USING GENETIC ALGORITHMS
TO FIND TECHNICAL TRADING RULES

by

Franklin Allen
Risto Karjalainen

20-93

RODNEY L. WHITE CENTER FOR FINANCIAL RESEARCH
The Wharton School
University of Pennsylvania
Philadelphia, PA 19104-6367

Using Genetic Algorithms to Find Technical Trading Rules

Franklin Allen and Risto Karjalainen[†]
Wharton School
University of Pennsylvania
December 22, 1993

In this paper, a genetic algorithm is used to find technical trading rules for Standard and Poor's Composite Stock Index in 1963–89. Compared to a simple buy-and-hold strategy, these trading rules lead to positive excess returns in the out-of-sample test period of 1970–89. In addition, the rules appear to reduce the variability of the returns. The results are compared to benchmark models of a random walk, an autoregressive model, and a GARCH–AR model. Conventional statistical tests and bootstrapping simulations are carried out to study the robustness of the results. It is found that the excess returns are both statistically and economically significant, even when transaction costs are taken into account.

[†]Department of Finance and Operations and Information Management, respectively. Helpful comments and suggestions were made by Adam Dunsby, Steven Kimbrough, Paul Kleindorfer, Michele Kreisler, James Laing, George Mailath, and seminar audiences at Wharton School and at Santa Fe Institute. Financial support from the National Science Foundation is gratefully acknowledged by the first author.

1. Introduction

One aspect of stock markets that has intrigued investors for many years is whether there exist technical trading rules based on patterns in prices which can be relied on to make money. Although technical analysis has been widely used among practitioners for many years, academic opinion on this issue has traditionally been almost unanimous that such rules did not exist and technical analysis was not useful. The available empirical evidence strongly suggested markets were weak form efficient and reflected all the information in past prices. Tests of the profitability of simple trading rules indicated these could not make money.

In a recent paper, Brock, Lakonishok and LeBaron (1992) (henceforth BLL) have provided evidence which suggests that the traditional academic conclusions concerning technical analysis may be premature. They studied some commonly used trading rules and found that these had significant forecasting ability for the Dow Jones index from 1897 to 1986. The results were not consistent with some simple time series models and could not be explained by conditional heteroskedasticity. LeBaron (1991) has replicated and extended these results in foreign exchange markets.

How should BLL's results be interpreted? One issue in the debate on market efficiency is whether markets should be expected to be efficient from a theoretical point of view. Grossman and Stiglitz (1980) have suggested that the traditional interpretation of market efficiency is flawed. If prices fully reflect information then it cannot be worthwhile for investors to expend resources to gather it; they are better off to simply deduce it from prices. But if nobody gathers costly information it cannot be reflected in prices. Grossman and Stiglitz argued that, in equilibrium, prices will reflect costly information imperfectly and this will enable investors to cover their costs of gathering information. One interpretation of technical analysis is that it is a costly form of gathering information. Grossman and Stiglitz's argument suggests that it would not be surprising if tests which ignore such costs found technical trading rules to be profitable. In this view, BLL's results are consistent with market efficiency.

BLL studied technical trading rules which practitioners have used for extensive periods of time. Rather than taking such rules as given, we use a genetic algorithm (discussed in detail below) to develop them. The employed algorithm provides a way to quickly search extremely large rule spaces, allowing a multitude of potential rules to be tested in a practical way. The algorithm provides a way to formalize the trading rules in a symbolic form, as opposed to techniques such as neural networks which are limited to a numerical representation.

The genetic algorithm is applied to the evolution of trading rules for Standard & Poor's Composite Index (S&P 500) in 1963–89. The rules are tested out of sample and are evaluated similarly to BLL. They are compared to a buy-and-hold strategy, as well as to benchmark models of a random walk, an autoregressive model, and a GARCH–AR model. Conventional statistical tests and bootstrapping simulations are carried out to study the robustness of the results. It is found that the excess returns are both statistically and economically significant, even taking transaction costs into account. In addition to the techniques used by BLL, tests of market timing ability developed by Henriksson and Merton (1981) and Cumby and Modest (1987) are also utilized. The results are consistent with the rules having timing ability.

The paper is organized as follows. Section 2 describes technical analysis and previous trading rule tests. Section 3 discusses genetic algorithms. Section 4 shows how genetic algorithms can be used to find trading rules and evaluates the rules found in this way. Finally, Section 5 contains concluding remarks.

2. Technical Analysis

Technical analysis refers to the use of past prices, trading volume, and other variables to forecast future price changes. Despite academic skepticism, technical analysis continues to be widely used in practice[1]. Examples of technical trading rules can be readily found from a large number of textbooks targeted to traders in various asset markets (see e.g. Edwards and Magee, 1992; Kaufman, 1978; Teweles, Harlow and Stone, 1974). In earlier days, technical analysis was more or less equated with "chartism", practised with the aid of pencil and paper. During the past decades, the practice of technical analysis has been transformed by the increasing use of computers both to search for profitable trading rules and to implement trading strategies.

Typically, trading rules generate buying or selling signals on the basis of summary statistics computed from price data and other publicly available information. For instance, a moving-average rule might direct an investor to buy a stock if the price rises above the 200-day moving average of past prices, and sell the stock when the price drops back below the moving average. Other rules are based on the idea that asset prices have "support" or "resistance" levels — once a price breaks through a previous local maximum, say, it continues to move in the same direction. Still more elaborate rules are often devised, taking into account the open interest (for futures), recent trading volume, etc.

Previous Trading Rule Tests

Mechanical trading rules have been used by investors at least since the turn of the century (BLL, footnote 6). Academic interest in testing specific forms of technical analysis originated in the 1960's. Early studies of technical analysis focused on relatively simple trading rules. For instance, Alexander (1961) tested a number of "filter rules", which advise a trader to buy if the price rises by a fixed percentage (5%, say), and sell if the price declines by the same percentage. Although such rules appeared to yield returns above the buy-and-hold strategy for Dow Jones and Standard & Poor's stock indices, Alexander (1964) later concluded that adjusting for transaction costs, the filter rules would not have been profitable. These conclusions were supported by the results of Fama and Blume (1966), who found no evidence of profitable filter rules for the 30 Dow Jones stocks. Cootner (1962), on the other hand, found that some of the 40-week moving average rules that he tested were profitable and resulted in a lower variance than the buy-and-hold strategy. On the whole, the studies during the 1960's provided little evidence of profitable trading rules, and led Fama (1970) to dismiss technical analysis as a futile undertaking. Subsequently, little academic effort was addressed to the topic, until a number of studies in late 1980's and early 1990's led to a revival of scholarly interest in technical analysis.

Sweeney (1988) re-examined the filter rules studied by Fama and Blume (1966), focusing on those of the 30 Dow-Jones industrial stocks that looked most promising at the earlier study. Using similar filter rules (but avoiding short positions that had performed poorly in the study of Fama and Blume), Sweeney (1988) found statistically significant excess returns over a buy-and-hold strategy. The excess returns remained positive for transaction costs obtainable by floor traders.

Lukac and Brorsen (1990) argued that there is no evidence of traders using the simple filter rules that were favored by researchers. They applied 23 technical trading systems that were being used in practice to contracts in 30 commodity futures markets. They found that most systems produced statistically significant excess returns

1. For instance, Taylor and Allen (1992) found that at least 90% of the chief foreign exchange dealers in London placed some weight on technical analysis in forming their expectations. The reliance on technical analysis was pronounced for short horizons, while the dealers paid more attention to fundamental analysis for the longer term (see also Frankel and Froot, 1990).

(returns for a filter rule of the type studied by Alexander (1961) were a little below the median). However, the results were sensitive to transaction costs and were generally inconclusive.

BLL applied a number of simple technical trading rules to a 90-year long period of daily Dow Jones data[2]. The rules they studied included moving-average rules and so-called trading range break rules (buy when the price rises above a local maximum and sell when it drops below a local minimum). They found significant excess returns (before transaction costs) over the whole period and over non-overlapping subperiods. BLL also addressed the question of whether the excess returns could be explained by plausible and popular models of equilibrium returns. The models tested included the random walk, an autoregressive model, and two different GARCH-models accounting for heteroskedasticity of returns. The bootstrapping simulations indicated that none of the models could explain the results. Furthermore, it appeared that the trading rules picked long positions when the volatility of returns was lower than the average.

LeBaron (1991) applied similar trading rules to foreign exchange data, and found evidence of excess returns that could not be explained by plausible null models. The excess returns also appeared to be economically significant. For other evidence of profitable trading rules in foreign exchange markets, see e.g. Dooley and Shafer (1983), Sweeney (1986) and Taylor (1992); for evidence of unprofitable rules, see e.g. Diebold and Nason (1990), Meese and Rogoff (1983) and Meese and Rose (1990).

Bulkley and Tonks (1989, 1992) tied the trading rule tests to the debate on the excess volatility of stock returns. They addressed the question whether price fluctuations are large enough so that buying underpriced securities and selling overpriced ones leads to an excess return above the return from a buy-and-hold policy. Bulkley and Tonks (1989) tested a rule that directed an investor to buy if the price was a fixed percentage below a trend, and sell otherwise (the percentage was optimized for the past data). Applied to U.K. stock market data for the period of 1929–1985, such a trading strategy would have yielded an annual post-tax excess return of 1.5%. Bulkley and Tonks (1992) conducted a similar test for over a century of Standard & Poor's index data, and found annual excess returns of 1.18%.

Theoretical Considerations

In the literature, two kinds of theoretical arguments have been put forth to explain why technical analysis might potentially be useful. The arguments are related to the role of technical analysts in facilitating the spreading of news in the market, and to the nonlinearity of many financial time series. These two issues are discussed in turn below.

The transmission of information through prices in competitive markets gives rise to two separate but related problems: inferring (perhaps imperfectly) the private information of more knowledgeable investors, and estimating the time when news has been incorporated into prices. Brown and Jennings (1989) studied the former problem; Treynor and Ferguson (1985) addressed the latter. Technical analysis may be a useful tool in either case, albeit for different reasons.

Brown and Jennings (1989) analyzed a three-period economy, where investors observe private information during the first two periods about the payoff received in the last period. Each investor knows when the private information is released, but they don't know the realisation of the private signals. The first period price is useful for

2. See also Goldberg and Schulmeister (1988) for a study of a wide variety of trading rules for the S&P 500 index.

inferences about the private signals because it is not perturbed by the random variation in the second period supply. Therefore, a study of both the current and the past prices dominates using the most recent price only. Technical analysis is useful in the setting of Brown and Jennings because it allows investors make a more accurate assessment of other investors' private information.

Treynor and Ferguson (1985) considered a situation where every investor eventually receives a new piece of information and everybody agrees about the implications of the news. They argued that technical analysis may be useful because it allows investors to improve their assessment of the likelihood that they have received the information before it is discounted in the market price.

At an intuitive level, technical trading rules will be profitable if they predict the movement of stock prices and invest taking this into account. This suggests there is a connection between ability to forecast and the profitability of technical trading rules. The optimal method of forecasting stock returns depends on whether the return series is linear or nonlinear. A return series is linear if the expected return can be expressed as a linear combination of the (possibly infinite) sequence of past returns. One example of a linear return generating process is a random walk where the returns are independent, though not necessarily identically distributed. The best forecasts for linear processes are provided by vector autoregressive models. Neftci (1991) has shown that technical trading rules are not useful in this context. LeBaron (1992b) has pointed out that this assumes the true parameters for the model are known.

There is empirical evidence that many financial return series are nonlinear (see e.g. Akgiray, 1989; Hinich, 1985; Hsieh, 1991; LeBaron, 1992; Scheinkman and LeBaron, 1989). Such nonlinear dependence may arise from the complex dynamics of speculative markets. Neftci (1991) has argued that technical analysis may be an informal attempt by practitioners to exploit nonlinearity of the return series. However, the link between nonlinearity and trading rule returns is not well established. LeBaron (1992b) addressed the question of whether trend–following rules based on moving averages of past prices exploit nonlinearity of returns in the foreign exchange markets. Using the simulated method of moments, he showed that linear models matching the trading rule results and exhibiting autocorrelations consistent with the data could be found. While these results may not generalize to other markets and to other kinds of trading rules, they suggest that nonlinearity of returns is neither a necessary nor a sufficient condition for profitable trading rules.

One important point is that trading rules may be profitable because the strategies involve bearing risk. The fact that returns are above the market return may simply reflect a high level of risk. In principle, it is important to try to measure the degree of risk borne and evaluate it. In practise, it is often difficult to do this because of the lack of an applicable asset pricing theory. Besides the problems involved in measuring the risk of trading rules, the problem of how to devise trading rules is a non–trivial one. Machine learning techniques such as genetic algorithms provide one practical way to develop rules for investment decisions.

3. Genetic Algorithms

Genetic algorithms comprise a class of search, adaptation, and optimization techniques based on the principles of natural evolution[3]. Genetic algorithms were developed by John Holland (1962; 1975). Other evolutionary algorithms include evolution strategies (Rechenberg, 1973; Schwefel, 1981), evolutionary programming (Fogel, Owens and Walsh, 1966), classifier systems (Holland, 1976; 1980), and genetic programming (Koza, 1992).

In an evolutionary algorithm, a population of solution candidates is maintained. The quality of each solution candidate is evaluated according to a problem-specific fitness function, which defines the environment for the evolution. New solution candidates are created by selecting relatively fit members of the population and recombining them through various operators. Specific evolutionary algorithms differ in the representation of solutions, the selection mechanism, and the details of the recombination operators.

In a genetic algorithm, solution candidates are represented as character strings from a given (often binary) alphabet. In a particular problem, a mapping between these genetic structures and the original solution space has to be developed, and a fitness function has to be defined. The fitness function measures the quality of the solution corresponding to a genetic structure. In an optimization problem, the fitness function simply computes the value of the objective function. In other problems, fitness may be determined by a co-evolutionary environment consisting of other genetic structures.

A genetic algorithm[4] starts with a population of randomly generated solution candidates. The next generation is created by recombining promising candidates. The recombination involves two parents, which are chosen at random from the population, biasing the selection probabilities in favor of the relatively fit individuals. The parents are recombined through a "crossover" operator, which splits the two genetic structures apart at randomly chosen locations, and joins a piece from each parent to create a new genetic structure. The fitness of the "offspring" is evaluated, and the offspring replaces one of the relatively unfit members of the population. New genetic structures are produced until the generation is completed. Successive generations are created in the same manner until a well-defined termination criterion is satisfied. The final population provides a collection of solution candidates, one or more of which can be applied to the original problem.

The theoretical foundation of genetic algorithms was layed out by Holland (1975). Maintaining a population of solution candidates makes the search process parallel, allowing an efficient exploration of the solution space. In addition to this explicit parallelism, genetic algorithms are implicitly parallel: The evaluation of the fitness of a specific genetic structure yields information about the quality of a very large number of "schemata", or building blocks. The algorithm automatically allocates an exponentially increasing number of trials to the best observed schemata. This leads to a favorable tradeoff between exploitation of promising directions of the search space and exploration of less frequented regions of the space. The stochastic nature of the selection and recombination operators is also important, ensuring that the algorithm is unlikely to become stuck at local optima[5].

Many of the alternative machine learning methods focus on heuristic rules that reduce the complexity of the search process. In contrast, genetic algorithms work by repeatedly generating and testing promising solution candidates. This parallel trial-and-error process is an efficient way to reduce the uncertainty about the search space

3. Although evolutionary algorithms have been inspired by natural evolution, there is little similarity between the computational algorithms and actual biological processes.

4. There are many variations of the basic genetic algorithm. The version described here employs continuous reproduction, as opposed to a more conventional approach where the whole generation is replaced by a new one. The idea of continuous reproduction was originally proposed by Holland (1975), and developed by Syswerda (1989) and Whitley (1989). Similarly to Whitley (1989), we also use rank-based selection, instead of the more common method of computing the selection probabilities on the basis of scaled fitness values.

5. Occasionally, random mutations are introduced to modify the genetic structure of the offspring. This is a further safeguard against the loss of genetic diversity and resulting premature convergence to suboptimal solutions. However, the role of mutations is relatively minor; the power of genetic algorithms stems mainly from the recombination of relatively fit solution candidates.

(Booker, Goldberg, and Holland, 1989). The parallelism is also the main difference between evolutionary algorithms and other popular machine learning paradigms such as neural networks and simulated annealing (see e.g. Koza, 1992, ch. 27). Of course, none of the alternative approaches is likely to dominate the others in all circumstances, and similar end results can often be obtained using different techniques. Using hybrids (such as using a genetic algorithm to construct a neural network) often provides a convenient way to proceed.

Evolutionary algorithms are weak methods, embodying very little problem–specific knowledge. Consequently, they are unlikely to perform better than special–purpose algorithms in well–understood domains. Evolutionary algorithms are most useful in problems that are difficult or impractical to solve through traditional methods, due to the size of the search space, non–differentiability of the objective function, the presence of multiple local optima, or non–stationarity of the environment.

Evolutionary algorithms have been applied to a large number of different problems in engineering, computer science, cognitive science, economics, management science, and other fields (for references, see Goldberg, 1989; Booker, Goldberg, and Holland, 1989). The number of practical applications has been rising steadily, especially since the late 1980's. Typical business applications involve production planning, job–shop scheduling, and other difficult combinatorial problems (for a recent list of applications, see Nissen, 1993). Genetic algorithms have also been applied to theoretical questions in economic markets by Andreoni and Miller (1990), Arthur (1992), and Rust, Palmer and Miller (1992), and to time series forecasting by Packard (1990) and Meyer and Packard (1992).

Genetic Programming

In traditional genetic algorithms, genetic structures are represented as character strings of fixed length. This representation is quite adequate for many problems, but it is restrictive when the size or the form of the solution cannot be assessed beforehand. Genetic programming developed by John Koza (1992) is an extension of genetic algorithms which partly alleviates the restrictions of the fixed–length representation of genetic structures. As it also provides a natural way to represent decision rules, it is used in this study to find technical trading rules. However, the choice of genetic programming is a matter of convenience, and not crucial to the approach taken in this paper.

In genetic programming, solution candidates are represented as hierarchical compositions of functions. In these tree–like structures, the successors of each node provide the arguments for the function identified with the node. The terminal nodes (i.e. nodes with no successors) correspond to the input data. The entire tree is also interpreted as a function, which is evaluated recursively by simply evaluating the root node of the tree. The structure of the solution candidates is not specified *a priori*. Instead, a set of functions is defined as building blocks to be recombined by the genetic algorithm.

The function set is chosen in a way appropriate to the particular problem under study. Much of the work of Koza (1992) is focused on genetic structures that include only functions of a single type. However, genetic programming possesses no inherent limitations about the types of functions, as long as a so–called "closure" property is satisfied. This property holds if all possible combinations of subtrees result in syntactically valid composite functions. Closure is needed to ensure that the recombination operator is well–defined.

As in genetic algorithms, a population of genetic structures is maintained. The initial population consists of random trees. The root node of a tree is chosen at random among functions of the type of the desired composite function. Each argument of that function is then selected among the functions of the appropriate type, proceeding recursively down the tree until a function with no arguments (a terminal node) is reached. The evolution takes place much as in the basic genetic algorithms, selecting relatively fit solution candidates to be recombined and replacing unfit individuals by the offspring.

In genetic programming, the crossover operator recombines two solution candidates by replacing a randomly selected subtree in the first parent by a subtree from the second parent[6]. If different types of functions are used within a tree, the appropriate procedure involves choosing the crossover node at random within the first parent, and then choosing the crossover node within the second parent among the nodes of the same type as the crossover node in the first tree.

Genetic programming has been applied by Koza (1992) to a diverse array of problems, ranging from symbolic integration to the evolution of ant colonies to the optimal control of a broom balanced on top of a moving cart. As one specific example illustrating the effectiveness of the algorithm, Koza (1992, ch. 8) applied genetic programming to learning the correct truth table for the so-called 6-multiplexer problem. In this problem, there are six binary inputs and one binary output. The correct logical mapping must specify the correct output for each of the 2^6 input combinations. Hence, the size of the search space is $2^{2^6} \approx 10^{19}$. Using genetic programming, no more than 160 000 individual solution candidates needed to be generated in order to find the correct solution with a 99% probability. For comparison, the best solution found in a random search over 10 million truth tables produced the correct output for only 52 out of 64 possible input combinations.

4. Finding and Evaluating Trading Rules

In this paper, genetic programming is used to learn technical trading rules for a composite stock index. The goal of the algorithm is to find decision rules that divide days into two disjoint categories, either "in" the market (earning the market rate of return) or "out" of the market (earning a risk-free rate of return). The decisions are based on past prices only. One way to apply genetic programming to finding trading rules is described below, followed by a description of the composite stock index data used and the results obtained.

Applying Genetic Programming to Finding Trading Rules

Each genetic structure represents a particular technical trading rule. A trading rule is a function that returns either a "buy" or a "sell" signal for any given price history. The trading strategy specifies the position to be taken the following day, given the current position and the trading rule signal. The trading strategy is implemented as a simple automaton, which works as follows: If the current state is "in" (a long position) and the trading rule signals "sell", switch to "out" (move out of the market). If the current state is "out", and the trading rule signals "buy", switch back to "in". In the other two cases, the current state is preserved (see Figure 1 for an illustration). This kind of trading strategy is perhaps the simplest one that can be imagined; more sophisticated ways to allocate funds between different asset classes could certainly be devised and learned using a genetic algorithm.

6. As in the basic genetic algorithms, mutations can also be used to introduce new genetic material to the population. In this study, mutations are implemented by using a randomly generated tree in place of the second parent with a small probability.

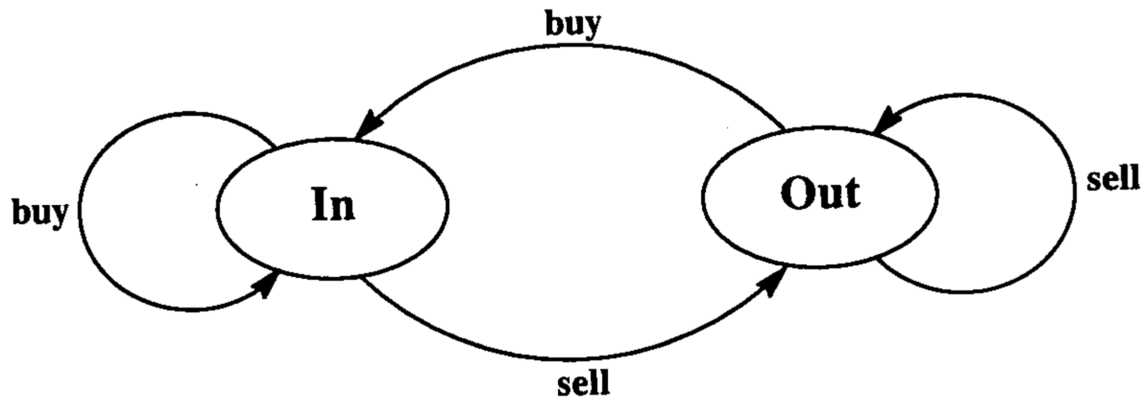

Figure 1. The trading strategy.

Building blocks for trading rules include simple functions of past price data, numerical and logical constants, and logical functions that allow the combination of low-level building blocks to more complicated expressions. The root node of each genetic structure corresponds to a boolean function; this restriction ensures that the trading strategy is well-defined.

The function set includes two kinds of functions: real and boolean. The real-valued functions include a function that computes a moving average of past prices (`"average"`) in a time window specified by an argument. There are also functions that return the local extrema of prices (`"maximum"` and `"minimum"`) during a time window of a given length[7]. Other real-valued functions include arithmetic operators (+ , - , * , ÷) and a function returning the absolute value of the difference between two real numbers (`"norm"`). Boolean functions include logical functions (`"if-then"`, `"if-then-else"`, `"and"`, `"or"`, `"not"`) and comparisons of two real numbers (`">"`, `"<"`). In addition, there are boolean constants (`"true"`, `"false"`) and real constants. The boolean constants are initialized randomly to either of the two truth values, and the real constants are initialized to values drawn from the uniform distribution between 0 and 2 when the initial population of genetic structures is created (and fixed thereafter). There is also a real-valued function (`"price"`) that returns the closing price of the current day. Finally, there is a function (`"lag"`) that causes its argument function to be applied to a price series lagged by a number of days specified by another argument.

These functions can be used to implement many commonly used technical trading rules. For instance, Figure 2 shows a 50-day moving average rule (on the left) and a simple 30-day trading range break rule (on the right). When the moving average rule is evaluated, the root node (`"<"`) first evaluates its arguments. When the first argument is evaluated, the corresponding node (`"average"`) evaluates its own single argument to find out the length of the moving average window. The corresponding terminal node (`"50.0"`) simply returns a real constant. The moving average function then computes the average of the past 50 days' prices, and returns the result to the root node. In the right-hand subtree corresponding to the second argument of the root node, the function `"price"` returns the closing price of the current day. The root node then compares the two arguments, returning

7. The choice of these building blocks is supported by the analysis of Neftci (1991), who showed that many trading rules relying on specific patterns can be expressed in terms of local maxima and minima of past prices. Moving average rules are useful as a way to model potential short-term of long-term trends. Real-valued arguments that specify the length of the time window for these functions are rounded to integers when the rules are evaluated.

a "buy" signal if the first argument is smaller than the second, and a "sell" signal otherwise. The 30–day trading range break rule is evaluated in the same recursive manner.

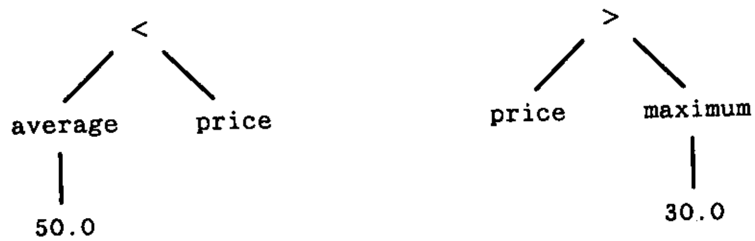

Figure 2. Genetic structures corresponding to a 50–day moving average rule (left) and a 30–day trading range break rule (right). The moving average rule returns a "buy" signal if the 50–day moving average of past prices is greater than the closing price, and a "sell" signal otherwise. The trading range break rule returns a "buy" signal if the price is greater than the local maximum of the past 30 days' prices, "sell" otherwise.

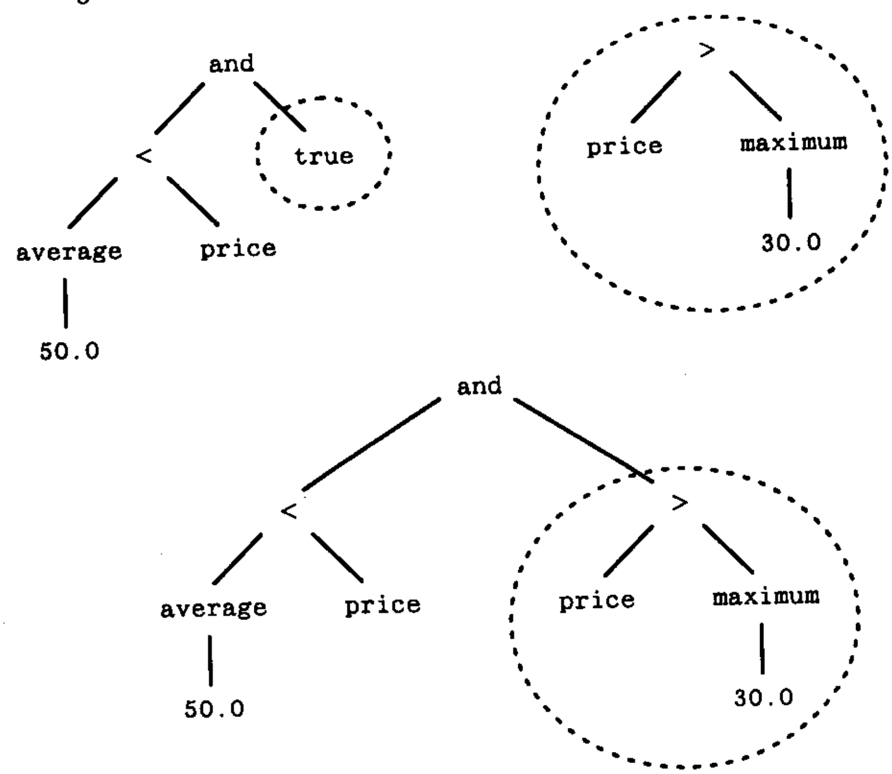

Figure 3. Illustration of the crossover operator that creates a new trading rule shown in the bottom by recombining the two rules shown in the top. In the crossover, the subtree designated by the dotted line in the first parent is replaced by the subtree designated in the second parent.

An illustration of how the genetic algorithm combines simple rules to create more complicated ones is shown in Figure 3. In this example, the two trading rules shown in the top of Figure 3 are used as the parents. In the first parent, the crossover node chosen at random corresponds to the subtree (`"true"`) designated by the dotted line. In the second parent, the crossover node coincides with the root node (">"). When the subtree in the first parent is replaced by the (sub)tree of the second parent, the trading rule shown in the bottom of Figure 3 is obtained. This new rule returns a "buy" signal if both the moving average rule and the trading range break rule are satisfied, and a "sell" signal otherwise. Proceeding in the same way, complex and subtle nonlinear rules can be created, although nothing precludes the discovery of quite elementary decision rules.

Fitness Measure

In this study, the fitness measure is based on the excess return over a buy-and-hold strategy. While this is perhaps the most obvious choice, different alternatives would obviously be useful in other situations. For instance, the fitness function might include a term that penalizes for large daily losses or for large drawdowns of wealth, according to the risk attitude of a particular investor.

The fitness of a rule is computed as the excess return over a buy-and-hold strategy during a *training period*. To evaluate the fitness of a trading rule, it is applied to each trading day to divide the days into periods "in" (earning the market return) or "out" of the market (earning a risk-free return). The continuously compounded return is then computed, and the buy-and-hold return and the transaction costs are subtracted to determine the fitness of the rule. In more detail, the simple return from a single trade (buy at date b_i, sell at s_i) is

$$\begin{aligned}\pi_i &= \frac{P_{s_i}}{P_{b_i}} \times \frac{1-c}{1+c} - 1 \\ &= \exp\left[\sum_{t=b_i+1}^{s_i} r_t\right] \times \frac{1-c}{1+c} - 1 \\ &= \exp\left[\sum_{t=b_i+1}^{s_i} r_t + \log\frac{1-c}{1+c}\right] - 1 \end{aligned} \tag{1}$$

where P_t is the closing price (or level of the composite stock index) on day t, $r_t = \log P_t - \log P_{t-1}$ is the daily continuously compounded return, and c denotes one-way transaction cost (expressed as a fraction of the price). Let T be the number of trading days, and let $r_f(t)$ denote the risk-free rate on day t. Define two indicator variables $I_b(t)$ and $I_s(t)$, equal to one if a rule signals "buy" and "sell", respectively, zero otherwise (obviously, the indicator variables satisfy the relationship $I_b(t) \times I_s(t) = 0 \ \forall t$). Lastly, let n denote the number of trades, i.e. the number of buying signals followed by a subsequent selling signal (an open position in the last day is forcibly closed). Then, the continuously compounded return for a trading rule can be computed as

$$r = \sum_{t=1}^{T} r_t\, I_b(t) + \sum_{t=1}^{T} r_f(t)\, I_s(t) + n\,\log\frac{1-c}{1+c} \tag{2}$$

and the total (simple) return is $\pi = e^{r_b} - 1$. The return for a buy-and-hold strategy (buy the first day, sell the last day) is

$$r_{bh} = \sum_{t=1}^{T} r_t + \log\frac{1-c}{1+c} \tag{3}$$

and the excess return — or the fitness — for a trading rule is given by

$$\Delta r = r - r_{bh} \tag{4}$$

As short sales can only be made on an up-tick, the implementation of simultaneous short sales for a composite stock index is rather difficult. Consequently, no short positions are considered here. Results by Sweeney (1988) suggest that large institutional investors can achieve one-way transaction costs in the range of 0.1 to 0.2 percent (at least after the middle of 1970's), and floor traders can achieve considerably lower costs (lower transaction costs could also be achieved in futures markets for the S&P 500 index). A one-way transaction cost of $c = 0.1$ percent is used below.

One issue that needs to be addressed in the design of the genetic algorithm is the possibility of overfitting the training data. The task of inferring technical trading rules relies on the assumption that there are some underlying regularities in the data (if the price changes truly are random, finding profitable technical trading rules is of course impossible). However, there are going to be patterns arising from noise, and the trick is to find trading rules that generalize beyond the training sample. The problem is common to all methods of nonlinear statistical inference, and several approaches have been proposed to avoid overfitting. These include reserving a part of the data as a validation set to test the predictions on, increasing the amount of training data, penalizing for model complexity, and minimizing the amount of information needed to describe both the model and the data (for a discussion of overfitting, see Gershenfeld and Weigend, 1993).

Although the current task is different from time series prediction (the fitness function reflects excess returns, not prediction error), the methods of avoiding overfitting nonlinear statistical models can be adapted to the current study. Here, a *selection period* immediately following the training period is reserved for validation of the inferred trading rules. Validation works as follows: After each generation, the fittest rule (based on the excess return in the training period) is applied to the selection period. If the excess return in the selection period improves upon the previously saved best rule, the new rule is saved.

Step 1

Create a random rule.
Compute the **fitness** of the rule as the excess return in the **training period** above the buy-and-hold strategy.
Do this 500 times (this is the initial **population**).

Step 2

Apply the fittest rule in the population to the **selection period** and compute the excess return.
Save this rule as the initial best rule.

Step 3

Pick two parent rules at random, using a probability distribution skewed towards the best rule.
Create a new rule by breaking the parents apart randomly and recombining the pieces (this is a **crossover**).
Compute the fitness of the rule as the excess return in the training period above the buy-and-hold strategy.
Replace one of the old rules by the new rule, using a probality distribution skewed towards the worst rule.
Do this 500 times (this is called one **generation**).

Step 4

Apply the fittest rule in the population to the selection period and compute the excess return.
If the excess return improves upon the previously best rule, save as the new best rule.
Stop if there is no improvement for 25 generations or after a total of 50 generations. Otherwise, go back to Step 3.

Table 1. One trial of the genetic algorithm used to find technical trading rules.

To summarize, the algorithm used to find trading rules is the following (see Table 1): To start with, an initial population of rules is created at random. The fitness of each trading rule is determined by applying it to the daily data for the S&P 500 index in the training period. A new generation of rules is then created by recombining parts of relatively fit rules in the population. After each generation, the best rule in the population is applied to a selection period. If the rule leads to a higher excess return than the best rule so far, the new rule is saved. The evolution is terminated when there has been no improvement in the selection period for a predetermined number of generations, or when a maximum number of generations has been reached. The best rule is then applied to the out-of-sample test period immediately following the selection period.

In this paper, a population size of 500 is used. The size of the genetic structures is limited to 100 nodes and to a maximum of 10 levels of nodes. Evolution is allowed to continue for a maximum of 50 generations, or until there is no improvement for 25 generations. One hundred independent trials are carried out with the same parameters, each trial starting from a different random population.

Data

The daily data for the Standard & Poor's Composite Index (S&P 500) from January 2, 1963 to December 29, 1989 were obtained from Center for Research in Security Prices (CRSP). The one-month risk-free rates corresponding to Treasury Bills were obtained from the same source.

Descriptive statistics indicate that the data consisting of the compounded daily returns are negatively skewed and strongly leptokurtotic[8]. The first lag of the sample autocorrelation function is significantly different from zero. Higher lags up to 5 are marginally significant, as are again lags of an order around 15 (Figure 4). In addition, the Ljung-Box-Pierce statistics are highly significant (p-value ≈ 0 for all lags up to 20), indicating that the autocorrelations are generally too high to make the hypothesis of white noise tenable. This conclusion can be confirmed by studying the sample autocorrelations of the series obtained by taking the absolute value or the square of the original returns. If the return series is a strict white noise process (i.e., subsequent daily price changes are independent and identically distributed (IID) with mean zero), then the absolute and the squared return series are strict white noise, too (Taylor, 1986). As seen from Figure 4, however, autocorrelations in these series die out very slowly. Although the customary confidence intervals may be too narrow because of non-normality, the first few autocorrelations of the squared process are of a magnitude ten times larger than the 95% confidence interval of $\pm 2/\sqrt{T}$.

It is instructive to take a look at how a simple autoregressive model filters out linear dependence, but is unable to convert the residuals to strict white noise. Figure 5 shows the sample autocorrelation function for the residuals from an AR(5) model. It can be seen that the autocorrelation structure of the residuals closely resembles strict white noise, whereas the autocorrelations of the absolute and the squared residuals remain significantly different from zero.

8. There are 6789 observations with mean = 0.0002538, standard deviation = 0.0089531, skewness = -2.4984 and kurtosis = 71.34427. Skewness and kurtosis are highly influenced by a few outliers. If the seven observations with the absolute value higher than 0.05 are excluded, skewness drops to 0.00825 and kurtosis to 5.712 (five out of the seven outliers occur during October, 1987). Excluding the 87 observations with absolute value greater than 0.025 leads to kurtosis (3.704) even closer to the normal distribution (3.0).

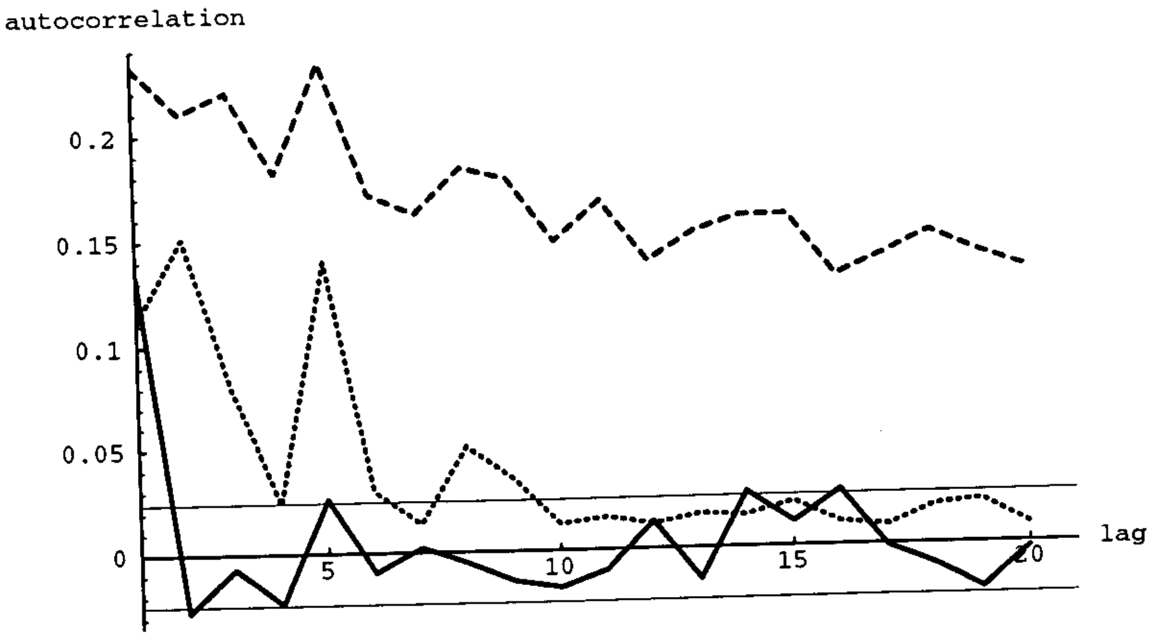

Figure 4. The sample autocorrelation function for the daily S&P 500 index from January 2, 1963 to December 29, 1989. The solid line corresponds to compounded daily returns, the dashed line to the absolute and the dotted line to the squared returns. The 95% confidence interval for the hypothesis of strict white noise is indicated by the two thin horizontal lines.

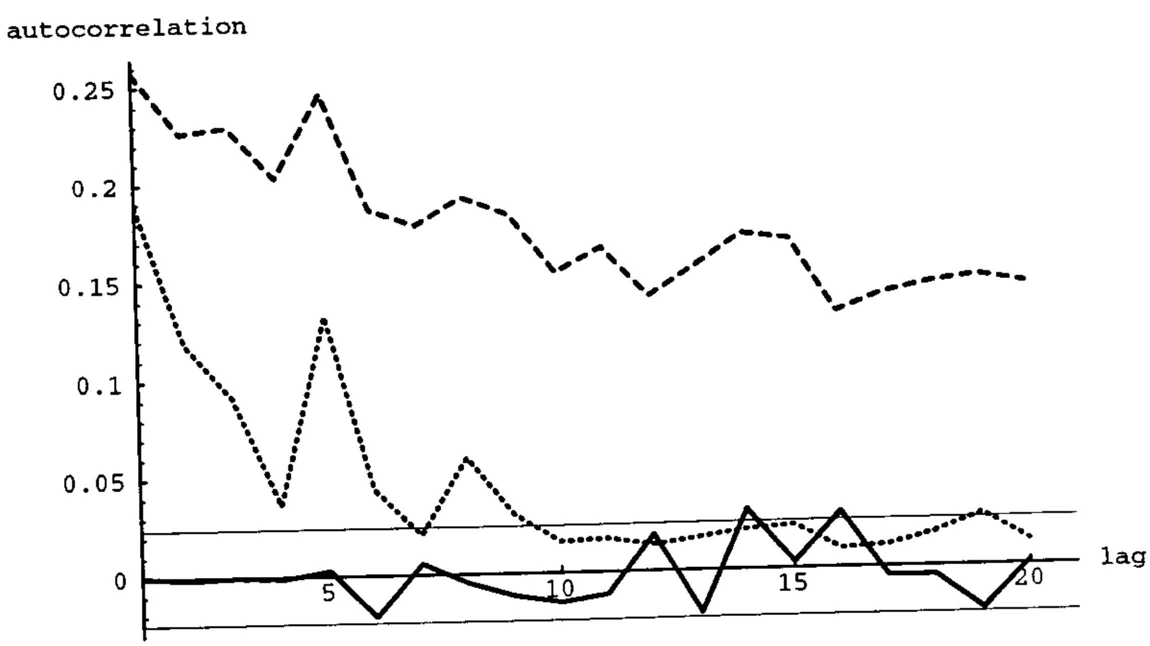

Figure 5. The sample autocorrelation function for the residuals from an AR(5) model fitted to the daily S&P 500 index in 1963–89. The solid line corresponds to compounded daily returns, the dashed line to the absolute and the dotted line to the squared residuals. The 95% confidence interval for the hypothesis of strict white noise is indicated by the two thin horizontal lines.

These observations are consistent with the results of Hsieh (1991), who applied the statistical test developed by Brock, Dechert, and Scheinkman (1987) (henceforth BDS) to several return series, including the daily S&P 500 from 1963 to 1987. The BDS test is meant to capture potential nonlinearity in the data. However, rejection of IID hypothesis by BDS test is consistent with four explanations: linear dependence, non-stationarity, low-dimensional chaos, and nonlinear stochastic processes. Hsieh (1991) strongly rejected the null model of IID returns for the S&P 500, after removing the linear correlation structure. The rejection was not attributed to non-stationarity, as the IID hypothesis was also rejected for high frequency data sampled at 15 minute intervals. A nonlinear model using locally weighted regression (LWR, see Cleveland, 1979) was estimated to study the possibility of low-order chaos, and no improvement in forecasting ability over the random walk was found. Hsieh (1991) concluded that the nonlinearity in the data was due to conditional heteroskedasticity. As an exponential GARCH model could not account for all nonlinearity, Hsieh (1991) standardized the S&P series (from April 1982 to September 1989) by dividing the daily returns by the one-step forecast of daily standard deviation from an AR(5) model, estimated using the 15-minute returns data. A first-order autoregressive model was then estimated, and the BDS test did not reject the IID hypothesis for the residuals.

Hence, the rejection of a random walk in the return series does not necessarily imply anything about market efficiency. As the results of Hsieh (1991) show, any nonlinearity can potentially be explained by conditional heteroskedasticity. Moreover, the particular nonlinear model estimated by LWR method did not improve forecasts obtained from the random walk. On the other hand, the failure to reject nonlinearity in the final heteroskedastic model applies only to a subset of data (1982–89). Furthermore, BLL studied a much longer time series, and found that models of conditional heteroskedasticity could not explain the trading rule returns.

The S&P data exclude dividends. Consequently, any seasonality in the dividends may potentially distort the results, if trading rules happen to pick periods to be out of the market when a disproportionate number of stocks go ex-dividend. In such a case, of course, there would have to be characteristic patterns preceding clustered ex-dividend days (dividend data is not part of the information set of the genetic algorithm).

Results may also be affected by infrequent trading, which may induce spurious serial correlation in stock index returns (Fisher, 1966). Typically, low-order autoregressive models have been proposed to account for the effects of nonsynchronous trading and lagged adjustment of prices (see e.g. Lo and MacKinlay, 1990). On the other hand, there is evidence that nonsynchronous trading can only explain a relatively minor part of the serial correlation of stock market indices (Atchison, Butler and Simonds, 1987). Regardless of the source of the serial correlations, however, bootstrapping simulations of low-order autoregressive models can be used to study the possibility that results are an artifact of those aspects of the market microstructure that induce <u>linear</u> dependence in stock index returns.

Results

To find trading rules for the daily S&P 500 index data[9], 100 independent trials were conducted. Years 1964–67 were used for training and years 1968–69 for validation. From each trial, one rule was saved and then tested during the years 1970–89.

In the 100 trials, 82 different rules were found[10]. The mean cumulative excess return in the test period was +0.9023, with a standard deviation of 0.1688. Put in another way, trading simultaneously with these 100 rules would have yielded an average excess return of 4.51% above the annual buy-and-hold return of 6.72% during the 20-year test period. Testing the hypothesis whether the population mean of the 100 rules is zero yields a p-value essentially equal to zero. These excess returns were computed on the basis of a one-way transaction cost equal to 0.1%. On the average, the excess returns are positive as long as the transaction cost is below 0.18%.

As the stock market crash of October 1987 is included in the test period, it is possible that the excess returns are largely due to a few winning trades. In the entire year of 1987, the average excess return of the 100 rules is indeed high (+14.9%). However, the average excess return in the period 1970–86 is equal to 6.15% (the buy-and-hold return is 5.68% in the same period). In other words, dropping the crash year from the test period would have somewhat lowered the excess returns, but cutting the test period short would have led to significantly higher excess returns than those reported above.

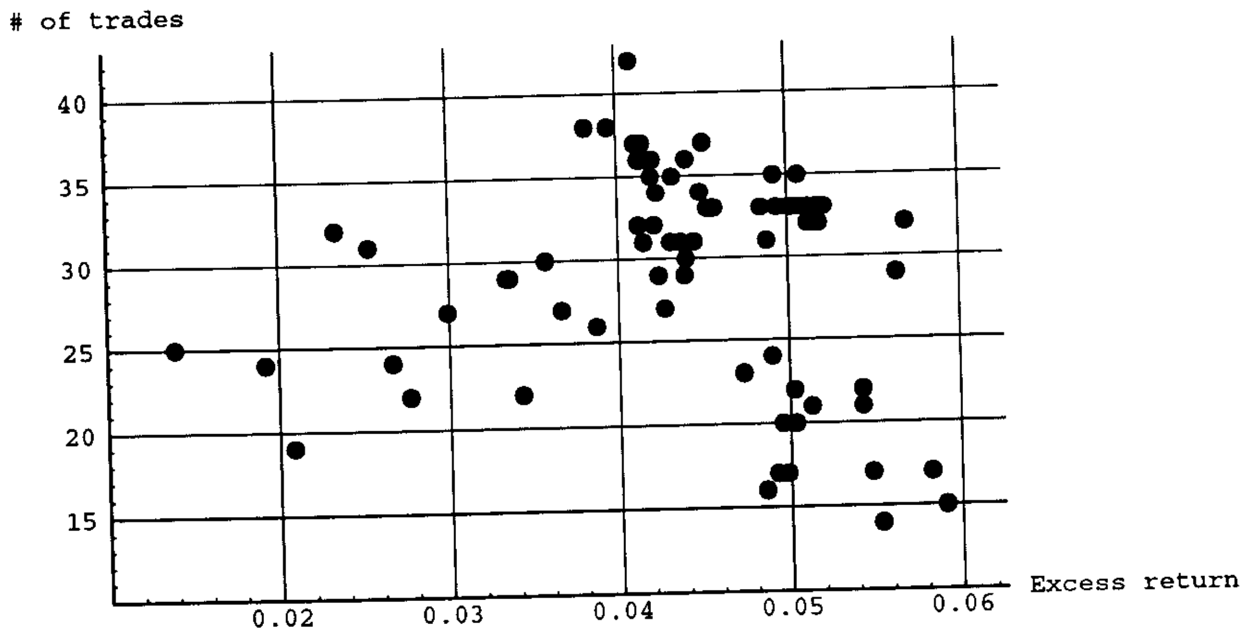

Figure 6. Yearly excess return and the average number of trades for the trading rules found by the genetic algorithm.

Figure 6 shows a scatter plot, where each point represents one of the 100 trading rules. The coordinates correspond to the yearly excess return and the average number of trades per year (each trade consists of one buy-signal and a subsequent sell-signal). It can be seen that the collection of rules is reasonably diverse: the yearly excess return ranges from +0.0139 to +0.0591, and the number of trades per year ranges from 14 to 42. The biggest

9. The S&P 500 series is clearly nonstationary, as the level of the index has risen from around 100 in the 1960's to close to 400 by the late 1980's. To compensate for the nonstationarity in a heuristic manner, the trading signals were generated from data that were normalized by dividing each day's price by a 250-day moving average. However, the excess returns (and all the test results) are based on the compounded returns corresponding to the original data.

10. To be more accurate, 82 rules with different trading patterns were found. Many of the rules with identical buy/sell signals do look quite different at first sight, indicating that the genetic structures contain a lot of redundant material (which may well be useful during the course of the evolution as raw material for the recombination operator).

cluster consists of rules that made about 30–35 trades/year, earning a yearly excess return of 4–5%. There is a smaller cluster of rules that made 15–25 trades/year with an annual excess return of 5–6%. However, some of the worst rules also made about 20–25 trades per year.

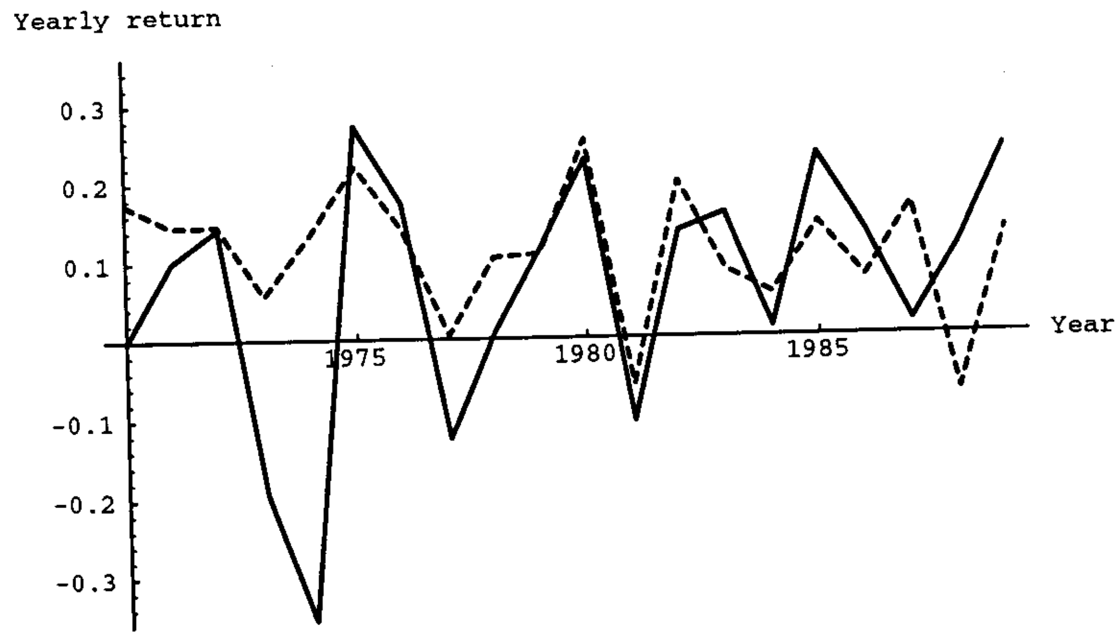

Figure 7. The average yearly return for the trading rules found by the genetic algorithm (dashed line) and for the buy-and-hold strategy (solid line) in 1970–89.

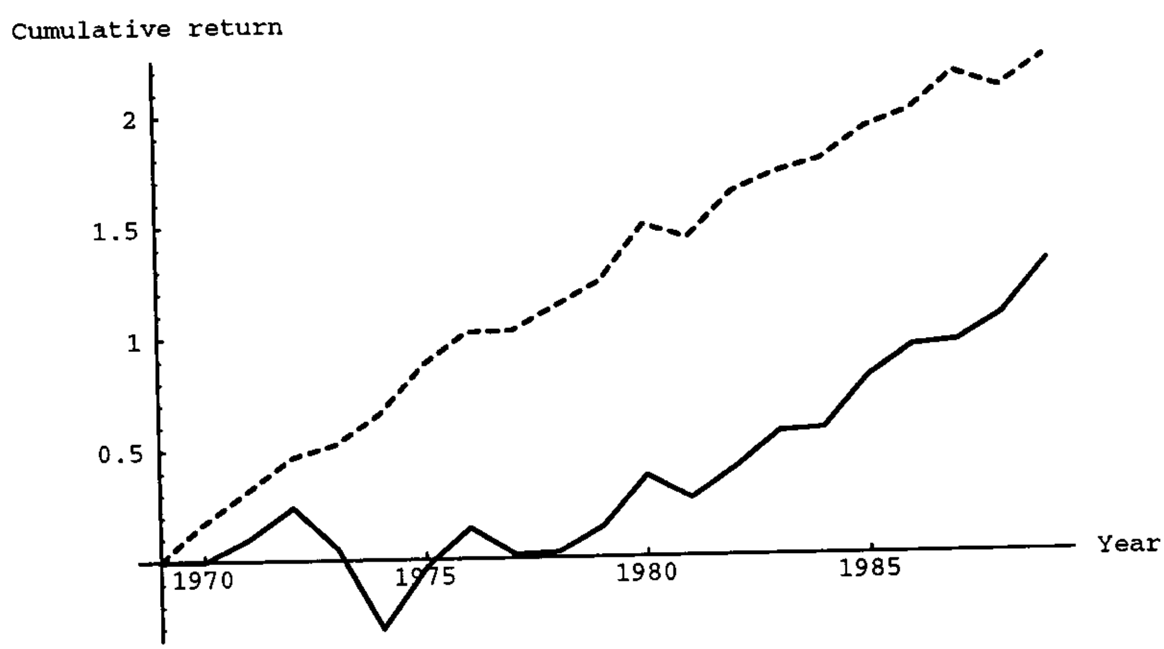

Figure 8. The cumulative return for the trading rules found by the genetic algorithm (dashed line) and for the buy-and-hold strategy (solid line) in 1970–89.

Figure 7 shows a comparison of the average yearly return for the 100 rules to the buy-and-hold return. The graph indicates that the trading rules tended to underperform the market in good years, but yielded positive excess returns in bad years. Overall, the variability of returns across the years is smaller than for a buy-and-hold strategy.

Figure 8 shows the average cumulative return for trading rules vs. the buy-and-hold return. It is evident that much of the excess return was accumulated during the 1970's. In the 1980's, the trading rules would not even have matched the market rate of return. These observations raise the possibility that the price patterns captured by the rules are no longer present in the latter part of the data. On the other hand, it should be kept in mind that the 1980's was a rising market on the whole, making it difficult for any kind of market timing rules to outperform the buy-and-hold strategy.

To study the trading rules in more detail, a subset of ten rules was selected. These rules were chosen by ranking the rules in descending order according to the excess return in the selection period of 1968–69, and retaining the rules 1,11,21,... in this ranking (i.e. rule 1 had the highest excess return in the selection period). The average excess return in the test period for the ten rules was +0.0436, ranging from +0.0208 to +0.0553. The number of trades ranges from 14 to 37 per year. These rules (and the corresponding test results) are quite representative of the 100 trials.

Table 2. Test results for trading rules found by the genetic algorithm. The second column shows the average yearly excess return above the buy-and-hold strategy in 1970–89. Daily returns during "in" and "out" periods are denoted by r_b and r_s, respectively, and the number of days during these periods is denoted by N_b and N_s. T-statistics are given in parentheses.

rule	excess	N_b	r_b	(t)	σ	N_s	r_s	(t)	σ	r_b-r_s	(t)
1	+0.0432	2436	+0.001278	(+4.161)	0.008854	2618	–0.000675	(-3.966)	0.010624	+0.001953	(+7.039)
2	+0.0512	2579	+0.001241	(+4.087)	0.008849	2475	–0.000750	(–4.201)	0.010718	+0.001991	(+7.178)
3	+0.0449	2742	+0.001149	(+3.777)	0.008875	2312	–0.000781	(–4.231)	0.010818	+0.001930	(+6.935)
4	+0.0453	2368	+0.001291	(+4.173)	0.008640	2686	–0.000637	(–3.837)	0.010739	+0.001928	(+6.937)
5	+0.0412	2613	+0.001137	(+3.665)	0.008890	2441	–0.000666	(–3.835)	0.010722	+0.001802	(+6.495)
6	+0.0411	2572	+0.001227	(+4.022)	0.008896	2482	–0.000729	(–4.119)	0.010676	+0.001956	(+7.051)
7	+0.0427	2764	+0.001031	(+3.279)	0.008866	2290	–0.000657	(–3.716)	0.010866	+0.001687	(+6.058)
8	+0.0208	2762	+0.000759	(+2.112)	0.008931	2292	–0.000327	(–2.391)	0.010843	+0.001086	(+3.900)
9	+0.0553	4139	+0.000686	(+2.033)	0.008776	915	–0.001634	(–5.366)	0.013570	+0.002321	(+6.444)
10	+0.0503	4149	+0.000717	(+2.182)	0.008865	905	–0.001800	(–5.807)	0.013320	+0.002517	(+6.960)
avg	+0.0436	2912	+0.001052		0.008844	2142	–0.000866		0.011290	+0.001917	

In 1970–89, the average daily return for 5054 trading days was +0.000266 with a standard deviation of 0.009858. Table 2 presents the results of statistical tests of daily returns for the ten rules. It can be seen that the average daily return during "in" periods is significantly higher than the unconditional return, while the return during "out" periods is lower than the unconditional return. The difference between the daily return in the two periods is positive at any reasonable signifigance level (the t-statistic for the difference between the "buy" and the unconditional return is $t = \frac{r_b - r}{s\sqrt{\frac{1}{N_b} + \frac{1}{N}}}$, where s^2 is the sample variance; the other two statistics for the "sell" mean and the difference between the two means are defined in an analoguous manner). In addition, the standard deviation during "in" periods is smaller than during "out" periods for each of the ten rules. The excess returns do not seem to be due to increased riskiness, although this point is addressed in more detail through bootstrapping simulations below.

Although the pattern of returns is mostly similar across the rules in Table 2, rules 9 and 10 are somewhat different from the rest. A total of 27 of the 100 trials converged to rules with trading patterns identical or very similar to either of these two rules. These rules are interesting because they make very few trades in rising markets, but signal frequent changes of position when prices are predominantly falling. In the subperiod of 1970–79, rules 9 and 10 would have earned an excess return of 9.63% and 7.48% above the annual buy-and-hold return of 1.57%, respectively. In 1980–89, the yearly excess returns would have been 1.44% and 2.59% above the buy-and-hold return of 11.84%. Over the entire test period, the break-even one-way transaction cost for these rules is 0.28% and 0.22%, respectively. As these rules are among those with the lowest excess return in the selection period, these findings can be interpreted as evidence that most of the other rules have to some extent been overfitted to the training data after all. In other words, the trading rule results in this paper probably give a conservative estimate of the returns at least theoretically obtainable using a similar methodology.

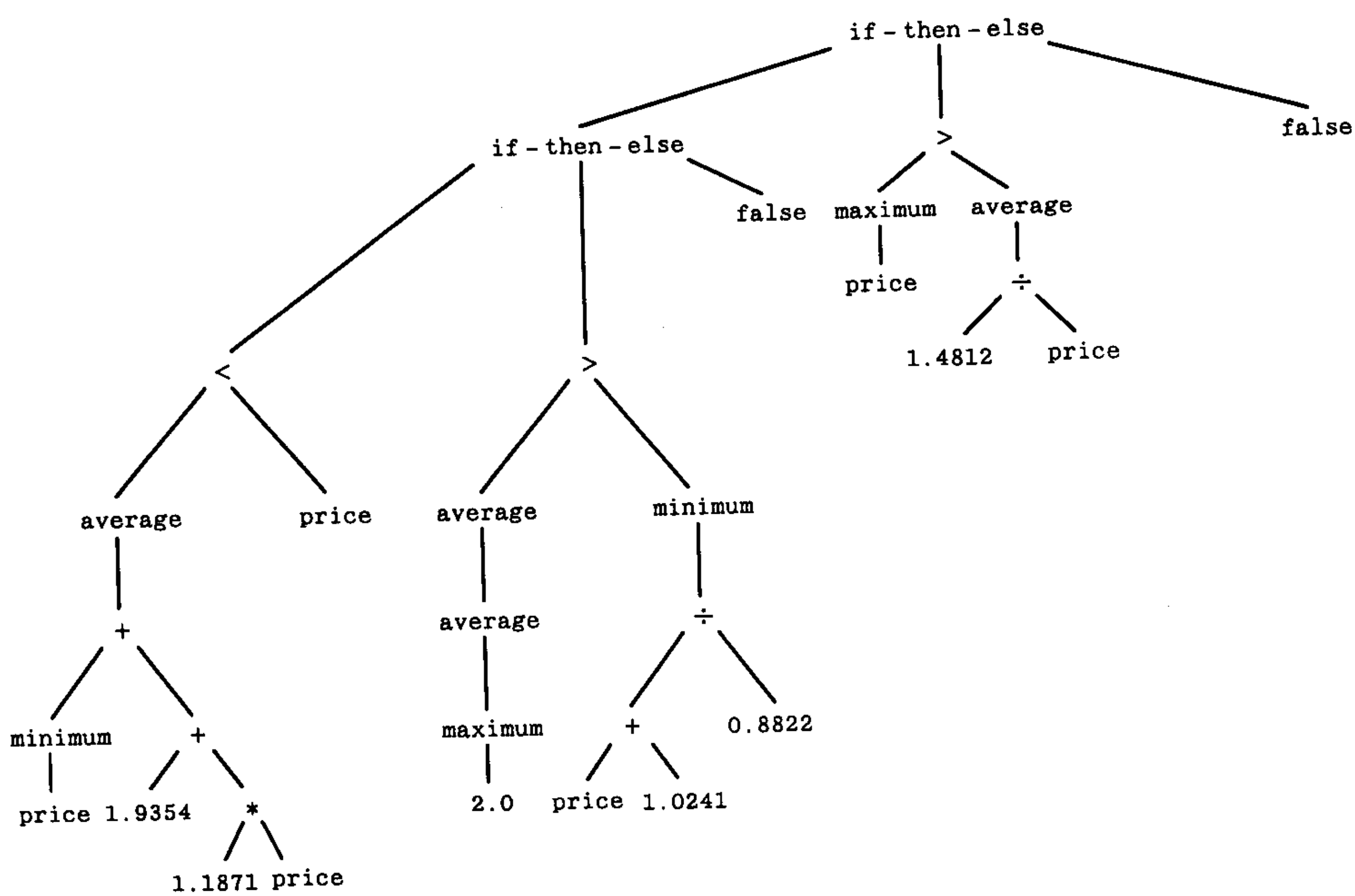

Figure 9. An example of the trading rules found by the genetic algorithm (rule 4 in Table 2).

There are two general characteristics of the trading rules found by the genetic algorithm: the representation is often highly redundant, and the rules often work in unexpected ways that a human analyst would hardly have thought of. Rule 4 in Table 2 is a typical example. The genetic structure consisting of 46 nodes is rather unwieldy to write out, but it can be simplified to an equivalent rule shown in Figure 9.

This rule works as follows: First, the leftmost subtree ("average") computes either a 4-day or a 5-day moving average of the past prices, depending on the current price history[11]. If the moving average is less than the closing price, then a shorter 1-2 day moving average is compared to the minimum of the past 3 days' prices. If the moving average is greater of the two, then the middle subtree (">") is evaluated. That subtree returns either a "buy" or a "sell" signal, depending on whether the maximum of 1-2 days' price is greater or smaller than the average of the past two days' prices. In all other cases, a "sell" signal is returned.

The rule in Figure 9 also illustrates how real-valued arguments are created in genetic programming. Although the initial population only includes real constants in the range of 0 to 2, these constants and other functions are combined by the algorithm to find additional numerical arguments that are needed to construct fit decision rules.

A simpler example is given by rule 7 (Figure 10). This rule is usually equivalent to a 2-3 moving-average rule. Sometimes, however, the length of the moving average window increases by several orders of magnitude, depending on the price history in a complex manner.

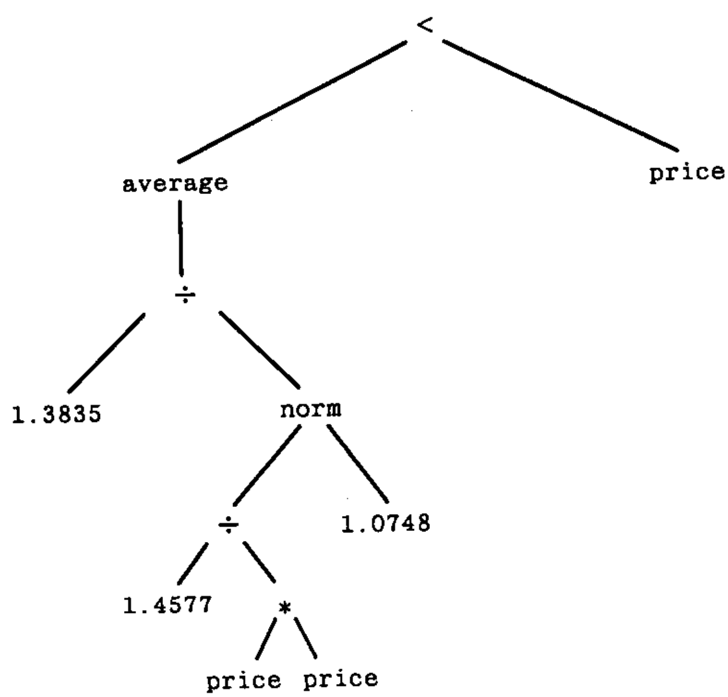

Figure 10. An example of the trading rules found by the genetic algorithm (rule 7 in Table 2).

11. Recall that the price series is normalized so that prices fluctuate around 1.0 (see footnote 9).

The trading rule results can be compared to the study by BLL. They tested different kinds of simple technical trading rules, including variable- or fixed-length moving-average rules and trading range break rules. Of those, the variable-length moving average rules can be composed from our building blocks in a rather straightforward manner. These rules generate a "buy" signal when a short-term moving average exceeds a long-term moving average by a pre-specified percentage, and a "sell" signal when the short moving average drops back below the long one (the band is either 0% or 1%). Table II of BLL presents results for ten such rules for the Dow Jones index. When the same ten rules were applied to the S&P 500 index in 1970–89, the average yearly excess return was 2.19%, taking the transaction cost of 0.1% into account (Table 3). Overall, these results corroborate the findings of BLL. It also appears that the trading rules found by the genetic algorithm are more profitable than the simple variable-length moving average rules.

Table 3. Test results for the variable-length moving average rules tested in Table II of Brock, Lakonishok, and LeBaron (1992, p. 1739). Rules are identified as (short, long, band), where short and long specify the length of the time windows, and band is the threshold percentage before a buy/sell signal is generated. The second column shows the average yearly excess return above the buy-and-hold strategy in 1970–89. Daily returns during "in" and "out" periods are denoted by r_b *and* r_s*, respectively, and the number of days during these periods is denoted by* N_b *and* N_s*. T-statistics are given in parentheses.*

rule	excess	N_b	r_b	(t)	σ	N_s	r_s	(t)	σ	$r_b - r_s$	(t)
(1,50,0)	+0.0163	2965	+0.000513	(+1.083)	0.008398	2089	-0.000084	(-1.367)	0.011612	+0.000597	(+2.122)
(1,50,1)	+0.0126	2455	+0.000555	(+1.190)	0.008536	2599	-0.000006	(-1.146)	0.010956	+0.000561	(+2.023)
(1,150,0)	+0.0268	3104	+0.000518	(+1.122)	0.008141	1950	-0.000135	(-1.527)	0.012090	+0.000654	(+2.294)
(1,150,1)	+0.0310	2841	+0.000573	(+1.327)	0.008187	2213	-0.000128	(-1.567)	0.011648	+0.000701	(+2.506)
(5,150,0)	+0.0168	3087	+0.000434	(+0.745)	0.008217	1967	+0.000003	(-1.006)	0.011986	+0.000431	(+1.517)
(5,150,1)	+0.0238	2834	+0.000499	(+1.005)	0.008230	2220	-0.000031	(-1.182)	0.011605	+0.000529	(+1.894)
(1,200,0)	+0.0203	3246	+0.000458	(+0.864)	0.008082	1808	-0.000078	(-1.273)	0.012421	+0.000536	(+1.851)
(1,200,1)	+0.0265	3047	+0.000509	(+1.072)	0.008114	2007	-0.000102	(-1.415)	0.012025	+0.000610	(+2.154)
(2,200,0)	+0.0243	3249	+0.000469	(+0.915)	0.008130	1805	-0.000099	(-1.350)	0.012369	+0.000568	(+1.962)
(2,200,1)	+0.0205	3040	+0.000463	(+0.872)	0.008172	2014	-0.000031	(-1.146)	0.011957	+0.000495	(+1.747)
average	+0.0219	2987	+0.000499		0.008221	2067	-0.000069		0.011867	+0.000568	

Bootstrapping

Results in Table 2 indicate that trading rules yield statistically significant excess returns. However, the student-t test statistics are derived under the assumption of normally distributed returns, which is scarcely supported by the descriptive statistics for the data. Following the lead of BLL, we apply the so-called bootstrapping methodology, which allows us to test specific null models, assuming only IID innovations (Efron, 1979; Efron and Tibshirani, 1986; Hall, 1992; Leger, Politis and Romano, 1992). In addition, bootstrapping provides a way to examine the riskiness of trading rules in more detail.

In bootstrapping, a hypothesized ("null") model is fitted to the data. A large number of simulated data sets are then obtained by generating time series according to the null model. Residuals from the null model are resampled with replacement and substituted for innovations in the simulated data sets. If the null model is true, the simulated data sets retain all the statistical properties of the original data, while all serial dependency (beyond any imposed by the model) is lost.

In bootstrapping simulations below, the trading rules are applied to each of the B simulated price series

$$P^*_{b,t} = P^*_{b,t-1} \exp(r^*_{b,t}), \quad t = 1, \ldots, T, \quad b = 1, \ldots, B \tag{5}$$

where the return series $r^*_b = \{r^*_{b,1}, \ldots, r^*_{b,T}\}$ is generated according to the null model, and $P^*_{b,0}$ is the price on the trading day immediately preceding the test period. For each r^*_b, $b = 1, \ldots, B$, a *resample* $e^*_b = \{e^*_{b,1}, \ldots, e^*_{b,T}\}$ of the original residuals are used. A resample is obtained by drawing T items with replacement from the residuals $e = \{e_1, \ldots, e_T\}$ from the null model. If the null model is true, each resample is an unordered collection of IID random variables, and the simulated price series replicate the statistical properties of the original time series. Consequently, the trading rule results should be similar between the original data and the bootstrapped price series. In particular, each simulated data set is equally likely to produce a statistic above or below of that obtained from the original data. Simulated p-values can be obtained by recording the fraction of times the value of the target statistic exceeds the original results[12].

Three different null models were tested: a random walk with drift, an autoregressive AR(1) model, and a GARCH(1,1)-AR(2) model. The test period from January 2, 1970 to December 29, 1989 includes $T = 5054$ data points, and $B = 1000$ simulated data sets were generated for each null model. Each of the ten test rules of Table 2 was applied to each of the simulated time series. The same statistics as in Table 2 were computed, with the addition of a statistic recording the number of times standard deviation during "in" periods was greater than during "out" periods.

In the case of a random walk with drift, the null model is

$$r_t = e_t \tag{6}$$

where the error terms $\{e_1, \ldots, e_T\}$ are IID random variables with non-zero mean. The simulated return series r^* were obtained by simply resampling the daily returns.

The autoregressive model is

$$r_t - \mu = \sum_{i=1}^{k} b_i \, (r_{t-i} - \mu) + e_t \tag{7}$$

where k is the order of the AR(k) process, and e_t are the IID residuals. Although Akaike's (1974) information criterion points to an AR(2) model, the improvement from AR(1) is negligible, and the coefficient of the second autoregressive term is only marginally significant. As an AR(1) model also facilititates a comparison of the results to BLL, it is used in the simulations below. Coefficients of the model were estimated through ordinary least squares (Table 4a), computing heteroskedasticity-consistent estimates of standard errors as suggested by White (1980) and Hsieh (1983). The scrambled residuals e^* for the bootstrapping simulations were obtained by resampling the residuals $\{e_1, \ldots, e_T\}$ given by (7).

12. See BLL (p. 1744-1745) for a justification of the bootstrap methodology for trading rule studies. BLL also tested the convergence properties of bootstrapping, and found that the simulated p-values were reliable after 500 replications.

As discussed above, any nonlinear dependence in the data may be due to conditional heteroskedasticity. To account for time-varying variance, Engle (1982) introduced the Autoregressive Conditional Heteroskedasticity (ARCH) model, which captures empirically observed volatility clustering (large (small) price changes tend to be followed by large (small) price changes of either sign). Various generalizations of the original model have been proposed during the past decade (for a review, see Bollerslev, Chou and Kroner, 1992). These include the Generalized ARCH model (GARCH) of Bollerslev (1986), which can be combined with an AR(k) model as follows:

$$r_t - \mu = \sum_{i=1}^{k} b_i \, (r_{t-i} - \mu) + e_t \qquad (8)$$

$$e_t \sim N(0, h_t) \qquad (9)$$

$$h_t = a_0 + \sum_{i=1}^{p} a_i \, e_{t-i}^2 + \sum_{i=1}^{q} \beta_i \, h_{t-i} \qquad (10)$$

While several low-order GARCH models were tried, GARCH(1,1)–AR(2) was significantly better than the alternatives by a likelihood ratio test. As opposed to the AR(1) case, the coefficient of the second autoregressive term was highly significant. Coefficients were estimated by maximum likelihood (Table 4b).

Because e_t in (8) can be rewritten as $e_t = \sqrt{h_t}\, z_t$, $z_t \sim N(0,1)$, the residual series can be expressed as

$$z_t = \frac{e_t}{\sqrt{h_t}} \qquad (11)$$

where e_t is specified by (8). Bootstrapped series were obtained from the null model defined by (8) and (10), resampling the standardized IID residuals $\{z_1, \ldots, z_T\}$ given by (11) and using the error terms $e_t^* = \sqrt{h_t^*}\, z_t^*$ in the simulations.

Table 4a. AR(1) parameters for the daily returns of the S&P 500 index in 1970–89, estimated through ordinary least squares (adjusted $R^2 = 0.0155$, log likelihood = 16212.8). T-statistics based on heteroskedasticity-consistent estimates of standard errors are given in parentheses.

μ	b_1
$0.231032 \; 10^{-3}$ (+1.636)	0.125322 (+2.553)

Table 4b. GARCH(1,1)–AR(2) parameters for the daily returns of the S&P 500 index in 1970–89, estimated through maximum likelihood (adjusted $R^2 = 0.0170$, log likelihood = 16910.0). T-statistics are given in parentheses.

μ	b_1	b_2	α_0	α_1	β_1	$h(0)$
$0.342352 \; 10^{-3}$ (+3.111)	0.166418 (+10.67)	−0.030719 (-1.952)	$0.124523 \; 10^{-5}$ (+7.188)	0.072846 (+39.66)	0.915440 (+229.5)	$0.262586 \; 10^{-4}$ (+1.263)

The choice of the AR(1) and the GARCH(1,1)–AR(2) models for bootstrapping is supported by results for the trading rules corresponding to these models (move in or out of the market depending on the sign of the one-step forecast). With zero transaction costs, a rule corresponding to the AR(1) model leads to a yearly excess return of 0.00026%, while the AR(2) rule yields an excess return of -1.183%. The GARCH(1,1)-AR(1) model gives an annual excess return of -0.033%, while the GARCH(1,1)-AR(1) model leads to a slightly positive excess return of 0.032%. With a one-way transaction cost of 0.1%, the annual excess return for these rules making 57-70 trades per year range from -11.71% to -15.36%.

Results for the random walk with drift are shown in Table 5. For the first seven rules, none of the simulated data sets yielded an annual excess return greater than for the original data. Even for the other three rules, only 5 to 10 out of the 1000 simulations led to greater excess returns. The simulated p-value for the difference in daily return between "in" and "out" periods is zero for all rules except for rule 8, for which the simulated p-value equals 0.001 (i.e. only one simulation out of the 1000 produced a larger difference than the S&P data). In 93.3 to 99.9 percent of the simulations, the difference in standard deviations between "in" and "out" periods exceeded the difference for the original data; the average p-value is 1 - 0.9597 = 0.0403. Overall, the trading rule results are not consistent with the random walk.

Table 5. Results from bootstrapping simulations of the random walk for trading rules found by the genetic algorithm. The second column specifies the average yearly excess return for the 1000 simulated data sets. The p-values in parentheses correspond to the fraction of simulated data sets for which the value of a statistic exceeds that computed for the original data.

rule	excess return	# of days in	# of days out	daily return in	daily return out	daily return in - out	standard deviation in	standard deviation out	standard deviation in - out
1	-0.0916	2415	2638	+0.000247	+0.000266	-0.000020	0.009895	0.009881	0.000014
	(0.0000)			(0.0000)	(1.0000)	(0.0000)	(0.9830)	(0.1620)	(0.9330)
2	-0.0827	2561	2492	+0.000268	+0.000268	-0.000000	0.009864	0.009862	0.000002
	(0.0000)			(0.0000)	(1.0000)	(0.0000)	(0.9820)	(0.1200)	(0.9530)
3	-0.0841	2729	2324	+0.000269	+0.000272	-0.000003	0.009869	0.009883	-0.000014
	(0.0000)			(0.0000)	(1.0000)	(0.0000)	(0.9780)	(0.1140)	(0.9600)
4	-0.0861	2352	2701	+0.000257	+0.000258	-0.000001	0.009903	0.009922	-0.000018
	(0.0000)			(0.0000)	(1.0000)	(0.0000)	(1.0000)	(0.1400)	(0.9670)
5	-0.0800	2571	2482	+0.000270	+0.000258	+0.000012	0.009864	0.009894	-0.000030
	(0.0000)			(0.0000)	(1.0000)	(0.0000)	(0.9780)	(0.1280)	(0.9410)
6	-0.0905	2558	2495	+0.000265	+0.000263	+0.000003	0.009903	0.009867	0.000037
	(0.0000)			(0.0000)	(1.0000)	(0.0000)	(0.9700)	(0.1400)	(0.9350)
7	-0.0755	2680	2373	+0.000253	+0.000259	-0.000006	0.009904	0.009902	0.000002
	(0.0000)			(0.0000)	(1.0000)	(0.0000)	(0.9810)	(0.1240)	(0.9640)
8	-0.0553	2775	2278	+0.000276	+0.000250	+0.000026	0.009888	0.009826	0.000062
	(0.0080)			(0.0080)	(0.9940)	(0.0010)	(0.9730)	(0.1090)	(0.9530)
9	-0.0405	4113	940	+0.000254	+0.000279	-0.000025	0.009907	0.009808	0.000099
	(0.0050)			(0.0010)	(1.0000)	(0.0000)	(1.0000)	(0.0100)	(0.9990)
10	-0.0517	4146	907	+0.000260	+0.000287	-0.000028	0.009878	0.009855	0.000023
	(0.0100)			(0.0010)	(1.0000)	(0.0000)	(0.9970)	(0.0220)	(0.9920)
avg	-0.0738	2890	2163	+0.000262	+0.000266	-0.000004	0.009888	0.009870	0.000018
	(0.0023)			(0.0010)	(0.9994)	(0.0001)	(0.9842)	(0.1069)	(0.9597)

Results for the bootstrapping simulations of the AR(1) model are shown in Table 6. Although the average of the simulated p–values for the yearly excess returns is 0.1006, the magnitude (0.57%) is considerably smaller than for the S&P 500 data. The average p–value for the difference in the daily returns between "in" and "out" periods is 0.0314, ranging from 0.001 to 0.068. The sign of the daily returns is correct, although the magnitude falls short of the original results. For the difference in the standard deviation, the average p–value is 0.0304. These results indicate that some of the excess returns can be explained by linear dependence, although the difference both in the daily return and in the volatility between "in" and "out" periods cannot be accounted for by the autoregressive process.

Table 6. Results from bootstrapping simulations of the AR(1) model for trading rules found by the genetic algorithm. The second column specifies the average yearly excess return for the 1000 simulated data sets. The p–values in parentheses correspond to the fraction of simulated data sets for which the value of a statistic exceeds that computed for the original data.

rule	excess return	# of days in	# of days out	daily return in	daily return out	daily return in - out	standard deviation in	standard deviation out	standard deviation in - out
1	+0.0067 (0.0910)	2416	2637	+0.001006 (0.0840)	–0.000401 (0.8970)	+0.001407 (0.0220)	0.009756 (0.9720)	0.009771 (0.1070)	–0.000015 (0.9570)
2	+0.0105 (0.0770)	2552	2501	+0.000954 (0.0810)	–0.000439 (0.9080)	+0.001393 (0.0170)	0.009765 (0.9650)	0.009747 (0.0790)	0.000017 (0.9650)
3	+0.0114 (0.1230)	2713	2340	+0.000929 (0.1320)	–0.000490 (0.9020)	+0.001419 (0.0350)	0.009780 (0.9740)	0.009753 (0.0760)	0.000027 (0.9730)
4	+0.0079 (0.0870)	2348	2705	+0.001006 (0.0810)	–0.000384 (0.8910)	+0.001390 (0.0290)	0.009755 (0.9990)	0.009784 (0.0840)	–0.000029 (0.9790)
5	+0.0088 (0.1130)	2561	2492	+0.000924 (0.1400)	–0.000395 (0.8720)	+0.001319 (0.0570)	0.009750 (0.9620)	0.009745 (0.0880)	0.000004 (0.9570)
6	+0.0085 (0.1340)	2546	2507	+0.000993 (0.1430)	–0.000472 (0.8750)	+0.001466 (0.0370)	0.009755 (0.9550)	0.009817 (0.1160)	–0.000063 (0.9480)
7	+0.0078 (0.1040)	2670	2383	+0.000837 (0.1490)	–0.000357 (0.8990)	+0.001194 (0.0470)	0.009760 (0.9860)	0.009764 (0.0830)	–0.000004 (0.9720)
8	–0.0133 (0.0880)	2616	2437	+0.000577 (0.1600)	–0.000080 (0.8190)	+0.000657 (0.0680)	0.009781 (0.9680)	0.009806 (0.0860)	–0.000025 (0.9540)
9	+0.0082 (0.1000)	4097	956	+0.000481 (0.0690)	–0.000570 (0.9990)	+0.001051 (0.0010)	0.009848 (0.9990)	0.009715 (0.0070)	0.000133 (0.9970)
10	+0.0002 (0.0890)	4112	941	+0.000499 (0.0360)	–0.000674 (0.9980)	+0.001173 (0.0010)	0.009773 (0.9940)	0.009796 (0.0120)	–0.000024 (0.9940)
avg	+0.0057 (0.1006)	2863	2190	+0.000821 (0.1075)	–0.000426 (0.9060)	+0.001247 (0.0314)	0.009772 (0.9774)	0.009770 (0.0738)	0.000002 (0.9696)

Results from bootstrapping simulations of the GARCH(1,1)-AR(2) model are given in Table 7. The simulated p-values for the excess return range from 0.075 to 0.226, with the average equal to 0.1545. The magnitude of the annual excess returns is still only about a third of that for the original data (+1.40% vs. 4.36%). The average p-value for the difference in the daily returns between "in" and "out" periods is 0.1582, ranging from 0.020 to 0.243. The model replicates the difference in standard deviation poorly, with an average p-value of 0.0368. Hence, the model cannot account for the the fact that the trading rules took long positions in days with relatively little variation in returns, but provides a better explanation of the excess returns than the AR(1) model.

Table 7. Results from bootstrapping simulations of the GARCH(1,1)-AR(2) model for trading rules found by the genetic algorithm. The second column specifies the average yearly excess return for the 1000 simulated data sets. The p-values in parentheses correspond to the fraction of simulated data sets for which the value of a statistic exceeds that computed for the original data.

rule	excess return	# of days		daily return			standard deviation		
		in	out	in	out	in - out	in	out	in - out
1	+0.0184 (0.1870)	2417	2636	+0.001106 (0.1950)	-0.000560 (0.7000)	+0.001666 (0.1910)	0.009995 (0.8030)	0.010152 (0.2430)	-0.000157 (0.9760)
2	+0.0216 (0.1270)	2558	2495	+0.001040 (0.1550)	-0.000585 (0.7580)	+0.001626 (0.1130)	0.010012 (0.7960)	0.010085 (0.2150)	-0.000073 (0.9870)
3	+0.0220 (0.2060)	2712	2341	+0.001011 (0.2270)	-0.000673 (0.6830)	+0.001684 (0.2220)	0.009940 (0.7770)	0.010015 (0.2140)	-0.000074 (0.9900)
4	+0.0190 (0.1560)	2350	2703	+0.001103 (0.1760)	-0.000535 (0.6880)	+0.001638 (0.1790)	0.009871 (0.8470)	0.010149 (0.2410)	-0.000278 (0.9790)
5	+0.0146 (0.1640)	2564	2489	+0.000972 (0.1960)	-0.000548 (0.7060)	+0.001520 (0.1740)	0.009949 (0.7970)	0.010082 (0.2330)	-0.000133 (0.9790)
6	+0.0199 (0.2260)	2554	2499	+0.001082 (0.2460)	-0.000658 (0.6260)	+0.001739 (0.2430)	0.010005 (0.7610)	0.010116 (0.2550)	-0.000111 (0.9860)
7	+0.0148 (0.1490)	2633	2420	+0.000904 (0.2410)	-0.000495 (0.7310)	+0.001399 (0.1900)	0.009883 (0.7810)	0.010261 (0.2440)	-0.000378 (0.9380)
8	-0.0050 (0.1670)	2806	2247	+0.000602 (0.1930)	-0.000231 (0.6420)	+0.000833 (0.2190)	0.009677 (0.6270)	0.010378 (0.2450)	-0.000701 (0.8680)
9	+0.0071 (0.0750)	4073	980	+0.000484 (0.0600)	-0.000824 (0.9580)	+0.001308 (0.0200)	0.009724 (0.7780)	0.010789 (0.0910)	-0.001065 (0.9690)
10	+0.0073 (0.0880)	4073	980	+0.000541 (0.0650)	-0.001050 (0.9530)	+0.001591 (0.0310)	0.009702 (0.7460)	0.010929 (0.1230)	-0.001228 (0.9480)
avg	+0.0140 (0.1545)	2874	2179	+0.000884 (0.1754)	-0.000616 (0.7445)	+0.001500 (0.1582)	0.009876 (0.7713)	0.010296 (0.2104)	-0.000420 (0.9620)

To study the effects of volatility clustering, another set of bootstrapping simulations was carried out for an AR(2) model. The average yearly excess return was -0.0116 with a p-value equal to 0.0285. As the autoregressive coefficients were very similar to the GARCH(1,1)-AR(2) model, the difference between those simulations and the results shown in Table 7 can be attributed to changing conditional variance[13]. This finding suggests that while some of the excess returns are due to short-term linear dependence, the trading rules also exploit nonlinear properties of the return series.

As observed above, the excess returns were not unduly driven by the stock market crash of 1987. However, the crash may affect the bootstrapping simulations in a more subtle way, as the few returns of large absolute value in October 1987 may have had a disproportionate influence on the parameter estimates of the null models tested above. To study this possibility, the AR(1) model and the GARCH(1,1)-AR(2) model were re-estimated for the subsample of 1970-86 (Table 8), and additional bootstrapping simulations were carried out for this period.

Table 8a. AR(1) parameters for the daily returns of the S&P 500 index in 1970-86, estimated through ordinary least squares (adjusted $R^2 = 0.0321$, log likelihood = 14315.7). T-statistics based on heteroskedasticity-consistent estimates of standard errors are given in parentheses.

μ	b_1
$0.183128\ 10^{-3}$	0.179688
(+1.388)	(+10.13)

Table 8b. GARCH(1,1)-AR(2) parameters for the daily returns of the S&P 500 index in 1970-86, estimated through maximum likelihood (adjusted $R^2 = 0.0327$, log likelihood = 14618.0). T-statistics are given in parentheses.

μ	b_1	b_2	α_0	α_1	β_1	$h(0)$
$0.267278\ 10^{-3}$	0.190853	-0.033242	$0.617638\ 10^{-5}$	0.050789	0.941469	$0.389607\ 10^{-4}$
(+2.371)	(+11.89)	(-2.072)	(+3.990)	(+11.04)	(+169.6)	(+1.706)

For the AR(1) model estimated for the period of 1970-86, the average excess return for the bootstrapped series was +0.0341 with the average p-value equal to 0.2593. The simulated p-value for the difference in the daily return between "in" and "out" periods was 0.2261. For the difference in the standard deviation, the p-value was equal to 0.0436.

For the GARCH(1,1)-AR(2) model in 1970-86, the average excess return in the bootstrap was +0.0241 with p-value equal to 0.1131. The p-value for the difference in the daily return between "in" and "out" periods (0.1563) is hardly changed from the earlier simulations, but the p-value for the difference in the standard deviation increases to 0.1302. All of these p-values are marginally significant at best, indicating that the results are to some extent sensitive to the impact of the 1987 crash on the coefficients of the null models[14].

To summarize the bootstrapping simulations, we found — similarly to BLL — that trading rules tended to take long positions when returns were positive and the volatility was relatively subdued; the rules stayed out of the market when returns were negative and relatively volatile. Overall, the simulated p-values were less significant

13. Akgiray (1989) similarly found that the autoregressive parameters changed very little between an AR(1) model and a GARCH(1,1)-AR(1) model for daily stock index returns in 1963-86.

14. Bootstrapping results for rules 9 and 10 in Table 2 are relatively robust to changes in the autoregressive parameters. In the simulations of the AR(1) model estimated for the subperiod of 1970-86, the p-values for the excess return for these two rules are 0.087 and 0.118, respectively. In the case of the GARCH(1,1)-AR(2) model, the p-values are 0.034 and 0.072, respectively.

than in the study of BLL (the time period was also much shorter), and somewhat sensitive to the impact of a few outlying observations on the coefficients of the hypothesized null models. Nevertheless, the pattern is robust across all the simulations carried out, indicating that none of the models tested provides a wholly adequate explanation of the results.

If only the variability of the daily returns is considered, then the bootstrapping results suggest that the excess returns are not due to increased riskiness. On the other hand, it is possible that other risk measures would be more appropriate, and finding a good explanation for the results is only a matter of finding an appropriate model of the return generating process. In this way, trading rule studies are useful because they spur the development of models leading to a better understanding of the behaviour of financial price series and the determinants of risk.

Evaluation Using Timing Tests

Trading rules found by the genetic algorithm can be interpreted as market-timing strategies. Therefore, statistical tests of market-timing ability provide another way to evaluate the signifigance of the trading rule results. While these tests have been initially developed to study the performance of mutual fund managers, they can easily be adapted to the current study. The alternative viewpoint provided by the timing tests is useful because the trading rule results reported in this paper differ from earlier studies by Fama and Blume (1966) and others discussed in Section 2.

Let $R_m(t)$ denote the return on a risky asset (the market portfolio) during the period t, and let $R_f(t)$ denote the return on the risk-free asset during the same period. Before each period, a market timer provides a forecast $X(t)$ that equals one if she thinks that the return on the risky asset will exceed the return on the risk-free asset during the next period, and zero otherwise. Define $p_1(t)$ as the conditional probability that a forecast is correct, given that the market return was above the risk-free return, and define $p_2(t)$ as the probability of a correct forecast, given that the market return was below the risk-free return in period t. Merton (1981) showed that the forecasts are valuable if and only if $p_1(t) + p_2(t) > 1$ (forecasts are also valuable — in a perverse way — if $p_1(t) + p_2(t) < 1$, i.e. if the predictions are consistently wrong). Valuable forecasts induce an investor to revise her prior beliefs about the distribution of returns on the market portfolio. Based on these results, Henriksson and Merton (1981) derived both nonparametric and parametric tests of market-timing ability, depending on whether forecasts are observable or not.

Cumby and Modest (1987) pointed out that the tests developed by Henriksson and Merton (1981) are based on the assumption that the conditional probability of a correct forecast is independent of the magnitude of subsequent returns. As they observed, this assumption is inappropriate in some situations, including studies of technical analysis. Predictive ability is neither a sufficient nor a necessary condition for profitable trading rules. Intuitively, it does not matter if you're mostly wrong, as long as you're right when it matters. For instance, a moving-average rule might lead to many whip-saw losses as a trader takes and quickly terminates short-term positions that place her on the wrong side of price movements. The resulting small losses are reflected in a poor predictive ability, but they may be more than offset if the rule places the trader on the right side of long, sustained price swings (it is of course an empirical question whether there are such price movements).

While the assumption of independence restricts the applicability of the timing tests of Henriksson and Merton (1981), the underlying reasoning is still valid: a forecast is valuable if it makes an investor revise her prior beliefs. Building on this insight, Cumby and Modest (1987) proposed a test based on the regression

$$R_m(t) - R_f(t) = \alpha + \beta\, X(t) + e_t \tag{12}$$

If forecasts are valuable, β in (12) should be significantly greater than zero. Cumby and Modest (1987) applied the test to a number of exchange rate advisory services, and found strong forecasting ability where the original Henriksson–Merton test failed to find any.

The regression test assumes that the expected equity premium is constant over time. In the light of the recent evidence of the predictability of stock returns, this assumption is questionable. If the forecast $X(t)$ is correlated with time–varying risk premia, the test may falsely reject the null hypothesis of no timing ability, as pointed out by Cumby and Modest (1987). However, if the forecasts $X(t)$ are based on past prices only (as is the case for the technical trading rules studied in this paper), the rejection of the null hypothesis implies that past prices can be used to forecast expected risk premia. In other words, if one attributes a significant coefficient of $X(t)$ to time–varying risk premia, one must also admit that risk premia are explained to a significant degree by past prices only, with no need to consider more fundamental indicators about the state of the underlying economy.

Table 9 presents the timing test results for the ten rules from Table 2. In these tests, $X(t) = 1$ for periods "in" the market and $X(t) = 0$ for periods "out" of the market. The coefficient β in the regression (12) is positive and highly significant for all of the trading rules (estimation was done through ordinary least squares, and t–statistics were computed on the basis of heteroskedasticity–consistent estimates of standard errors). R^2 is small for all rules, as should be expected — after all, price changes are mostly random. For comparison, the regression coefficients for rules corresponding to AR(1) and GARCH(1,1)–AR(2) forecasts are small and insignificant.

Table 9. Tests of market timing ability for the ten rules from Table 2, for the AR(1) *model and for the* GARCH(1,1)–AR(2) *model (rules* AR *and* G–AR*, respectively). The second column specifies the number of observations, the third the adjusted R^2 and the fourth the Durbin–Watson statistic. T–statistics corresponding to the estimates of α and β in (12) are given in parentheses.*

rule	# of obs.	R^2	D–W	α	(t)	β	(t)
1	1435	0.0328	1.872	–0.003199	(–4.381)	0.006872	(+7.045)
2	1333	0.0362	1.904	–0.003529	(–4.614)	0.007570	(+7.149)
3	1379	0.0337	1.867	–0.003295	(–4.695)	0.007084	(+7.009)
4	1351	0.0346	1.920	–0.003335	(–4.257)	0.007174	(+7.032)
5	1295	0.0300	1.861	–0.003266	(–4.074)	0.007058	(+6.415)
6	1491	0.0324	1.867	–0.003096	(–4.635)	0.006649	(+7.135)
7	1119	0.0294	2.054	–0.003518	(–3.663)	0.007645	(+5.913)
8	797	0.0218	2.001	–0.003033	(–2.967)	0.006921	(+4.332)
9	589	0.0322	2.123	–0.005737	(–4.920)	0.012639	(+4.539)
10	821	0.0281	2.248	–0.004439	(–4.731)	0.009711	(+4.974)
AR	2293	–0.0004	1.830	0.000045	(+0.103)	0.000205	(+0.336)
G–AR	2595	–0.0002	1.826	–0.000049	(–0.120)	0.000357	(+0.647)

There is evidence of misspecification, as the Durbin–Watson statistics are (at least marginally) significantly different from 2.0 for all rules except rule 8. The obvious first place to look for misspecification is the presence of first–order serial correlation in the residuals, as that is what the Durbin–Watson statistic was developed to measure. The results for the estimation of the model

$$\begin{cases} R_m(t) - R_f(t) = \alpha + \beta\, X(t) + u_t \\ u_t = e_t + \varrho\, u_{t-1} \end{cases} \qquad (13)$$

are shown in Table 10, and it can be seen that the first–order autocorrelation largely accounts for the serial dependence in the residuals. The estimates of α and β are essentially unchanged.

Table 10. Tests of market timing ability for the ten rules from Table 2, for the AR(1) *model and for the* GARCH(1,1)–AR(2) *model (rules* AR *and* G-AR, *respectively), accounting for first–order serial correlation of the residuals. The second column specifies the number of observations, the third the adjusted* R^2, *and the fourth the Durbin–Watson statistic. T–statistics corresponding to the estimates of* ϱ, α and β *in (13) are given in parentheses.*

rule	# of obs.	R^2	D-W	ρ	(t)	α	(t)	β	(t)
1	1435	0.0367	2.000	0.0641	(+2.431)	−0.003199	(−4.617)	0.006872	(+7.506)
2	1333	0.0384	1.997	0.0482	(+1.761)	−0.003529	(−4.701)	0.007570	(+7.496)
3	1379	0.0380	1.998	0.0666	(+2.476)	−0.003295	(−4.589)	0.007084	(+7.487)
4	1351	0.0361	1.999	0.0401	(+1.476)	−0.003335	(−4.614)	0.007174	(+7.314)
5	1295	0.0347	1.996	0.0695	(+2.508)	−0.003266	(−4.176)	0.007058	(+6.872)
6	1491	0.0367	1.997	0.0666	(+2.576)	−0.003096	(−4.676)	0.006649	(+7.622)
7	1119	0.0301	1.997	−0.0268	(−0.895)	−0.003518	(-3.843)	0.007645	(+5.751)
8	797	0.0218	1.999	−0.0006	(−0.016)	−0.003033	(-2.684)	0.006921	(+4.325)
9	589	0.0359	2.008	−0.0617	(-1.499)	−0.005737	(-2.903)	0.012637	(+4.264)
10	821	0.0430	1.985	−0.1238	(-3.573)	−0.004439	(-3.167)	0.009708	(+4.389)
AR	2293	0.0069	2.001	0.0851	(+4.088)	0.000045	(+0.103)	0.000205	(+0.366)
G-AR	2595	0.0073	1.999	0.0867	(+4.435)	−0.000049	(−0.124)	0.000357	(+0.705)

The presence of positive serial correlation indicates that there are slight persistent trends in the returns across subsequent holding periods for most of the trading rules. It appears that at least part of the trading rule returns can be attributed to positive serial correlation on a few days' horizon. In other words, trading rules pick up trends that start when they hold a position out of the market, and the rules switch to a long position early enough to be able to exploit the persistence of the price trend. However, rules 9 and 10 are an interesting exception, as there is evidence of significant negative serial correlation in returns across subsequent holding periods. These rules appear to exploit persistent long–term trends: they hold long positions most of the time (see Table 2), but switch out of the market when a price trend reverses itself.

Overall, these timing tests indicate that the trading rules found by the genetic algorithm have significant forecasting ability. Interestingly, the sum of the conditional probabilities is considerably less than one for each of the ten rules: on the average, $p_1(t) + p_2(t) = 0.448 + 0.449 = 0.896$ (averaged over the 100 trials, $p_1(t) + p_2(t) = 0.451 + 0.449 = 0.900$). These results support the intuition of Cumby and Modest (1987) that prediction ability is not a necessary condition for successful market timing.

5. Concluding Remarks

A genetic algorithm has been used to identify technical trading rules. By a number of measures, these trading rules appear to perform better out-of-sample than a simple buy-and-hold strategy. There is some evidence they perform better than the rules analyzed by BLL.

Bootstrapping simulations indicated that none of the tested null models could wholly explain the reduced variability of the returns during the periods when the trading rules held long positions in the market. As other measures of risk may be more appropriate, it appears premature to reject the possibility that the excess returns may still be due to increased riskiness. Once more sophisticated determinants of risk are identified, however, machine learning techniques like genetic algorithms can allow investors to incorporate those readily into their trading strategies.

Although profitable, the excess returns provided by the rules considered are not dramatic; the breakeven transaction costs are in the range of costs borne by large institutional investors. Moreover, a number of practical details were not taken into account, including the exact intra-day timing of the trading rule signals and the possibility of lumpy dividends. Overall, these results suggest that only large institutional investors are likely to find it profitable to use such techniques to develop technical trading rules (at least in equity markets); small investors certainly will not. In this sense the results are consistent with Grossman and Stiglitz's (1980) modified view of market efficiency that gathering information can lead to trading profits which in the long run cover costs.

There are several liquid financial markets where transaction costs are much lower than in equity markets. These include the futures markets for financial instruments (including the S&P 500 futures) and for various commodities, as well as the foreign exchange markets. Because of low transaction costs, these markets may offer more promising opportunities for machine learning techniques than a composite stock index. On the other hand, such markets may already be very efficient, making trading rules less profitable.

Finally, the genetic algorithm adopted is a relatively simple one. The parameters of the algorithm such as the length of the training period and the selection period are not necessarily optimal. More importantly, it only uses very limited information for its inputs. This suggests it should be possible to use different genetic algorithms to find better rules than those developed above.

References

Akaike, H. (1974), *"A new look at the statistical model identification"*, IEEE Transactions on Automatic Control, AC-19, 716–723

Akgiray, Vedat (1989), *"Conditional heteroscedasticity in time series of stock returns: Evidence and forecasts"*, Journal of Business, 62 (1), 55–80

Alexander, Sidney S. (1961), *"Price movements in speculative markets: Trends or random walks"*, Industrial Management Review, 2 (2), 7–26

Alexander, Sidney S. (1964), *"Price movements in speculative markets: Trends or random walks, No. 2"*, in: P. Cootner (ed.), *"The random character of stock market prices"*, MIT Press, Cambridge, Mass. 338–372

Andreoni, James, and John H. Miller (1990), *"Auctions with adaptive artificially intelligent agents"*, Santa Fe Institute, working paper 90–01–004

Arthur, W. Brian (1992), *"On learning and adaptation in the economy"*, Santa Fe Institute, working paper 92–07–038

Atchison, Michael D., Kurt C. Butler, and Richard R. Simonds (1987), *"Nonsynchronous security trading and market index autocorrelation"*, Journal of Finance, 42 (1), 111–118

Bollerslev, Tim (1986), *"Generalized autoregressive conditional heteroskedasticity"*, Journal of Econometrics, 21, 307–328

Bollerslev, Tim, Ray Y. Chou, and Kenneth F. Kroner (1992), *"ARCH modeling in finance"*, Journal of Econometrics, 52, 5–59

Booker, L.B., D.E. Goldberg and J.H. Holland (1989), *"Classifier systems and genetic algorithms"*, Artificial Intelligence, 40, 235–282

Brock, W., W. Dechert, and J. Scheinkman (1987), *"A test for independence based on the correlation dimension"*, working paper, University of Wisconsin at Madison, University of Houston, and University of Chicago

Brock, William, Josef Lakonishok, and Blake LeBaron (1992), *"Simple technical trading rules and the stochastic properties of stock returns"*, Journal of Finance, 47 (5), 1731–1764

Brown, David P. and Robert H. Jennings (1989), *"On technical analysis"*, Review of Financial Studies, 2 (4), 527–551

Bulkley, George and Ian Tonks (1989), *"Are UK stock prices excessively volatile ? Trading rules and variance bound tests"*, The Economic Journal, 99, 1083–1098

Bulkley, George and Ian Tonks (1992), *"Trading rules and excess volatility"*, Journal of Financial and Quantitative Analysis, 27 (3), 365–382

Cleveland, W.S. (1979), *"Robust locally weighted regression and smoothing scatterplots"*, Journal of the American Statistical Association, 74, 829–836

Cootner, Paul H. (1962), *"Stock prices: Random vs. systematic changes"*, Industrial Management Review, 3 (2), 24–45

Cumby, Robert E. and David M. Modest (1987), *"Testing for market timing ability"*, Journal of Financial Economics, 19, 169–189

Diebold, Francis X. and James A. Nason (1990), *"Nonparametric exchange rate prediction"*, Journal of International Economics, 28, 315–332

Dooley, Michael P. and Jeffrey R. Shafer (1983), *"Analysis of short-run exchange rate behavior: March 1973 to November 1981"*, in: D. Bigman and T. Taya (eds.), *"Exchange rate and trade instability: causes, consequences, and remedies"*, ch. 3, 43–69, Ballinger, Cambridge, Mass.

Edwards, Robert D. and John Magee (1992), *"Technical Analysis of Stock Trends"*, 6th ed., John Magee Inc., Boston, Mass.

Efron, B. (1979), *"Bootstrap methods: Another look at the Jackknife"*, Annals of Statistics, 7 (1), 1–26

Efron, B., and R. Tibshirani (1986), *"Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy"*, Statistical Science, 1 (1), 54–77

Engle, Robert F. (1982), *"Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation"*, Econometrica, 50, 987–1007

Fama, Eugene F. (1970), *"Efficient capital markets: a review of theory and empirical work"*, Journal of Finance, 25 (2), 383–417

Fama, E., and Blume, M. (1966), *"Filter rules and stock market trading"*, Security Prices: A Supplement, Journal of Business, 39 (1), 226–241

Fisher, L. (1966), *"Some new stock-market indexes"*, Journal of Business, 39, 191–225

Fogel, Lawrence J., Alwin J. Owens and Michael J. Walsh (1966), *"Artificial intelligence through simulated evolution"*, Wiley, New York

Frankel, Jeffrey A. and Kanneth A. Froot (1990), *"Chartists, fundamentalists, and trading in the foreign exchange market"*, American Economic Review, 80 (2), 181–185

Gershenfeld, Neil A. and Andreas S. Weigend (1993), *"The future of time series: Learning and understanding"*, in: Andreas S. Weigend and Neil A. Gershenfeld (eds.), *"Time series prediction: Forecasting the future and understanding the past"*, Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XV, 1–70, Addison–Wesley, Reading, Mass.

Goldberg, David E. (1989), *"Genetic algorithms in search, optimization, and machine learning"*, Addison–Wesley, Reading, Mass.

Goldberg, Michael D. and Stephan Schulmeister (1988), *"Technical analysis and stock market efficiency"*, New York University, C.V. Starr Center, working paper 88–21

Granger, Clive W.J. (1991), *"Developments in the nonlinear analysis of economic series"*, Scandinavian Journal of Economics, 93 (2), 263–276

Grossman, Sanford J. and Joseph E. Stiglitz (1980), *"On the impossibility of informationally efficient markets"*, American Economic Review, 70 (3), 393–408.

Hall, Peter (1992), *"The bootstrap and Edgeworth expansion"*, Springer–Verlag, New York

Henriksson, Roy D. and Robert C. Merton (1981), *"On market timing and investment performance. II. Statistical procedures for evaluation forecasting skills"*, Journal of Business, 54 (4), 513–533

Hinich, M.L. and Patterson, D.M. (1985), *"Evidence of nonlinearity in daily stock returns"*, Journal of Business and Economic Statistics, 3 (1), 69–77

Holland, John H. (1962), *"Outline for a logical theory of adaptive systems"*, Journal of the Association for Computing Machinery, 3, 297–314

Holland, John H. (1976), *"Adaptation"*, in: Robert Rosen and Fred M. Snell (eds.), *"Progress in theoretical biology"*, vol. 4, Academic Press, New York, 263–293

Holland, John H. (1975), *"Adaptation in natural and artificial systems"*, Univ. of Michican Press, Ann Arbor, Mich.

Holland, John H. (1980), *"Adaptive algorithms for discovering and using general patterns in growing knowledge–bases"*, International Journal of Policy Analysis and Information Systems, 4 (3), 217–240

Hsieh, David A. (1983), *"A heteroskedasticity–consistent covariance matrix estimator for time series regressions"*, Journal of Econometrics, 26, 281–290

Hsieh, David A. (1991), *"Chaos and nonlinear dynamics: Applications to financial markets"*, Journal of Finance, 46 (5), 1839–1877

Kaufman, Perry J. (1978), *"Commodity trading systems and methods"*, Wiley, New York

Koza, John R. (1992), *"Genetic programming: On the programming of computers by means of natural selection"*, MIT Press, Cambridge, Mass.

LeBaron, Blake (1991), *"Technical trading rules and regime shifts in foreign exchange"*, Santa Fe Institute, working paper 91–10–044

LeBaron, Blake (1992a), *"Nonlinear forecasts for the S&P stock index"*, in: Martin Casdagli and Stephen Eubank (eds.), *"Nonlinear modeling and forecasting"*, Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XII, 381–393, Addison–Wesley, Reading, Mass.

LeBaron, Blake (1992b), *"Do moving–average trading rule results imply nonlinearities in foreign exchange markets"*, University of Wisconsin–Madison, Social Systems Research Institute, working paper 9222

LeBaron, Blake (1993), *"Nonlinear diagnostics and simple trading rules for high–frequency foreign exchange rates"*, in: Andreas S. Weigend and Neil A. Gershenfeld (eds.), *"Time series prediction: Forecasting the future and under-*

standing the past", Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XV, 457–474, Addison-Wesley, Reading, Mass.

Leger, Christian, Dimitris N. Politis, and Joseph P. Romano (1992), *"Bootstrap technology and applications"*, Technometrics, 34 (4), 378–398

Lo, Andrew W. and A. Craig MacKinlay (1990), *"An econometric analysis of nonsynchronous trading"*, Journal of Econometrics, 45 (1,2), 181–211

Lukac, Louis P. and B. Wade Brorsen (1990), *"A comprehensive test of futures market disequilibrium"*, Financial Review, 25 (4), 593–622

Meese, R.A. and K. Rogoff (1983), *"Empirical exchange rate models of the seventies: Do they fit out of sample ?"*, Journal of International Economics, 14, 3–24

Meese, Richard A. and Andrew K. Rose (1991), *"An empirical assessment of non-linearities in models of exchange rate determination"*, Review of Economic Studies, 58, 603–619

Merton, Robert C. (1981), *"On market timing and investment performance. I. An equilibrium theory of value for market forecasts"*, Journal of Business, 54 (3), 363–406

Meyer, Thomas P. and Norman H. Packard (1992), *"Local forecasting of high-dimensional chaotic dynamics"*, in: Martin Casdagli and Stephen Eubank (eds.), *"Nonlinear modeling and forecasting"*, Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XII, 249–263, Addison-Wesley, Reading, Mass.

Neftci, Salih N. (1991), *"Naive trading rules in financial markets and Wiener-Kolmogorov prediction theory: A study of 'technical analysis'"*, Journal of Business, 64 (4), 549–571

Nissen, Volker (1993), *"Evolutionary algorithms in management science"*, Interdisziplinaeres Graduiertenkolleg, Universitaet Goettingen, Papers on Economics and Evolution, #9303

Packard, N.H. (1990), *"A genetic learning algorithm for the analysis of complex data"*, Complex Systems, 4, 543–572

Rechenberg, Ingo (1973), *"Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution"*, Frommann-Holzboog Verlag, Stuttgart

Rust, John, Richard Palmer, and John H. Miller (1992), *"Behavior of trading automata in a computerized double auction market"*, Santa Fe Institute, working paper 92-02-008

Scheinkman, Jose A. and Blake LeBaron (1989), *"Nonlinear dynamics and stock returns"*, Journal of Business, 62 (3), 311–338

Schwefel, Hans-Paul (1981), *"Numerical optimization of computer models"*, Wiley, Chichester

Sweeney, Richard J. (1986), *"Beating the foreign exchange market"*, Journal of Finance, 41 (1), 163–182

Sweeney, Richard J. (1988), *"Some new filter rule tests: Methods and results"*, Journal of Financial and Quantitative Analysis, 23 (3), 285–300

Syswerda, Gilbert (1989), *"Uniform crossover in genetic algorithms"*, in: David J. Schaffer (ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, Cal., 2–9

Taylor, Stephen J. (1986), *"Modelling financial time series"*, Wiley, New York

Taylor, Stephen J. (1992), *"Rewards available to currency futures speculators: Compensation for risk or evidence of inefficient pricing ?"*, Economic Record, supplement (special issue on futures markets), 105–116

Taylor, Mark P. and Helen Allen (1992), *"The use of technical analysis in the foreign exchange market"*, Journal of International Money and Finance, 11, 304–314

Teweles, Richard J., Charles V. Harlow, and Herbert L. Stone (1974), *"The commodity futures game"*, McGraw–Hill, New York

Treynor, Jack L. and Robert Ferguson (1985), *"In defence of technical analysis"*, Journal of Finance, 40 (3), 757–775

White, Halbert A. (1980), *"A heteroskedasticity–consistent covariance matrix estimator and a direct test for heteroskedasticity"*, Journal of Econometrics, 12, 3–21

Whitley, Darrell (1989), *"The GENITOR algorithm and selection pressure: Why rank–based allocation of reproductive trials is best"*, in: David J. Schaffer (ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, Cal., 116–121