of the 1997 SPIE International Meeting*, 1997.
- D. Ashlock and C. Richter, "The Effect of Splitting Populations on Bidding Strategies." *GP-97 Proceedings of the Second Annual Conference*, pp. 27-34. San Francisco, CA: Morgan Kaufmann Publishers, 1997.
- D. Ashlock. "GP-automata for Dividing the Dollar." Mathematics Department, Iowa State University, Ames, IA 1995.
- B. Barkovich and D. Hawk, "Charting a new course in California." *IEEE Spectrum*, vol. 33, p. 26, New York NY: IEEE, July 1996.
- D. Barry. "Socket to Them." in *Dave Barry's Bad Habits*. New York, NY: Henry Holt and Company, Inc., 1985.
- B.R. Binger and E. Hoffman. *Microeconomics with Calculus*. New York, NY: Harper Collins, 1988.
- R. E. Clayton and Rana Mukerji. "System Planning Tools for the Competitive Market." *IEEE Computer Applications in Power*, p. 50, July 1996.
- R. Dawkins. *The Blind Watchmaker*. New York, NY: W.W. Norton and Company, 1996.
- G. Fahd and G. Sheblé. "Optimal power flow emulation of interchange brokerage systems using linear programming." *IEEE Transactions on Power Systems*, Vol. 7, No. 2, pp. 497-504, 1992.
- D. Finlay, *Optimal Bidding Strategies in Competitive Electric Power Pools*. Masters thesis. University of Illinois. Urbana-Champaign, IL, 1995.
- A. J. Germond and D. Niebur. "Survey of Knowledge-Based Systems in Power Systems: Europe". *Proceedings of IEEE*, Vol. 80, No. 5, pp. 732-744, May 1992.
- W. Gilchrist, *Statistical Forecasting*. New York, NY: John Wiley and Sons, 1976.
- D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1989.
- S. Goonatilake, J. A. Campbell, and N. Ahmad. "Genetic-Fuzzy Systems for Financial Decision Making." *Lecture Notes in Computer Science*, vol. 1011, pp. 202-223. New York: Springer-Verlag, 1995.
- S. Hakin, *Neural Networks: A Comprehensive Foundation*. Englewood Cliffs, NJ: Macmillan Publishing Company, 1994.
- H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press, 1975.

- M. Huneault, C. Rosu, R. Manoliu, and F. D. Galiana, "A Study of Knowledge Engineering Tools in Power Engineering Applications", *IEEE Transactions on Power Systems*, Vol. 9. No. 4. pp. 1825-1832, November 1994.
- C. S. Indulkar and B. Raj, "Application of Fuzzy Controller to Automatic Generation Control." Department of Electrical Engineering, IIT-Delhi, New Delhi, India, 1995.
- Y. Ioannides, "Evolution of Trading Structures." Department of Economics, Tufts University, Medford, MA, 1995.
- S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, "A Genetic Algorithm Solution to the Unit Commitment Problem." 1995 IEEE/PES Winter Meeting 152-9 PWRs, New York: IEEE, 1995.
- S. Kondragunta, Genetic algorithm unit commitment program. M.S. Thesis, Iowa State University, Ames, IA, 1997.
- B. Kosko. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs, NJ: Prentice Hall, 1992.
- J. Koza. *Genetic Programming*, Cambridge, MA: The MIT Press, 1992.
- J. Koza. *Genetic Programming II*. Cambridge, MA: The MIT Press, 1994.
- E. Kristjánsson, D. Barak, K. Plummer, M. Jamshidi, N. Vadiiee, and T. J. Ross. *Fuzzy Logic and Control*. Englewood Cliffs, NJ: Prentice Hall, 1993.
- J. Kumar and G. Sheblé. "Framework for energy brokerage system with reserve margin and transmission losses." 1996 IEEE/PES Winter Meeting, 96 WM 190-9 PWRs, New York: IEEE, 1996a.
- J. Kumar and G. Sheblé. "Auction Game in Electric Power Market Place." *Proceedings of the 58th American Power Conference*, pp. 356-364, 1996b.
- J. Kumar, K. Ng, and G. B. Sheblé. "AGC Simulator For Price Based Operation, Part I: A Model." presented at IEEE PES Summer Meeting, 96 SM 588-4 PWRs, 1996a.
- J. Kumar, K. Ng, and G. B. Sheblé. "AGC Simulator For Price Based Operation, Part II: Case Study Results." presented at IEEE PES Summer Meeting, 96 SM 373-1 PWRs, 1996b.
- B. LeBaron. "Experiments in Evolutionary Finance." University of Wisconsin -Madison, Madison, WI, 1995.
- T. Maifeld and G. Sheblé, "Genetic-Based unit commitment." *IEEE Transactions on Power Systems*, Vol. 11, No. 3, p. 1359, August 1996.
- J. M. Mann. *Neural Networks: Genetic-Based Learning, Network Architecture, and Applications to Nondestructive Evaluation*, M.S. Thesis, Iowa State University, Ames, IA, 1990.
- Y. M. Mansur. *Fuzzy Sets and Economics*. Burlington, VT: Edward Elger Publishing Limited, 1995.
- J. Martin, and S. Oxman, *Building Expert Systems*, Englewood Cliffs, NJ: Prentice Hall, 1988.
- P. Milgrom, "Auctions and bidding: A Primer." *Journal of Economic Perspectives*, vol. 3, no. 3., pp. 3-22, 1989.

- A. Papalexopoulos, S. Hao, and T. Peng. "An implementation of a neural network based load forecasting model for the ems," Presented at the 1994 IEEE/PES Winter Meeting, 94 WM 209-7 PWRS. New York: IEEE, 1994.
- D. Post. *Electric power interchange transaction analysis and selection*. Master's thesis, Iowa State University, Ames, IA, 1994.
- S. Rajan. *Electric system operating strategies in an energy brokerage environment*. Ph.D. dissertation, Iowa State University, Ames, IA, 1997.
- D. Reichelt and H. Glavitsch. "Features of a Hybrid Expert System for Security Enhancement." *Proceedings of the PICA Conference*., pp. 330-336, May 1991.
- C. Richter and G. Sheblé. "Genetic algorithm evolution of utility bidding strategies for the competitive marketplace," 1997 IEEE/PES Summer Meeting, PE-752-PWRS-1-05-1997 New York: IEEE, 1997a.
- C. Richter and G. Sheblé. "Building fuzzy bidding strategies for the competitive generator," in *Proceedings of the 1997 North American Power Symposium*, 1997b.
- C. Richter and G. Sheblé. "Bidding strategies that minimize risk with options and futures contracts." in *Proceedings of the 1998 American Power Conference, session 25, Open Access II-Power Marketing, paper C*, 1998.
- C. W. Richter, Jr., T. T. Maifeld, and G. B. Sheblé. "Genetic Algorithm Development of a Healthcare Expert System." *Proceedings of the 4th Annual Midwest Electro-Technology Conference*, pp. 35-38. Ames, IA, 1995.
- C. Richter, D. Ashlock, and G. Sheblé. "Effects of tree size and state number on GP-Automata bidding strategies." *Proceedings of the GP-98 genetic programming conference*, Denver, CO: Morgan Kaufmann, 1998.
- C. Richter, G. Sheblé, and D. Ashlock. "Comprehensive Bidding Strategies with Genetic Programming/Finite State Automata." Accepted for publication in the *IEEE Transactions on Power Systems*, 1999.
- C. Richter. *Developing bidding strategies for electric utilities in a competitive environment*. Master's thesis, Iowa State University, Ames, IA, 1996.
- G. Sheblé and G. Fahd, "Unit commitment literature synopsis." *IEEE Transactions on Power Systems*, Vol. 9, No. 1, pp 128-135, February 1994.
- G. Sheblé and J. McCalley, "Discrete auction systems for power system management." Paper presented at the 1994 National Science Foundation Workshop, Pullman, Washington, 1994.
- G. Sheblé, M. Ilic, B. Wollenberg, and F. Wu. Lecture notes from: *Engineering strategies for open access transmission systems*, A two-day short course presented in San Francisco, California, December, 1996.
- G. Sheblé, "Unit Commitment for Operations," Ph.D. Dissertation, Virginia Polytechnic Institute and State University, March, 1985.

- G. Sheblé, "Simulation of discrete auction systems for power system risk management." Paper presented at the 1994 Frontiers of Power, OK, 1994a.
- G. Sheblé, "Electric energy in a fully evolved marketplace." Paper presented at the 1994 North American Power Symposium, Kansas State University, KS, 1994b.
- G. B. Sheblé, "Automatic Generation Control Standard Functional Specification." notes from Fall Semester Steady State Power Systems class, Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, 1995.
- G. Sheblé, "Priced based operation in an auction market structure." Paper presented at the 1996 IEEE/PES Winter Meeting, Baltimore, MD, 1996.
- K. Shimojima, Y. Hasegawa and T. Fukuda. "Unsupervised/Supervised Learning for RBF-Fuzzy System." *Lecture Notes in Computer Science*, vol. 1011, pp 127-147. New York, NY: Springer-Verlag, 1995.
- B. Soucek, *Dynamic, Genetic, and Chaotic Programming*, New York, NY: John Wiley and Sons, Inc., 1992.
- S. Takriti, B. Krasenbrink, and L. S.-Y. Wu. "Incorporating Fuel Constraints and Electricity Spot Prices into the Stochastic Unit Commitment Problem," IBM Research Report: RC 21066. Mathematical Sciences Department, T.J. Watson Research Center, Yorktown Heights, New York, December 29, 1997.
- L. Tesfatsion. "A Trade Network Game with Endogenous Partner Selection." *Economic Report Series*, Department of Economics, Iowa State University, Ames, IA, 1995.
- G. Thompson and S. Thore. *Computational Economics: Economic Modeling with Optimization Software*. San Francisco, CA: Scientific Press, 1992.
- A. Vojdani, C. Imparato, N. Saini, B. Wollenberg, and H. Happ. "Transmission access issues." Paper presented at the 1995 IEEE/PES Winter Meeting, 95 WM 121-4 PWRs, New York: IEEE, 1994.
- D. C. Walters and G. B. Sheblé. "Genetic Algorithm Solution of Economic Dispatch with Valve Point Loading." 1992 IEEE/PES Summer Meeting 414-3. New York: IEEE, 1992.
- A. Wood and B. Wollenberg, *Power Generation, Operation, and Control*. New York, NY: John Wiley and Sons, 1996.
- R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*. New York, NY: John Wiley and Sons, 1994.

# IMAGE EVALUATION TEST TARGET (QA-3)



**APPLIED IMAGE . Inc**  
 1653 East Main Street  
 Rochester, NY 14609 USA  
 Phone: 716/482-0300  
 Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved