Using Genetic Algorithms to Find Technical Trading Rules

Franklin Allen and Risto Karjalainen[†]

The Wharton School

University of Pennsylvania

Current version: October 9, 1995

A genetic algorithm is used to learn technical trading rules for Standard and Poor's composite stock index using data from 1963-69. In the out-of-sample test period 1970-1989 the rules are able to identify periods to be in the index when returns are positive and volality is low and out when the reverse is true. Compared to a simple buy-and-hold strategy, they lead to positive excess returns after transaction costs in the period of 1970-89. Using data for other periods since 1929, the rules can identify high returns and low volatility but do not lead to excess returns after transaction costs. The results are compared to benchmark models of a random walk, an autoregressive model, and a GARCH-AR model. Bootstrapping simulations indicate that none of these models of stock returns can explain the findings.

JEL classification codes: C61, G11, G14.

Keywords: Trading rules, genetic algorithms, excess returns, GARCH-models, bootstrapping.

[†] Department of Finance and Operations and Information Management, respectively. Helpful comments and suggestions were made by Adam Dunsby, Larry Fisher, Steven Kimbrough, Paul Kleindorfer, Michele Kreisler, James Laing, Josef Lakonishok, George Mailath, Bill Schwert and seminar audiences at Institutional Investor, J.P. Morgan, the NBER Asset Pricing Program, Ohio State University, the Santa Fe Institute, Rutgers University, Stanford University, University of California, Berkeley, University of Michigan, University of Pennsylvania and participants at the 1995 AFA Meeting in Washington, D.C. Financial support from the National Science Foundation is gratefully acknowledged by the first author and from the Academy of Finland by the second and from the Geewax-Terker Program in Financial Instruments by both. The second author is currently with Buchanan Partners Ltd in London. Correspondence should be addressed to Franklin Allen, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104-6367, tel. (215) 898 3629, fax: (215) 573 2207, email: allenf@wharton.upenn.edu.

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. Most tests of the profitability of simple trading rules indicated these could not make money. 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. An example of contrary evidence was Cootner (1962). 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, though, 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.

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 applied a number of simple technical trading rules to a 90-year long period of daily Dow Jones data. The rules they studied included moving-average rules (buy when the price rises above, say, the 200-day moving average and sell when it drops below) 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.

Bessembinder and Chan (1995) tested the rules used by BLL in six Asian markets, and found that the rules could predict changes in the stock market indices[1].

How should BLL's results be interpreted? One issue in the debate on market efficiency and technical analysis is why markets should be expected to be efficient from a theoretical point of view. The standard textbook argument is that competition among technical analysts ensures that past prices cannot be used to forecast future price changes. As soon as a price pattern is discovered it will self-destruct.[2] Early formal theoretical models of market efficiency did not capture this learning process. For example, Samuelson (1965) showed that if traders start by believing prices are random, competition ensures they are random. More recent contributions have focused on the information revelation aspect of technical analysis. Brown and Jennings (1989) analyzed a three-period rational expectations 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 realization of the private signals. The first period price is useful for 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 past prices dominates using the most recent price only. Technical analysis is useful in the setting of Brown and Jennings because it allows investors to 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.

In the sequel to his classic 1970 survey, Fama (1991) acknowledged Grossman and Stiglitz's (1980) critique of the traditional view of market efficiency which ignored costs of gathering information and learning and wrote (p. 1575) "A ... sensible version of the [market efficiency] hypothesis says that prices reflect information to the point where the marginal benefits of acting on information (the profits to be made) do not exceed the marginal costs." In this revised view, technical trading rules which are profitable in the sense that they allow unmeasured costs of learning to be recouped are consistent with market efficiency. We argue below that the profitability of

---

[1] See also Sweeney (1988) and Goldberg and Schulmeister (1988).

[2] See, for example, Brealey and Myers (1991), pp. 293-294.

rules can be regarded as a measure of the extent to which markets are efficient.

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 non-linear. 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 but 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 non-linear (see e.g. Akgiray, 1989; Hinich and Patterson, 1985; Hsieh, 1991; LeBaron, 1992a; Scheinkman and LeBaron, 1989). Such non-linear 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 rather than markets being inefficient. In principle, it is important to try to measure the degree of risk borne and evaluate it. In practice, it is often difficult to do this because of the lack of an applicable asset pricing theory. The approach adopted by BLL which is also used below is to use a market index as the benchmark and either be in this index or in a risk free asset. Since it is found that market volatility is lower while in the index and higher when in the risk free asset it is difficult to explain excess returns over the index as being compensation for risk.

BLL studied simple 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 learn them using a period of training data. 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. Using a genetic algorithm removes much of the arbitrariness of the rules used in BLL and other previous studies.

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 analyze the results. These tests indicate that the rules tend to be long in the market when returns are positive and volatility is low, and stay out when the returns are negative and volatility is high. In the original out-of-sample test period of 1970-89, the rules lead to excess returns above the buy-and-hold strategy. The break-even transaction costs are small, however, and probably beyond the reach of most market participants. When applied to earlier time periods between 1929 and 1963, the genetic algorithm rules still result in being long when returns are high and volatility is low and out when the reverse is true but do not lead to excess returns after transaction costs.

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

2. Genetic algorithms

2.1. Introduction to evolutionary algorithms

Genetic algorithms comprise a class of search, adaptation, and optimization techniques based on the principles of natural evolution. 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[3] 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 laid 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 trade-off between exploitation of promising directions of the search space and

[3] 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.

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[4].

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 (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). String-based genetic algorithms have been applied to finding market timing strategies based on fundamental data for stock and bond markets by Bauer (1994).

[4] 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.

2.2. 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 same type as 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[5]. 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 $2^{64} \approx 10^{19}$. 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.

3. Finding and evaluating trading rules

3.1. *Ex ante* optimal trading rules

Most of the previous trading rule studies have sought to test whether particular kinds of technical rules have forecasting ability. Using a machine learning approach to find testable trading rules sheds light on a question that is subtly different. Instead of asking whether specific rules work, we seek to find out whether in a certain sense optimal rules could be used to forecast future returns. Allowing a genetic algorithm to compose the trading rules avoids much of the arbitrariness involved in choosing which rules to test.

We should point out, however, that the rules found by the genetic algorithm are optimal in a very specific sense. The rules are the ones which maximize one particular — though intuitive — measure of fitness. In addition, some structure must be imposed on the rule space explored by the genetic algorithm. In this paper, the search space consists of logical combinations of simple rules that look at moving averages and maxima and

---

[5] 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.

minima of past prices. The choice of these rules as 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 extrema of past prices. Moving average rules are used to model potential short-term or long-term trends. Of course, using rules similar to those studied in the past poses the danger of data-snooping. We try to mitigate this problem by restricting the building blocks to simple and sensible rules which have been used in practice for several decades.

3.2. Applying genetic programming to finding trading rules

In this paper, genetic programming is used to find 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 the risk-free rate of return). 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.

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. 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[6]. 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 1 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 a "buy" (stay in) signal if the first argument (the 50-day moving average of past prices) is smaller than the second (the current price), and a "sell" (stay out) signal otherwise. The 30-day trading range break rule is evaluated in the same recursive manner.

An illustration of how the genetic algorithm combines simple rules to create more complicated ones is shown in figure 2. In this example, the two trading rules shown in the top of figure 2 are used as the parents. In the first parent, the crossover node chosen at random corresponds to the boolean subtree designated by the dotted line. In the second parent, the crossover node is selected at random among nodes of the boolean type. When the subtree in the first parent is replaced by the subtree of the second parent, the trading rule shown in the bottom of figure 2 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 non-linear rules can be created, although nothing precludes the discovery of quite elementary decision rules.

---

[6] Real-valued arguments that specify the length of the time window for these functions are rounded to integers when the rules are evaluated. It can be noted that the `if-then` building block is only well-defined because it implicitly evaluates to `false` if the condition is not satisfied.

3.3. 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. The simple return from a single trade (buy at date b_i, sell at s_i) is thus

$$\pi_i = \frac{P_{s_i}}{P_{b_i}} \times \frac{1-c}{1+c} - 1 = \exp\left[\sum_{t=b_i+1}^{s_i} r_t\right] \times \frac{1-c}{1+c} - 1 = \exp\left[\sum_{t=b_i+1}^{s_i} r_t + \log\frac{1-c}{1+c}\right] - 1 \qquad (1)$$

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

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

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

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

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

$$\Delta r = r - r_{bh} \qquad (4)$$

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 non-linear 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 non-linear 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.

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[7].

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.

3.4. 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. In addition, the Ljung-Box-Pierce statistics are highly significant (p-value $\approx$ 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). 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

---

[7] The computer code for finding trading rules for the S&P 500 index can be obtained using so-called anonymous ftp from `grace.wharton.upenn.edu` (log in as `anonymous` and type your e-mail address as password). The C code is located in the directory `pub/programs/risto`.

[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).

squared process are of a magnitude ten times larger than the 95% confidence interval of $\pm 2/\sqrt{T}$. The autocorrelations of the absolute and squared residuals remain significantly different from zero even if the linear structure is removed using an autoregressive model, as illustrated in figure 3.

These observations are consistent with the results of Hsieh (1991), who also rejected the null model of IID returns for the daily S&P 500 returns from 1963 to 1987. The rejection of the IID hypothesis is consistent with four explanations: linear dependence, non-stationarity, low-dimensional chaos, and non-linear stochastic processes. Hsieh (1991) concluded that the nonlinearity in the data was due to conditional heteroskedasticity in the subperiod of 1982-89 he studied in more detail. As his results demonstrate, the rejection of a random walk in a return series does not necessarily imply anything about market efficiency.

3.5. 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 the training period and years 1968-69 for the selection period. 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 for the entire test period 1970-89 was +90.23%, with a standard deviation across rules of 16.88%. 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%.

---

[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).

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 4 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 +1.39% to +5.91%, and the number of trades per year ranges from 14 to 42. The biggest 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 5 shows a comparison of the average yearly return for the 100 rules (dashed line) to the buy-and-hold return (solid line). 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. The sample standard deviation of the annual returns in 1970-89 for the buy-and-hold strategy is 16.0%. The volatility for the trading rules ranges from 4.6% to 14.4%, with an average of 9.6%. Were the rules used as an equal-weighted "portfolio", the standard deviation would drop to 8.2%.

Figure 6 shows the average cumulative return for trading rules vs. the buy-and-hold return from 1970 to 1989. 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 which do not use short sales or borrowing 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.

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 significance level (the t-statistic for the difference between the "buy" and the unconditional return is $t = (r_b - r) \Big/ \left(s\sqrt{\frac{1}{N_b} + \frac{1}{N}} \right)$, 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 analogous manner). In addition, the standard deviation during "in" periods is smaller than during "out" periods for each of the ten rules. On the average, the difference is about 22%. The excess returns do not seem to be due to increased riskiness, although this point is addressed in more detail through bootstrapping simulations below.

Since trading rules can be interpreted as market-timing strategies, statistical tests of timing ability provide another way to evaluate the significance of the results. In an earlier version of this paper (Allen and Karjalainen, 1993), we carried out market-timing tests proposed by Cumby and Modest (1987). These tests indicated that all the rules in table 2 had statistically significant forecasting ability. However, the proportion of variance explained by the trading rule signals was small, in the range of 0.03 to 0.04.

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.

A comparison to a simple benchmark rule is perhaps helpful in placing the results in a proper perspective. We use a benchmark rule corresponding to the so-called weekend effect, one of the seasonal anomalies widely documented in the literature[11]. Stock returns tend to be, on the average, negative on Mondays. This effect has been observed in the S&P Composite Index (Cross, 1973; French, 1980), in the Dow-Jones index (Gibbons and Hess, 1981), and in OTC stocks (Keim and Stambaugh, 1984). The weekend effect has been present for at least 90 years (Lakonishok and Smidt, 1988), and has been documented in many countries across the world (for a summary, see Hawawini and Keim, 1995).

The weekend effect was turned into a trading rule that stays long in the market from Tuesday to Friday, sells the market portfolio and buys T-bills when the markets close on Friday, and switches back to stocks at the closing on Monday. In the same test period of 1970-89 as used for the genetic algorithm, the average return for the 967 Mondays was -0.001256 (the standard deviation was 0.012538), significantly smaller than the average of 0.000626 for the rest of the week (the standard deviation for the other 4087 days was 0.009082). Without transaction costs, the trading rule for the weekend effect had an average return of 14.2% per year, corresponding to an annual excess return of 7.46%. With the same one-way transaction cost of 0.1% as used for the genetic algorithm, however, the rule would have led to a return of only 4.52% per year, less than for the buy-and-hold strategy. The break-even transaction cost was equal to 0.08%.

The results for the weekend rule are weaker than for the genetic algorithm. The daily returns during the "in" periods (Tuesday to Friday) are smaller than for any of the rules in table 2. The returns during the "out" periods (i.e. Mondays), however, are more negative than for the rules found by the genetic algorithm. Interestingly, the pattern of results for the weekend rule resembles that for rules 9 and 10 in table 2, despite the fact that the weekdays were not in the information set of the algorithm. Compared with the weekend rule, these two rules had a roughly similar number of days in and out of the market, and comparable daily returns and standard deviations. On the other hand, rules 9 and 10 did achieve these results with significantly fewer trades (14 and 20 per year, respectively) than the number of Mondays in a year. In any case, genetic algorithm rules are uncorrelated

[11] This comparison was suggested to us by Josef Lakonishok.

with the weekend rule, with long positions roughly equally distributed across the week (176 out of 915 and 172 out of 905 days long were Mondays for rules 9 and 10, respectively).

The trading rule results can also be compared to the study by BLL. They tested different kinds of simple technical trading rules, including variable- and 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 band, and a "sell" signal when the short moving average drops back below the long one (the band is either 0% or 1%, the short-term moving average window 1,2, or 5 days, and the long between 50 and 200 days). 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). While the pattern of reduced volatility during long positions is very similar to rules in table 2, the difference between returns during "in" and "out" periods is only about 30% of that observed for the genetic algorithm rules Overall, these results corroborate the findings of BLL. It also appears that the trading rules found by the genetic algorithm have better forecasting ability than the variable-length moving average rules tested by BLL.

3.6. Characterizing the trading rules

There are many interesting questions regarding the nature of the trading rules found by the genetic algorithm. This section describes what the rules look like and analyzes what kind of price behavior triggers the trading rule signals.

As an example, let us first consider rule 4 in table 2. The genetic structure consisting of 46 nodes is rather unwieldy to write out, but it can be simplified to an equivalent tree structure shown in figure 7. 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[12]. 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 the greater of the two, then the middle subtree (>) is evaluated. That subtree returns either a "buy" or a "sell" signal,

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

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 7 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. Another example of the rules is shown in figure 8. The structure corresponds to rule 7, which is usually equivalent to a 2-3-day 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.

Although the genetic algorithm is capable of forming highly complicated and non-linear functions of past prices, the complexity of the final rules is not pre-determined. Measuring the complexity of the trading rules is useful, because it sheds light on the predictability of future returns using the price history only. If the algorithm came up with very complicated rules, the results would be consistent with a view that there is some kind of a "hidden" structure that could be discovered from past prices. If only relatively simple rules were found, the results would be more consistent with a view that past prices have limited value in predicting future returns.

Measuring the complexity of the trading rules is difficult, because it cannot be done by inspection of the tree structures. While many of the rules appear to be quite complicated at first sight, there is often a lot of redundant material in the form of duplicated subtrees. If some of these subtrees never get visited when the rules are evaluated, seemingly complex tree structures may be effectively similar to much simpler ones. To find out whether this was the case, we compared the "behavior" of the rules — not the rules themselves — to the behavior of a class of simple technical trading rules.

The technical rules that the genetic algorithm rules were compared to were of the form illustrated in figure 1. These rules compare one indicator to another, and take a position in the market if the value of the first indicator is greater than the value of the second one, and a position out of the market otherwise. Each indicator was chosen from the set {`max`, `min`, `average`}, and the length of the time window for an indicator was chosen from the set {1,2,3,4,5,10,15,20,40,60,80,100,150,200,250}. For each rule in table 2, the combination of indicators and time windows that led to the highest correlation between the daily positions was retained, after evaluating all the $3^2 \times 15^2 = 2025$ combinations.

For rules 1, 4, 5, and 7, the closest match was a rule which takes a long position if the 5-day moving average is greater than today's price[13], with a correlation of 0.787, 0.841, 0.803, and 0.675, respectively. For rules 2, 3, and 6, the 4-day moving average rule provided the best match, with a correlation of 0.831, 0.862, and 0.862, respectively. Rule 8 was ill matched with any of the simple rules with a correlation of only 0.378, corresponding to a rule that compared the maximum of 250 days' prices to the maximum of 20 days. For rules 9 and 10, the closest counterpart among the simple rules was a 250-day moving average rule, with a correlation of 0.632 and 0.630, respectively.

The test results in 1970-89 for the four different simple rules are shown in table 4. It is apparent that the results for the short-term moving average rules are similar to the genetic algorithm rules in table 2, although the difference between the returns between "in" and "out" periods is slightly smaller. The combination closest to rule 8 leads to results that are quite different from the original. For rules 9 and 10, the closest is only moderately successful in reproducing the original findings.

One possible interpretation of these results is the following: Given the opportunity of creating almost arbitrarily complicated functions of past prices, the genetic algorithm converged to rules only modestly different from some very simple technical trading rules. This is not to say that the genetic algorithm rules are worthless; the results of the previous section do indicate, after all, that the rules can predict future returns. What the simplicity of the rules does suggest is a conclusion that the degree of potential predictability is small, so that more complex technical rules are unlikely to lead to much higher excess returns.

The projection of the trading rules into a space of simpler ones shows that most of the rules tend to be of a trend-following kind. To get a better picture of what kind of price behavior actually triggers the trading rule signals, we analyzed two of the rules — rule 1 and rule 10 in table 2 — in more detail. Rule 1 is an example of a short-term moving average rule, and rule 10 is an example of a longer-term moving average rule. Presenting a detailed analysis for these two rules only should be sufficient, since the results of table 4 divide the rules rather naturally into two disjoint sub-classes.

For rule 1 many of the trades lead to small losses or modest gains, but once every so often the prices move by 5 to 10% in the "right" direction after the rule switches to a different position. Trades in a long position typically last for one to two weeks. Without transaction costs, most of the trades are profitable, and the losing

---

[13] Note that today's price is returned for each of the indicators for the case where the length of time window equals one.

trades are quickly covered. The rule switches out of the market after a large negative return, consistently with its trend-following nature. On the average, small negative returns tend to persist for the next 4 days or so. The rule switches from "out" to "in" after a large positive return, which tends to be followed by small positive returns during the next few days. The size of the average price change needed to trigger a switch of position is about 70 to 90 basis points, or roughly 80-90% of the daily volatility.

Rule 10 is different in a number of interesting ways. To begin with, there are fewer trades (410 vs. 717 for rule 1). Consistently with the relatively long-term nature of this rule, the big gains occur mostly during a small number of winning trades. The rule stays long in the market for periods of up to three years at a time. Almost all of the trades that stay in the market for longer than a few days are profitable. All of the trades out of the market are closed in less than two weeks. Rule 10 effectively combines two kinds of building blocks. It takes a long position when the price rises above a short-term moving average, and then stays in the market until the price drops below a much longer moving average.

Figures 9 and 10 present typical price patterns corresponding to trading rule signals. Figure 9 shows the cumulative returns (without transaction costs) for each trade for rule 1 in a long position, superimposed on top of each other. This figure illustrates how rule 1 exploits price trends that last about one to two weeks, on the average. Without transaction costs, most of the trades are profitable, and the losing trades are quickly covered. Figure 10 shows the cumulative market returns during periods when rule 1 stayed out of the market. The pattern is roughly a mirror image of that in figure 9, indicating that both the "buy" and the "sell" signals carry similar information about future returns. The corresponding figures for rule 10 have been omitted for brevity. Compared to rule 1, the main difference is the breakdown of symmetry between "buy" and "sell" signals. As discussed above, the positions in the market last much longer than positions in the risk-free asset.

To summarize the results of this section, it was found that the genetic algorithm rules may look deceptively complicated, but can often be closely replicated with rather simple technical trading rules. This indicates that while past prices can be used to predict future returns to a small extent, the limit of predictability is quickly reached. If this were not the case, one would expect a powerful search algorithm like genetic programming to come up with more complicated and more profitable rules. Lastly, analyzing the aggregated trade data indicated that the rules had predictive ability because the large daily returns that made them switch side carried over for a few days, on the average.

3.7. The effects of biased data

While the excess returns for the trading rules appear to be statistically significant, there is a possibility that much of these findings can be explained by a combination of biases in the data and flaws in the design of the algorithm[14]. It should also be kept in mind that the one-way transaction costs of 0.1% are small, and probably not realistic for most market participants, especially before the deregulation of the New York Stock Exchange on May 1, 1975. Even if members of the stock exchange could achieve low trading costs, their advantage is partly eroded by the taxes they have to pay on the trades. In any case, the excess returns are not huge, as the break-even one-way transaction cost is 0.18%, on the average.

Using a stock index may lead to biased results. Delays before the prices of the infrequently traded stocks adjust to news can lead to spurious serial correlation in a stock index (Fisher, 1966). It is possible that the genetic algorithm finds trading rules which exploit such spurious time series structure[15]. One way to address this issue would be to use futures data, where there is little evidence of positive autocorrelation. Another way to account for spurious serial correlation is to model the effects of non-synchronous trading using low-order autoregressive models, as proposed by Lo and MacKinlay (1990). The bootstrapping tests of an AR(1) model in the next section are used to study the possibility that the results are an artifact of those aspects of market microstructure that give rise to linear dependence in the returns.

Another source of bias is the use of closing prices. The tests in this paper are based on the implicit assumption that the trading signals could be generated on the basis of the daily closing price, and that the same price could also be used for executing the trades. Clearly, this could not be done in practice, and the only justification is the assumption that the results would remain essentially the same even if a more realistic scheme were used. One way to achieve more realistic timing of trades would involve generating signals on the basis of prices some time before closing (one hour, say), and executing the trades after a delay of perhaps 10-15 minutes. Since historical intra-day data for the S&P 500 index are not widely available, such a test was not carried out.

---

[14] The authors are grateful for Larry Fisher for pointing out many of the potential biases.

[15] There is evidence, however, that nonsynchronous trading can only explain a relatively minor part of the serial correlation in stock market indices (Atchison, Butler and Simonds, 1987).

The payment of dividends on the component stocks is ignored in the computation of the S&P 500 index. Ignoring dividends introduces two sources of bias. Any seasonality in the dividends may 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. There is, in fact, a rather pronounced yearly pattern in dividend payments, which tend to be concentrated in February, May, August, and November (Luskin, 1987, pp. 140-141). Nevertheless, it would be rather a coincidence if the excess returns were explained by clustered dividends, since the calendar dates are not part of the information set of the genetic algorithm.

The dividends do, however, have a significant impact on the trading rule results. Ignoring the dividend yield leads to under-estimation of the return for the buy-and-hold strategy. The trading rule returns are under-estimated, too, but to a lesser extent. In order to get a rough estimate of the impact of dividends, the monthly Stocks, Bonds, Bills, and Inflation data from Ibbotson and Associates were obtained from CRSP. In the test period of 1970 to 1989, the dividend yield was 4.22% on a continuously compounded basis. The capital appreciation was 6.72%, leading to the total return of 10.9% in this period.

On the average across the 100 trials, the trading rules stayed out of the market for 39.45% of the time in the test period, or 1993.7 out of 5054 trading days. The aggregate effect of the dividends can be estimated by subtracting $0.3945 \times 4.22\% = 1.66\%$ from the average excess returns of 4.51%. Consequently, about 37% of the excess returns can be explained by the dividends. Taking the dividends into account, the average break-even transaction costs turn out to be 0.15%.

The effects of dividends were studied in more detail using a value-weighted index of NYSE and AMEX stocks from the CRSP data set[16]. This index should provide a reasonable proxy for the S&P 500: both are value-weighted, and the S&P 500 consists mainly of stocks with a fairly large market capitalization. Table 5a presents the results for the ten test rules without dividends. Comparing the results to the S&P 500 data in table 2, it can be seen that the overall pattern of findings is similar. The returns are positive during days when rules stay long in the market, and negative when the rules stay out. The variability of the returns also follows the same pattern as before in table 2. The excess returns are somewhat higher, though, and the t-statistics are uniformly much more significant than for the S&P 500 index.

---

[16] The index levels were reconstructed from the CRSP data by setting the index to 100 in the beginning of 1963.

Table 5b shows the trading rule results with the dividends. As might be expected, the effect of the dividends is to increase the daily returns in periods both in and out of the market by roughly the same amount. This finding indicates that while there is a seasonal pattern in the dividends, that pattern is independent of the trading rule signals. The t-statistics in table 5b are slightly lower than in table 5a, but remain highly significant. The main effect of the dividends is to lower the annual excess returns by about 24%.

The significance of the statistical tests in table 2 can be questioned, since many of the rules are similar to each other. The similarity is of course due to the fact that the same training period was used in all the trials. Consequently, the trials are not truly independent of each other, even though each starts from random initial conditions. In principle, it is possible that the algorithm only happens to work for the particular training period studied in detail in this paper. To study this possibility, it would be desirable to address the stability of the results across different training environments and different markets.

Since the first version of this paper was completed, we have gained access to a longer price history for the S&P 500 index[17], starting at the beginning of 1926. These data give us a different out-of-sample test period, and can also be used to study the robustness of the results across different training periods. It is likely, though, that a number of switches of economic regimes has taken place over the years, so that it would be unrealistic to expect the same kind of rules to work across several decades. Nevertheless, using a longer data set should provide some insight into the stability of the results over time.

As a first test over the longer price history, the rules found from 1964-67 were applied to the data from 1929 to 1963. In this period[18], there are 9800 trading days with an average return of 0.000115 and a standard deviation of 0.013010. In this period, the buy-and-hold strategy had an annual return of 3.21%. With the same 0.1% one-way transaction cost as used before, the mean excess return over the buy-and-hold strategy for the 100 rules found by the genetic algorithm was -0.72% per year, ranging from -3.96% to +3.91. The sample standard deviation of annual returns for the buy-and-hold strategy was 23.4%, while the standard deviation of the trading rule returns ranges from 12.4% to 26.7%, with an average of 19.0% (the volatility of an equal-weighted portfolio was 18.0%).

---

[17] The authors are grateful to Bill Schwert for providing us with these data.

[18] Years 1926-27 were dropped because the data were recorded on a weekly frequency in that period. Since one year of price history is needed to normalize the prices, 1929 is the first year available for the genetic algorithm.

Table 6 shows the results in more detail for the ten rules tested in table 2. As in 1970-89, the rules tended to stay long in the market when daily returns were positive, and stay out when they were negative. The daily volatility was also lower during the "in" days than during the "out" days. Compared to days "out" of the market, the daily volatility was reduced by about 21% when the rules held a long position. The characteristics of individual rules remain similar across different test periods. For instance, rules 9 and 10 stayed invested in the market about 80% of the time, while the other rules stayed long only a little over 50% of the time. Overall, the findings corroborate the results for 1970-89, even though the rules did not lead to excess returns over the buy-and-hold strategy in 1929-63[19]. In particular, volatility is lower when the rules are long in the market compared to when they are out.

To address the sensitivity of the results for different training environments, the genetic algorithm was applied to three different 10-year training periods: 1929-38, 1939-48, and 1949-58. In each case, a 5-year selection period immediately following the training period was used, and the rules were tested in the remainder of the data up to the end of the original test period in 1989. The parameters for the genetic algorithm were exactly the same as before, with the exception that only 25 generations were allowed for evolution, and only 10 trials were carried out for each training period.

The results for the training period of 1929-38 and selection period of 1939-43 are shown in table 7a. There are 11854 days in the test period of 1944-89, with an average return of 0.000288 and a standard deviation of 0.008393. The annual return for the buy-and-hold strategy in the test period is 7.41%, and it can be seen that even the best of the 10 rules barely matched that return. Considering the fact that the 1929 crash was included in the training period, it is not surprising that three of the rules learned to stay out of the market altogether. One rule learned a strategy very close to the buy-and-hold one. The remaining six rules greatly resemble the rules in table 2. For all but one of these rules, the difference between the daily return during "in" and "out" periods is statistically

---

[19] We also compared the results in 1929-63 to the rules of BLL's table II. With one-way transaction costs of 0.1%, these rules resulted in an average annual excess return of 1.3%. The average difference in daily returns between "in" and "out" periods was 0.000532, or about 47% of the difference for the genetic algorithm rules (t-statistics for the difference ranged from 1.25 to 2.44). BLL's rules led to higher excess returns because their holding periods were much longer. On the average, their rules made 4.2 trades per year, while the rules from table 2 made 33.9 trades per year. Because of the difference in trading frequency, the BLL rules can sustain higher transaction costs. Across the entire history we had access to (1929-89), the BLL rules break even as long as the one-way transaction costs are smaller than 0.29%.

significant, and the volatility is lower when the rules are long in the market. Figure 11 compares the average cumulative returns for the ten rules to the buy-and-hold strategy during the test period. The sequence of yearly returns for these rules is much smoother than for a passive long strategy. It can also be observed that the performance of the trading rules does not show any clear tendency to deteriorate further out after the end of the training period.

Table 7b shows the results for the training period of 1939-48 and selection period of 1949-53. In the test period of 1954-89, there are 9055 trading days with the average return of 0.000293 and standard deviation of 0.008580. The return for the buy-and-hold strategy is 7.37% per year. The rules found by the genetic algorithm did not, on the average, lead to excess returns. Six of the ten rules appear to have at least some forecasting ability, although the results are weaker than in table 2. There is evidence that rules 7 and 10 learned features of the data particular to the training data, with poor results in the out-of-sample test period. Again, the return sequence appears to be less volatile than for the buy-and-hold strategy (see figure 11).

The results for the last training period of 1949-58 and selection period of 1959-63 are shown in table 7c. The test period of 1964-89 consists of 6538 trading days. The average daily return was 0.000237 and the standard deviation was 0.009059; the annual return for the buy-and-hold strategy was 5.95%. A total of 16 trials were required for this training period, because the algorithm failed to find a rule better than the buy-and-hold in six of the trials[20]. The ten rules found in the end were mostly long in the test period (see figure 11). Since the training period consists of a strong bull market (see figure 12), this is as much as could be expected. Perhaps more interestingly, all the rules managed to stay out of the market for a small number of days with large negative returns even in the 26-year out-of-sample period.

Taken together, the above findings indicate that the excess returns after transaction costs in 1970-89 were more of an exception than the rule. It would be an exaggeration to say that the rules worked in this period only, since the other statistical measures of trading rule performance were roughly similar — albeit not quite as strong — in 1929-63. The explanation of the relatively good performance during the latest two decades is probably related to the return for the buy-and-hold strategy. To see why this is likely to be the case, consider figure 12,

[20] As the algorithm is set up, it does not save any rules before the average fitness (in the training period) in the population exceeds zero, corresponding to the return for the buy-and-hold strategy.

which shows the average cumulative returns in 1929-89 for the 100 rules trained in 1964-67. In the 1930's, the trading rules went down with the market. Beginning in 1942, the rules yielded a comparatively smooth sequence of returns, regardless of the return for the market as a whole. Consequently, they tended to outperform the buy-and-hold strategy when the market was flat or declining, and underperformed when the markets were rising rapidly. Given that the post-war years consist of an almost uninterrupted bull market, it is perhaps not that surprising that the rules failed to yield a return above a passive buy-and-hold strategy. About the only exception is provided by the bear market of the 1970's, where the trading rules did outperform the market by a wide margin.

As discussed in this section, there are a number of potential sources of bias in the trading rule results. The impact of dividends was studied in some detail. It was found that dividends do indeed affect the excess returns, but cannot explain the fact that the rules tended to take long positions when daily returns were positive, and stay out of the market when returns were negative. The stability of the results across different periods was also addressed. When the genetic algorithm rules from table 2 were applied to a test period of 1929-63, the results were roughly similar to the original test period of 1970-89 except they did not lead to excess returns after transaction costs. This was also true when the algorithm was applied to different ten-year training periods. In some of the trials, the algorithm picked up strong features particular to the training period and absent from the subsequent test periods. In other cases, rules similar to those in table 2 were found. However, the original finding that the trading rules reduced volatility was robust across different time periods. The standard deviation of daily returns was roughly 20% smaller in days when the rules were long in the market, compared to days when they held the risk-free asset. The annual volatility was about 4% to 8% smaller, in absolute terms, than for the buy-and-hold strategy.

3.8. Bootstrapping models of stock returns

Results in table 2 indicate that trading rules have statistically significant ability to predict future returns. As discussed in the previous section, these results cannot be wholly attributed to biases in the data or flaws in the design of the algorithm. However, the findings may still be explained by spurious time series structure in the market returns. Another problem is caused by the fact that 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

Following the lead of BLL, we apply the so-called bootstrapping methodology, which allows us to test specific null models of stock returns, assuming only IID innovations (Efron, 1979; Efron and Tibshirani, 1986; Hall, 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

$$P_{b,t}^{*} = P_{b,t-1}^{*} \exp\left(r_{b,t}^{*}\right), t = 1,\ldots,T, b = 1,\ldots,B \tag{5}$$

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^{*}, a resample $e_b^{*} = \{e_{b,1}^{*},\ldots,e_{b,T}^{*}\}$ of the original residuals is 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[21].

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 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

[21] See BLL (pp. 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.

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

$$r_t - \mu_t = \sum_{i=1}^{k} b\left(r_{t-i} - \mu\right) + e_t \tag{7}$$

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 facilitates 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 8a), 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).

As discussed above, any non-linear 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:

$$\begin{cases} r_t - \mu_t = \sum_{i=1}^{k} b\left(r_{t-i} - \mu\right) + e_t & (8) \\ e_t \sim N(0, h_t) & (9) \\ h_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i e_{t-i}^2 + \sum_{i=1}^{q} \beta_i h_{t-i} & (10) \end{cases}$$

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 8b).

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.

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(2) 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 9. 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.

Results for the bootstrapping simulations of the AR(1) model are shown in table 10. 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 returns and in the volatility between "in" and "out" periods cannot be accounted for by the autoregressive process.

Results from bootstrapping simulations of the GARCH(1,1)-AR(2) model are given in table 11. 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 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.

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 11 can be attributed to changing conditional variance. This finding suggests that while some of the excess returns are due to short-term linear dependence, the trading rules also exploit non-linear 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 12), and additional bootstrapping simulations were carried out for this period.

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[22].

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 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. These results imply that while the trading rule returns can be partly explained by linear dependence and clustered volatility of the returns, some of the returns can be attributed to nonlinearity not captured by the models we tested.

4. Concluding remarks

A genetic algorithm has been used to learn technical trading rules rather than having them exogenously specified as in previous studies. In the original out-of-sample test period of 1970-89, these price-based rules for the S&P 500 index led to excess returns over a simple buy-and-hold strategy. However, the excess returns vanished in earlier time periods between 1929 and 1963. In most of the periods studied, the rules took long positions when the returns were positive and daily volatility was low, and stayed out of the market when returns were negative and volatility high. In these aspects, the genetic algorithm rules performed better than the moving average rules analyzed by BLL.

The profitability of technical trading rules can be regarded as a measure of market efficiency. For instance, the variable-length moving average rules of BLL earned excess returns in 1929-89 as long as the one-way transaction costs were smaller than 0.3%. Bessembinder and Chan (1995) found that BLL's rules worked well in four emerging Asian markets, but had less explanatory power in two of the more developed ones. In contrast to results for BLL's rules in the U.S., the break-even one-way transaction costs for the same rules were 1.45%,

[22] 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.

averaged across the six Asian markets. These results are consistent with the view that these markets — especially the less liquid ones — are considerably less efficient than the U.S. stock market.

If we treat trading rule results as a measure of market efficiency, then the question of how to select the rules to test becomes important. It can be argued that using a genetic algorithm is a sensible way of doing this, since the rules so chosen are *ex ante* optimal (conditional, of course, on the structure imposed on the rule space). Testing such rules out-of-sample gives a good estimate about the usefulness of historical prices in predicting future returns. In this paper, we started with building blocks corresponding to simple and commonly used technical rules. The genetic algorithm combined these building blocks to rules that mostly relied on short-term trends in prices. Because the out-of-sample trading rule profits were small and not robust across different time periods, the results indicate that the market for S&P 500 stocks is, indeed, fairly efficient.

While our results suggest that profits from short-term technical rules are small, at best, there are still some open questions. The trading rules considerably reduced the volatility of both daily and annual returns. This reduction could not be explained by any of the models we tested through bootstrapping simulations. It is also possible that using higher transaction costs — and thus focusing on rules relying on longer-term price trends such as those studied by BLL — could lead to different conclusions. There are certainly other opportunities for testing rules learned by evolutionary algorithms. These may be found from liquid markets with low transaction costs, including financial futures, commodities, and foreign exchange markets. One may develop the methodology further: The genetic algorithm we used is a relatively simple one, and the parameters such as the length of the training period and the selection period are not necessarily optimal. More importantly, the current algorithm uses very limited information for its inputs. It would be interesting to apply a similar technique to learn fundamental trading rules, by changing the building blocks to include the desired fundamental variables.

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, 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, MA), 338-372.

Allen, Franklin and Risto Karjalainen, 1993, Using genetic algorithms to find technical trading rules, Rodney L. White Center, The Wharton School, University of Pennsylvania, working paper 20-93.

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, 111-118.

Bauer, Richard J. Jr., 1994, Genetic algorithms and investment strategies (Wiley, New York, NY).

Bessembinder, Hendrik and Kalok Chan, 1995, The profitability of technical trading rules in the Asian stock markets, Pacific-Basin Finance Journal, 3, 257-284.

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.

Brealey, Richard A. and Stewart C. Myers, 1991, Principles of corporate finance (McGraw Hill, New York, NY).

Brock, William, Josef Lakonishok, and Blake LeBaron, 1992, Simple technical trading rules and the stochastic properties of stock returns, Journal of Finance, 47, 1731-1764.

Brown, David P. and Robert H. Jennings, 1989, On technical analysis, Review of Financial Studies, 2, 527-551.

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

Cross, F., 1973, The behavior of stock prices on Fridays and Mondays, Financial Analysts Journal, 29 (November-December), 67-69.

Cumby, Robert E. and David M. Modest, 1987, Testing for market timing ability, Journal of Financial Economics, 19, 169-189.

Efron, B., 1979, Bootstrap methods: Another look at the Jackknife, Annals of Statistics, 7, 1-26.

Efron, B., and R. Tibshirani, 1986, Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy, Statistical Science, 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, 383-417.

Fama, E. F., 1991, Efficient Capital Markets: II, Journal of Finance, 46, 1575-1617.

Fama, E., and Blume, M., 1966, Filter rules and stock market trading, Security Prices: A Supplement, Journal of Business, 39, 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, NY).

French, Kenneth R., 1980, Stock returns and the weekend effect, Journal of Financial Economics, 8, 55-70.

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 (Addison-Wesley, Reading, MA), 1-70.

Gibbons, M.R. and P. Hess, 1981, Day of the week and asset returns, Journal of Business, 54, 579-596.

Goldberg, David E., 1989, Genetic algorithms in search, optimization, and machine learning (Addison-Wesley, Reading, MA).

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

Grossman, Sanford J. and Joseph E. Stiglitz, 1980, On the impossibility of informationally efficient markets, American Economic Review, 70, 393-408.

Hall, Peter, 1992, The bootstrap and Edgeworth expansion (Springer-Verlag, New York, NY).

Hawawini, Gabriel and Donald B. Keim, 1995, On the predictability of common stock returns: World-wide evidence, in: Jarrow, Robert A., Vojislav Maksimovich and William T. Ziemba, eds., North-Holland handbooks of operations research and management science, finance (North-Holland, Amsterdam), ch. A17, forthcoming.

Hinich, M.L. and Patterson, D.M., 1985, Evidence of nonlinearity in daily stock returns, Journal of Business and Economic Statistics, 3, 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, NY), 263-293.

Holland, John H., 1975, Adaptation in natural and artificial systems (University of Michigan Press, Ann Arbor, MI).

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, 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, 1839-1877.

Keim, Donald B. and Robert F. Stambaugh, 1986, A further investigation of the weekend effect in stock returns, Journal of Finance, 39, 819-835.

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

Lakonishok, Joseph and Seymour Smidt, 1988, Are seasonal anomalies real? A ninety-year perspective, Review of Financial Studies, 1, 403-425.

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 (Addison-Wesley, Reading, MA), 381-393.

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.

Lo, Andrew W. and A. Craig MacKinlay, 1990, An econometric analysis of nonsynchronous trading, Journal of Econometrics, 45, 181-211.

Luskin, Donald L., 1987, Index options and futures (Wiley, New York, NY).

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 (Addison-Wesley, Reading, MA), 249-263.

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

Nissen, Volker, 1993, Evolutionary algorithms in management science, Interdisziplinäres Graduiertenkolleg, Universität 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.

Samuelson, P.A., 1965, Proof that properly anticipated prices fluctuate randomly, Industrial Management Review, Spring, 41-49.

Scheinkman, Jose A. and Blake LeBaron, 1989, Nonlinear dynamics and stock returns, Journal of Business, 62, 311-338.

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

Sweeney, Richard J., 1988, Some new filter rule tests: Methods and results, Journal of Financial and Quantitative Analysis, 23, 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, CA), 2-9.

Taylor, Stephen J., 1986, Modeling financial time series (Wiley, New York, NY).

Treynor, Jack L. and Robert Ferguson, 1985, In defense of technical analysis, Journal of Finance, 40, 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, CA), 116-121.

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

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 probability 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 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)
average	+0.0436	2912	+0.001052		0.008844	2142	-0.000866		0.011290	+0.001917	

Table 3. Test results for the variable-length moving average rules 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)	0.01209	+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	

Table 4. Test results for the closest counterparts of the rules in Table 2 among simple technical trading rules. Rules 1, 4, 5, and 7 were best matched with a 5-day moving average rule; rules 2, 3, and 6 with a 4-day moving average rule, rule 8 with a rule that compared the maximum of 250 days' prices to the maximum of 20 days, and rules 9 and 10 with a 250-day moving average rule. 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)
1,4,5,7	+0.0424	2561	+0.001102	(+3.497)	0.008916	2493	-0.000593	(-3.560)	0.010674	+0.001695	(+6.111)
2,3,6	+0.0287	2564	+0.001081	(+3.409)	0.008935	2490	-0.000573	(-3.477)	0.010662	+0.001654	(+5.964)
8	-0.0128	4061	+0.000218	(-0.234)	0.010088	993	+0.000465	(+0.580)	0.008858	-0.000247	(-0.708)
9,10	+0.0219	2465	+0.000564	(+1.232)	0.008354	2589	-0.000018	(-1.192)	0.011097	+0.000582	(+2.099)

Table 5a. Test results for the trading rules applied to a value-weighted index of NYSE and AMEX stocks, excluding dividends. 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 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	+0.0725	2482	+0.001463	(+5.288)	0.008277	2572	-0.000904	(-5.164)	0.010041	+0.002367	(+9.052)
2	+0.0760	2608	+0.001389	(+5.045)	0.008244	2446	-0.000947	(-5.266)	0.010156	+0.002335	(+8.930)
3	+0.0780	2768	+0.001343	(+4.935)	0.008313	2286	-0.001054	(-5.606)	0.010202	+0.002397	(+9.129)
4	+0.0780	2409	+0.001508	(+5.432)	0.008073	2645	-0.000880	(-5.104)	0.010144	+0.002388	(+9.124)
5	+0.0721	2636	+0.001330	(+4.799)	0.008288	2418	-0.000909	(-5.083)	0.010148	+0.002239	(+8.558)
6	+0.0682	2602	+0.001391	(+5.052)	0.008315	2452	-0.000943	(-5.256)	0.010090	+0.002335	(+8.927)
7	+0.0790	2763	+0.001270	(+4.601)	0.008330	2291	-0.000961	(-5.212)	0.010202	+0.002231	(+8.498)
8	+0.0524	2772	+0.000938	(+3.094)	0.008542	2282	-0.000567	(-3.522)	0.010068	+0.001505	(+5.730)
9	+0.0626	4170	+0.000704	(+2.293)	0.008265	884	-0.001844	(-6.207)	0.012889	+0.002548	(+7.407)
10	+0.0694	4168	+0.000784	(+2.702)	0.008390	886	-0.002212	(-7.301)	0.012411	+0.002996	(+8.716)
average	+0.0708	2938	+0.001212		0.008304	2116	-0.001122		0.010635	+0.002334	

Table 5b. Test results for the trading rules applied to a value-weighted index of NYSE and AMEX stocks, including dividends. 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 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	+0.0515	2484	+0.001598	(+5.155)	0.008236	2570	-0.000709	(-5.040)	0.010075	+0.002307	(+8.829)
2	+0.0568	2603	+0.001550	(+5.021)	0.008276	2451	-0.000770	(-5.226)	0.010120	+0.002320	(+8.874)
3	+0.0543	2772	+0.001469	(+4.758)	0.008299	2282	-0.000844	(-5.417)	0.010223	+0.002313	(+8.812)
4	+0.0626	2433	+0.001692	(+5.530)	0.008078	2621	-0.000752	(-5.262)	0.010146	+0.002444	(+9.346)
5	+0.0552	2628	+0.001501	(+4.820)	0.008273	2426	-0.000741	(-5.083)	0.010149	+0.002243	(+8.576)
6	+0.0485	2592	+0.001557	(+5.046)	0.008313	2462	-0.000767	(-5.223)	0.01008	+0.002325	(+8.893)
7	+0.0644	2759	+0.001432	(+4.583)	0.008411	2295	-0.000787	(-5.181)	0.010113	+0.002219	(+8.457)
8	+0.0303	2365	+0.001198	(+3.343)	0.008640	2689	-0.000256	(-3.069)	0.009774	+0.001454	(+5.553)
9	+0.0518	4296	+0.000813	(+2.012)	0.008283	758	-0.001773	(-6.077)	0.013450	+0.002586	(+7.068)
10	+0.0632	4297	+0.000902	(+2.476)	0.008377	757	-0.002284	(-7.483)	0.013008	+0.003186	(+8.702)
average	+0.0539	2923	+0.001371		0.008319	2131	-0.000968		0.010714	+0.002339	

Table 6. 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 1929-63. 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	-0.0217	4861	+0.000655	(+2.368)	0.011593	4939	-0.000417	(-2.343)	0.014249	+0.001072	(+4.080)
2	-0.0039	5154	+0.000690	(+2.569)	0.011808	4646	-0.000523	(-2.753)	0.014199	+0.001213	(+4.610)
3	-0.0065	5514	+0.000643	(+2.413)	0.011827	4286	-0.000565	(-2.854)	0.014363	+0.001208	(+4.561)
4	+0.0091	4696	+0.000862	(+3.236)	0.011181	5104	-0.000573	(-3.062)	0.014458	+0.001435	(+5.454)
5	-0.0099	5165	+0.000635	(+2.327)	0.011753	4635	-0.000465	(-2.501)	0.014260	+0.001101	(+4.181)
6	-0.0158	5144	+0.000663	(+2.447)	0.011784	4656	-0.000491	(-2.615)	0.014219	+0.001154	(+4.385)
7	-0.0006	5213	+0.000627	(+2.296)	0.010732	4587	-0.000467	(-2.501)	0.015170	+0.001094	(+4.154)
8	-0.0148	5200	+0.000376	(+1.169)	0.013303	4600	-0.000180	(-1.269)	0.012666	+0.000556	(+2.112)
9	+0.0096	7882	+0.000358	(+1.237)	0.011742	1918	-0.000886	(-3.080)	0.017238	+0.001244	(+3.756)
10	-0.0065	7873	+0.000348	(+1.182)	0.011615	1927	-0.000836	(-2.934)	0.017567	+0.001184	(+3.580)
average	-0.0061	5670	+0.000586		0.011734	4130	-0.000540		0.014839	+0.001126	

Table 7a. Test results for trading rules found by the genetic algorithm in the training period 1929-38 and selection period 1939-43. The second column shows the average yearly excess return above the buy-and-hold strategy in 1944-89. 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	+0.0047	8263	+0.000400	(+0.936)	0.007534	3591	+0.000029	(-1.620)	0.010091	+0.000371	(+2.215)
2	-0.0431	0	+0.000000	(+0.000)	0.000000	11854	+0.000288	(+0.000)	0.008393	+0.000000	(+0.000)
3	-0.0050	7021	+0.000390	(+0.807)	0.007673	4833	+0.000140	(-1.034)	0.009339	+0.000250	(+1.594)
4	+0.0103	8216	+0.000436	(+1.233)	0.007101	3638	-0.000048	(-2.109)	0.010747	+0.000484	(+2.895)
5	-0.0006	6850	+0.000426	(+1.086)	0.007202	5004	+0.000098	(-1.338)	0.009788	+0.000328	(+2.099)
6	+0.0044	8407	+0.000391	(+0.860)	0.007568	3447	+0.000037	(-1.546)	0.010123	+0.000354	(+2.086)
7	+0.0031	8318	+0.000388	(+0.831)	0.007542	3536	+0.000053	(-1.460)	0.010113	+0.000335	(+1.986)
8	-0.0006	1183	+0.000286	(-0.013)	0.008345	24	+0.001009	(+0.420)	0.022068	-0.000723	(-0.421)
9	-0.0431	0	+0.000000	(+0.000)	0.000000	11854	+0.000288	(+0.000)	0.008393	+0.000000	(+0.000)
10	-0.0431	0	+0.000000	(+0.000)	0.000000	11854	+0.000288	(+0.000)	0.008393	+0.000000	(+0.000)
average	-0.0113	5891	+0.000272		0.005297	5964	+0.000218		0.010745	+0.000140	

Table 7b. Test results for trading rules found by the genetic algorithm in the training period 1939-48 and selection period 1949-53. The second column shows the average yearly excess return above the buy-and-hold strategy in 1954-89. 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	+0.0163	6753	+0.000441	(+1.068)	0.008123	2302	-0.000139	(-2.159)	0.009787	+0.000580	(+2.800)
2	+0.0060	6531	+0.000419	(+0.900)	0.008988	2524	-0.000031	(-1.679)	0.007412	+0.000450	(+2.236)
3	-0.0220	3916	+0.000351	(+0.352)	0.009182	5139	+0.000249	(-0.294)	0.008091	+0.000102	(+0.559)
4	+0.0008	5407	+0.000396	(+0.699)	0.007806	3648	+0.000141	(-0.908)	0.009611	+0.000256	(+1.391)
5	+0.0119	6306	+0.000457	(+1.166)	0.008406	2749	-0.000083	(-2.014)	0.008956	+0.000540	(+2.756)
6	+0.0223	5850	+0.000529	(+1.639)	0.007699	3205	-0.000137	(-2.442)	0.009975	+0.000667	(+3.535)
7	-0.0190	6938	+0.000267	(-0.194)	0.009000	2117	+0.000380	(+0.420)	0.007027	-0.000113	(-0.533)
8	+0.0004	8750	+0.000305	(+0.094)	0.008579	305	-0.000054	(-0.696)	0.008610	+0.000360	(+0.720)
9	-0.0172	5339	+0.000313	(+0.133)	0.008860	3716	+0.000265	(-0.170)	0.008161	+0.000048	(+0.262)
10	-0.0145	8272	+0.000254	(-0.304)	0.008665	783	+0.000712	(+1.309)	0.007615	-0.000458	(-1.428)
average	-0.0015	6406	+0.000373		0.008531	2649	+0.000130		0.008525	+0.000243	

Table 7c. Test results for trading rules found by the genetic algorithm in the training period 1949-58 and selection period 1959-63. The second column shows the average yearly excess return above the buy-and-hold strategy in 1964-89. 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	+0.0028	6391	+0.000264	(+0.172)	0.009054	147	-0.000952	(-1.574)	0.009209	+0.001217	(+1.610)
2	+0.0059	6368	+0.000270	(+0.207)	0.009084	170	-0.001001	(-1.759)	0.007983	+0.001271	(+1.805)
3	+0.0081	6354	+0.000280	(+0.268)	0.009073	184	-0.001242	(-2.184)	0.008451	+0.001522	(+2.246)
4	+0.0138	6163	+0.000330	(+0.576)	0.009058	375	-0.001287	(-3.168)	0.008938	+0.001616	(+3.355)
5	+0.0064	6377	+0.000271	(+0.213)	0.009083	161	-0.001107	(-1.860)	0.007965	+0.001378	(+1.906)
6	+0.0050	6386	+0.000265	(+0.173)	0.009085	152	-0.000919	(-1.556)	0.007792	+0.001378	(+1.906)
7	+0.0070	6339	+0.000274	(+0.233)	0.009079	199	-0.000949	(-1.819)	0.008317	+0.001223	(+1.875)
8	+0.0047	6387	+0.000265	(+0.172)	0.009086	151	-0.000924	(-1.557)	0.007769	+0.001189	(+1.594)
9	+0.0065	6365	+0.000271	(+0.216)	0.009074	173	-0.001028	(-1.813)	0.008409	+0.001300	(+1.862)
10	+0.0075	6360	+0.000276	(+0.245)	0.009071	178	-0.001160	(-2.030)	0.008510	+0.001436	(+2.086)
average	+0.0068	6349	+0.000277		0.009075	189	-0.001057		0.008334	+0.001353	

Table 8a. 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
$0.231032\ 10^{-3}$	0.125322
(+1.636)	(+2.553)

Table 8b. 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)
$0.342352\ 10^{-3}$	0.166418	-0.030719	$0.124523\ 10^{-5}$	0.072846	0.915440	$0.262586\ 10^{-5}$
(+3.111)	(+10.67)	(-1.952)	(+7.188)	(+39.66)	(+229.5)	(+1.263)

Table 9. 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 (0.0000)	2415	2638	+0.000247 (0.0000)	+0.000266 (1.0000)	-0.00002 (0.0000)	0.009895 (0.9830)	0.009881 (0.1620)	0.000014 (0.9330)
2	-0.0827 (0.0000)	2561	2492	+0.000268 (0.0000)	+0.000268 (1.0000)	-0.00000 (0.0000)	0.009864 (0.9820)	0.009862 (0.1200)	0.000002 (0.9530)
3	-0.0841 (0.0000)	2729	2324	+0.000269 (0.0000)	+0.000272 (1.0000)	-0.000003 (0.0000)	0.009869 (0.9780)	0.009883 (0.1140)	-0.000014 (0.9600)
4	-0.0861 (0.0000)	2352	2701	+0.000257 (0.0000)	+0.000258 (1.0000)	-0.000001 (0.0000)	0.009903 (1.0000)	0.009922 (0.1400)	-0.000018 (0.9670)
5	-0.0800 (0.0000)	2571	2482	+0.000270 (0.0000)	+0.000258 (1.0000)	0.000012 (0.0000)	0.009864 (0.9780)	0.009894 (0.1280)	-0.00003 (0.9410)
6	-0.0905 (0.0000)	2558	2495	+0.000265 (0.0000)	+0.000263 (1.0000)	0.000003 (0.0000)	0.009903 (0.9700)	0.009867 (0.1400)	0.000037 (0.9350)
7	-0.0755 (0.0000)	2680	2373	+0.000253 (0.0000)	+0.000259 (1.0000)	-0.000006 (0.0000)	0.009904 (0.9810)	0.009902 (0.1240)	0.000002 (0.9640)
8	-0.00553 (0.0080)	2775	2278	+0.000276 (0.0080)	+0.000250 (0.9940)	+0.000026 (0.0010)	0.009888 (0.9730)	0.009826 (0.1090)	0.000062 (0.9530)
9	-0.0405 (0.0050)	4113	940	+0.000254 (0.0010)	+0.000279 (1.0000)	-0.000025 (0.0000)	0.009907 (1.0000)	0.009808 (0.0100)	0.000099 (0.9990)
10	-0.0517 (0.0100)	4146	907	+0.000260 (0.0010)	+0.000287 (1.0000)	-0.000028 (0.0000)	0.009878 (0.9970)	0.009855 (0.0220)	0.000023 (0.9920)
average	-0.0738 (0.0023)	2890	2163	+0.000262 (0.0010)	+0.000266 (0.9994)	-0.000004 (0.0001)	0.009888 (0.9842)	0.00987 (0.1069)	0.000018 (0.9597)

Table 10. 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		daily return			standard deviation		
		in	out	in	out	in - out	in	out	in - out
1	+0.0067	2416	2637	+0.001006	-0.000401	+0.001407	0.009756	0.009771	-0.000015
	(0.0910)			(0.0840)	(0.8970)	(0.0220)	(0.9720)	(0.1070)	(0.9570)
2	+0.0105	2552	2501	+0.000954	-0.000439	+0.001393	0.009765	0.009747	0.000017
	(0.0770)			(0.0810)	(0.9080)	(0.0170)	(0.9650)	(0.0790)	(0.9650)
3	+0.0114	2713	2340	+0.000929	-0.00049	+0.001419	0.00978	0.009753	0.000027
	(0.1230)			(0.1320)	(0.9020)	(0.0350)	(0.9740)	(0.0760)	(0.9730)
4	+0.0079	2348	2705	+0.001006	-0.000384	+0.001390	0.009755	0.009784	-0.000029
	(0.0870)			(0.0810)	(0.8910)	(0.0290)	(0.9990)	(0.0840)	(0.9790)
5	+0.0088	2561	2492	+0.000924	-0.000395	+0.001319	0.00975	0.009745	0.000004
	(0.1130)			(0.1400)	(0.8720)	(0.0570)	(0.9620)	(0.0880)	(0.9570)
6	+0.0085	2546	2507	+0.000993	-0.000472	+0.001466	0.009755	0.009817	-0.000063
	(0.1340)			(0.1430)	(0.8750)	(0.0370)	(0.9550)	(0.1160)	(0.9480)
7	+0.0078	2670	2383	+0.000837	-0.000357	+0.001194	0.00976	0.009764	-0.000004
	(0.1040)			(0.1490)	(0.8990)	(0.0470)	(0.9860)	(0.0830)	(0.9720)
8	-0.0133	2616	2437	+0.000577	-0.00008	+0.000657	0.009781	0.009806	-0.000025
	(0.0880)			(0.1600)	(0.8190)	(0.0680)	(0.9680)	(0.0860)	(0.9540)
9	+0.0082	4097	956	+0.000481	-0.00057	+0.001051	0.009848	0.009715	0.000133
	(0.1000)			(0.0690)	(0.9990)	(0.0010)	(0.9990)	(0.0070)	(0.9970)
10	+0.0002	4112	941	+0.000499	-0.000674	+0.001173	0.009773	0.009796	-0.000024
	(0.0890)			(0.0360)	(0.9980)	(0.0010)	(0.9940)	(0.0120)	(0.9940)
average	+0.0057	2863	2190	+0.000821	-0.000426	+0.001247	0.009772	0.00977	0.000002
	(0.1006)			(0.1075)	(0.9060)	(0.0314)	(0.9774)	(0.0738)	(0.9696)

Table 11. 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 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	+0.0184	2417	2636	+0.001106	-0.00056	+0.001666	0.009995	0.010152	-0.000157
	(0.1870)			(0.1950)	(0.7000)	(0.1910)	(0.8030)	(0.2430)	(0.9760)
2	+0.0216	2558	2495	+0.001040	-0.000585	+0.001626	0.010012	0.010085	-0.000073
	(0.1270)			(0.1550)	(0.7580)	(0.1130)	(0.7960)	(0.2150)	(0.9870)
3	+0.0220	2712	2341	+0.001011	-0.000673	+0.001684	0.00994	0.010015	-0.000074
	(0.2060)			(0.2270)	(0.6830)	(0.2220)	(0.7770)	(0.2140)	(0.9900)
4	+0.0190	2350	2703	+0.001103	-0.000535	+0.001638	0.009871	0.010149	-0.000278
	(0.1560)			(0.1760)	(0.6880)	(0.1790)	(0.8470)	(0.2410)	(0.9790)
5	+0.0146	2564	2489	+0.000972	-0.000548	+0.001520	0.009949	0.010082	-0.000133
	(0.1640)			(0.1960)	(0.7060)	(0.1740)	(0.7970)	(0.2330)	(0.9790)
6	+0.0199	2554	2499	+0.001082	-0.000658	+0.001739	0.010005	0.010116	-0.000111
	(0.2260)			(0.2460)	(0.6260)	(0.2430)	(0.7610)	(0.2550)	(0.9860)
7	+0.0148	2633	2420	+0.000904	-0.000495	+0.001399	0.009883	0.010261	-0.000378
	(0.1490)			(0.2410)	(0.7310)	(0.1900)	(0.7810)	(0.2440)	(0.9380)
8	-0.0050	2806	2247	+0.000602	-0.000231	+0.000833	0.009677	0.010378	-0.000701
	(0.1670)			(0.1930)	(0.6420)	(0.2190)	(0.6270)	(0.2450)	(0.8680)
9	+0.0071	4073	980	+0.000484	-0.000824	+0.001308	0.009724	0.010789	-0.001065
	(0.0750)			(0.0600)	(0.9580)	(0.0200)	(0.7780)	(0.0910)	(0.9690)
10	+0.0073	4073	980	+0.000541	-0.00105	+0.001591	0.009702	0.010929	-0.001228
	(0.0880)			(0.0650)	(0.9530)	(0.0310)	(0.7460)	(0.1230)	(0.9480)
average	+0.0140	2874	2179	+0.000884	-0.000616	+0.001500	0.009876	0.010296	-0.00042
	(0.1545)			(0.1754)	(0.7445)	(0.1582)	(0.7713)	(0.2104)	(0.9620)

Table 12a. 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 12b. 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)

Figure 1. 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 smaller 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 and a "sell" otherwise.

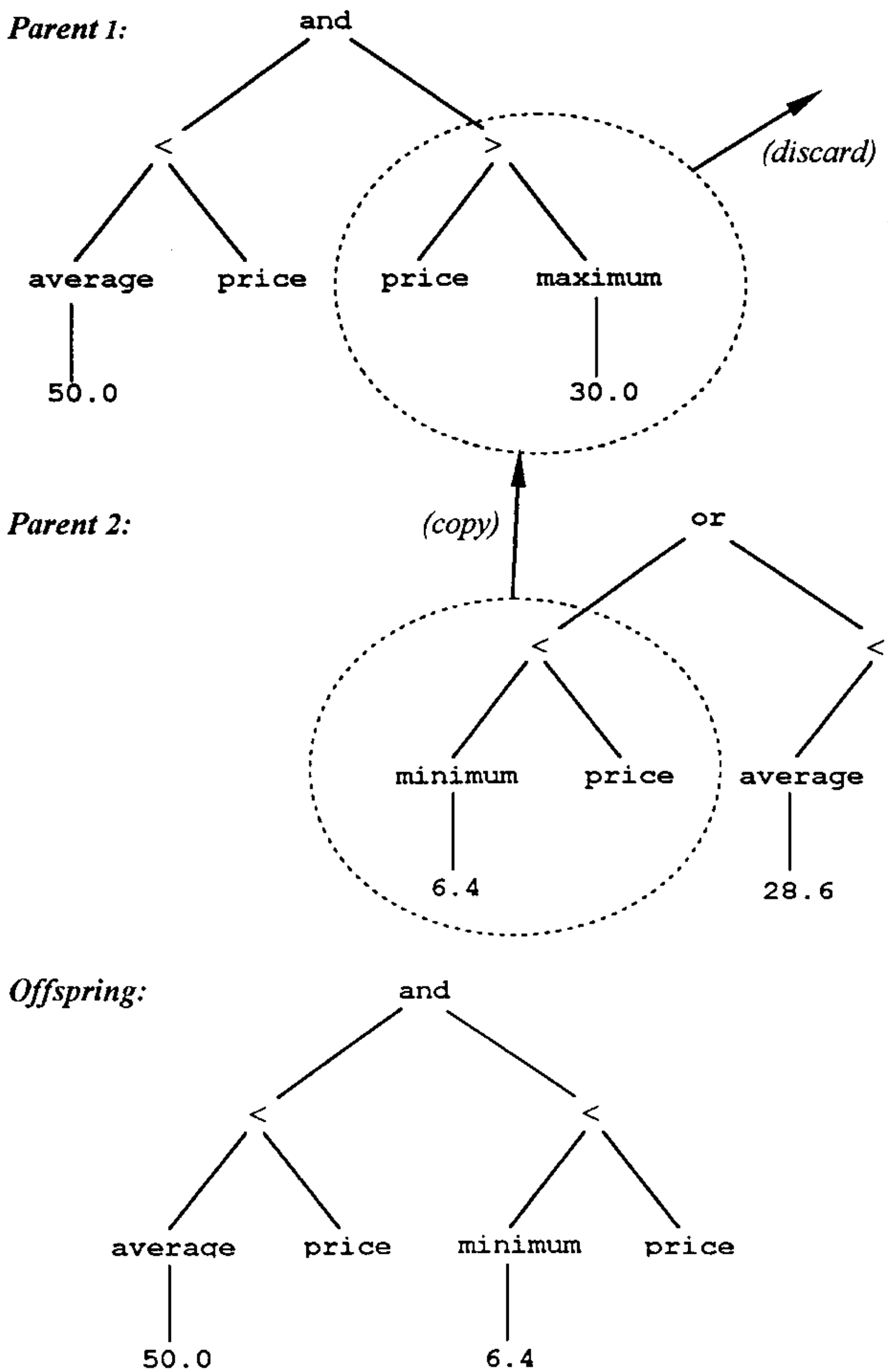

Figure 2. Illustration of the crossover operator of the genetic algorithm. A crossover creates a new trading rule by recombining two parent rules. A randomly chosen subtree (designated by the dotted line) in the first parent is discarded. The discarded subtree is replaced by a subtree of the same type, chosen from the second parent. The resulting offspring rule is shown in the bottom of the figure.

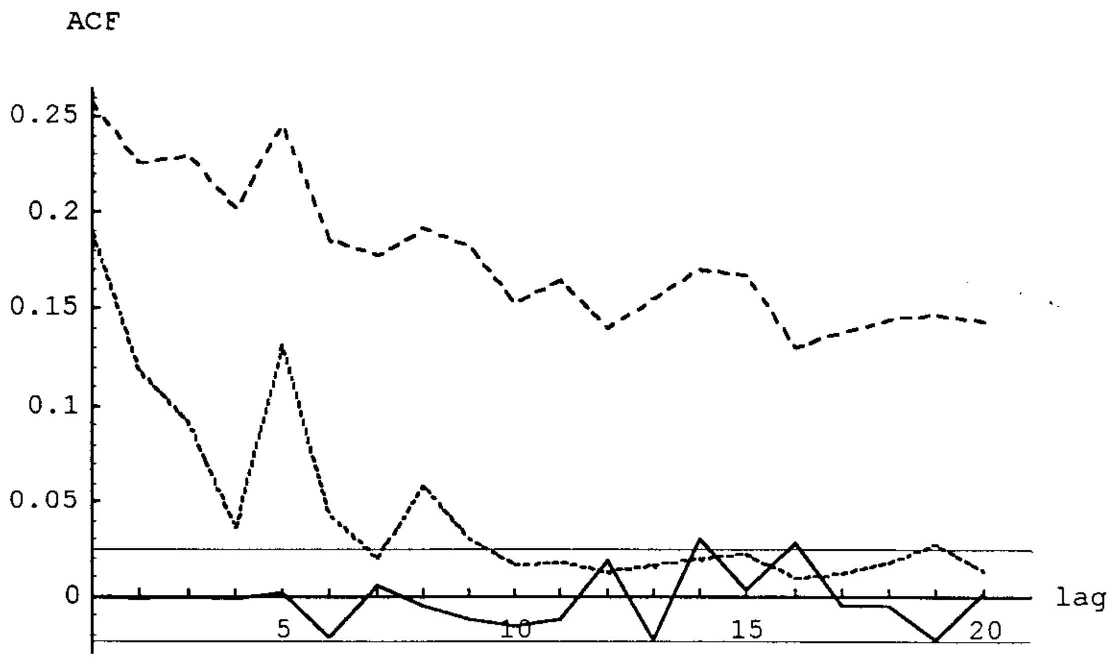

Figure 3. The sample autocorrelation function (ACF) for the residuals from an AR(5) model fitted to the daily S&P 500 index in 1963-89. The solid line corresponds to compounded daily returns, the dashed line to the absolute and the dotted line to the squared residuals. The 95% confidence interval for the hypothesis of strict white noise is indicated by the two thin horizontal lines.

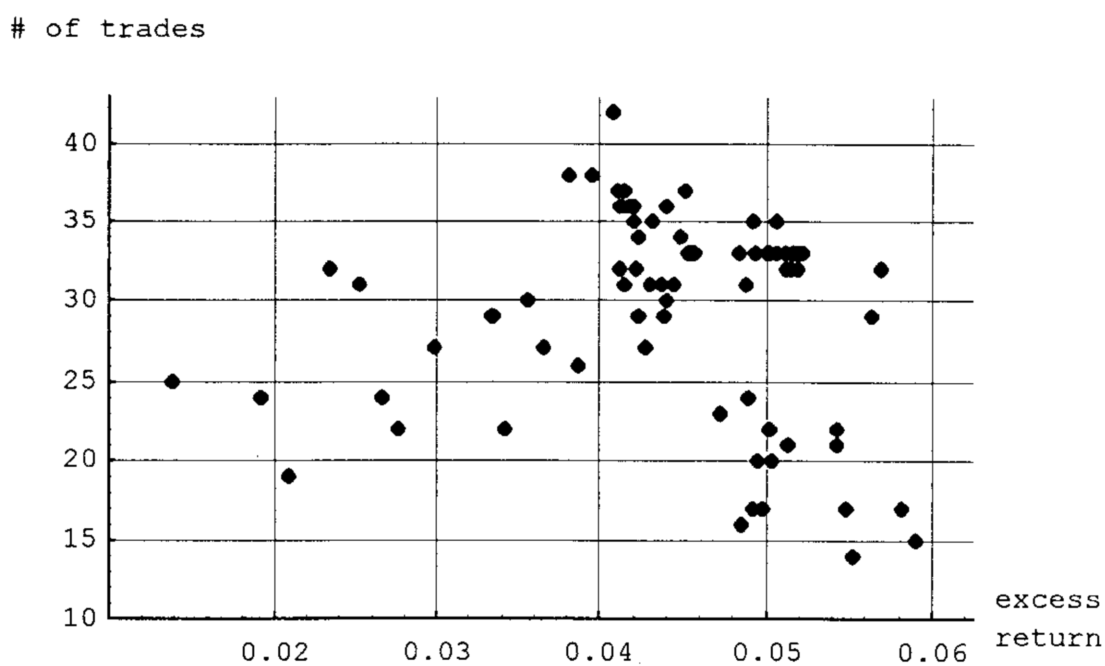

Figure 4. The yearly excess return and the average number of trades in the test period 1970-89 for the trading rules found by the genetic algorithm. The rules were found using the training period of 1964-67 and the selection period of 1968-69.

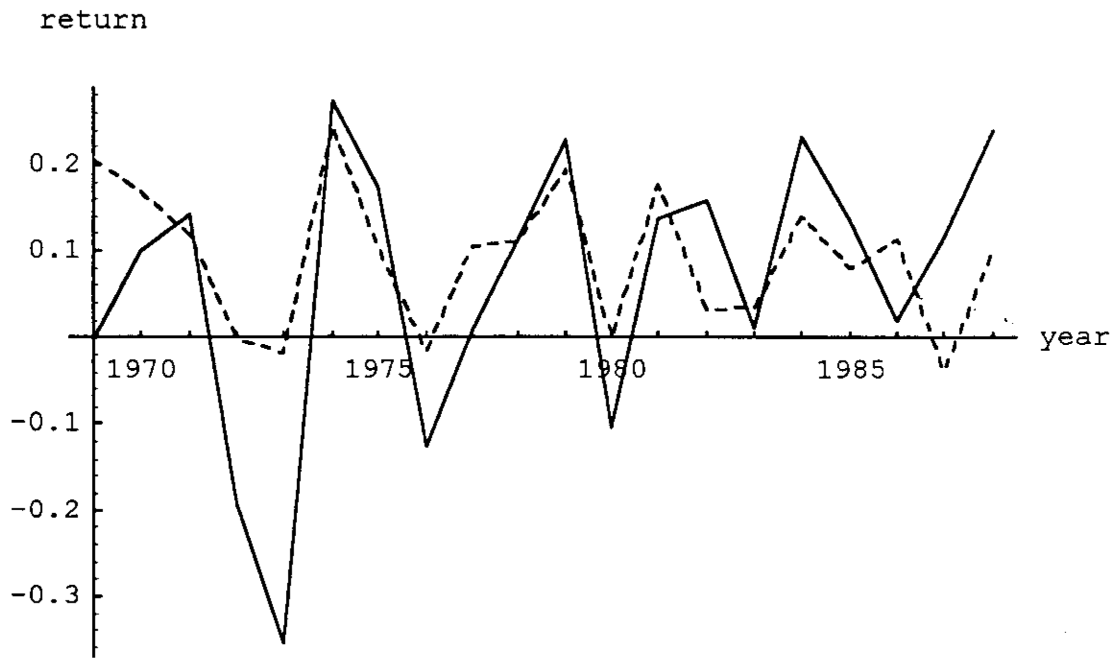

Figure 5. The average yearly return in the test period 1970-89 for the trading rules found by the genetic algorithm (dashed line), compared to the buy-and-hold strategy (solid line). The rules were found using the training period of 1964-67 and the selection period of 1968-69.

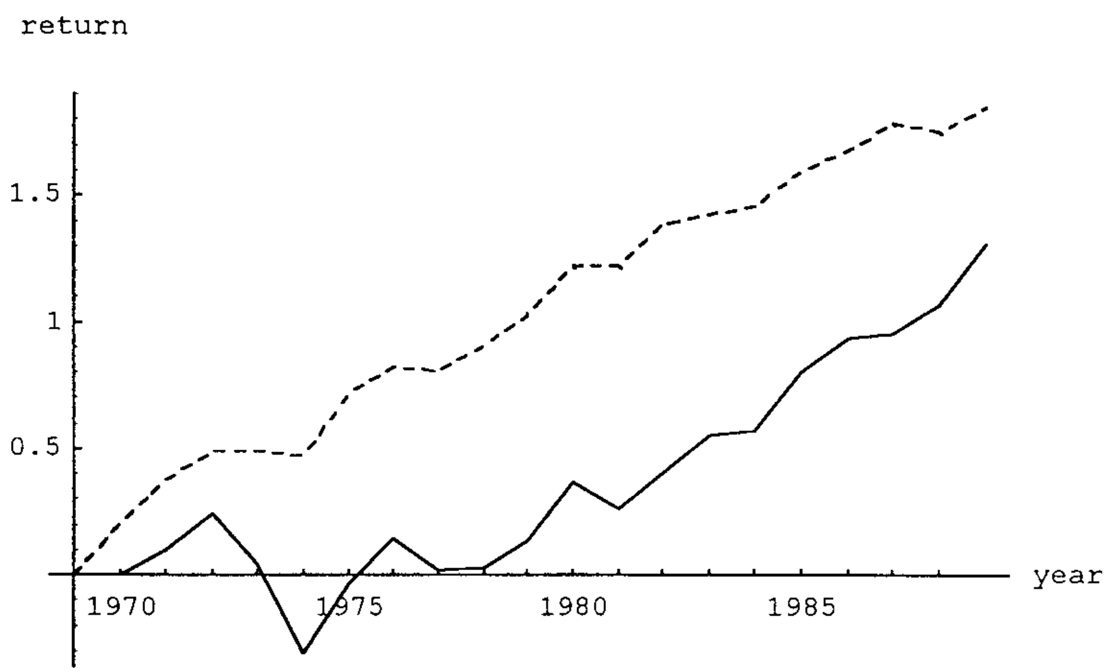

Figure 6. The cumulative return in the test period 1970-89 for the trading rules found by the genetic algorithm (dashed line), compared to the buy-and-hold strategy (solid line). The rules were found using the training period of 1964-67 and the selection period of 1968-69.

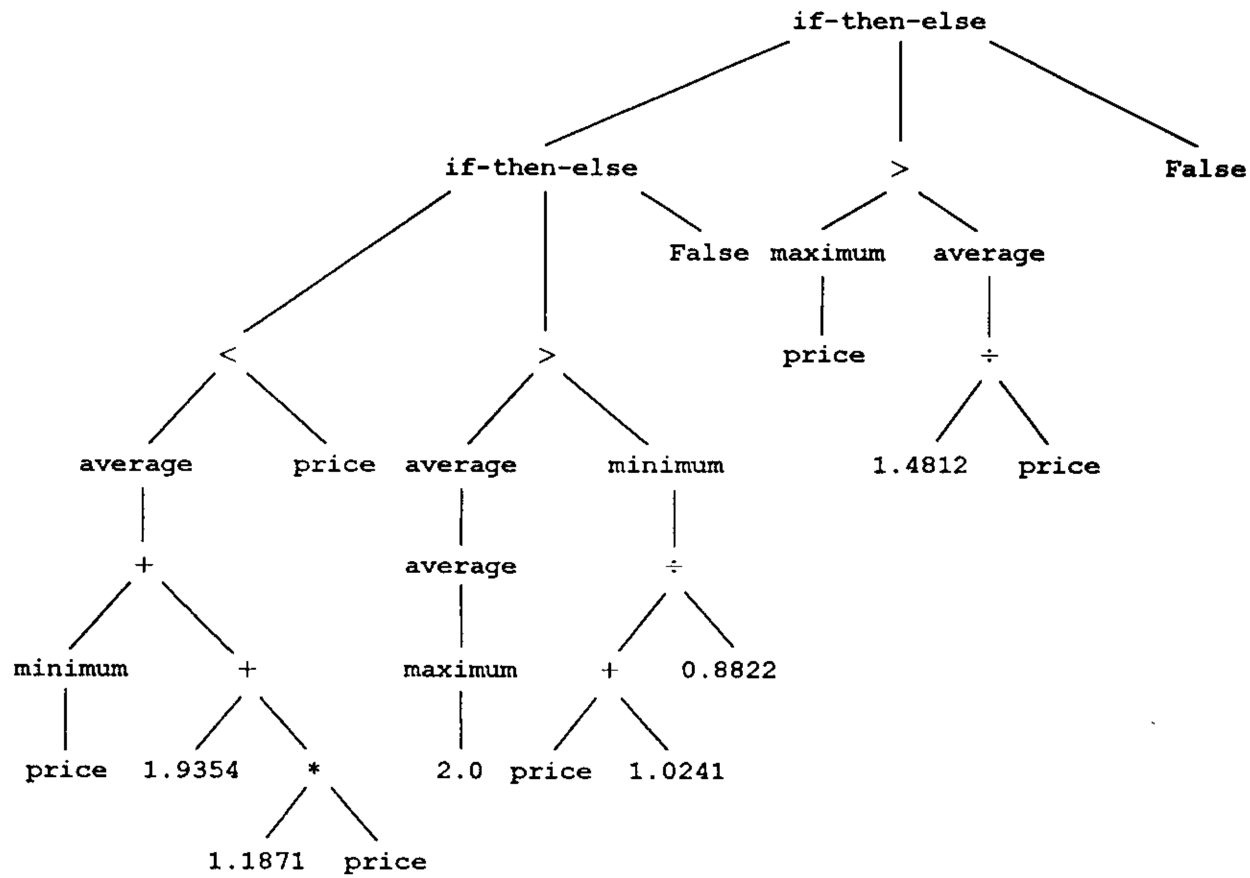

Figure 7. An example of the trading rules found by the genetic algorithm (rule 4 in table 2).

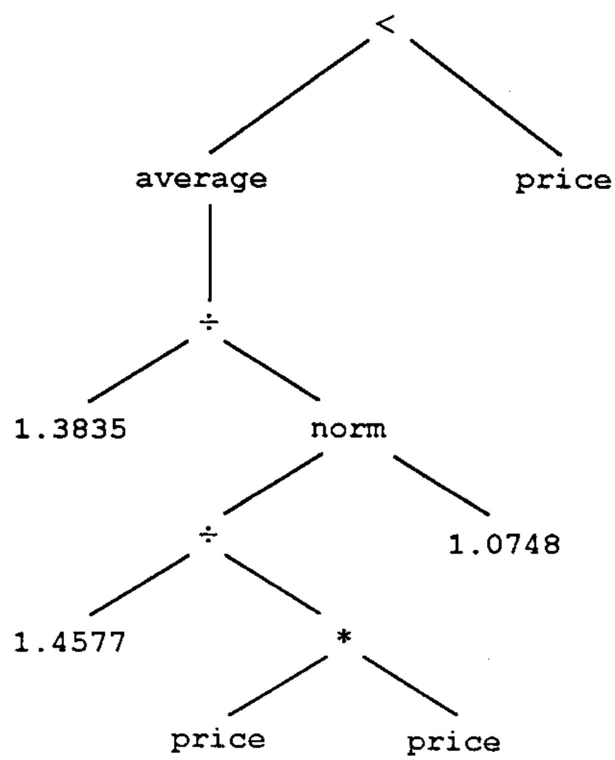

Figure 8. An example of the trading rules found by the genetic algorithm (rule 7 in table 2).

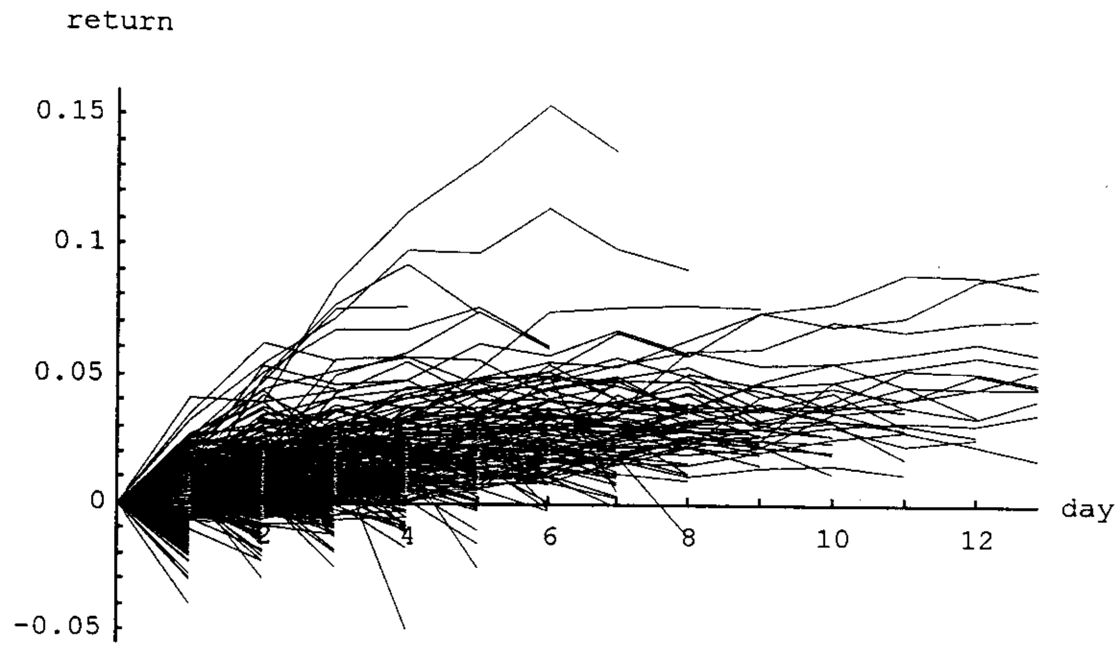

Figure 9. The cumulative market return during each trade "in" the market for rule 1 in table 2. The graphs for each of the 716 trades in the test period of 1970-89 are superimposed on top of each other.

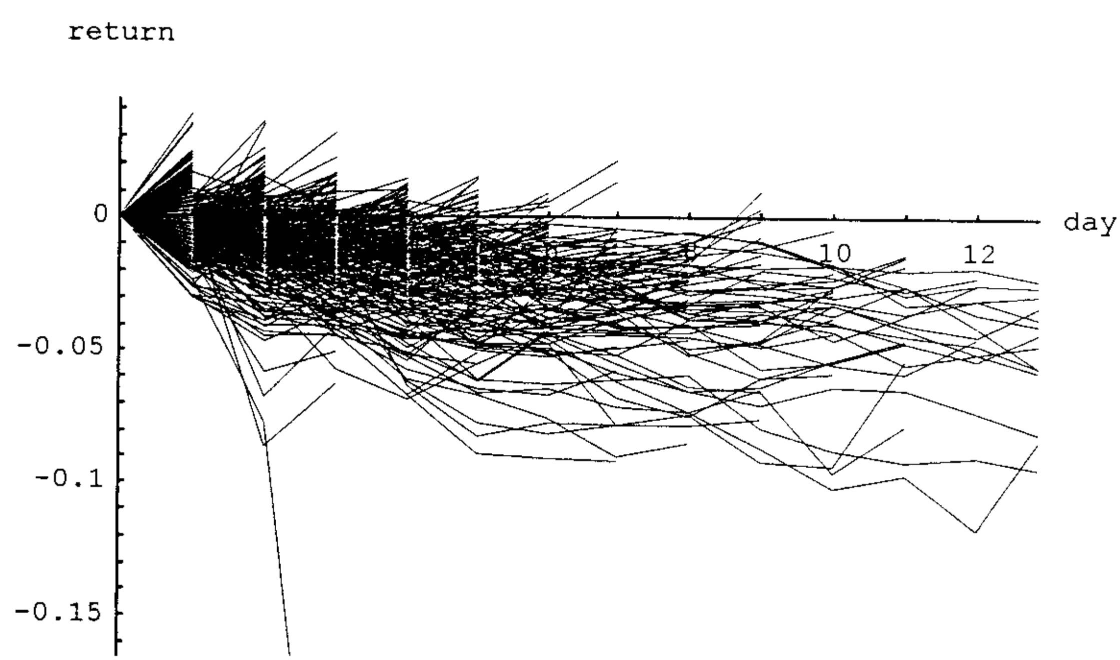

Figure 10. The cumulative market return during each trade "out" of the market for rule 1 in table 2. The graphs for each of the 715 trades in the test period of 1970-89 are superimposed on top of each other.

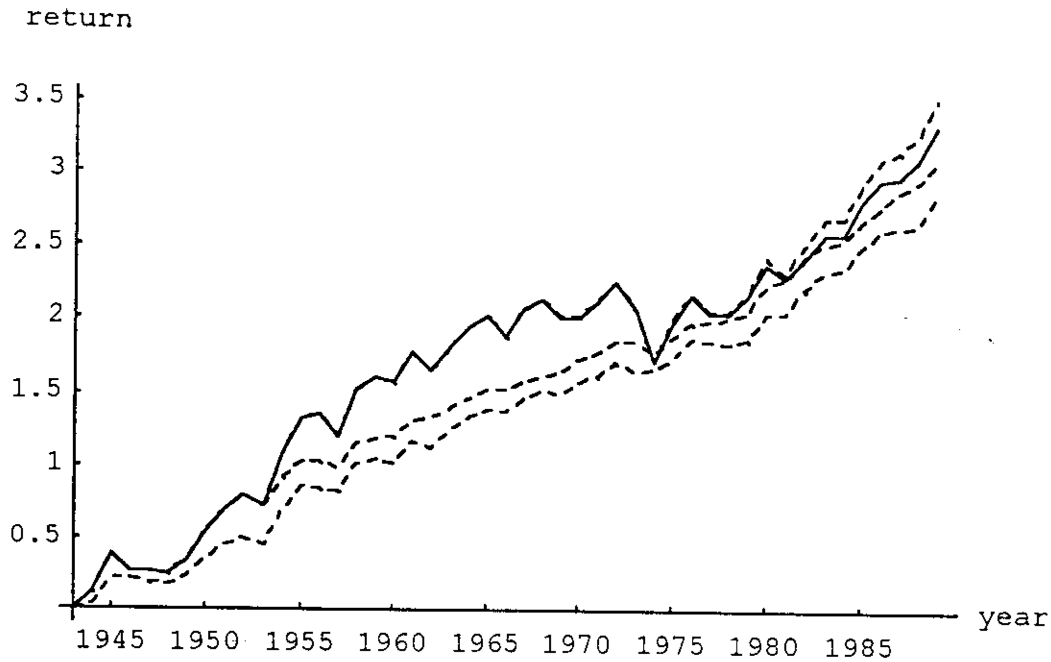

Figure 11. The cumulative return in the test periods 1944-89, 1954-89, and 1964-89 for the trading rules found by the genetic algorithm (dashed lines), compared to the buy-and-hold strategy (solid line). The rules were found using the training periods of 1929-38, 1939-48, and 1949-58, with selection periods of 1939-43, 1949-53, and 1959-63, respectively.

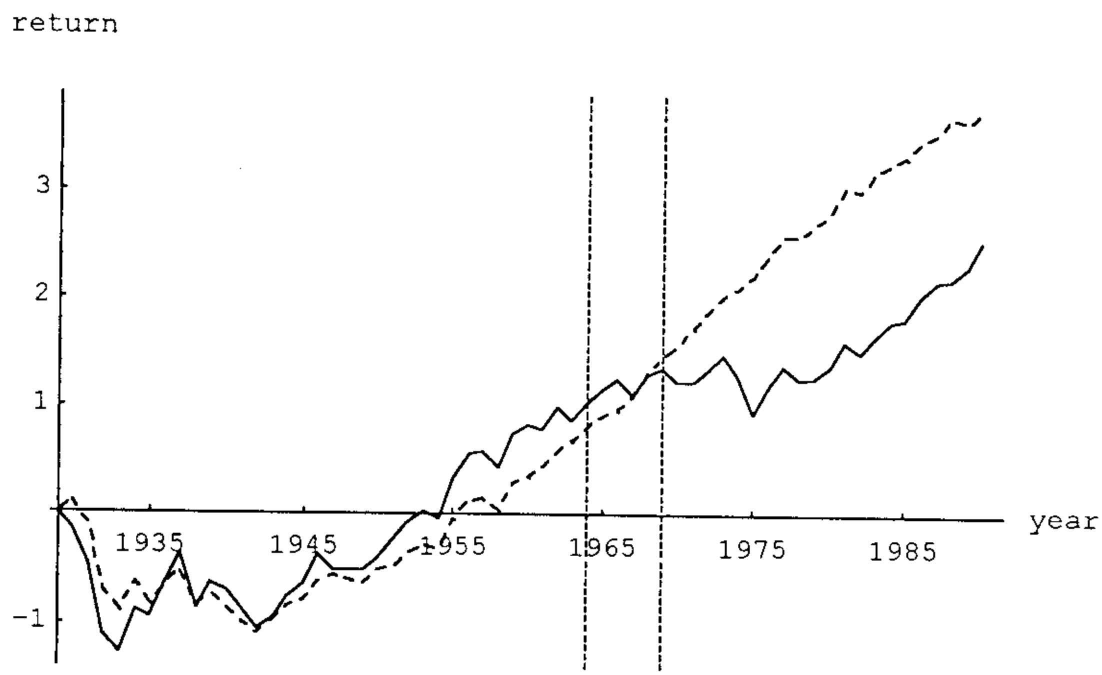

Figure 12. The cumulative return in 1929-89 for the trading rules found by the genetic algorithm (dashed line), compared to the buy-and-hold strategy (solid line). The rules were found using the training period of 1964-67 (indicated by the two dotted vertical lines) and the selection period of 1968-69.