Chapter 1

GENETIC ALGORITHMS AND GENETIC PROGRAMMING IN COMPUTATIONAL FINANCE: AN OVERVIEW OF THE BOOK

Shu-Heng Chen
AI-ECON Research Center, Department of Economics, National Chengchi University
Taipei, Taiwan 11623
chchen@nccu.edu.tw

Abstract This chapter reviews some recent advancements in financial applications of genetic algorithms and genetic programming. We start with the more familiar applications, such as forecasting, trading, and portfolio management. We then trace the recent extensions to cash flow management, option pricing, volatility forecasting, and arbitrage. The direction then turns to agent-based computational finance, a bottom-up approach to the study of financial markets. The review also sheds light on a few technical aspects of GAs and GP, which may play a vital role in financial applications.

Keywords: Genetic Algorithms, Genetic Programming, Agent-based Computational Finance, Financial Engineering

Introduction

It has been exactly ten years since the first published application of genetic algorithms to computational finance (Bauer and Liepins (1992)). After a decade of development, it is now time to reflect upon how genetic algorithms (**GAs**) and genetic programming (**GP**) have contributed to computational finance. Even though, to date, there are only about 150 publications in this area, their application coverage is continuously increasing. The twenty-one chapters presented in this volume give us a general picture of the current state. In this volume you will see the application of genetic algorithms and genetic programming to a large domain in computational finance. In addition to the conventional applications to financial forecasting, trading strategies, trading system development, and portfolio management, there are novel applications to cash flow management, volatility modeling, option pricing, index options, and futures arbitrage.

The materials presented in this book are divided into two halves. The eleven chapters in the first half (Chapters 4- 14) discuss the modeling of financial optimization. The seven chapters (Chapters 15-21) in the second half focus on the modeling of financial markets. These two halves are, however, related as what microeconomics (individuals) is to macroeconomics (aggregation). The first half provides a blueprint for modeling individual *financial agents*, whereas their collective interacting behavior is dealt with in the second half. The connection of these divisions is fascinatingly described in Kwok Yip Szeto's and Markose's chapter in this volume. To give a background review of this research area, the volume starts with three introductory chapters (Chapters 1-3), which provide some basic materials, literature reviews, and computing practices.

1. Introductory Chapters

A great number of introductory materials to this research area is available. Bauer (1994a) is the oldest yet still the best textbook on introducing genetic algorithms to finance people, helping them to see the relevance of genetic algorithms to computational finance. Unfortunately, there are no equivalent textbooks on genetic programming. Nonetheless, Smith (1998) and Chen (1998a) provide a comprehensive review of the financial applications of genetic programming. Furthermore, in a more broader content, genetic algorithms and genetic programming are a branch of evolutionary computation and, even in a more broader sense, a branch of computational intelligence. Hence, Deboeck (1994) and Chen (2002a), while not exclusively devoted to GAs and GP, also include many overview articles. In particular, Chen and Kuo (2002) provides a bibliography covering almost 400 papers on evolutionary computation in economics and finance. Among the 400 papers, there are about 150 which are directly related to the subject of this volume. Given this rich resource, it is no longer necessary to duplicate too many introductory materials on this already sizeable volume. The two chapters presented in **Part I** of the volume do, however, make this volume self-contained.

Chapter 2, **Genetic Algorithms in Economics and Finance: Forecasting Stock Market Prices and Foreign Exchange**, by Adrian Drake and Robert Marks is a good start for those who have no background in genetic algorithms. In addition to a nice introduction to the basics of genetic algorithms, the chapter also reviews the *first half* of a decade's development of this area, from Bauer and Liepins (1992) to Pereira (1996). The review leads us through the earliest three financial domains to which genetic algorithms are applied, including financial status identification (classification), portfolio selection, and trading strategies.

A basic issue which may interest beginners is *implementation.* Implementation of genetic algorithms may not be difficult thanks to the many commercially available software packages. However, at this moment, very few packages on computation finance have a module of GP. While some GP software can be downloaded from websites, it is not written for finance people. To use it, one needs to know some programming language, and for those who are not equipped with programming skills, it may be a daunting task to apply GP. To help general finance people to overcome any technical obstacles and to exploit this novel tool, this volume provides a menu-driven program on GP.

Chapter 3, **Genetic Programming: A Tutorial with the Software Simple GP**, by Shu-Heng Chen, Tzu-Wen Kuo and Yun-Pyng Shieh is written to acquaint those beginners without a programming background with the six essential elements of genetic programming, i.e., the survival-of-the fittest principle, selection schemes, disruption avoidance, search intensity, primitives, and genetic operators. None of them are far-fetched for readers, because they can directly play and interact with each of these elements via the software's main menu, **Simple GP**. The software is demonstrated with a series of simulations to justify *a set of simple rules of thumb* in order to conduct an effective implementation of GP. A special feature of this chapter is that the authors describe the behavior of GP with the application of *production theory* from economics and *portfolio theory* from finance.

2. Main Application Domains

Financial *forecasting* and *trading* are the most active financial application domains of GAs and GP. Based on the bibliography prepared by Chen and Kuo (2002), there are about 40 publications for the former and 35 publications for the latter.[1] These two application domains are tightly connected, because one of the main purposes in making high-quality financial forecasting is to enhance the profitability of *trading.* For that purpose, one can first apply GAs or GP to evolve and build forecasting models, and then one can base trading decisions upon the resultant forecasts. Chapters 4 and 6 are applications of this style. Alternatively, one can also use GAs or GP to evolve trading decisions *directly.* Chapters 7 to 9 are cases in point.

2.1. Forecasting Financial Time Series

Forecasting financial time series can be generally described as follows. Given a time series data, $\{x_t\}$, we look for a mathematical function, $f(\ )$, such that

$$x_{t+1} = f(x_t, x_{t-1}, x_{t-2}, ...) + \epsilon_{t+1} = \hat{x}_{t+1} + \epsilon_{t+1}, \tag{1.1}$$

where the series $\{\epsilon_t\}$ is statistically independent or patternless. Term $\hat{x}_{t+1}$ is the forecast of x_{t+1}. A trading decision $g(\ )$ is the mapping,

$$g : \hat{x}_{t+1} \rightarrow \{buy, sell, hold\}. \tag{1.2}$$

Two issues can arise here. First, which series $\{x_t\}$ should interest us? Second, what is the effective characterization of the function $f(\ )$ which we are looking for? These two issues are not separate as they are evidently related through Equations 1.1 and 1.2. Ideally, the series $\{x_t\}$ should be very informative as far as market timing is concerned. Nonetheless, that information generally is hidden and cannot be effectively extracted without an appropriate choice of function $f(\ )$. In **Part II** of the volume, Chapter 4 will address the first issue, as the authors assert that activities on the internet may be an interesting series to forecast. Chapters 5 and 6 will show *two progresses* made for the second issue. The former is motivated by a *function approximation approach*, while the latter is based on the concept of *multi-stationarity.*

As already mentioned, one important issue in forecasting financial time series is the *set of variables* upon which a forecast is made. In the case of predicting stock prices, the variables range from the history of the stock price and trading volumes to technical indicators. However, attention has never been given to the information (communication) flows among market participants. Would it be a *signal* for us to predict the stock price if we see a case of *unusually* active communication among traders, or for that matter, a case of unusual quietness? In brief, would *communication density* help predict the stock price? This question has never been addressed in the literature partially because direct observations on communication density are not available. Interestingly, Chapter 4, **Genetic Programming and the Predictive Power of Internet Message Traffic**, by James Thomas and Katia Sycara, uses *the message board volume* data as a proxy for communication density, and addresses the question: *whether the message board volume data has predictive power.*

From yahoo.com and ragingbull.com, the authors collect *the volume of message board postings* on 68 stocks from Russell 1000. They then construct a time series of message traffic volume numbers, and use GP to search for trading signals. Their initial results are quite encouraging. A standard GP applied to the message traffic volume can generate trading rules for which the performance is superior to that of a buy-and-hold strategy in terms of excess returns, the excess Sharpe ratio, and the differential Sharpe ratio. Based on the bootstrap test, its dominance is statistically significant. The authors further show that message traffic volume provides genuinely new information which may not be revealed by returns and trading volumes.

The paper also deals with a few technical issues of GP. The authors place *proper representation* and *overfitting avoidance* as the keys to GP success, and consider tinkering with the parameters or using sophisticated search techniques of only secondary importance.

While genetic programming is extensively applied to financial forecasting, its mathematical or statistical foundation has not been rigorously laid. In particular, there are no rules to follow in the choice of *primitives*, and more often than not it is quite an arbitrary decision. Over the past few years, Hitoshi Iba has contributed some foundation works based on the *function approximation approach.* Chapter 5, **Genetic Programming of Polynomial Models for Financial Forecasting**, by him and Nikolay Nikolaev, shows a concrete application of the function approximation approach.

Based on *power series expansion*, or more precisely, the *Kolmogorov-Gabor polynomial*, the paper uses a set of *transfer polynomials* as the function set. This function set makes the GP system studied by the authors dramatically different from the standard GP, which uses basic arithmetic operations, e.g., $+$, -, $\times$, /, as the function set. The GP system with the transfer polynomials, called **STROGANOFF**, is then compared to the standard GP in forecasting financial time series.

In addition to the function set, another crucial issue addressed by the authors is the design of the *fitness function*, as the vanilla error function generally leads to overfitting solutions. Generally speaking, there are two approaches to tackling this issue. One is to add a *validation* step between the training and testing step. The other is to incorporate a parsimony and/or smoothness criterion into the fitness function, and that is what is done by the authors. They find that the elaborated fitness function enables one to find better forecasting solutions than the vanilla error function.

The authors' attention is also drawn to *data pre-processing.* Extremely noisy raw data may give GP a hard time here. The chapter considers three different transformations of raw series, and their empirical findings contribute to our understanding of the following issues. First, should one use raw data or *moving-average transformations*? Second, should one use price series or return series to make forecasts? The answers depend on the GP system which we apply. It is found that simple moving-average transformations may not help the GP of polynomials (**STROGANOFF**) to evolve profitable polynomial models from given price movements, though they are helpful to the standard GP system. On the other hand, GP of polynomials produces models that outperform those from standard GP on *return series*, and these polynomials yield the highest profits in all the experiments.[2]

Nikolaev and Iba's application of GP to financial forecasting is based on *universal approximation.* In contrast to the *global* modeling strategy, Chapter 6, **NXCS: A Hybrid Approach to Stock Indexes Forecasting**, by Giuliano Armano, Andrea Murru and Michele Marchesi proposes *local approximation* based on a novel technique on domain decomposition. Their chapter contributes to the volume in several different ways. Firstly, like the previous one, the chapter is motivated by a theoretical consideration, i.e., *multi-stationarity* in a financial time series, or to put it differently, *piecewise stationarity* or *quasi-stationarity.* The authors assert that *under the hypothesis that financial time series are multi-stationary, obtaining a single model that holds for different regimes can be extremely difficult.* Therefore, instead of identifying a global model, they attempt to identify different local models, known in the literature as the *guarded experts framework.*

Secondly, the authors give a magnificent presentation of the idea of *guarded experts*, tracing the origin of the concept. The literature review broadly covers a significant proportion of *computational intelligence* and *time series analysis*, from the early nearest neighborhood (classification and regression trees, decision trees, threshold autoregressive models, multivariate adaptive regression splines), to the most recent neural networks and extended classifier systems. The idea of guarded experts is shown to have long been pursued throughout the history of machine learning. It is composed of two parts, namely, *building the guards* (input domain decomposition) and *inviting the experts* (domain-specific models). Approaches vary in how guards and experts are established.

The system proposed by the authors, **NXCS**, is an *evolutionary system* whereby a population of NXCS experts, each characterized by an XCS (eXtended Classifier System) classifier and corresponding ANN (artificial neural network) predictor, is raised in a typical XCS-like environment. The XCS classifiers are then renewed and revised by the standard genetic algorithm. The system has been tested on COMIT, S&P500, and Nasdaq. In terms of the normalized Sharpe ratio, the results point to the good forecasting capability of the system, which repeatedly outperforms the "Buy and Hold" strategy.

This chapter is also the only application of an extended classifier system (**XCS**) in the volume. Unlike genetic algorithms and genetic programming, XCS may sound less familiar to readers with a background in finance, and the appendix and references given in the chapter may help interested readers to gain some familiarity with it.

2.2. Trading

Like financial forecasting, one of the most fundamental issues in the application of GAs or GP to financial trading is still *representation*, i.e., *how to effectively characterize a trading strategy which one is looking for.* Research on this issue is very much motivated by the format of existing trading strategies, and there are generally two approaches to this. The first approach, called the *decision tree* approach, was pioneered by Bauer and Liepins (1992), Bauer (1994a) and Bauer (1995). In this approach each trading strategy is represented by a decision tree. At the early stage, Bauer used the bit string to encode these decision tress, and generated and evolved them with genetic algorithms. However, since the expression power of the bit-string representation is very limited, it makes GAs very difficult to represent and to evolve decision trees with various shapes and sizes. Allen and Karjalainen (1999), Neely et al. (1997), Neely and Weller (1999) and many others started to use the parse trees of genetic programming to represent the decision trees of trading strategies, as GP enables users to explore a significantly larger space of trading strategies.[3]

The second approach, called the *combinatoric* approach, was first seen in Palmer et al. (1994).[4] The combinatoric approach treats each trading strategy as one realization of $\binom{n}{k}$ combinations, where $1 \leq k \leq n$, and n is the total number of trading rules. Using GAs, one can encode the *inclusion* or *exclusion* of a specific trading rule as a bit and the whole trading strategy as a chromosome. However, this approach can only *combine* the rules known to the users, but cannot *generate* anything that sounds *novel* to the users.

The three chapters presented in **Part III** of the volume provide some of the most recent advancements for both of the two representations. Chapter 7 proposes a *novel fitness function* to evolve financial decision trees, whereas Chapter 8 uses the *Backus-Norm Form* to address the semantic issue of financial decision trees. Chapter 9 deals with the novelty issue of the combinatoric approach via the *fuzzification* of the trading rules.

Chapter 7, **EDDIE for Financial Forecasting**, by Edward Tsang and Jim Li introduces a financial trading system developed at the University of Essex, called **EDDIE**, which is used to generate trading rules in the form of decision trees. To evaluate and evolve trading strategies, one has to notice that financial agents usually pursue more than just one goal. They care about profits, but they hate risks. They do not want to miss any golden opportunity, but they are afraid of catastrophe. Usually, these conflicting desires are compromised via a fitness measure which assigns different weights to different goals, and GAs or GP maximizes or minimizes these *unconstrained* fitness functions. In this chapter the authors introduce an alternative way to solve the conflicts based on a *constrained* fitness function, called **FGP-2**. The constraints, in their case, are the minimum and maximum percentage of investment recommendations,whereby FGP-2 takes two parameters from the users and then trades the rate of precision with the rate of missing opportunities. They test the return performance of FGP-2 against three artificial neural networks and a linear classifier on 10 stocks.

Chapter 8, **Forecasting Market Indices Using Evolutionary Automatic Programming**, by Michael O'Neill, Anthony Brabazon, and Conor Ryan introduces a non-standard application of GP which was seldom seen in previous financial applications. This non-standard application is based on the *Backus-Naur Form* (**BNF**) in computer theory.[5]

Financial users have long complained about the *semantic meaning* of the programs evolved from the standard GP. Chen 2002 included this issue as one of the four main issues in the use of genetic programming in economics and provided a lengthy discussion of it. Recent efforts made to cope with this issue subject the standard GP with *grammar*. Two pioneering applications of this kind are Bhattacharyya et al. (1998) and Nikolaev and Iba (2000), but Duffy and Engle-Warnick (2002) is the first one that explicitly referred to the *Backus-Naur Form*. What makes O'Neill et al's chapter further different from these early applications is their *representation*. Rather than representing the programs as *syntax trees*, *a linear genome representation* is used. Each trading rule is encoded as a *variable-length* binary string, containing in its condons (groups of 8 bits) the information to select production rules from **BNF** grammar.

Chapter 9, **Genetic Fuzzy Expert Trading System for NASDAQ Stock Market Timing**, by Sze Sing Lam, K. P. Lam and Hoi Shing Ng applies the conventional combinatoric approach to encode and evolve a trading expert system with a genetic algorithm. However, in their application, trading rules are not crispy, but *fuzzy*. The advantages of using fuzzy trading rules over the crispy ones are well discussed in Tay and Linn (2001). The fuzzy approach to modeling agents' behavior is intuitively sound, and its combined use with genetic algorithms or genetic programming should be a promising direction for further research.

Financial markets are complex in the sense that the underlying law of motion, if it exists, is *non-linear* and *time-variant*. The *non-linear* characteristic justifies the use of GAs or GP as the data mining toolkit, but that is not good enough. The *time-variant* characteristic requires the use of GAs or GP in an adaptive way, usually called *dynamic learning* or *adaptive learning*. The authors of this chapter also notice the significance of this issue and show the extra gains which one may have if GAs are combined with the use of an adaptive learning scheme.[6].

3. Miscellaneous Application Domains

Financial forecasting and trading take up 60% of the published applications of GAs and GP to financial engineering. The other 40% of them do not belong to any single application domain. Instead, they are unevenly distributed. The five chapters presented in **Part IV** of the book show these miscellaneous applications. They are *portfolio optimization* in Chapter 10, *cash flow management* in Chapter 11, *option pricing* in Chapter 12, *volatility modeling* in Chapter 13, and *arbitrage* in Chapter 14.

3.1. Portfolio Management

Portfolio management is not a new application domain, given that the first journal publication appeared in 1995 (Leinweber and Arnott, 1995).[7] *Portfolio optimization* is a very standard financial problem. The traditional approach to portfolio optimization is the mean variance framework, which is known as a *quadratic optimization* problem and can be solved analytically. However, financial reality may complicate both the *objective function* and *constraints* facing financial agents, which transforms the standard problem into one that is difficult to solve analytically. For example, Baglioni et al. (2000) showed how financial regulation can complicate the objective function and constraints in the case of a pension fund; Hiemstra (1996) exemplified how the short-run fluctuation in excess returns and volatility can modify the standard problem into a more difficult tactical asset allocation problem. They all suggested the use of GAs to tackle the optimization problem.

Chapter 10, **Portfolio Selection and Management Using a Hybrid Intelligent and Statistical System**, contributed by Juan Lazo, Marco Pacheco, and Marley Vellasco continues this line of research. They proposed a hybrid-system approach to portfolio selection and management. The distinguishing feature of this chapter is applying artificial neural nets to forecasting returns and the GARCH model to forecasting volatility. Based on the estimated returns and volatility, the authors use GAs to determine the optimal portfolios under different objectives, e.g., maximizing the Sharpe ratio and minimizing risks under a target return. The chapter is probably the first one that uses VaR (*Value at Risk*), one of the most popular risk measures, to evaluate the risk of GA-based portfolios.

3.2. Cash Flow Management

A subject related to portfolio management is *cash flow management*, a standard issue in corporate finance. While it should not be a surprise to see the application of GAs or GP to this area, any such work has never been done. Chapter 11, **Intelligent Cash Flow: Planning and Optimization Using Genetic Algorithms**, by Marco Pacheco, Marley Vellasco, Maíra F. de Noronha, and Carlos Henrique P. Lopes initiates this application to cash flow planning. Like many other financial issues, cash flow planning deals with an enormous amount of search space, which can be computationally demanding. As shown by the authors, cash flow planning for a period of 90 days can face 68^{90} possibilities if we consider 68 options of investment products for each day of a 90-day period.

Pacheco et al.'s paper also leads us to a very technical and important issue in GAs, namely, *epistasis*. Epistasis refers to *the lack of independence among bits* with respect to a fixed fitness function. A precise description of this phenomenon was given by Davidor (1991). With the presence of epistasis, high-performance schemata may point toward a poor area of the space: good low-order building blocks lead to poor higher-order building blocks. There are many studies regarding how the performance of GAs is affected by epistasis. These studies frequently center on the usefulness of operators such as crossover and mutation in solving epistatic problems. Nonetheless, the relevance of the epistatic issue to finance was not noted in financial applications of GAs until the appearance of their paper. To address the epistatic problem, they use the *partially matched crossover* (**PMX**) derived by Goldberg and Lingle (1985), which is a binary operator that combines ordering building blocks from above-average parents in a sensible way.

3.3. Option Pricing

Optional pricing is another new application domain. It was not until 1998 that the first journal article on this area was published. Jay White (1998) utilized *genetic adaptive neural networks* (**GANNs**) for pricing *interest rate futures call and put options.* In his application the option pricing formulae were *not* encoded *directly* by bit strings (chromosomes). Instead, they were represented by three-layer, feedforward, artificial neural networks. Genetic algorithms were then used to evolve and determine the weights of the neural nets. This representation is known as *indirect representation.* In indirect representation, a chromosome is not mapped directly to a *solution*; rather it is mapped to *a structure of a solution.* This indirect representation has become a common practice to enhance the expression power of genetic algorithms in many financial applications.

Option pricing formulae can also be directly represented by bit strings, though it is not that straightforward. Chen and Lee (1997) use a series expansion approach to represent a European call option formula, and truncated the infinite series to a finite one. The coefficients of the series were then encoded by bit strings, and evolved with genetic algorithms. However, since determining the size and shape of option pricing formulae is in general very difficult, the majority of recent studies have all adopted the parse-tree representation [Chen, Lee and Yeh (1999); Chidambaran et al. (2000); Keber (2000); Keber (2001)], as does the next chapter.

Chapter 12, **The Self-Evolving Logic of Financial Claims**, by Thomas Noe and Jonathan Wang demonstrates a standard application of genetic programming to option pricing. Using genetic programming, they first show that the Black-Sholes formula can be recovered from the simulated data. They then apply GP to S&P 500 futures options, finding that the performance of GP in pricing options is at least comparable to the performance of artificial neural nets in Hutchinson et al. (1994).[8]

The chapter also touches on a few technical issues in the financial application of GP. As in many other financial applications, the design of the fitness function is not always a trivial issue. To accurately measure the performance of a pricing formula from the range of *out-of-the-money* to the range of *in-the money*, the authors include in the fitness function both the *absolute error* and the *absolute percentage error.* The inclusion of the latter is particularly important when the option price is deep out of the money and the price is small. Another inclusion to the fitness function is a penalty based on the *program size.* The program size is measured by the number of nodes in the parse tree, the so-called *node complexity.* This modification is intended to bias the search toward functions with fewer nodes, which are simpler and therefore less prone to overfit the data.

An alternative approach taken to deal with overfitting is to add a *validation step* immediately after the training step. This approach does not impose a penalty to the program size directly, but uses a *selection period* to confirm that the expression power of GP has not been abused. In financial applications, the *selection* approach is first taken up by Allen and Karjalainen (1999) and Neely et al. (1997) and further used in Neely and Weller (1999) and Wang (2000). It has now become a standard procedure for GPers in financial data mining.[9]

3.4. Volatility

While the focus of financial forecasting is either price or return, for many financial decisions, these two factors are relatively less important than *fluctuation*, known as *volatility* in finance. A mathematical description of volatility, $\{v_t\}$, is given in Equations 1.3 to 1.5.

$$R_t = f_t(R_{t-1}, ..., R_{t-p}, \epsilon_{t-1}, ..., \epsilon_{t-q}) + \epsilon_t, \tag{1.3}$$

$$\epsilon_t = v_t \mu_t, \tag{1.4}$$

where R_t is the stock return, μ_t is a standard normal random variable and

$$v_t = h(R_{t-1}, ..., R_{t-r}, \epsilon_{t-1}, ..., \epsilon_{t-s}, h_{t-1}, ..., h_{t-m}). \tag{1.5}$$

We saw earlier the application of GP for modeling the function $f(\ )$ (Equation 1.1). However, few studies have extended the application of GP to modeling $h(\ )$.[10] Chen and Yeh 1997 and Chen (1998b) are the only such publications.[11] Based on the *jump process*, Chen and Yeh propose a non-parametric approach, called *adaptive genetic programming* (**AGP**), to model $h(\ )$.

Chapter 13, **Using a Genetic Program to Predict Exchange Rate Volatility**, by Christopher Neely and Paul Weller is an extension of the authors' early studies on forecasting exchange rates [Neely et al. (1997), Neely and Weller (1999)]. Those successful applications of genetic programming motivated the authors to advance from forecasting the conditional mean to forecasting the conditional variance. Obviously, one is curious as to whether genetic programming can forecast volatility better than the well-known *Generalized Autoregressive Conditionally Heteroskedastic* (**GARCH**) model. The authors conducted a series of careful experiments and tested the forecasting performance of GP against that of the GARCH model over different *time horizons*, using various *accuracy criteria*. Their results are mixed, with GP often outperforming the GARCH model on longer horizons and consistently returning lower mean absolute forecast errors. However, on short horizons the GARCH model outperforms GP in terms of the mean squared error.

Running GP involves many technical issues to which there are no general answers. Among the tricky ones are the determination of the function set, the fitness function, and the over-fitting avoidance strategy. In this chapter the authors provide two tests on these technical designs. The first test concerns the role that the function set plays. They consider two function sets, one with *primitive functions* and the other with primitive functions *plus* advanced functions. The second test concerns the benefits that one can gain by avoiding over-fitting. For this test, the authors incorporate into the fitness function a penalty depending on *node complexity*, as shown in the previous chapter. Interestingly, neither imposing a penalty for complexity nor expanding the set of data functions leads to any appreciable improvement in the performance of the genetic program. Adding a penalty function does not help in this case, probably because the program already has a *validation* step, which itself is a design for over-fitting avoidance.

3.5. Arbitrage

The last chapter of Part IV, **Evolutionary Decision Trees for Stock Index Options and Futures Arbitrage**, by Sheri Markose, Edward Tzang and Hakan Er applies genetic programming to *stock index options and futures arbitrage*; more precisely, the short **P-C-F** arbitrage. The ex post analysis of efficiency violations for short arbitrage positions shows on average that for all periods to maturity the profits from the arbitrage is substantial and statistically significant. It would be interesting to know whether GP can correctly identify and exploit profitable short arbitrage opportunities in a real time setting. The performance of **EDDIE-ARB**, a system developed by the authors, is compared with that of a naive strategy which executes an arbitrage trade whenever there is a contemporaneous profit signal. Due to the execution delay, the naive strategy faces execution price risk, as not every contemporaneous **P-C-F** profit signal will continue to be profitable. Therefore, with the assumed time delay in the execution of an arbitrage from an observed contemporaneous profit signal, an effective forecasting tool is needed to assess the success rate of such a strategy.

4. Agent-Based Computational Finance

Part V represents another active application domain of genetic algorithms and genetic programming in computational finance, namely *agent-based computational finance* (**ACF**). The term**ACF** implies a computational study of financial behavior and financial markets modeled as *evolving decentralized systems of autonomous agents.*[12] Introductory material can be found in LeBaron (2000). He also constructs a website

http://www.unet.brandeis.edu/ blebaron/acf for this new area in finance. Interested readers are referred directly to this website.

A central element of **ACF** is the simulation of financial agents' evolution in financial markets. This is not just about a *single* financial agent. It is about a *population* of interacting (competing) agents. The previous three parts, usually known as *financial engineering*, only model an individual agent, but in ACF we need to "aggregate" these individual so that they evolve together or use *co-evolving*.

The purpose of GA and GP is to drive the evolution of a population. They can be applied towards evolving a population of financial strategies for an individual financial agent, as we see in financial engineering; so that they can be applied to a population of financial agents for a financial market, as we shall see in this part of ACF. While GA and GP are not the only tools used in ACF, they do play the most prominent role in the development of ACF. In fact, the earliest application domain of GA in computational finance is ACF.

As the publications of ACF pile up, the significance of it shall become gradually clearer for finance people. It is a promising approach for studying behavior finance, micro-structure, experimental finance, and psychological finance, while most of its current fruitful findings concentrate on the *financial econometrics*. The seven chapters in this part offer wide coverage of the recent progress in ACF, including the mathematical foundation of finance, modeling techniques of bounded rationality, price dynamics, market efficiency, trading mechanism, financial regulations, patterns of survival financial strategies or behavior, and methodological and philosophical issues. Below provides a quick grasp of them.

The foundation of mainstream financial economics or mathematical finance was built upon the *Walrasian general equilibrium analysis*, which is mainly concerned with the existence (or the non-existence) of equilibria associated with their characterizations. However, scant attention is drawn to the *market process* converging to these equilibria.[13] The usual argument that agents will *eventually learn* one of these equilibria (in particular, the Pareto superior one) is anything but well grounded.

The chapter by Thomas Riechmann, "**A Model of Bounded Rational Consumer Choice**", uses a standard general equilibrium model to show that even finding the *optimal consumption bundle* of three goods can be an extremely complicated issue for consumers. The model is very simple. It has 500 consumers. Each is endowed with the same utility function and the same budget constraint. These consumers interact with each other in an economy of three commodities, where the supply schedule of each commodity is exogenously fixed with the same elasticity. The prices are determined by equating supply with demand, which are not known to the consumers upon submission of their consumption plan. The trial-and-error process of consumers is driven by *genetic algorithms*. Under the circumstances, Riechmann examines two essential characterizations of consumer optimization. First, how well is the budget control done? Second, how good is the chosen consumption bundle?

Riechmann finds that the aggregate performance of consumers' choices crucially depends on how genetic algorithms are used to model the learning process of consumers. In particular, he evaluates the performance of the canonical GA and that with the addition of the *election operator* or the *elitist operator*. The last two operators basically prevent agents from rushing into any *new* idea without testing it first, making agents behave more prudently. While the economic meaning of these operators is intuitively sound, there is no guarantee that they will lead to desirable results.[14] The election operator tests the new ideas by estimating its preference, the so-called *potential fitness*. In some situations, this ex-ante fitness can be quite different from the realized one, and that may cause a problem.[15] In the author's words, for the case of flexible prices and low elasticity of supply, the election is far from leading to any kind of sensible consumer choice.

The performance with the elitist operator, called the *preselection operator* by the author on the other hand shows a degree of robustness. This result is interesting, because the elitist operator is usually neglected in agent-based computational economic models. Riechmann's description of the significance of the elitist operator reminds us that *memory* works in such a way as to enhance agents' learning capability.[16]

There is another technical novelty that should not go unnoticed, and that is the way Riechmann copes with the constraints in constrained optimization. Despite the many approaches that Michalewicz (1996) introduced to deal with this issue, few have been applied to financial engineering, not to mention agent-based computational finance. In this chapter the author gives a concrete example of how to use these techniques in an economic context by modifying fitness (utility) via a *penalty function*.

If boundedly-rational interacting heterogeneous agents cannot replicate the *equilibrium* in a simple general equilibrium model, then it would be no surprise if they cannot do the same thing in a more complicated financial models. The chapter by Shu-Heng Chen and Chung-Chih Liao, **Price Discovery in Agent-Based Computational Modeling of Artificial Stock Markets**, shows that indeed this is the case. They start with a standard asset pricing model with its homogenous rational expectations equilibrium price, and augment the standard asset pricing model with its *agent-based extension.* They then examine how well a population of financial agents can track the equilibrium price in the **AIE-ASM**, which is a variant of the Santa Fe Institute Artificial Stock Market. By simulating the artificial stock market with different dividend processes, interest rates, risk attitudes, and market sizes, they find that the market price is not an unbiased estimator of the equilibrium price. Except in a few extremely worse cases, the market price deviates from the equilibrium price moderately from minus four per cent to sixteen per cent.

The pricing errors are in fact not *patternless.* They are actually negatively related to *market sizes*: a thinner market size tends to have a larger pricing error, and a thicker market tends to have a smaller one. For the thickest market which they have simulated, the mean pricing error is only 2.17%. This figure suggests that the new classical simplification of a complex world may still provide a useful approximation if some conditions are met, such as, in this case, the market size.

At the end of the chapter, the authors sketch a research agenda for agent-based financial modeling. To make an agent-based computational approach a prolific tool for doing finance, they propose to include in the current artificial stock market a large variety of *financial products* and *financial agents.* Certainly, a trading mechanism should also be added to the list. In fact, the trading mechanism adopted in most agent-based stock markets either follows the *Walrasian tatonnement scheme* or the *rationing scheme.* Few studies have been done with a double auction.[17] The chapter by Shu-Heng Chen, Chung-Chin Tai, and Bin-Tzong Chie, **Individual Rationality as a Partial Impediment to Market Efficiency**, contributes to an agent-based version of the double auction market. To our best knowledge, this is the first paper to simulate the evolution of *bargaining strategies* within the context of an agent-based double auction market.[18]

The chapter re-visits a fundamental surprise in economics, i.e., the *inconsistency* between individual rationality and aggregate rationality. The usual way to put the surprise is that individual irrationality can lead to aggregate rationality. For example, in their computerized double auction market, Gode and Sunder (1993) show that a near 100% allocative efficiency can be generated from a group of zero-intelligence traders, who can only randomly bid or ask. Chen et al.'s chapter presents another way to see the surprise: individual rationality may also lead to aggregate irrationality. They considers two types of traders: smarts ones and mediocre ones. The smart traders are distinguished from the mediocre traders by a *privilege* which gives them the potential to learn sophisticated bargaining strategies with genetic programming. They then examine the allocative efficiency of 20 double auction markets composed exclusively of either smart traders or mediocre traders. Their simulation evidences that higher allocative efficiency is not achieved from a trading room of smart traders, but from one of mediocre traders. Financial regulations, from this paper, can be read as an annihilation to the evil-side of smartness.

In addition to asset pricing, another area to which agent-based simulations can be applied, is *portfolio theory.* This connection is particularly clear from Chapter 18, **A Numerical Study on Portfolio Optimization**, by Guido Caldarelli, Marina Piccioni, and Emanuela Sciubba. This chapter is motivated by an old debate in the theory of portfolio choice: *the normative appeal of logarithmic utility maximization* versus *the mean-variance approach.* The authors believe that an effective solution to the debate can come from the *evolution approach* which aims at studying long-run financial market outcomes as a result of a process akin to *natural selection.* The evolutionary approach to portfolio behavior was first taken up by Blume and Easley (1992), and was followed by Sandroni (2000). While both studies attempt to single out the key factors which determine the surviving portfolio rules, their restrictive assumptions make their analysis difficult to extend to the consideration of some interesting portfolio behavior, such as the CAPM rule.

The departure of Caldarelli et al's chapter provides more general insights into the debate based on numerical computation with less restrictive assumptions. They consider two types of traders, namely, *logarithmic traders* and *CAPM traders.* They then let these two groups of traders compete against each other in wealth. The dominance, survival, and extinction of these traders are then examined based on the asymptotic wealth shares of the traders. While the authors do not conduct their analysis with agent-based modeling, and hence do not use either genetic algorithms or genetic programming to evolve agents' portfolio behavior, it would be worthwhile to give it a try in the next step of the research.

One essential element of agent-based computational financial modeling is the *complex heterogeneity* of agents. While agents can be heterogeneous in many aspects, most agent-based computational financial models only address heterogeneity in *expectations.* Little attention has been drawn to other aspects of heterogeneity. From the previous chapter, we saw that in order to tackle the debate on the dominance of the MEL rules, it would be necessary to consider agents with different *preferences.* More generally, the artificial life of financial agents can be enriched if different *human characters* are taken into account. Chapter 19 by Kwok

Yip Szeto, **Adaptive Portfolio Managers in Stock Markets**, opens up an avenue for this direction.[19]

In Szeto's artificial stock market, the heterogeneity of portfolio managers is heterogeneous in two psychological measures, namely, a degree of *fear* and a degree of *greed*. These managers are otherwise homogeneous, including that they share the same forecasting rule. The forecasting rule is obtained by applying a genetic algorithm to extract patterns from financial time series. He finds a universal property in all his time series data: *greedy and confident investors are the winners*. The conclusion itself, while quite interesting, may not be the most important, because different enrichments of this rather simple model may lead to different results. What, however, really motivates us in this paper is the research opportunity of using agent-based financial modeling to address issues in psychological finance: e.g., *what types of personality determine a successful portfolio manager?*

While ACF brings us a great research opportunity, there is a known weakness: ACF is largely a computational model without immediate prospects for rigorous mathematical results. Since small changes may engender radically different results[20], a sensitivity analysis is required before one can ascertain a finding. Chapter 20, **Learning and Convergence to Pareto Optimality**, by Chris Birchenhall and Jie-Shin Lin provides perhaps the most extensive coverage of robustness checks ever seen in the **ACF** literature. A specific question is posed in this chapter. *How can we be sure that a certain kind of observed interesting behavior from an ACF model is attributed to real economic forces rather than to technical (genetic) parameters?* It is therefore up to them to check whether the observed behavior is *robust* to different designs of genetic operators.

Their work covers two different levels of GA designs: one is *genetic operators*, and the other is *architecture*. For the former, they consider different implementations of the four main GA operators, i.e., selection, crossover, mutation, and election. For the latter, they consider a *single-population GA* (population learning or social learning) vs. a *multi-population GA* (individual learning). They then apply all these different designs to re-run *the model of inflation* in Bullard and Duffy (1999). They find that Bullard and Duffy's results are sensitive to two main factors: the *election operator* and *architecture*. Their experimental results in fact lend support to some early findings, e.g., the significance of the election operator (Arifovic, 1994), and the different consequences of social learning and individual learning (Vriedn, 2000; Vriend, 2001; Yeh and Chen, 2001). What is particularly interesting is that *individual learning reduces the rate of convergence to the same belief*. This is certainly an important finding, because most studies on the convergence of GAs to Pareto optimality are based on the social learning version, e.g., Dawid (1996), Riechmann (1999), Riechmann (2001).

Since the result is sensitive to the design of GA, the immediate question is whether we can still attribute the economic behavior (in this case, the Pareto superior low inflation) to some economic forces. The answer depends on whether we have an economic-theoretic foundation to support a particular GA design; in their case, the use of the election operator and that of the social-learning architecture. Unfortunately, at this moment, such a foundation does not exist, and any specific GA design can be *ad hoc*.[21] To avoid *arbitrariness*, one can have the design determined *endogenously*. While this idea does not sound peculiar, it has never been tried in the context of ACF. This chapter is probably the first one to give it a try.

The authors propose an approach of *meta learning* (*open learning*), using an individual learning scheme to model agents' forecasts. The genetic operators for agents are neither homogeneous nor fixed. Instead, they evolve through *social learning*. Thus, some agents use tournament selection to evolve their forecasts, while others use a roulette-wheel selection to do the work. The same goes for other genetic operators. Over time, the market will determine which design is the best. The interesting findings from these experiments are two-fold. First, the authors find that the GA design used in Bullard and Duffy (1999) is one of the most popular survivors. Second, all runs converge and they converge to the Pareto superior low inflation equilibrium. While meta learning still has its limits, one cannot but acknowledge the novelty of this approach.

5. Concluding Chapter

After a decade's development of financial applications of GAs and GP, it is time to reflect what has been done and to examine what has been taken for granted. The volume starts with the question: *what is it?* We, however, are aware all the time of the more basic issue: *why is it so?* There are probably good reasons not to start with the fundamental issue, but there is no excuse to end this volume without touching on it. The concluding chapter, **The New Evolutionary Computational Paradigm of Complex Adaptive Systems: Challenges and Prospects for Economics and Finance**, by Sheri Markose presents a thought-provoking discussion of the issue: *were the financial applications of GAs and GP well anticipated?*

The author places the financial agents and financial markets in the context of modern *complex sciences*, and examines the research methodologies for them from the development of *computational theory* and *history of economic thoughts.* The message of the chapter is clear and strong: *dynamical system outcomes produced by algorithmic agents need not be computable.* Generally we have no way (*algorithm*) of inferring what would result from a system except by running its course. Whatever will be will come to us as *emergent properties.* Neoclassical economics fails to see this, but it was clearly identified at the provenance of the economics in the 18th century.

Given the nature of uncomputability, it would be more appropriate to treat financial agents, not as *neo-classical optimizing agents*, but as *adaptive agents* whose goals and means are changing over time. Part II to Part IV show how these adaptive agents can be built with genetic algorithms and genetic programming, whereas Part V demonstrates simulations of markets composed of these adaptive agents. In the last chapter, the author examines some theoretical issues concerning the complexity of minority games, double-auction markets and stock markets.

6. Concluding Remarks

What is the current state of financial applications of genetic algorithms and genetic programming? From a review of the 20 chapters distributed over the five parts of the volume, one can see the following observations. First, *the application coverage is continuously enlarging.* There is little doubt that new application domains will emerge in the next few years. In fact, apart from what have been said on the volume, the recent publication Noe (2000), which applies genetic algorithms to the study of *takeover* behavior, shows another novel application.

Second, to reflect credits of GAs and GP in financial applications, a *rigorous statistical analysis is imperative.* This point is well taken by the authors of the volume, as we see the involvement of various statistical procedures applied to the performance evaluation, such as Monte-Carlo simulation, bootstrap testing, and kernel estimation of error density. The statistical rigors also extend to experimental designs, as we see from Chapter 2 where the performance of GAs and GP sensitively depends on the design. Wang (2000), who casts doubt on the superior performance of GP to that of the buy-and-hold strategy, is another recent example to show such rigor.

Third, *several technical issues are still the main concerns of financial applications of GAs and GP.* The choice of *fitness function* which can correctly measure the quality of solutions is not trivial, and neither is the *representation* of solution candidates. The expression power of GP is striking, but should not be abused. It becomes clear that the canonical genetic algorithms and standard genetic programming would not be so productive for financial applications without further modifications, and chapters of this volume suggest ways to do so.

Notes

1. See Chen and Kuo (2002) pp. 425-426 for details.

2. These results can be compared to the main finding in Kaboudan (2000), which shows that one should use price series rather than return series to forecast the price.

3. There is, however, another approach to enhance the flexibility of Bauer's trading strategies. That is, to *parameterize trading rules*, and then encode them with bit strings. The GA is then used to evolve these strings. Examples can be found in Pereira (2002).

4. The combinatoric approach was also frequently seen is other application domains. See Farley and Jones (1994).

5. For those who are not familiar with the Bcakus-Naur Form, ? could be a useful reference.

6. For a systematic study of the significance of adaptive learning schemes in trading applications of GAs, one is referred to Chen and Lin (1997).

7. Bauer (1994b) applied genetic algorithms to solve a related problem, namely, selection of mutual funds.

8. Hutchinson et al. (1994) is the first journal publication of an application of artificial neural nets to option pricing.

9. For instance, also see Chapters 4 and 13 of the volume.

10. In addition to genetic programming, there are other non-parametric or semiparametric approaches to modeling the function $h(\ )$. The interested reader is referred to Chen (1998b).

11. Not in the context of a financial time series, but in the context of option pricing, Christian Keber has conducted a series of studies on *implied volatility* using genetic programming. See Keber (1999), Keber (2000) and Keber (2001).

12. Italics are borrowed from the Tesfatsion (2001) in the definition of *agent-based computational economics.*

13. While *computational general equilibrium models* do provide a *constructive* proof of the existence of the equilibrium, the construction itself is not a real market process. For a full discussion, see the Markose chapter in this book.

14. It is true that the use of the election operator is already a standard procedure in agent-based computational economics. It has also been shown in many cases that without the inclusion of the election operator, one can have quite disappointing results, but there are also cases where the election operator is preferred not to be used. For details, the interested reader is referred to Chen 2002.

15. Birchenhall (1995) is probably the first one who acknowledges this problem.

16. Economic studies comparing the performance of the two operators are limited.Novkovic 1998 is the only paper that shows some advantages of the elitist operator over the election operator.

17. See Chen (2001) for an account of this development.

18. Chen (2000) provided a literature review of some experimental and computerized double auction markets.

19. The first half of the chapter is devoted to a review of the author's early applications of genetic algorithms to financial time series prediction, including Fong and Szeto (2001), and Szeto and Luo (1999).

20. See Fogel et al. (2002) for two interesting examples.

21. See Chen (2001) and Chen 2002 for an in-depth discussion.

References

Allen, F. and R. Karjalainen (1999). "Using genetic algorithms to find technical trading rules," *Journal of Financial Economics*, 51(2), 245–271.

Arifovic, J. (1994). "Genetic Algorithm Learning and the Cobweb Model," *Journal of Economic Dynamics and Control*, 18(1), 3–28.

Baglioni, S., D. Sorbello, C. C. Pereira, and A. G. B. Tettamanzi (2000). "Evolutionary Multiperiod Asset Allocation," in Whitley D., Goldberg D., Cantú-Paz E., Spector L., Parmee I., Beyer H.-G. (eds.), *Proceedings of the Genetic and Evolutionary Computation Conference*, 597–604. Morgan Kaufmann.

Bauer, R. J. Jr. (1994a). *Genetic Algorithms and Investment Strategies.* New York: John Wiley & Sons.

Bauer, R. J. Jr. (1994b). "An Introduction to Genetic Algorithms: A Mutual Fund Screening Example," *Neurove$t Journal*, 2(4), 16–19.

Bauer, R. J. Jr. (1995). "Genetic Algorithms and the Management of Exchange Rate Risk," in Biethahn J., Nissen V. (eds.), *Evolutionary Algorithms in Management Applications.* 253–263, Heidelberg and New York: Springer.

Bauer, R. J. Jr. and G. E. Liepins (1992). "Genetic Algorithms and Computerized Trading Strategies," in O'leary D. E., Watkins R. R. (eds.), *Expert Systems in Finance.* North Holland.

Bhattacharyya, S., O. Pictet, and G. Zumbach (1998). "Representational Semantics for Genetic Programming Based Learning in High-Frequency Financial Data," in Koza J. R., Banzhaf W., Chellapilla K., Deb K., Dorigo M., Fogel D. B., Garzon M. H., Goldberg D. E., Iba H., Riolo R. (eds.), *Genetic Programming 1998: Proceedings of the Third Annual Conference*, 11–16. Morgan Kaufmann.

Birchenhall, C. R. (1995). "Modular Technical Change and Genetic Algorithms," *Computational Economics*, 8(3), 233–253.

Blume, E. and E. Easley (1992). "Evolution and Market Behavior," *Journal of Economic Theory*, 58, 9–40.

Bullard, J. and J. Duffy (1999). "Using Genetic Algorithms to Model the Evolution of Heterogeneous Beliefs," *Computational Economics*, 13(1), 41–60.

Chen, S.-H. (1998a). "Evolutionary Computation in Financial Engineering: A Roadmap to GAs and GP," *Financial Engineering News*, 2(4).

Chen, S.-H. (1998b). "Modeling Volatility with Genetic Programming: A First Report," *Neural Network Worlds*, 8(2), 181–190.

Chen, S.-H. (2000a). "Toward an Agent-based Computational Modeling of Bargaining Strategies in Double Auction Markets with Genetic

Programming," in K.S. Leung, L.-W. Chan, and H. Meng (eds.), *Intelligent Data Engineering and Automated Learning- IDEAL 2000: Data Mining, Financial Engineering, and Intelligent Agents*, Lecture Notes in Computer Sciences 1983. 517–531. Springer.

Chen, S.-H. (ed. 2002a). *Evolutionary Computation in Economics and Finance.* Physica-Verlag.

Chen, S.-H. (2002b). "Fundamental Issues in the Use of Genetic Programming in Agent-Based Computational Economics," in A. Namatame T. Terano and K. Kurumatani (eds.), *Agent-based Approaches in Economic and Social Complex Systems*, 208–220. IOS Press.

Chen, S.-H. (2002c). "Evolutionary Computation in Economics and Finance: An Overview of the Book," in Chen, S.-H. (ed.), *Evolutionary Computation in Economics and Finance*, 1–26. Physica-Verlag.

Chen, S.-H. and T.-W. Kuo (2002). "Evolutionary Computation in Economics and Finance: A Bibliography," in S.-H Chen (ed.), *Evolutionary Computation in Economics and Finance*, 419–444. Physica-Verlag.

Chen, S.-H. and W.-C. Lee (1997). "Option Pricing with Genetic Algorithms: The Case of European-Style Options," in T. Back (ed.), *Proceedings of the Seventh International Conference on Genetic Algorithms*, 704–711, San Francisco, CA: Morgan Kaufmann Publishers.

Chen, S.-H. and W.-Y. Lin (1997). "Financial Data Mining with Adaptive Genetic Algorithms," in Philip T. (ed.), *Proceedings of the 10th International Conference on Computer Applications in Industry and Engineering*, 154–159

Chen, S.-H. and C.-H. Yeh (1997). "Using Genetic Programming to Model Volatility in Financial Time Series," in Koza J. R., Banzhaf W., Chellapilla K., Deb K., Dorigo M., Fogel D. B., Garzon M. H., Goldberg D. E., Iba H., Riolo R. (eds.), *Genetic Programming 1997: Proceedings of the Second Annual Conference.* 58–63. Morgan Kaufmann.

Chen, S.-H., W.-C. Lee, and C.-H. Yeh (1999). "Hedging Derivative Securities with Genetic Programming," *International Journal of Intelligent Systems in Accounting, Finance and Management*, 8(4), 237–251.

Chen, S.-H., W.-Y. Lin, and C.-Y. Tsao (1999). "Genetic Algorithms, Trading Strategies and Stochastic Processes: Some New Evidence from Monte Carlo Simulations," in Banzhaf W., Daida J., Eiben A. E., Garzon M. H., Honavar V., Jakiela M., Smith R. E. (eds.), *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference.* 114–121. Morgan Kaufmann.

Chidambaran, N., C.-W. J. Lee, and J. Trigueros (2000). "Option Pricing via Genetic Programming," in Abu-Mostafa Y. S., LeBaron B., Lo

A. W., Weigend A. S. (eds.), *Computational Finance – Proceedings of the Sixth International Conference*, Cambridge, MA: MIT Press.

Cohen, D. I. A. (1991). *Introduction to Computer Theory.* Wiley.

Davidor, Y. (1991). "Epistasis Variance: A Viewpoint on GA Hardness," in Rawlins, G. J. E. (ed.) *Foundation of Genetic Algorithms*, 23–35. San Mateo: Morgan Kaufmann Publishers.

Dawid, H. (1996). *Adaptive Learning by Genetic Algorithms: Analytical Results and Applications to Economical Models*, Heidelberg and New York. Springer.

Deboeck, G. J. (ed. 1994). *Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets.* John Wiley & Sons.

Duffy, J. and J. Engle-Warnick (2001). "Using Symbolic Regression to Infer Strategies from Experimental Data," in S.-H. Chen (ed.), *Evolutionary Computation in Economics and Finance.* Physica-Verlag.

Farley, A. M. and S. Jones (1994). "Using a Genetic Algorithm to Determine an Index of Leading Economic Indicators," *Computational Economics*, 7(3), 163–173.

Fogel, D. B., K. Chellapilla, and P. J. Angeline (2002). "Evolutionary Computation and Economic Models: Sensitivity and Unintended Consequences," in S.-H. Chen (ed.), *Evolutionary Computation in Economics and Finance.* Physical-Verlag.

Fong, A. L. Y. and K. Y. Szeto (2001b). "Rules Extraction in Short Memory Time Series Using Genetic Algorithms," *European Physics Journal B*, 20, 569–572.

Gode, D. K. and S. Sunders (1993). "Allocative Efficiency of Market with Zero-Intelligence Trader: Market as a Partial Substitute for Individual Rationality," *Journal of Political Economy*, 101(1), 119–137.

Goldberg, D. E. and R. Lingle (1985). "Alleles, Loci, and the Traveling Salesman Problem," *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, 154–159.

Hiemstra, Y. (1996). "Applying Neural Networks and Genetic Algorithms to Tactical Asset Allocation," *Neurove$t Journal*, 4(3), 8–15.

Hutchinson, J. M., A. W. Lo, and T. Poggio (1994). "A Nonparametric Approach to Pricing and Hedging Derivative Securities via Learning Networks," *Journal of Finance*, 851–889.

Jay White, A. (1998). "A Genetic Adaptive Neural Network Approach to Pricing Options: A Simulation Analysis," *Journal of Computational Intelligence in Finance*, 6(2), 13–23.

Kaboudan, M. A. (2000). "Genetic Programming Prediction of Stock Prices," *Computational Economics*, 16(3), 207–236.

Keber, C. (1999). "Genetically Derived Approximations for Determining the Implied Volatility," *OR Spektrum*, 205–238.

Keber, C. (2000). "Option Valuation with the Genetic Programming Approach," in Abu-Mostafa Y. S., LeBaron B., Lo A. W., Weigend A. S. (eds.), *Computational Finance – Proceedings of the Sixth International Conference*, 689–703, Cambridge, MA: MIT Press.

Keber, C. and M. G. Schuster (2001). "Evolutionary Computation and the Vega Risk of American Put Options," *IEEE Transactions on Neural Networks*, 12(4), 704–715.

LeBaron, B. (2000). "Agent Based Computational Finance: Suggested Reading and Early Research," *Journal of Economic Dynamics and Control*, 24, 679–702.

Leinweber, D. and R. Arnott (1995). "Quantitative and Computational Innovation in Investment Management," *Journal of Portfolio Management*, 21(2), 8–15.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer.

Neely, C. J. and P. A. Weller (1999), "Technical Trading Rules in the European Monetary System," *Journal of International Money and Finance*, 18(3), 429–458.

Neely, C. J., P. A. Weller, and R. Dittmar (1997). "Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach," *Journal of Financial and Quantitative Analysis*, 32(4), 405–426.

Nikolaev, N. I. and H. Iba (2000). "Inductive Genetic Programming of Polynomial Learning Networks," in Yao X. (ed.), *Proceedings of the IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, 158–167. IEEE Press.

Noe, T. H. and L. Pi (2000). "Learning Dynamics, Genetic Algorithms, and Corporate Takeovers," *Journal of Economic Dynamics and Control*, 24(2), 189–217.

Novkovic, S. (1998). "A Genetic Algorithm Simulation of a Transition Economy: An Application to Insider-Privatization in Croatia," *Computational Economics*, 11(3), 221–243.

Palmer, R. G., W. B. Arthur, J. H. Holland, B. LeBaron, and P. Tayler (1994). "Artificial Economic Life: A Simple Model of a Stockmarket," *Physica D*, 75, 264–274.

Pereira, R. (1996). "Selecting Parameters for Technical Trading Rules Using Genetic Algorithms," *Journal of Applied Finance and Investment*, 1(3), July/August 27–34.

Pereira, R. (2002). "Forecasting Ability But No Profitability: An Empirical Evaluation of Genetic Algorithm-Optimized Technical Trading Rules," in S.-H. Chen (ed.), *Evolutionary Computation in Economics and Finance*, 287–309. Physica-Verlag.

Riechmann, T. (1999). "Learning and Behavioural Stability: An Economic Interpretation of Genetic Algorithms," *Journal of Evolutionary Economics*, 9(2), 225–242.

Riechmann, T. (2001). "Genetic Algorithm Learning and Economic Evolution," in Chen, S.-H (ed.), *Evolutionary Computation in Economics and Finance*, 45–59. Physica-Verlag.

Sandroni, A. (2000). "Do Markets Favor Agents Able to Make Accurate Prediction?" *Econometrica*, 68(6), 1303–1341.

Sciubba, E. (1999). "The Evolution of Portfolio Rules and the Capital Asset Pricing Model," DAE Working Paper No. 9909, University of Cambridge.

Smith, S. N. (1998). "Trading Applications of Genetic Programming," *Financial Engineering News*, 2(6).

Szeto, K. Y. and P. X. Luo (1999). "Self-Organizing Behavior in Genetic Algorithm for the Forecasting of Financial Time Series," *Proceeding of the International Conference on Forecasting Financial Markets*, FFM99, CD-Rom.

Tay, N. and S. Linn (2001). "Fuzzy Inductive Reasoning, Expectation Formation and the Behavior of Security Prices," *Journal of Economic Dynamics and Control*, 25, 321–361.

Tesfatsion, L. (2001). "Introduction to the Special Issue on Agent-Based Computational Economics," *Journal of Economic Dynamics and Control*, 25, 281–293.

Vriend, N. (2000). "An Illustration of the Essential Difference between Individual and Social Learning, and Its Consequence for Computational Analysis," *Journal of Economic Dynamics and Control*, 24(1), 1–19.

Vriend, N. (2001). "On Two Types of GA-Learning," in Chen, S.-H. (ed.) *Evolutionary Computation in Economics and Finance*, 233–243, Heidelberg: Physica-Verlag.

Wang, J. (2000). "Trading and Hedging in S&P 500 Spot and Futures Markets Using Genetic Programming," *Journal of Futures Markets*, 20(10), 911–942.

Yeh, C.-H and Chen S.-H. (2001). "Market Diversity and Market Efficiency: The Approach Based on Genetic Programming," *Journal of Artificial Simulation of Adaptive Behavior* (**AISB Journal**), 1(1).