

ISECON'02

Heuristic Programming for Business Solutions

Professor Kurt F. Lauckner

Phone: 734-487-2140

Department of Computer Science

Messages: 734-487-1063

Eastern Michigan University

Fax: 734-747-7464

Ypsilanti, MI 48103

Email: csc_lauckner@online.emich.edu

Intended Track: Leading Edge and Emerging Technologies

Author: Kurt F. Lauckner received his Ph.D. in Physics from the University of Michigan in 1968. His experience with computers started with extensive computer computations for his thesis work in theoretical physics. While a member of the Mathematics Department he took the major role in creating the Computer Science program at Eastern Michigan University. During the past twenty years he has developed a concepts approach to teaching computer literacy. The computer literacy course COSC 136-Computers for the Non-specialist has grown into the University's primary computer literacy course and currently enrolls approximately 2500 students per year. It was used as the model for the University wide computer literacy requirement in the EMU basic studies program. Many materials were developed for this freshman level computer literacy course. They include the textbooks Computers: Inside & Out (5 editions) and the Computer Continuum in its 2nd edition with Prentice Hall.

He is currently working in the area of heuristic programming or "soft computing," with most efforts being spent in evolutionary paradigms.

Heuristic Programming in Business Solutions

Abstract: Heuristic programming often called soft computing consists of several paradigms, which include: expert systems, fuzzy logic, neural networks, genetic algorithms, genetic programming and other evolutionary systems. In spite of the many successes they are not all fully accepted in solving business problems. It is important for problem solvers to be aware of these somewhat unorthodox approaches to problem solving and be prepared to use them. This paper gives an overview of the several soft computing paradigms and some examples of successful use in solving business problems.

1 Introduction

Heuristic programming is often referred to as soft computing and includes the areas of expert systems, fuzzy logic, neural networks, genetic algorithms, genetic programming and other evolutionary systems. It has a rich history beginning with such notable figures as John von Neumann and Alan Turing. Soft computing seemed to come out of the field of artificial intelligence, but as AI matured those following the soft computing were not readily accepted into the fold of the mainstream. In fact, a rather bleak period beginning in the late 1960s and continuing well into the 1980s created an research atmosphere that kept the field from a normal growth. It began with a series of events that threw a dark cloud over the various areas in the soft computing field, especially fuzzy logic and neural networks. One of the first of these events was the 1969 book *Perceptrons* by Marvin Minsky and Seymour Papert. It discouraged research in the area of neural networks for well over a decade. In spite of some major successes, funding in the soft computing

area is considered by some to be less than acceptable. John Koza, a leading genetic programming researcher stated in an open letter to the Genetic Programming mail-list:

"...it is apparent to me that there is a strong anti-evolution cast to almost everyone and everything in artificial intelligence, machine learning, and computer science in general." (July 2000).

Part of the responsibility of those working in the field of heuristic programming is to educate other computer professionals regarding the benefits and promise of the various areas in soft computing.

The intention of this paper is to give an overview of the major aspects of the several specialty areas in the field and to give some examples of successful applications in business.

2 Expert Systems

In 1965 Ed Feigenbaum and others at Stanford University created a program called DENDRAL that was used to analyze the mass spectrum of organic molecules. It consisted of a collection of rules and was the beginning of what ultimately became the expert system or rule based system. It consisted of a set of rules with the form

IF (condition) THEN (action)

To create an expert system the expertise of an expert in the desired field is put into a collection of these rules. For example, in a loaning institution the following could be an experts opinion:

IF (interest rate increases by 1%) THEN (increase loans by 10%)

After carefully creating a number of these rules by "picking an experts" brain a program referred to as an inference engine was used to examine all of the rules *conditions* to find those that were satisfied. The engine then used the *action*, *consequent*, or THEN part of the rule to look for other rules that had the particular *action* as a *condition*. As each rule is used, it is said to have *fired*. The process was continued until no other rules had *conditions* which matched the current *consequents*.

This is referred to as *forward chaining*. Expert system inference engines often have situation where the result is known, but it is desirable find out the path through the rules that led to that result. This process is referred to as *backward chaining*.

One of the largest and most successful expert systems was Digital Equipment Corporation's XCON. Using several thousand rules it could give the configuration for putting together a custom VAX computer minicomputer almost error free and 300 times faster than the human engineers. The program saved DEC tens of millions of dollars per year.

Another more recent business application of an expert system is the business strategy expert system. Business Resource Software, Inc. (BRS) has a series of expert systems products that it claims are a Knowledge Base of Business Experts. Their clientele reads like a who's who in the corporate world and includes notables such as: American Express, AT&T, Boeing, Century 21, Dun & Bradstreet, E.I.DuPont, Eastman Kodak, Ernst & Young, Federal Express, Harvard University, IBM Corp., J.C.Penny, Kinney Shoe, McDonalds, Motorola, NEC, New York Life, Olympus, Pfizer, Rand McNally Roadway Express, Shell Oil, Texas Instruments, Toyota, Wells Fargo Bank, Xerox.

One of BRS's products is called *Plan Write for Marketing* and draws on experts in the field to assist in creating a marketing plan. Quoting from their sales literature: "Marketing strategy evaluated by expert system software for risks, weaknesses and roadblocks." And it uses "Michael Porter Five Forces Model and 40 other experts." With guidance from the expert system program you identify the market, create a document strategy, prepare a budget, produce a marketing plan document and finally execute the plan. There is no shortage of testimonials, such as the one from Richard Wiser, VP Financial Planning, Mary Kay Cosmetics: " Using **Business Insight** is like having Michael Porter on call. It is like a refresher course for my MBA education."

There are five parts to the planning and documentation process as stated in their literature:

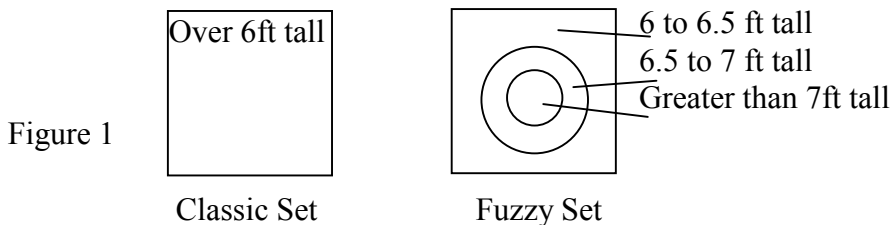
- Text** that describes your philosophy, assumptions and strategies
- Financial** projections that address marketing & sales operational expenses, revenue sources & amounts and (hopefully) profits
- Charts** to graphically portray key aspects of your plan
- Milestones** for tracking and measuring progress towards the plan objectives
- Marketing Plan** to be organized, edited and printed

At each stage the expert system rules guide the user to a completion of the task at hand.

3 Fuzzy Logic

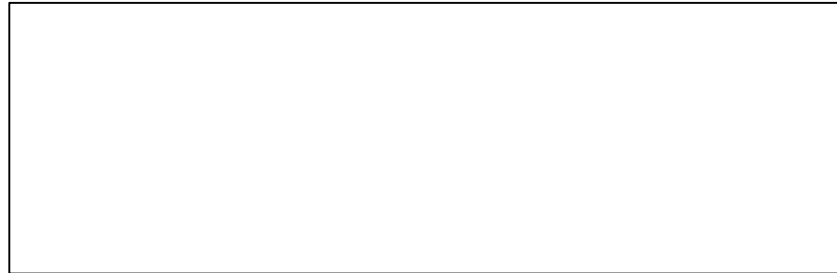
The classic expert system is based on Aristotelian or crisp logic. In other words, the results are either true or false. The concept of approximate reasoning or fuzzy logic was studied by Bertrand Russell and other mathematicians, but the application of fuzzy logic to practical problems was championed by Lotfi Zadeh. His work was not well received in the United States until the 1990s. In fact, the 1980s are considered to be the dark age of fuzzy logic. The Japanese, however, did recognize the potential of fuzzy logic and began using it in cameras, microwave ovens and a large number of appliance control systems. It is now considered respectable and important in US engineering circles.

Fuzzy logic consists of fuzzy sets and approximate reasoning. It lends itself very nicely to non-linear systems. Unlike a classic set where an object is either in the set or not in the set, the fuzzy set has a degree of membership. There is no sharp transition from membership to non-membership. Figure 1 shows a classic set and fuzzy set that refers to height of people.



In the classic set someone is either over 6 feet tall and a member of the set, or less than 6 feet tall and therefore not a member of the set. In the fuzzy set it is possible to differentiate degrees of tallness. Making this gradation continuous and labeling regions of membership with variables called linguistic variables it is possible to use them to create regions of application. For example, in Figure 2, height is described by linguistic variables that identify the fuzzy membership functions.

Figure 2



The vertical axis represents the degree of membership in a certain fuzzy set. From Figure 2, a 7 foot tall person would be 100% *Very Tall*, a 6.5 foot person would be about 45% *Very Tall* but only about 25% *Tall*. Two or more of these fuzzy membership function definitions can be combined with the rules regarding the system being described in a process called defuzzification. The rules referred to are created by an expert in the field of application. An simple example of a rule could be that *if a person is tall then they can be assigned to warehouse restocking*. The major use of fuzzy logic is in control systems such as automobile ABS brakes. However, there are some very interesting business applications.

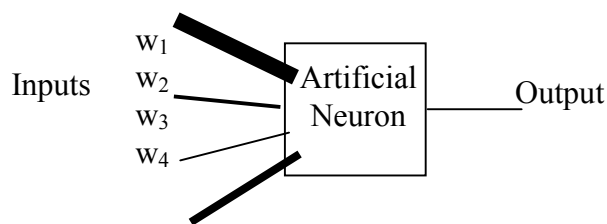
One unique business application of fuzzy logic is in assessing risk management of fraud. This application of fuzzy logic to risk management done by A. Deshmukh and J. Romine made use of a spreadsheet and used rules that were gleaned from auditing standards and a list of red flags on business and economic factors created by several major accounting firms. These red flags include such things as "management operations and financial decisions dominated by a single person, management's unduly aggressive attitude toward financial reporting, and management's poor

reputation in the business community." The authors summarized the benefits and costs to a fuzzy system approach to the red flag implementation:

There are a number of benefits to this approach: (1) the spreadsheet is relatively inexpensive, (2) the training and usage costs are relatively low when compared to expert systems and this approach can be used by small practitioners, (3) the spreadsheet can be used to capture impressions, beliefs, and the intuition of the auditors in a variety of settings, for example, it can be used to capture measurements of red flags associated with employee fraud, (4) trying to quantify red flags using fuzzy numbers (instead of a mere yes or no) is a worthwhile exercise in itself and sheds light on the interdependencies and interactions of red flags, and (5) it aids in structured decision making. However, there are costs associated with this approach: (1) a basic understanding of fuzzy numbers and fuzzy sets is required, (2) quantifying red flags using fuzzy numbers requires more time than the traditional checklists, and finally (3) the spreadsheet output is only as good as the input; the spreadsheet only helps in capturing measurements and mathematically manipulating these measurements. The spreadsheet does not supplant but supplements the auditor's judgment.

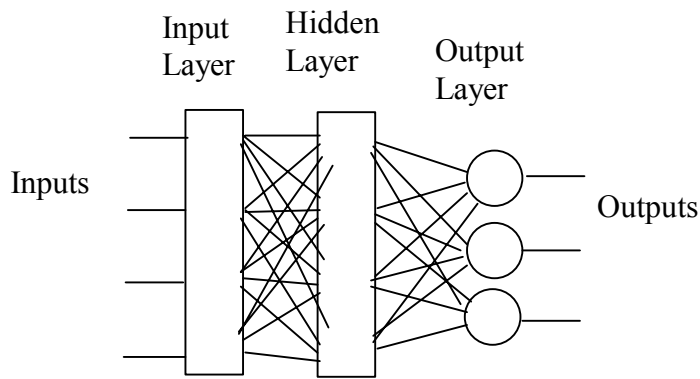
4 Back Propagation Neural Networks

The back-propagation neural network was inspired by the human brain. Modeled loosely on the human brain's neurons, they are the most commonly used neural networks. In Figure 3 is a model of an artificial neuron. It has inputs that are weighted (e.g., w_j) and often are just summed to give an input value to a function, which in turn outputs a value on the output side of the artificial neuron.



By ganging together several neurons into layers a neural network is formed. Figure 4 illustrates a very common 3 layer network, which has an input layer, hidden layer and an output layer.

Figure 4



Neural networks must be trained, this means that the weights have to be adjusted to correspond to the application being developed. A training set of data usually consists of several hundred example cases where both inputs and outputs are known. The training process starts by applying the first data point. Initially the weights are chosen randomly and the back propagation technique involves adjusting these weights so that the output value errors are minimized using a least squares method. Another data point is applied and the same process is done. This will destroy some of the original data point's weight adjustments. But, by doing this process hundreds or even thousands of times repeating the same data points over and over again will ultimately result in weights that represent the information of those data points. The network can then be used to predict the outputs for other given inputs whose output values are unknown. Neural networks have been very successfully used in business in anything from mortgage lending decisions to risk management.

An interesting case study is the use of neural networks in real estate appraisal. Traditionally a certified expert appraiser will use the sale price and some computer assistance to appraise a home. The major difficulties with this is the inconsistency between appraisers, changing market conditions and computer software that is not capable of going beyond rules and mathematical formulas. Neural networks use current information assuming they are continually trained with the current market conditions. Properly managed neural network real estate appraisal programs regularly predict actual sale prices of properties with 90% accuracy as is done by Richard Borst, a Senior Vice President at Day & Zimmerman, Inc., a major provider of appraisal services to state and local

governments. Most of his work was done in the New York area and incorporates eighteen data elements shown in Figure 9.

	Name	Description	Range
Figure 9	SALEPRIC	Actual sales price of home	\$103,000-250,000
	DWLUN	Number of dwelling units	1-3
	RDOS	Reverse (months since sale)	0-23
	YRBLT	Year built	1850-1986
	TOTFIXT	Number of plumbing fixtures	5-17
	HEATING	Heating system type	coded as A or B
	WBFPSTKS	Wood burning fireplace stacks	0-1
	BMNTGAR	Basement garage	0-2
	ATTFRGAR	Attached frame garage area	0-228
	TOTLIVAR	Total living area	714-4185
	DECK/OFP	Deck / open porch area	0-738
	ENCLPOR	Enclosed porch area	0-452
	NBHDGRP	Neighborhood group	coded as A or B
	RECREOM	Recreation room area	0-672
	FINBSMT	Finished basement area	0-810
	GRADE%	Grade factors	0.85-1.08
	CDU	Condition / desirability / useful	3-5
	TOTOBY	Total other value (bldg & yard)	0-16400

Mr. Borst's neural network was developed using the Brainmaker Neural Network software and data from 217 sales records with prices ranging from \$103,00 to \$282,000.

5 Self Organizing Neural Networks

The second most popular neural network model is the self-organizing or Kohonen network. In the simplest form of this scheme there is an input layer and an output layer. Figure 5 shows a two layer self-organizing network.

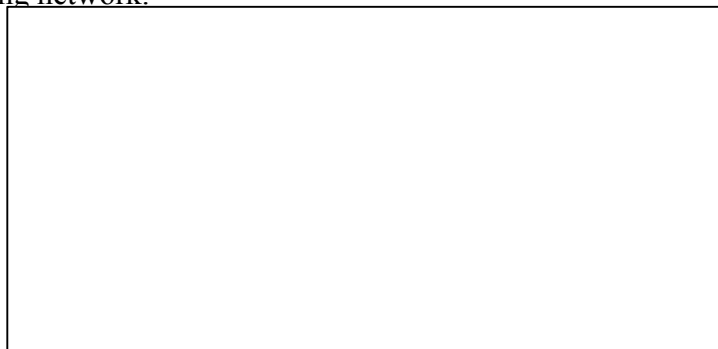


Figure 5

To give the basic idea on how it works an example will be most efficient. In a most trivial case suppose we have a two dimensional surface with a circle drawn within a square. Given a few dozen points (x,y) on the edges of both the circle and square, we feed them into the Kohonen network one at a time. The node with an (x,y) output value that is a minimum least square distance from that point is called the winner. The weights of the network are modified to further make this value even closer. If this is done several hundreds of times, then the network nodes will cluster around the edges of both the square and the circle. Note we are using a visual crutch by letting the nodes position indicate its actual output value (x,y) . This is easy to do in two or even three dimensions. But suppose your data has 30 input values, which may or may not be independent of each other. In fact, it may only be possible to represent it in a 30 dimensional space. Using a Kohonen network the data will cluster around points that minimize the least squares distance for the cluster. From this clustering it is possible to separate the data points into classes and possibly discover some relationships.

One of the most challenging problems in today's vast document collections is making some sense of the mass of knowledge by categorizing and organizing. It wasn't too long ago that libraries were able to individually catalog books and documents as they came into the collection. The Dewey decimal cataloging was simple, and finding books in subject areas such as astronomy or economics was a simple task. Now with the millions of electronic documents it is almost impossible to organize things in this way. In fact new subject areas are created daily. A grand attempt using Kohonen self organizing neural networks to automatically organizing vast collections of documents is being done by none other than Teuvo Kohonen himself and his team of researchers. One reported experiment was to organize some 6,840,568 patent abstracts. "As feature vectors we

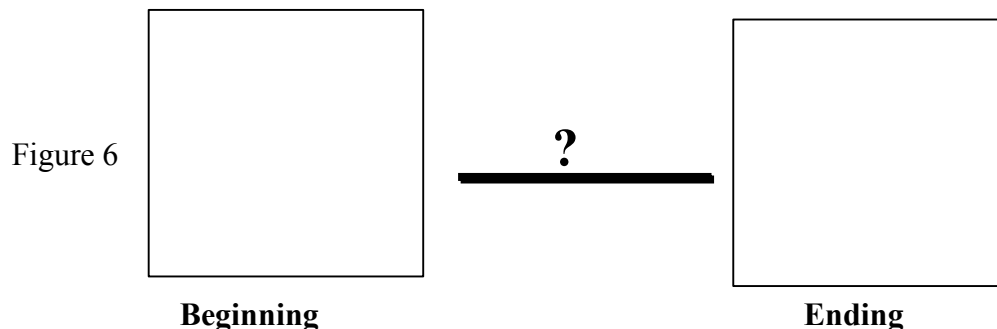
used 500-dimensional vectors of stochastic figures obtained as random projections of weighted word histograms." There were 1,002,240 nodes in the Kohonen network. The discovery of related categories and the scaling up to larger collections is ongoing research.

6 Genetic Algorithms

The last two areas to be examined make use of one of the most powerful, least understood, and least accepted scientific theories: evolution. Evolutionary techniques are having great success in the business arena, but for cultural reasons the general acceptance of the theory is low. The power of evolution in computing has been demonstrated in many ways. One of the first paradigms to have practical application is John Holland's concept of the genetic algorithm.

The essential idea is to create a bit string representation that is capable of describing a solution to the problem being solved. With this representation a solution is evolved by using the classic concepts from evolution, such as mutation, fitness and crossover. This may not result in the best solution, but it will be a solution. An example will help specify a general overview of the details.

Suppose we would like to solve the 8-puzzle as shown in Figure 6. Starting from the beginning configuration and moving one square at a time into the empty square it is desired to put the tiles in order as shown in the end configuration.



In playing the 8-puzzle one possible representation of a solution could be a string of bits taken two at a time, which represents the movement of the empty square. Figure 7

shows a bit string that indicates that the empty square should move down, then left, then up and then right.

00 -- up 10 -- left
 01 -- right 11 -- down

Figure 7 **11 10 00 01**

An arbitrary string length of maybe 50 is chosen, which would result in 25 individual movements of the empty square. The general scheme would start with an arbitrary number of say 500 randomly generated strings. Each string is evaluated by what is termed the fitness function. This fitness function could be in many forms, one possibility is to score each square by how close it is to the final desired solution. If a larger score means the tile is closer to its final position, then adding up all of the individual tile scores it would seem reasonable that the highest scoring string would be closest to the final solution.

After evaluating all 500 strings we might make copies of the top 200 scoring strings and then perform 100 swaps between portions of pairs of the 200 copies. This is referred to as crossover. The two strings before and after the crossover is shown in Figure 8 where the crossover point is chosen at random.

	Old String 55	110101011100110101010001101001010		1101010100110101
	Before crossover			
Figure 8	Old String 56	01011110100010101110101010101111100101010110101010		
	New String 55	110101011100110101010001101001011		00101010110101010
	After crossover			
	New String 56	010111101000101011101010101011110		11101010100110101

Crossover Point

The next generation population will consist of the 200 original strings, the 200 crossed over strings, but to maintain a population of 500 strings we need an additional 100 strings. These strings may be a random selection of 100 strings from the 300 low scoring strings not used, and may or may not be crossed over. Finally, a random single bit change is done to about 3% to 5% of the strings. All of this may seem rather arbitrary, and it is arbitrary. However, through experience researchers in the genetic algorithms will use intuition and creativity to make these decisions. There is no proof of a best way.

This process is repeated over and over again until some maximum number of generations is reached or maybe a solution is found. This particular problem isn't as clean as it could be, because we might reach a solution half way through the string moves, should the system be designed to stop at this point? One might even identify the extra moves of a string that are not necessary as akin to junk DNA in human genes. The information is there, but not used anymore.

It should be noted that this is the first of the heuristic paradigms that can go beyond what we teach it. In other words, it could come up with solutions we have never seen. This isn't especially notable in the above example, but can be very important in more complex examples.

Genetic algorithms have been used to improve models for tactical asset allocation and international equity strategies. In fact, reported improvements of 82% in cumulative portfolio value as compared with a passive benchmark model. This seems to indicate genetic algorithms are quite suitable for financial modeling applications. There are three reasons for this:

- "1. They are payoff driven. Payoffs can be improvements in predictive power or returns over a benchmark. There is an excellent match between the tool and the problems addressed.
2. They are inherently quantitative, and well-suited to parameter optimisation (unlike most symbolic machine learning techniques).

3. They are robust, allowing a wide variety of extensions and constraints that cannot be accommodated in traditional methods."

7 Genetic Programming

Genetic programming uses the same evolutionary concepts as the genetic algorithm, which include fitness, crossover and mutation. However, instead of creating bit string representations, programs are generated which might result in solutions to a problem. Genetic programming may be more general in the sense that less assumptions are made in the choice of representation. As long as the basic building blocks (i.e., instructions) are capable of creating a solution, then the problem is not restricted to subsets of possible solutions. A rather common approach to creating programs that are obtained through evolution is through the use of tree structures. LISP is the most obvious choice for this, and it indeed seems to be a common choice. Tree structures that easily translate to programs are not only easy to construct, but the operations of crossover and mutation are also relatively easy to perform.

A particular application of the GP paradigm is a tool called STROGANOFF, which was developed in the early 1990s. It is used for system identification such as learning the basic structure of financial time series that lead to well forecasting models. The STROGANOFF tool was used in predicting USD/SF currency exchange rates, movements of the Dow Jones industrial index and the Nikkei225 index of the Tokyo Stock Exchange.

Future application of the STROGANOFF tools as stated by the company are

- derivative markets*-- including: analysis and prediction of option prices, risk estimation, foreign currency exchange rates, etc.;
- banking and finance*-- financial risk control; identification of economy factors; fraud detection, bankruptcy prediction, financial scheduling;
- marketing*-- modeling artificial stock markets; forecasting customer demands; pricing of financial derivatives (option pricing).

8 References

- Deshmukh, Ashutosh and Jeff Romine. "Assessing the Risk of Management Fraud Using Red Flags: A Fuzzy Number Based Spreadsheet Approach. *Journal of Accounting and Computers*. http://www.swcollege.com/acct/jac/jac12/jac12_article1.html
- Von Altrock, Constantin. *Fuzzy Logic and Neurofuzzy Applications in Business and Finance* Prentice-Hall ESH Professional, 1997.
- Borst, R. A.. *Artificial Neural Networks: The Next Modeling / Calibration Technology for the Assessment Community?*, *Property Tax Journal* (International Association of Assessing Officers), 10(1):69-94, 1991.
- Kohonen, T., S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela.. *Self Organization of a Massive Document Collection*. *IEEE Transactions on Neural Networks*, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, volume 11, number 3, pages 574-585. May 2000.
- Deschaine, Larry and Jennifer McCormack, Frank Francone, and Dorian Pyle. *Genetic Algorithms and Intelligent Agents team Up: Techniques for Data Assembly, Preprocessing, Modeling, and Optimizing Decisions*. *PC-AI Magazine*, May-June, 2001.
- Dulay, Naranker, Editor. *Surveys and Presentations in Information Systems Engineering (SURPRISE)1996*. <http://www.doc.ic.ac.uk/~nd/surprise.html>
- Nikolaev, N., and Iba, H. *Genetic Programming of Polynomial Models for Financial Forecasting*. In: Shu-Heng Chen (Ed.), *Genetic Algorithms and Genetic Programming in Computational Finance*, Kluwer Academic Publ., Boston, MA. (2002).