Trading and Hedging in S&P 500 Spot and Futures Markets Using Genetic Programming

JUN WANG

In this study, genetic programming, an optimization technique based on the principles of natural evolution, was used to generate trading and hedging rules in Standard & Poor's 500 spot and futures markets. I adopted a realistic trading process that included reasonable transaction costs, obtainable execution prices, and all the unique features of futures trading. The results suggested that the spot market was quite efficient with most genetically generated trading rules duplicating the buy-and-hold strategy. Most of the trading activities of these trading programs were in the futures market, where transaction costs were substantially lower. The out-of-sample performance of these trading rules varied from year to year, indicating that genetic programming could not consistently find out-performing technical trading rules. Some evidence was found for the superior market-timing abilities of these rules. © 2000 John Wiley & Sons, Inc. Jrl Fut Mark 20:911–942, 2000

The author thanks Gerald D. Gay, Jayant Kale, Thomas Noe, Sailesh Ramamurtie, Michael Rebello, Rubin Saposnik, and Stephen Smith for their helpful comments. Special thanks are extended to Robert Webb and two anonymous referees for very insightful comments. The views expressed in this article do not reflect those of SAS Institute, Inc. All errors are the author's.
For correspondence, Jun Wang, SAS Institute, Inc., SAS Campus Drive, Cary, North Carolina, 27513; e-mail: Jonathan.Wang@sas.com

Received October, 1999; Accepted March, 2000

■ *Jun Wang is at SAS Institute, Inc. in Cary, North Carolina.*

INTRODUCTION

The traditional view of market efficiency suggests that price incorporates all available information, so it would not be profitable to trade on the basis of past information. For this reason, technical analysis, which is based on studying past prices and other information to forecast future prices, has not been well received by academics. However, technical analysis continues to receive wide support from investment practitioners, and the prevailing attitude of academics may also be changing after several empirical proofs suggesting that some strategies can beat the market.[1]

Empirical methods for studying financial markets range from ordinary least-square regressions to high-tech computing methods. Brock, Lakonishok, and LeBaron (1992) studied two of the simplest and most popular trading rules, moving average and trading range break, and found significant positive returns by utilizing the trading rules. Cooper (1999) showed that the filter strategy can outperform the buy-and-hold strategy under relatively low transaction costs. Others have documented stock return anomalies such as the weekend effect, the turn-of-the-month effect, the holiday effect, and the January effect.[2] A significant relationship between return and variables such as price-earning ratios, market-to-book ratio, size, and institutional ownership has also been documented.[3]

Driven by the development of computing technology, researchers have applied computationally intensive methods to the financial markets. The most popular method is the neural network.[4] The other method that is attracting much attention is genetic algorithms and genetic programming (GP). Neely, Weller, and Dittmar (1997; henceforth, NWD) used GP techniques to find technical trading rules in the foreign exchange market, and they found that the generated rules earned significant out-of-sample excess returns over the period from 1981 to 1995. Allen and Karjalainen (1999; henceforth, AK) applied GP to trade Standard & Poor's (S&P) 500 index from 1928 to 1995, but they did not find consistent excess returns over a simple buy-and-hold strategy in the holdout sample periods. Given this conflicting evidence on the capabilities of GP to find successful technical trading rules, several natural questions arise:

[1]There are many theories on why technical analysis can be useful, including the beauty contest theory by Keynes (1936), the informationally inefficient market by Grossman and Stiglitz (1980) and Grossman (1995), the information discovery by Brown and Jennings (1989), and the information herding by Froot, Scharfstein and Stein (1992).

[2]See, for example, Lakonishok and Smidt (1988).

[3]See, for example, Fama and French (1993) and Badrinath, Kale, and Noe (1995).

[4]See, for example, Trippi and DeSieno (1992) and Peterson and Peterson (1996).

- Is GP really useful in finding technical trading rules in financial markets?
- Are the inconclusive results in AK confined to the S&P 500 index only? What can be said about index futures?
- Can a real investor benefit from applying GP?

In this article, I address these questions by including the S&P 500 futures market. The S&P 500 index studied by AK is an index and is not directly tradable until the very recent emergence of index trusts, whereas the foreign exchange markets studied by NWD are highly liquid. The transaction cost incorporated by AK was also much higher than the cost used by NWD.[5] These may be the reasons one group did not find consistent out-performance of GP-generated trading rules after transaction cost and one group did. S&P 500 futures are directly traded and are highly liquid. The transaction cost in futures trading is also significantly lower than the cost in equity trading. There is also evidence that the futures market and the spot market are highly connected with strong price feedback.[6] By applying GP to find trading rules in S&P 500 futures, I shed more light on whether GP can be useful in another liquid market with lower transaction cost.

The focus of this article is whether a real investor can benefit from GP. As pointed out by AK and NWD, one big advantage of using GP as a search procedure is to assess the true out-of-sample performance of technical trading rules. The trading rules are generated *ex ante,* and GP serves as a search procedure to find the optimal rules without any hindsight. This contrasts with the common approach of studying technical analysis through the evaluation of the performance of some known trading rules. That these rules are known to the researchers may be due to their good *ex post* performance. This is the collective data snooping bias pointed out by Sullivan, Timmermann, and White (1999). Although researchers may find significant excess returns from these rules, an investor still does not know which rule to use at the beginning of a period. If an algorithm or a method with the ability to pick trading rules *ex ante* can be found, a real investor can use such an algorithm with much higher confidence. AK and NWD found conflicting evidence on whether GP is such an algorithm in two different markets. In this article, I examine whether GP can be useful to a real investor trading in the S&P 500 futures and spot markets. To assess this, I made sure that the transaction

[5]AK used a one-way transaction cost of 0.1, 0.25, and 0.5%, whereas NWD used a one-way transaction cost of 0.05%.

[6]See, for example, Kawaller, Koch, and Koch (1993).

costs for both spot trading and futures trading were realistic and that the prices at which trades took place were the ones obtainable by the investor.

Unlike AK and NWD, who evaluated the performance of the trading rules with only buy and sell signals on daily return series, I adopted a trading process in which trades were executed with prices with dollar profits or losses. Futures trading has some unique features such as mark to market, margin requirement, and contract expiration. My trading process with prices and dollar profit/loss allowed me to exactly duplicate the trading process of a real investor trading indexes and futures. All the features in futures trading and cash borrowing and lending were included in my trading process. In addition, the dollar-trading process enabled me to use more signals from the trading rules than buy and sell signals. Hence, a real investor will be able to execute the trading strategies generated by GP in the market. I then tested whether the trading rules found by GP could consistently outperform the buy-and-hold strategy.

To avoid data snooping bias from the selection of the time periods, I adopted the rolling forward approach first proposed by Pesaran and Timmerman (1995). When applying GP to learn trading rules for futures trading only or for both index and futures trading, I found that the performances of the trading rules in the out-of-sample periods varied over time. In some years, a majority of the rules beat the buy-and-hold strategy, whereas in other years, most rules underperformed the market. Because an investor has to use this procedure *ex ante*, it would be difficult for the investor to earn positive excess returns consistently. This result is similar to the findings of AK but contrary to the conclusions of NWD.

In addition, I studied a new application of GP that was not investigated by AK and NWD. Because many investors use futures to hedge positions in spot market, I tested whether GP could be used for finding hedging strategies. I specified the fitness measure that GP optimized as the mean return minus the variance multiplied by the risk-aversion factor. This fitness measure was the μ–σ^2 criterion and was the objective function of the investors if the distributions of asset returns were normal or if the investors were strictly variance averse.[7] In this way, the GP technique could be employed to find trading strategies that maximized returns and controlled the risk to a specified level. Although I found more trading rules that outperformed the market, the variation over time was still too big to give a conclusive answer.

Although my results may be disappointing to traders who would like to use GP, this article makes several contributions to the understanding

[7]For a detailed discussion on conditions supporting a μ–σ^2 criterion, see Loffler (1996).

of the markets and the evaluation of technical trading rules. In the spot market, the majority of the trading programs generated to maximize returns are duplicates of the buy-and-hold strategy. This is evidence that the stock market, at the index level, is quite efficient such that one cannot outperform the buy-and-hold strategy with realistic transaction costs. However, the GP-generated trading rules in the index futures market are more active at exploring lower transaction costs in futures trading. The annual variation of out-of-sample performances of these rules is again evidence for efficient markets. The rules display some market-timing abilities, and GP can find trading rules maximizing risk-adjusted returns. Finally, this article highlights the importance of the time-period selection in evaluating technical trading rules. Without the rolling forward scheme for period selection, one may come to a totally different conclusion. It is essential for anyone studying technical trading rules to use multiple samples for training, validation, and out-of-sample evaluation.

The article is organized as follows. The next section provides an introduction to GP. The setup is discussed in the third section, and the results are presented in the fourth section. The final section concludes the article.

GP

Koza (1992) developed GP, which shares many of the features of the genetic algorithms developed by Holland (1962, 1975). Both are optimization techniques based on the principles of natural evolution. In biological terms, there is an environment in which evolution occurs. This environment consists of a population of solution candidates, each of which is evaluated according to a problem-specific function, *fitness*, which measures how good the solution is to the problem. Then, the solution candidates are recombined through crossover and mutation to generate new solution candidates, or *offspring*. Similarly to natural selection, the candidates with higher fitness measures are more likely to reproduce offspring and keep their genes alive. The new generation of solution candidates are evaluated again, and they reproduce their offspring following the same selection process. This evolution process is stopped whenever certain terminating criteria are met.

The major difference between genetic algorithms and GP is the representation of the solution candidates. In genetic algorithms, a solution candidate is called a *chromosome*, which is a string of characters with a fixed length. For a specific problem, chromosomes are mapped to the specific solution domain of the problem. In GP, a solution candidate is represented as a treelike hierarchically structured computer program. Each node of the tree is a *function*, or building block, that takes the output of its branch nodes as inputs and produces an output. If a function requires no input, it is a terminal function that becomes a terminal node of the tree. The whole program is computed recursively from the terminal nodes to the root of the tree. The size, shape, and content of the tree can change dynamically during the evolution process. The dynamic nature of the solution structure makes GP more flexible and potentially more useful than genetic algorithms.

The success of genetic techniques as optimization tools has been well documented. Genetic methods conduct a parallel search of many points in the searching space. In addition, they implicitly process in parallel a large amount of useful information contained in each solution candidate. A solution candidate is composed of many pieces of *schemata*, or building blocks. The evaluation of the solution candidate generates information about the quality of these schemata. The Darwinian fitness-proportionate reproduction allocates an exponentially increasing weight to better schemata. This leads to an optimal search in the stochastic environment balancing between a promising search direction and a less visited direction. This parallelism makes the genetic method an efficient optimization technique for a number of complex problems. Several economists have used genetic algorithms to study the learning dynamics of genetic agents in different game settings.[8] It is well established that agents using genetic algorithms are able to find the optimal strategy in these games. The application of GP in economics is mainly in the area of generating smart computer traders because the tree structure of the solution candidates can be easily interpreted as the decision tree of human traders.[9]

SETUP

Using GP to Find Trading Rules

In this article, one solution candidate or program is represented as four hierarchical trees. The first two trees are automatically defined functions (ADFs; Koza, 1994). An ADF is one way to capture the idea of subroutine and reusable codes in computer programming. The ADF differs from other predefined functions in that it evaluates by traversing its own tree.

[8]Examples include Arifovic (1994, 1996), Ho (1996), and Noe and Pi (1995).

[9]See, for example, Allen and Karjalainen (1999), Andrews and Prager (1994), and Neely et al. (1997).

ADFs follow the same evolution process so they have the same feature of parallel search for the optimal subroutine. The advantage of ADF is that the same complex function can be called from multiple nodes by a single reference, so higher efficiency can be achieved. The third tree generates a signal to be used in the futures market, and the fourth tree generates a signal to be used in the spot market.

The building blocks consist of two types of functions, Boolean-valued functions (AND, OR, $>$, $<$, etc.) and real-valued functions ($+$, $-$, $\times$, /, maximum, minimum, average, lag, etc.). A complete list of the functions used can be found in Table I. The first tree is a Boolean-valued ADF, and the second tree is a real-valued ADF. The root of the third and fourth tree is restricted to be the function if–then–else. This ensures that most of the simple trading rules can be generated. Figure 1 shows two simple trading strategies, moving average and trading range break, represented in treelike form. The output of these two trees is transformed into the range [0,1] by $1/(1 + e^{-x})$, which is a common transformation used in neural networks.[10] Then, the transformed values are converted to trading signals. For the S&P 500 futures market, five signals are generated by the range [0,1] being divided into five equal segments. In descending order, the signals represent two contracts held long, one contract held long, no position, one contract held short, and two contracts held short.[11] For the S&P 500 spot market, five signals are allowed, corresponding to the five signals in the futures trading. As one S&P 500 futures contract represents 500 times the exposure of S&P 500, the trading signals are defined in the spot market as holding 1,000 shares, 500 shares, 0 shares, 500 short shares, and 1,000 short shares of the S&P 500 index. Short positions have been allowed in the spot market. For institutional investors, selling short does not appear to be a big problem. The other interpretation is that investors may own shares if the signal is neutral. In that case, short selling simply means reducing the exposure to stocks by selling a portion of the shares that would be owned with a neutral signal. In an earlier setup, short selling was restricted in the spot market, and the results did not change quantitatively.

Two operators of GP are used in addition to reproduction. They are crossover and mutation. *Crossover* is an operator to generate one new program from two existing programs (see Figure 2 for an illustration). In crossover, one tree, denoted the *n*th tree, is randomly picked from one program, as is one node from that tree. Then, another node of the same

[10]We used this transformation because it was easier to divide the range [0,1] into five segments than to divide the range $(-\infty,\infty)$. It did not have any effect on the outcome.

[11]Conforming to the practice, one contract represents the index multiplied by 500.

TABLE I

Building Blocks Used in Genetic Programming

Building Blocks	*Return Type*	*Input*	*Output*
AND	Boolean	2 Boolean inputs, c_1, c_2	c_1 AND c_2
OR	Boolean	2 Boolean inputs, c_1, c_2	c_1 OR c_2
XOR	Boolean	2 Boolean inputs, c_1, c_2	c_1 XOR c_2
$>$	Boolean	2 real inputs, c_1, c_2	True if $c_1 > c_2$, else false
$<$	Boolean	2 real inputs; c_1, c_2	True if $c_1 < c_2$, else false
True	Boolean	No input	Always true
False	Boolean	No input	Always false
ADF-1	Boolean	No input	Output by traversing the first tree
If–then–else	Real	One Boolean input, c_1, two real input, c_2 and c_3	If c_1 is true, output c_2, else output c_3
$+$	Real	Two real inputs, c_1 and c_2	$c_1 + c_2$
$-$	Real	Two real inputs, c_1 and c_2	$c_1 - c_2$
$\times$	Real	Two real inputs, c_1 and c_2	$c_1 \times c_2$
/	Real	Two real inputs, c_1 and c_2	c_1/c_2
Maximum of two	Real	Two real inputs, c_1 and c_2	Maximum (c_1, c_2)
Minimum of two	Real	Two real inputs, c_1 and c_2	Minimum (c_1, c_2)
Absolute value	Real	One real input, c_1	$\lvert c_1 \rvert$
Lag spot price	Real	One real input, c_1	c_1 lagged spot prices[a]
Moving upper bound	Real	One real input, c_1	Maximum of the spot prices in the past c_1 days[a]
Moving lower bound	Real	One real input, c_1	Minimum of the spot prices in the past c_1 days[a]
Moving average	Real	One real input, c_1	Average of the spot prices in the past c_1 days[a]
Lag futures price	Real	One real input, c_1	c_1 lagged futures prices[a]
Moving futures average	Real	One real input, c_1	Average of the futures prices in the past c_1 days[a]
Open interest	Real	One real input, c_1	Open interest on day c_1[a]
Futures volume	Real	One real input, c_1	Futures volume on day c_1[a]
Spot price	Real	No input	Closing spot price today
Futures price	Real	No input	Closing futures price today
ADF-2	Real	No input	Output by traversing the second tree
Constants	Real	No input	Real number from -1 to 1

[a]If c_1 is negative, we used $\lvert c_1 \rvert$ to get the value and multiplied by -1 as the output.

type is found randomly from the nth tree of the other program. The first picked node and its branches are replaced with the second picked node and its branches to generate a new program. *Mutation* is performed on one program by the replacement of the program with a randomly generated one.

The fitness measure of a program is its performance of the portfolio constructed based on its trade signals. In particular, I chose the μ–σ^2 criterion, that is, the mean daily return minus the variance of the return multiplied by a risk-aversion coefficient, k:

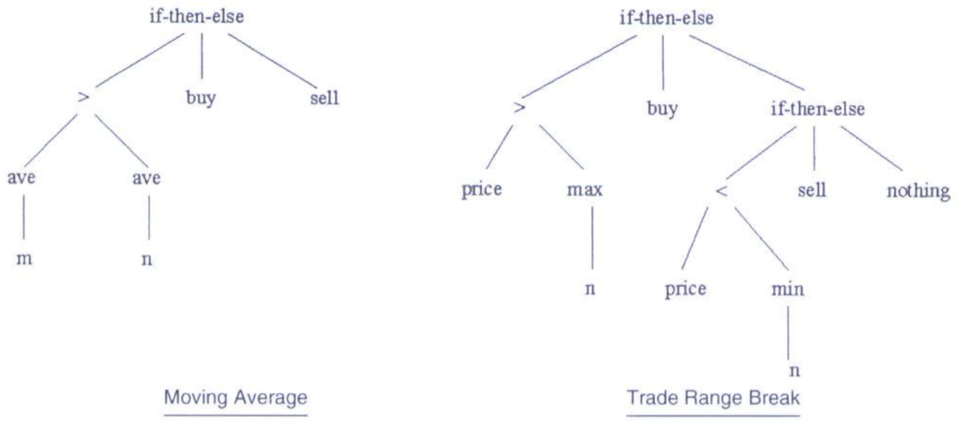

FIGURE 1
Examples of tree representations of simple trading rules.

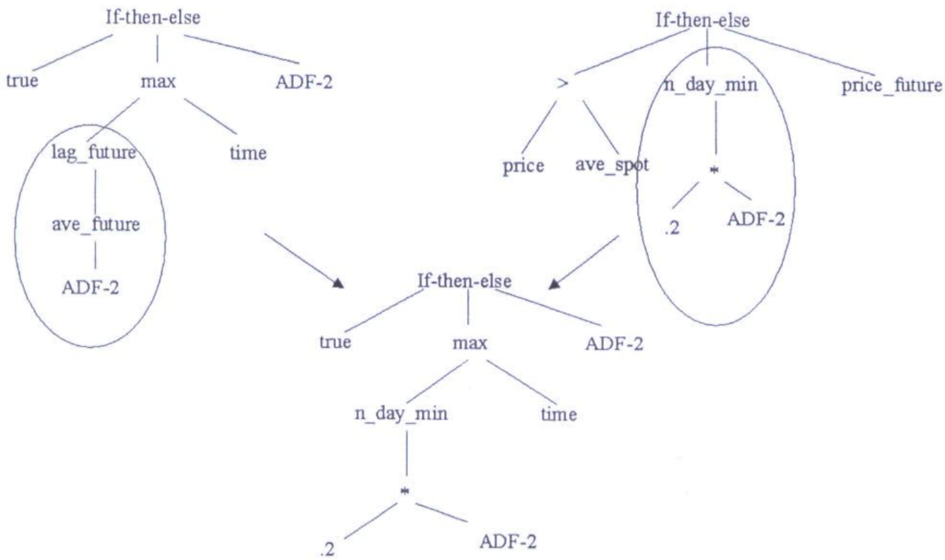

FIGURE 2
Example of a crossover operation.

$$F(\alpha) = E_s(r_t) - kVar_s(r_t) \tag{1}$$

where E_s and Var_s are the sample mean and sample variance, respectively. Details of how daily returns are computed is discussed in the following section. This fitness measure is the objective function of the investors if the distributions of asset returns are normal or if the investors are strictly variance averse. By using this fitness measure, the computer trader can be trained to trade with different risks tolerance. If k is zero, the investor is risk neutral; therefore, maximizing the return is his or her only trading objective. For a larger k, the investor is more risk averse so the fluctuation of daily returns becomes important in choosing a trading program.

As with other computationally intensive methods, there is a problem of *overfitting* in GP. This means that spurious relations can be picked up in the data series. To mitigate this problem, one run of GP uses two time periods. The first period is called the *training period* and is used to train the genetic programs. The second period is the *validation period*, which is used to select the best performing programs and decide when to stop the training.

One run of GP includes the following steps:

1. An initial population of 100 randomly generated programs is created. The programs are evaluated to obtain their fitness in the training period.
2. The program with the highest fitness is stored as the best rule of this run. This program is evaluated with the validation period to obtain its fitness in the validation period.
3. Two parent programs and a dropout program are selected from the population. The probability of a program being selected as a parent program is proportional to its adjusted fitness. The adjusted fitness equals the fitness if all fitness measures of the population are non-negative. If some fitness measures are negative, the adjusted fitness is the fitness of the program minus the lowest fitness of the whole population. The probability of becoming a parent program is then the adjusted fitness of this program over the sum of adjusted fitness of all the programs in the population. The probability of a program being selected as a dropout program is one minus the probability of becoming a parent program. In short, a program is more likely to be selected as a parent program if it has a higher fitness and more likely to be selected as a dropout program if it has a lower fitness. The two parent programs perform a crossover operation to generate a new child program to replace the dropout program in the population. With a probability of .001, the child program is mutated by the mutation operator. This child program is then evaluated in the training period to obtain its fitness. This selection–crossover–replacement sequence is repeated 100 times to form a new generation of genetic programs.
4. The fittest program in this generation is then evaluated in the validation period. If this program performs better in the validation period than the best program so far, it is saved as the best program of the

current run. This run is stopped if 100 generations have been generated or if the performance of the best program does not improve for 40 generations. Otherwise, return to Step 3 for the next generation.

The Trading Process

Because I wanted to model the trading costs and trading process as realistically as possible, all the bookkeeping of the trades is in dollars instead of returns. At the start, the trader receives a cash asset A. At day t, two trading suggestions are computed with all the information up to $t - 1$. Then, the trade is executed at the closing price of t, on the basis of the suggestions from the trading rule. This is a different trading process than the ones employed by AK and NWD. In both articles, the trading suggestions were generated on the basis of the information up to $t - 1$, and then the return from $t - 1$ to t was added to the total return, depending on the long or short position. This implies that the trader can establish the position with the closing price at $t - 1$; in other words, the opening price at day t is the same as the closing price of the previous day. This assumption certainly does not hold for most financial markets.[12] Actually, when AK used 1-day lagged return, that is, the return from t to $t + 1$, to compute the performance of the trading rules, they found that a large portion of the excess returns disappeared. The treatment of the trading environment in this article ensures that the profit can be actually realized by an investor using these trading rules.

In particular, I define the trading activities of 1 day as follows. First, interest is added to the cash account on the basis of the cash balance at $t - 1$ and the 3-month Treasury bill (T-bill) rate. This cash balance can be positive or negative depending on the past trading results. If the cash balance is negative, the trader is borrowing money to cover previous trading losses and trading costs. I made the assumption that the borrowing rate is the same as the lending rate. This is not a bad assumption for institutional traders. Second, future positions are marked to market on the basis of the position at $t - 1$ and the difference between the futures price of t and that of $t - 1$. The gain (loss) is added to (deducted from) the cash account. Only the most recent futures contract, which is the most liquid in the market, is used for trading. The switch from one contract to another contract occurs on the 1st trading day after the 15th of the expiring month (March, June, September, December) of the former

[12] In traders' jargon, the opening price higher than the closing price of the previous day is called *gap-up*, and the opening price lower than the closing price of the previous day is called *gap-down*.

contract.[13] If it is the switching day, the old position is set to zero. Then, the new futures position is established by following the futures trading signal. If this new position is different from the old position, transaction costs may be incurred. In futures trading, a two-way transaction cost structure is adopted, with F_f per transaction and F_p per contract. If the new position requires adding n contracts, a transaction cost of $F_f + nF_p$ is deducted from the cash account.[14] Finally, trade in the spot market is taken. The trader buys or sells the necessary shares to establish the suggested position. The transaction cost structure in the spot market is a one-way cost structure. Any time the trader buys or sells n shares, he or she incurs a total cost of $S_f + nS_p$, where S_f is the fixed cost per transaction and S_p is the cost per share. The cash for buying (selling) shares is drawn from (deposited into) the cash account, and the transaction cost is also deducted from the cash account at the transaction time. At time T, trading stops, all futures positions are closed, and all shares are sold.

The daily wealth of the trader (denoted w_t) is defined as the value of the stock holdings at day t and the cash balances after all the trading activities at day t. Future contracts are not counted in the daily wealth because they are marked to market every day. The daily return (r_t) of the trading program at t is then the continuously compounded return, $\log(w_t/w_{t-1})$. A bankruptcy constraint is also imposed so that whenever the daily wealth is negative, the trader is out of the markets and can no longer trade in either futures or spot markets.

The parameters used in this article are as follows: The starting asset value A is 1,000 times the index on the 1st day of the evaluation period. In the spot market, the trading costs are \$0.50 per share plus \$25.00 per transaction one way. In the futures market, the trading costs are \$36.00 per contract plus \$25.00 per round-trip transaction. This transaction cost structure is quite realistic in that the brokerage commission cost and the bid–ask-spread cost have been incorporated. Because the average index level during the study period was 427, the cost of \$0.50 per share plus the fixed cost corresponds to a one-way percentage transaction cost of 0.12%. This is close to the transaction costs assumed by AK and is certainly a realistic assumption for institutional investors. In the futures market, Fleming, Ostdiek, and Whaley (1995) pointed out that the av-

[13]The futures price series used by GP was constructed with the futures price of the most recent futures contract and a switch to the next contract on the switching day. The switching date and the futures price of the contract close to expiration were also stored.

[14]The margin requirement in futures trading is ignored in this setup. Because the trader can borrow at the T-bill rate and the initial margin can be posted with the T-bill, it will not have any wealth effect on the traders. If the margin does not earn interest, there will be a wealth effect. A margin requirement in the trading was included in an earlier setup, and the results were similar.

erage bid–ask spread was .0558, with round-trip commissions and exchange fees amounting to $6 per contract in the S&P 500 futures market. Thus, the total round-trip cost per futures contract is $33.90 (0.0558 × 500 + 6), less than the cost of $36.00 assumed in this test. In addition, by trading in the nearest futures contract only, one has enough liquidity to ensure the execution of the futures trade.

RESULTS

Data

I used the daily data of the S&P 500 index, S&P 500 futures, and 3-month T-bill rates from 1983 to 1998. The index data are from the Center for Research in Security Prices. The yield of 3-month T-bills was obtained from the data sources at the Federal Reserve Bank, St. Louis. The prices, open interests, and volumes of the S&P 500 futures contracts were obtained from the Commodity Futures Trading Commission (CFTC) and TurtleTrader.com.[15] Because at any date there usually existed four futures contracts, a single time series of future prices, open interests, and volumes was constructed with the nearest futures contract. The switch from one contract to the next occurred on the 1st trading day after the 15th of the expiring month of the old contract. For example, in February futures contracts expiring in March, June, September, and December of that particular year were traded. The price, open interest, and volume of the March contract were included in the time series. On March 16, the June contract was used for filling the time series of futures prices, open interest, and volumes. Figure 3 displays the time series of S&P 500 index and futures price. To make the input data of the trading rules reasonably scaled, I divided the index spot and futures prices by 100 and the open interest and volumes by 10,000 before feeding to the trading rules.

To guide against data snooping in the choice of time periods and to study whether a real investor can use this procedure to trade, I rolled forward the training period, validation period, and out-of-sample period similarly to Pesaran and Timmerman (1995), AK, and Cooper (1999).[16] In particular, I considered an investor at the beginning of 1987. The investor had 3 years of data available, and he or she defined the training

[15] I give special thanks to Dae Jung for providing me with the futures data from the CFTC from 1986 to 1993. The futures data from 1983 to 1985 and from 1994 to 1998 are from the Free Resources of TurtleTrader.com (http://www.turtletrader.co.uk/). The downloaded data were verified with the data from the CFTC to ensure their correctness.

[16] Special thanks to an anonymous referee for suggesting this roll forward approach.

FIGURE 3
Standard & Poor's 500 index and futures prices.

period to be 1984 and 1985 and the validation period to be 1986. Ten trials were run with periods specified this way, and 10 trading programs were generated. These 10 programs were used to trade in 1987, the out-of-sample period, and the results are reported later. At the beginning of 1988, all the periods rolled forward a year. The training period was 1985 and 1986, and the validation period was 1987. Ten trials were run, and the resulting 10 trading programs were used to trade in 1988. In this way, I rolled forward until 1998. Altogether, I ran 120 independent trials for one set of scenarios. The results reported are all out-of-sample results.

Trading in Futures Only

I first let GP generate 120 trading rules for S&P 500 futures only with the rolling forward scheme discussed previously. The risk-aversion coefficient k was set to 0, so the object was to achieve maximum returns. This was the same fitness measure used by AK and NWD. Hence, my results complement theirs in providing evidence on whether GP can be helpful in generating trading rules in a different financial market. The buy-and-hold strategy in this context was holding long two futures contracts all

TABLE II

Summary of the Results of the Trading Rules Generated to Trade in Futures Markets Only ($k = 0$)

Out-of-Sample Year	Buy and Hold				M			Mdn		
	r	σ	Nb	Ne	r	σ	Cost	r	σ	Cost
87	0.00277	2.9199	3	2	−0.01212	3.2597	4094.4	0.00113	3.0347	4462
88	0.03826	1.1812	1	1	0.01999	0.9606	2298.2	0.02626	1.1620	1067
89	0.11095	0.9636	2	0	0.01183	0.8851	2860.2	0.02346	1.0668	2187.5
90	−0.02696	1.0928	6	0	−0.00098	0.9741	6883.5	−0.01082	1.0236	7390.5
91	0.11202	0.9029	3	0	0.08689	0.9720	4309.3	0.07787	0.9981	5044
92	0.02598	0.6213	7	0	0.03444	0.5481	5593.2	0.03412	0.5828	5836
93	0.03666	0.5462	2	0	0.02157	0.5179	6228.1	0.01859	0.5453	6159.5
94	0.00443	0.6623	3	6	0.00804	0.6554	1513.5	0.00443	0.6623	485
95	0.11974	0.5390	1	0	0.01342	0.6459	4905.5	−0.01343	0.6641	4947
96	0.07311	0.8274	2	0	0.04316	0.8148	4645.8	0.04585	0.8114	4559
97	0.10797	1.2882	2	1	0.06858	1.3541	2602.1	0.10149	1.2896	1455
98	0.09577	1.4237	4	5	0.10342	1.3698	1830.9	0.09577	1.4237	582
sum			36	15						

The first column contains the out-of-sample years for which the trading rules were evaluated. The second and third columns contain the average daily returns (r) and standard deviation (σ) of the daily returns of the buy-and-hold strategy (holding long two futures contracts all the time) in the year. For each year, 10 trading rules were generated with the previous year as the validation period and 2 more years before the training period. *Nb* is the number of rules that produced better average daily returns than the buy-and-hold strategy in the out-of-sample year. *Ne* is the number of rules that produced the same average daily returns as the buy-and-hold strategy. The next three columns contain the mean of average daily returns, standard deviation of daily returns, and transaction costs of the 10 trading rules. The final three columns contain the median of average daily returns, standard deviation of daily returns, and transaction costs of the 10 trading rules. All daily returns are in percentages.

the time. I compared the GP-generated trading rules with this buy-and-hold strategy in the futures market. In Table II, I report the out-of-sample results of the 120 trading rules. Overall, 36 trading rules performed better than the buy-and-hold strategy in their corresponding out-of-sample years, and 15 rules were just buy-and-hold. The performance varied a lot for different years. For example, 7 out of the 10 rules outperformed the buy-and-hold strategy in 1992. The average daily return of the 10 rules was 0.034%. This was 32% higher than the daily returns of the buy-and-hold strategy, 0.026%. In 1995, the trading rules generated by the application of GP to learn in the previous 3 years performed much worse than the buy-and-hold strategies. Only 1 out of the 10 rules outperformed the buy-and-hold strategies. The performance of the GP-generated trading rules appeared to be sensitive to the time periods. The rules employed in 1995 were generated with 1992 and 1993 as the training period and 1994 as the validation period. As can be seen from the returns of buy-and-hold strategy from 1992 to 1994, the returns were very modest during this period. In 1995, however, the market rallied, and the strategies that were trained from the previous low return period did not perform well in this changing environment. On the contrary, when the out-of-sample period was similar to the training period and the validation period, GP could at least learn to use a buy-and-hold strategy if it could not find a better one. This was the case for 1998, when the returns in the previous 3 years were high. Five of the 10 trading rules trained from 1995 to 1997 were just buy-and-hold, and the other four rules showed better performance in the out-of-sample period, 1998.

Because a real investor would have no way of knowing whether the next period would be similar to the training periods he or she could see, the results in Table II cast doubt on whether GP can be used by a real investor to beat the buy-and-hold strategy in the futures market. This is consistent with the finding of AK but contrary to the results of NWD. The rolling forward scheme is very important for coming to this conclusion both in this article and in AK. In their early working paper (Allen & Karjalainen, 1995), they used one training period, one validation period, and one out-of-sample period, and they found that GP-generated rules outperformed the buy-and-hold strategy in the out-of-sample period. Once they used the rolling forward mechanism to prevent data-snooping bias in the selection of time periods, the results were no longer conclusive. Similarly, in a previous draft of this article I defined the training period as April 1986 to September 1987 and validation period as 1988 and 1989. Then GP-generated rules consistently beat the buy-and-hold strategy in the only out-of-sample period, 1990 to 1993. However, a real investor who would use GP is more likely to use the rolling forward scheme specified in this study. In this case, as shown in Table II, the investor cannot outperform the buy-and-hold strategy consistently. NWD did not use the rolling forward approach to test their results for different time periods. Hence, it is a subject for future research to test whether their results hold for only one period or GP can learn useful technical trading rules in the foreign exchange markets.

The other possible reason for the different conclusions drawn by NWD versus AK and this article is the choice of benchmarks. In this article and AK, the benchmark is the buy-and-hold strategy in the indexes or futures. In NWD, the benchmark is zero return because there is no well-defined buy-and-hold strategy in the foreign exchange market. To a certain extent, the buy-and-hold strategy can be considered a surviving strategy derived mainly from the U.S. equity market. As shown by Jorion and Goetzmann (1999), the real U.S. equity returns in the twentieth century are the highest among all countries. Hence, it may be difficult to beat the buy-and-hold strategy in the S&P 500 spot and futures markets,

TABLE III

Summary of the Results of the Trading Rules Generated to Trade in Both Spot and Futures Markets ($k = 0$)

Out-of-Sample Year	*Buy and Hold*				*M*			*Mdn*		
	r	σ	*Nb*	*Ne*	r	σ	*Cost*	r	σ	*Cost*
87	−0.03229	5.3564	4	2	−0.03216	5.3456	2994.2	−0.03229	5.3554	2549.5
88	0.03512	2.2470	4	1	0.03872	1.8286	4924.7	0.03476	2.1341	2512
89	0.16744	1.6059	0	0	0.01182	1.2036	6769.8	0.04246	0.9660	7491
90	−0.10112	2.4371	9	0	−0.04386	1.5043	6390.8	−0.04504	1.2317	5692.5
91	0.17659	1.5998	3	1	0.16708	1.4783	4008.7	0.16903	1.5717	2546
92	0.02666	1.2573	5	0	0.03988	1.0171	8388.7	0.03716	0.9669	9570.5
93	0.05136	1.0519	2	0	0.03281	0.8276	9339.6	0.03922	0.8929	5802.5
94	−0.02093	1.3171	3	6	−0.01833	1.2742	2147.5	−0.02093	1.3171	1535
95	0.19684	0.8953	0	0	0.06614	0.5381	9140.8	0.06768	0.4894	4348
96	0.11649	1.4986	1	0	0.07269	1.0399	7502.6	0.07924	1.1382	4065.5
97	0.17769	2.1766	5	1	0.17931	2.0915	2721.1	0.17774	2.1757	2214
98	0.15676	2.6061	3	6	0.16067	2.5351	5914.3	0.15676	2.6061	1535
sum			38	17						

The first column contains the out-of-sample years for which the trading rules were evaluated. The second and third columns contain the average daily returns (r) and standard deviation (σ) of the daily returns of the buy-and-hold strategy (holding long 1,000 shares of index and two futures contracts all the time) in the year. For each year, 10 trading rules were generated with the previous year as the validation period and 2 more years before as the training period. *Nb* is the number of rules that produced better average daily returns than the buy-and-hold strategy in the out-of-sample year. *Ne* is the number of rules that produced the same average daily returns as the buy-and-hold strategy. The next three columns contain the mean of average daily returns, standard deviation of daily returns, and transaction costs of the 10 trading rules. The final three columns contain the median of average daily returns, standard deviation of daily returns, and transaction costs of the 10 trading rules. All daily returns are in percentages.

but the task may be easier in other markets where GP can be more helpful.

Trading in Both Spot and Futures

In this section, I use GP to search for trading rules to trade simultaneously in both the S&P 500 spot market and futures market. I still assume a risk-neutral investor, i.e., $k = 0$. In this case, the buy-and-hold strategy is holding long 1,000 shares of index and two futures contracts all the time. The GP-generated trading rules using the rolling forward scheme are thus compared with this maximum exposure strategy in both markets. Table III provides a summary of the results of these 120 trading rules found by GP to trade in both markets.

The results were similar to the ones in the previous section. Out of the 120 trading rules, 38 outperformed the buy-and-hold strategy in the out-of-sample year, and 17 did the same as the buy-and-hold strategy. The same variation of performance in different time periods existed as well. In both 1994 and 1998, 3 of the 10 rules performed better than the buy-and-hold strategy, and the other 6 rules did the same as the buy-and-hold strategy. In 1990, the market was a down year, and the training period for the rules included the crash in 1987. The out-of-sample period was similar to the training period. Hence, rules learned from 1987 to 1989 were more likely to do well in 1990, as 9 out of 10 beat the buy-and-hold strategy. On the contrary, when the out-of-sample year was drastically different from the training and validation periods, the performance of GP-generated trading rules deteriorated. This was the case for 1989 and 1995.

Trading and Hedging in Both Spot and Futures

An investor can hedge his or her positions in the spot market using futures. In this section, I examine whether GP can find trading rules that provide better risk-adjusted returns. As previously specified, I chose the objective to be the μ–σ^2 criterion, that is, the mean daily return minus the variance of the return multiplied by a risk-aversion coefficient, k. In particular, I set k equal to 0.03.[17] When this risk-aversion coefficient was set at 0.03, the μ–σ^2 criterion for the buy-and-hold strategy in the index only was higher than that for the buy-and-hold strategy in both index and futures markets simultaneously. It was also higher than the μ–σ^2 criterion of investing in T-bills only. Usually, one would not suggest the strategy of obtaining as much exposure as possible in both the index and futures markets as a prudent strategy for most investors. Investing in T-bills only is also considered very conservative, and the performance of S&P 500 index has been the benchmark for a lot of portfolio analyses. In the sample period I studied , if an investor had a risk-aversion coefficient of 0.03 and maximized the μ–σ^2 criterion, he or she preferred the buy-and-hold strategy in the S&P 500 index to other strategies such as buy-and-hold in both the index and futures or just holding T-bills.

I trained 120 trading rules to maximize the μ–σ^2 criterion with k = 0.03, using the rolling forward approach previously specified. All these rules were then compared to the buy-and-hold strategy in the index for the respective out-of-sample year. Table IV provides the summary results of this scenario. Out of the 120 rules, 57 performed better than the buy-and-hold counterpart in risk-adjusted returns. In general, the GP-gen-

[17]In an earlier draft of this article, I varied k from 0 to 0.1 by 0.01 increments. Trading rules generated with high k had lower daily variations than rules found when k was low. Hence, GP appears to recognize the need to hedge when presented with such an objective.

TABLE IV

Summary of the Results of the Trading Rules Generated to Trade in Both Spot and Futures Markets ($k = 0.03$)

Out-of-	Buy and Hold					M				Mdn			
Sample Year	Fitness	r	σ	Nb	Ne	Fitness	r	σ	Cost	Fitness	r	σ	Cost
87	−0.15191	−0.01045	2.1719	6	0	−0.16380	0.03543	2.2933	5169.6	−0.08252	0.01598	2.0143	4613.5
88	−0.01121	0.02317	1.0706	4	0	−0.04691	0.01644	1.3310	4657.4	−0.03092	0.00899	1.1903	3419
89	0.07895	0.09998	0.8372	0	0	0.02262	0.02556	0.2712	2104.4	0.02890	0.02913	0.3286	1815
90	−0.06378	−0.03393	0.9975	8	0	−0.00235	0.00718	0.4106	3344.8	0.02771	0.02791	0.2704	1687.5
91	0.07753	0.10080	0.8807	6	0	0.06992	0.10513	1.0434	3014.9	0.08717	0.10945	0.9496	2570.5
92	0.00292	0.01421	0.6134	9	0	0.02325	0.02858	0.3439	3964.6	0.02191	0.02278	0.1971	1693
93	0.01859	0.02739	0.5414	8	0	0.01938	0.02405	0.3092	4252	0.01886	0.01955	0.1513	1535
94	−0.01923	−0.00764	0.6217	9	0	−0.00669	0.01766	0.8846	4626.2	−0.00605	0.01456	0.9027	4486
95	0.11065	0.11796	0.4934	0	0	−0.02688	−0.01356	0.6635	4626.2	−0.03655	−0.02213	0.6920	4470
96	0.05240	0.06900	0.7438	5	0	0.05299	0.07372	0.7860	10362.6	0.04834	0.06333	0.9066	4583
97	0.06511	0.10527	1.1569	2	0	0.00991	0.13230	2.0107	3713.7	0.03889	0.14837	1.9284	3135.5
98	0.04215	0.09176	1.2860	0	0	−0.04379	0.15295	2.5565	4249.6	−0.04753	0.15653	2.6061	2147.5
Sum				57	0								

The first column contains the out-of-sample years for which the trading rules were evaluated. The second, third, and fourth columns contain the fitness, average daily returns (r), and standard deviation (σ) of the daily returns of the buy-and-hold strategy (holding long 1,000 shares of index all the time) in the year. The fitness was computed as the average daily return minus k multiplied by the variance of daily returns. For each year, 10 trading rules were generated with the previous year as the validation period and 2 more years before as the training period. *Nb* is the number of rules that produced higher fitness than the buy-and-hold strategy in the out-of-sample year. *Ne* is the number of rules that produced the same fitness as the buy-and-hold strategy. The next four columns contain the mean of fitness, average daily returns, standard deviation of daily returns, and transaction costs of the 10 trading rules. The final four columns contain the median of fitness, average daily returns, standard deviation of daily returns, and transaction costs of the 10 trading rules. All daily returns are in percentages.

erated rules were likely to do better in those years when the market was down or neutral. For example, in the 3 down years of the S&P 500 index, 6 out of 10 rules outperformed the buy-and-hold strategy in 1987, 8 out of 10 outperformed in 1990, and 9 out of 10 outperformed in 1994. In 1992 and 1993, when the index moved up moderately, there were 9 and 8 rules, respectively, that provided higher risk-adjusted returns. In those years when the index moved up by a large percentage, very few rules did better than buying the index and holding it. The rules were generated with the training period before the year they were evaluated. Hence, all the performances were out-of-sample performances. Similar to the results in previous sections, the variation in the performance of GP-generated rules was quite big across different years.

To better demonstrate the performance variation of individual trading rules, Figures 4 and 5 plot the means and standard deviations of the trading rules for 2 out-of-sample years, 1992 and 1995. In 1992, 21 out of the 30 trading rules outperformed their buy-and-hold benchmark, whereas in 1995, only 1 out of 30 beat the benchmark.

Market-Timing Abilities

The portfolio created by following the GP-generated trading rules could be evaluated in the same way as mutual funds. The only difference was that here the portfolio manager was a computer trading rule. One traditional measure of performance is market-timing ability, that is, the ability to predict overall market returns. There are two traditional ways to measure market-timing ability. One is the quadratic regression of Treynor and Mazuy (1966):

$$r(t) - r_f(t) = \alpha + \beta(r_m(t) - r_f(t)) + \gamma(r_m(t) - r_f(t))^2 + \varepsilon(t) \quad (2)$$

where r, r_f, and r_m are the returns of the portfolio in evaluation, risk-free portfolio, and market portfolio. The coefficient γ is the measure of market-timing ability. The hypothesis of no abnormal market-timing ability implies that γ is zero, whereas a positive γ is an indication of high market-timing ability. The other model of market-timing is based on the regression proposed by Merton and Henriksson (1981):

$$r(t) - r_f(t) = \alpha + \beta(r_m(t) - r_f(t)) + \phi(r_m(t) - r_f(t))^+ + \varepsilon(t) \quad (3)$$

where the coefficient ϕ is the measure of market-timing ability. A positive ϕ indicates the ability to forecast when the market return will exceed the risk-free rate.

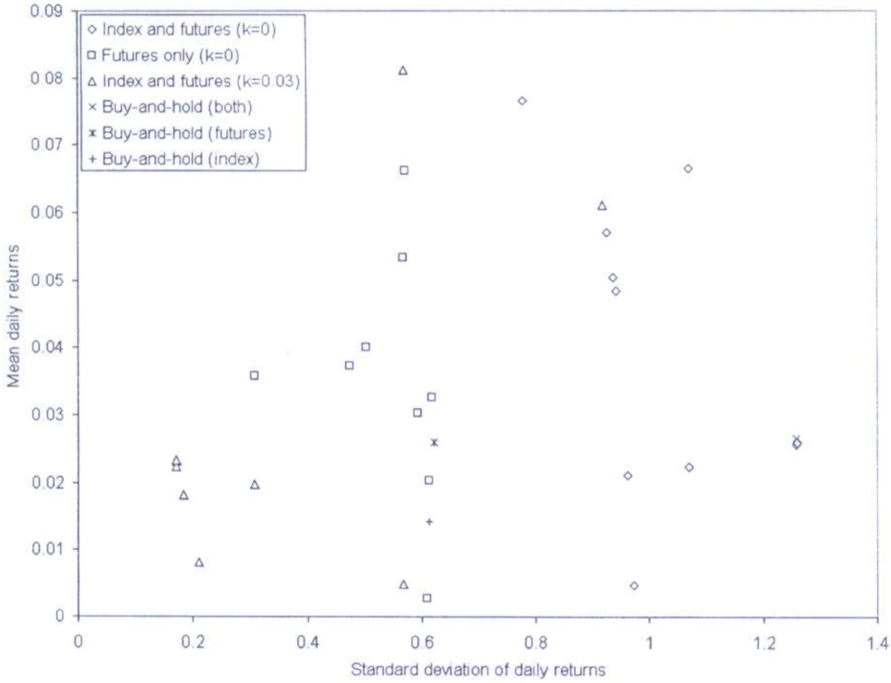

FIGURE 4
Return mean–variance for trading strategies in 1992.

These regressions were run for all 360 trading rules generated under the previous three scenarios with the return of the S&P 500 index as the market return and the return on T-bills as the risk-free return. Table V presents a summary of the market-timing ability measures of these trading rules. There was some evidence that GP-generated trading rules had some market-timing capability. Over 40% of the timing coefficients were significantly positive at the 5% level. This result was consistent with AK, who also found that GP-generated rules were able to identify periods when daily returns were positive and volatility was low.

I also evaluated the performance of these computer-generated trading rules using the conditional beta approach proposed by Ferson and Schadt (1996). The timing ability equations were run with the conditional beta. The results, however, did not change qualitatively from the unconditional version.

Trading Patterns

In this section, the trading patterns of these genetically generated trading programs are investigated. Table VI provides a summary of the average

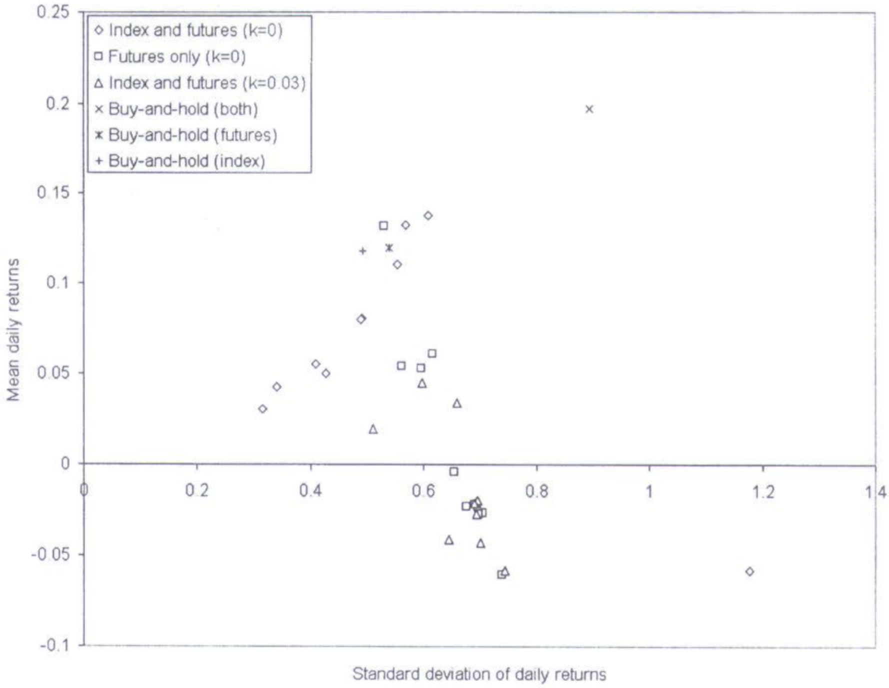

FIGURE 5
Return mean–variance for trading strategies in 1995.

daily positions and the total number of position changes of the 360 trading rules previously generated. When the objective was return only and both the index and futures markets were allowed to trade, the trading rules generated by GP were more likely to trade in the futures market because the transaction cost there was substantially lower. In the spot market, the average position of all the trading rules was close to 1,000, the position of maximum exposure allowed in my setup. The average number of position changes was less than three for 8 out-of-sample years and close to six in the other 4 years. A buy-and-hold strategy requires two position changes, one to establish the position at the beginning of the period and one to close the position at the end. Hence, the low average of position changes indicated that the trading rules did not trade much beyond the buy-and-hold strategy. In addition, 93 of them were exactly duplicates of buy-and-hold. In the futures market, most trading rules differed from the buy-and-hold strategy, with only 21 out of 120 rules duplicating the buy-and-hold strategy in futures market. The average number of position changes was much higher than two, the number of position changes for the buy-and-hold strategy. The results were similar for the trading rules that were allowed to trade only in the futures market.

TABLE V

Market-Timing Abilities of the Trading Rules

	Treynor and Mazuy			*Merton and Henriksson*		
Out-of-Sample Year	*M* γ	*Np*	*Nps*	*M* ϕ	*Np*	*Nps*
Panel A: Trading Rules Generated to Trade in Futures Markets Only (k = 0)						
87	−1.24657	1	1	−0.03515	5	1
88	0.17704	4	4	−0.03392	4	3
89	−0.87100	5	4	0.05791	5	4
90	2.86007	6	5	0.63102	7	5
91	7.09950	9	8	0.21318	9	4
92	9.10003	9	5	0.21454	9	5
93	1.61615	3	2	0.07832	4	2
94	−0.53756	2	1	−0.01969	2	0
95	11.96870	8	7	0.77937	7	7
96	2.44988	8	8	0.80626	8	8
97	0.52336	2	2	−0.06308	3	0
98	0.24513	8	1	0.07555	9	2
Sum		65	48		72	41
Panel B: Trading Rules Generated to Trade in Both Spot and Futures Markets (k = 0)						
87	−1.66486	1	1	−0.15428	1	1
88	−0.76438	3	3	−0.01225	3	3
89	0.06627	6	6	−0.14884	5	3
90	6.33187	7	7	1.38544	7	7
91	5.43551	9	8	0.20731	9	8
92	9.84977	10	3	0.23242	10	2
93	−5.54203	5	2	−0.06732	5	2
94	−1.01630	3	0	−0.03189	3	0
95	20.69446	10	10	1.27994	10	10
96	3.22876	10	9	1.04792	9	9
97	1.93607	2	2	0.09071	2	2
98	0.21933	1	1	0.06616	10	1
Sum		67	52		74	48
Panel C: Trading Rules Generated to Trade in Both Spot and Futures Markets (k = 0.03)						
87	1.84371	7	7	0.55929	8	6
88	−3.31400	3	3	−0.15811	3	3
89	−1.15309	5	0	−0.07228	4	0
90	1.26071	6	5	0.24236	5	4
91	5.80572	10	9	0.16070	9	7
92	5.57647	9	2	0.12791	9	2
93	8.93575	10	10	0.24674	10	3
94	9.77313	9	6	0.20673	9	4
95	10.26986	9	8	0.72633	9	8
96	3.94504	10	10	1.33527	10	10
97	−3.08857	0	0	−0.22135	1	0
98	0.35399	2	1	0.07053	9	2
Sum		80	61		86	49

The first column contains the out-of-sample years for which the trading rules were evaluated. The market-timing measure of Treynor and Mazuy (1966) is the γ in the following equation:

$$r(t) - r_f(t) = \alpha + \beta(r_m(t) - r_f(t)) + \gamma(r_m(t) - r_f(t))^2 + \varepsilon(t)$$

The second column contains the mean of γ of the 10 trading rules. The third column (*Np*) contains the number of trading rules with positive γ, and the fourth column (*Nps*) contains the number of trading rules with positive γ significant at the 5% level. The market-timing measure of Merton and Henriksson (1981) is ϕ in the following equation:

$$r(t) - r_f(t) = \alpha + \beta(r_m(t) - r_f(t)) + \phi(r_m(t) - r_f(t))^+ + \varepsilon(t)$$

The fifth column contains the mean of ϕ of the 10 trading rules. The third column (*Np*) is the number of trading rules with positive ϕ, and the fourth column (*Nps*) contains the number of trading rules with positive ϕ significant at the 5% level.

TABLE VI

Summary of the Positions and the Position Changes of the Trading Rules

Out-of-Sample Year	*Futures Only (k = 0)*			*Index and Futures (k = 0)*						*Index and Futures (k = 0.03)*					
	Futures			*Index*			*Futures*			*Index*			*Futures*		
	P	*Nc*	*Nh*	*P*	*Nc*	*Nh*	*P*	*Nc*	*Nh*	*P*	*Nc*	*Nh*	*P*	*Nc*	*Nh*
87	1.01	48.2	2	1000	2	10	1.55	19.3	2	508	3.3	6	0.13	44.6	1
88	0.92	33.5	1	997	2.8	8	1.08	46.6	1	650	2.2	5	0.57	53.4	1
89	−0.41	44.6	0	600	1.6	7	−0.48	65.3	0	−375	3.5	1	0.62	10.4	0
90	0.44	78.1	0	452	2.8	5	0.69	51.8	1	−163	3.6	1	0.60	26.5	2
91	0.78	42.7	0	946	2.8	8	1.47	26.3	2	251	1	2	1.23	31.8	2
92	0.86	71.1	0	999	2.8	9	0.71	72	0	−289	2.4	2	0.86	27.6	2
93	0.07	77.8	0	706	5.6	5	0.92	68.5	0	−850	2	0	1.35	31.1	6
94	1.71	13.5	6	1000	2	10	1.84	14.9	6	649	1.8	4	1.03	42.8	1
95	−0.55	52.9	0	860	6.3	9	−1.13	40.7	0	0.6	0.8	0	−0.99	44	0
96	0.78	52.2	0	572	5.2	4	1.08	40.2	1	145	9.5	3	0.62	31.2	3
97	1.69	24.9	1	999	2.2	9	1.70	13.7	2	949	2.2	8	1.03	31.5	2
98	1.75	20.5	5	961	6.2	9	1.99	4	6	982	4.7	7	1.84	12.8	6
Sum			15			93			21			39			26

The first column contains the out-of-sample years for which the trading rules were evaluated. The section *Futures Only (k = 0)* summarizes 120 trading rules in Table II. The section *Index and Futures (k = 0)* summarizes 120 trading rules in Table III. The section *Index and Futures (k = 0.03)* summarizes 120 trading rules in Table IV. The futures column contains the positions for trading in the futures market. The index column contains the positions for trading in the spot market. *P* is the average position over the out-of-sample year and across all 10 trading rules. (The possible positions in index trading are from −1000 to 1000, and the possible positions in futures trading are from −2 to 2.) *Nc* is the average position change over the out-of-sample year and across all 10 trading rules. *Nh* is the number of rules that mimic buy-and-hold strategy out of the 10 trading rules.

Hence, GP recognized that transaction costs were lower in the futures market than they were in the spot market, and the generated trading rules traded more often in the futures market while adopting the buy-and-hold strategy most of the time in the spot market.

When the object was maximizing risk-adjusted returns, the number of rules that mimicked the buy-and-hold strategy in the spot market dropped to 39. The average position was far less than 1,000 and even turned negative for 4 years. This indicates that the rules were taking less exposure and sometimes negative exposure to the index to reduce risk. The number of position changes was still low for the index market. This implies that most trading rules took on a position and maintained the position for a long period. The position may not be the maximum exposure because the goal was to maximize risk-adjusted returns. However, the negative effect of frequent trading given existing transaction costs was clearly recognized by these trading rules. The trading metrics in the futures market when $k = 0.03$ were similar to those when $k = 0$. Trading

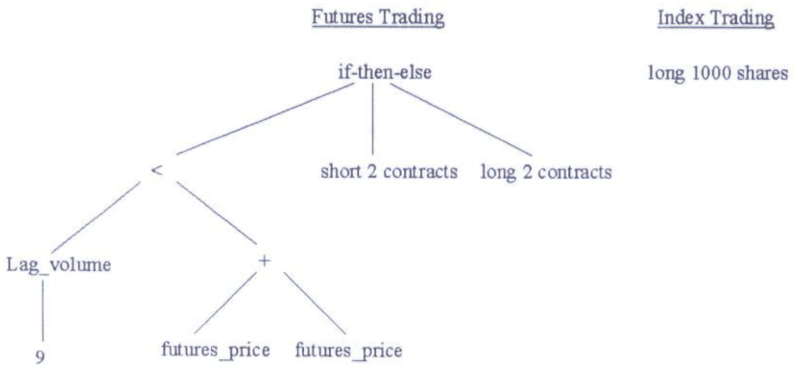

FIGURE 6
The reduced form of one trading rule found by genetic programming to trade in both spot and futures markets with $k = 0$.

was more active in the futures market, with average annual position changes ranging from 10 to 60.

One issue related to the time when the rules chose to be in the market is the dividend. Because the S&P 500 index was used to compute the market returns, the returns that investors received from the stock dividends were excluded. However, because the genetically generated trading programs with $k = 0$ prescribed an index trading strategy extremely close to the buy-and-hold strategy, they collected all the stock dividends. Hence, the performance did not change for these trading programs when the dividends were included in the return. When the risk-aversion coefficient k was higher, the periods that these rules chose to be in the market were shorter, and the dividend might change the relative performance over the benchmarks. However, because all these programs were generated under the assumption that there was no dividend, one could generate trading programs that achieved the same goal with the dividend included in the returns.

The program trees of the GP-generated trading rules were very complex and contained many redundant branches. I broke down two rules to provide a better understanding of the trading rules. Figure 6 shows the reduced form of the index and futures trading rule that had the best out-of-sample performance with a risk-averse coefficient of 0. This rule was generated with 1994 to 1995 as the training period and 1996 as the validation period. In the out-of-sample year, 1997, it achieved an average daily return of 0.216% with a standard deviation of 1.49. As shown in Figure 6, this rule actually was quite simple in its reduced form.[18] It

[18]Before reduction, the rule had 221 nodes.

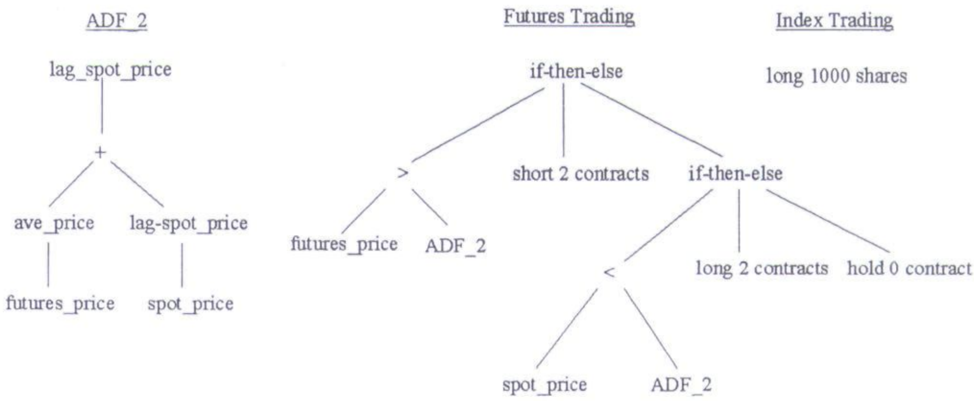

FIGURE 7
The reduced form of one trading rule found by genetic programming to trade in both spot and futures markets with $k = 0.03$.

adopted the buy-and-hold strategy in the index. In futures trading, it took two contracts held long all the time unless two times the futures price was greater than the 9-day lagged futures volume,[19] in which case it took two contracts held short. At first glance, this rule did not make too much sense in comparing price to volume. However, there is an explanation for the rule if I separate the relation between volume and futures price. That is, this rule predicted a market downturn when futures prices rose very high and the rise was achieved with low volume. The direct comparison of price and volume was probably the result of scaling coincidence.

The other trading rule I studied was the rule with the highest fitness measure when trading in both spot and futures market and the risk-aversion coefficient was 0.03. This rule was generated with 1993 to 1994 as the training period and 1995 as the validation period. In 1996, its average daily return was 0.144% with a standard deviation of 0.94. The rule in its reduced form is shown in Figure 7. In the futures market, this rule compared the futures price to some lag spot price computed from an ADF. If the futures price was greater, then it shorted two contracts. Otherwise, if the spot price was less than the lag price, it longed two contract and held no position if the spot price was greater than the lag price. The use of ADF made the representation of the trading rule simpler by avoiding duplication of the same branches. In the spot market, its strategy was also buy and hold. This strategy appeared to use futures to hedge the position in the spot market. When the futures price was too high, it shorted the futures to reduce exposure. When both the futures price and spot price dropped down from some previous levels, it longed two con-

[19]The volumes were divided by 10,000, and the prices were divided by 100 as input to trading rules.

tracts to double the exposure. If one looks at the trading strategy for the futures market only, this rule appears to have bet on market reversal. However, combined with the buy-and-hold strategy in the index, this rule was clearly hedging when the market got too high but became very aggressive during market dips.

DISCUSSION

It is important to recognize that the trading strategies, once created, were being evaluated *ex post*. The issue of finding trading strategies *ex ante* is an open question and attracts much research attention.[20] In this article, I am trying to address the question of whether a real investor can use GP as a search procedure to find useful trading strategies in the S&P 500 index and futures markets *ex ante*. The answer appears to be inconclusive. Because the out-of-sample performances of the trading strategies varied over the years, an investor would not have very high confidence at the beginning of the new year that the trading strategies trained from the past would do better than the buy-and-hold strategy. This result is consistent with the findings of AK that GP is not a magic procedure capable of outperforming the buy-and-hold strategies consistently. The rolling forward approach was essential to arriving at this conclusion. Without the rolling forward test over a long period, results could be biased by the selection of time periods.

There are several explanations for this result. One is market efficiency. The markets for the S&P 500 index and futures may be quite efficient so that it is difficult, if not impossible, for any search procedure to consistently find useful technical trading rules *ex ante*. The other explanation is related to the GP procedure in this article. It may be that there exist profitable technical trading rules but GP or my application of GP cannot find these rules. It is easy to find a computational trading method that does not work. For example, a trading rule generated randomly is most likely to have bad performance. However, I do not think this is the main cause of my results. As shown by all the references in the second section, genetic algorithms and GP have been applied to a wide range of tasks and have demonstrated their capability of finding optimal solutions. The parameter values I used in this application, such as parent selection probability, mutation probability, population size, and length of the run, are standard in most GP applications. I did not optimize these

[20]LeBaron and Weigend (1995) provided one method of finding a statistical distribution through a resampling method similar to bootstrapping.

parameters for my application, but my results did not change quantitatively for other sets of parameters.[21]

As shown from the timing test, the trading rules generated by GP displayed some timing capability. More than 40% of the rules had significant positive timing coefficients, indicating that the learning process of GP did have some power. In addition, because I did not add any buy-and-hold strategies in the initial population at the start of the run, the fact that many GP-generated rules adopted the buy-and-hold strategy in the index market was an indication of learning in progress. Given the transaction costs in the spot market, the best strategy that GP gravitated to was buy and hold. This was both an indication of the learning ability of GP and evidence that the index market was efficient if transaction costs had been taken into account. However, GP found trading rules that were much more active in the futures market because of the low transaction costs there. The two trading rules studied in the previous section both used the buy-and-hold strategy in the index and explored trading opportunities in the futures market. Both rules achieved superior risk-adjusted returns. All these suggested that GP had the ability to learn and the ability to locate useful trading rules. When the market did not change dramatically from previous years, the trading rules generated by GP compared favorably to the buy-and-hold strategy. It was when the market trend changed, especially when the market was changing from a down or flat market to an up market, that GP-generated rules were least likely to outperform the buy-and-hold strategy. Overall, my results suggest that GP can find successful trading rules in some years but can also find bad trading rules in other years. Because *ex ante* one cannot differentiate the two types of rules, the usefulness of GP as a search procedure is limited. Overall, I think the lukewarm results of the trading rules have more to do with the market being efficient than to my application of GP. At the very least, in using GP to trade in the S&P 500 index and futures market, I did not find evidence against efficient markets.[22]

CONCLUSION

In this study, GP, an optimization technique based on the principle of natural evolution, was used to find technical trading and hedging rules in the S&P 500 spot market and S&P 500 futures market. Specifically, I

[21]Optimization of the parameters to improve performance may introduce more data snooping bias because this involves the attempt to obtain better *ex post* performance by the adjustment of *ex ante* parameters.

[22]White (1988) came to the same conclusion with the application of neural networks.

tested GP in three scenarios: generating technical trading rules to maximize returns in the S&P 500 futures market, learning trading rules to maximize returns in both the S&P 500 index and futures markets, and searching for trading programs that maximized risk-adjusted returns (the mean returns minus the variance multiplied by the risk-aversion coefficient) in both markets. I adopted a realistic trading environment and prevented any hindsight when GP learned the trading rules. The training period, validation period, and out-of-sample period were rolled forward each year to simulate the decision-making process of a real investor. In each out-of-sample year, the performances of GP-generated trading rules varied a lot. There were some rules that outperformed the buy-and-hold strategy and some that did not. Overall, I do not have conclusive evidence that GP is a magic procedure capable of finding technical trading rules *ex ante*. My results extend AK's similar findings in the S&P 500 index to the S&P 500 futures and hedging strategies.

Interestingly, most trading rules are buy and hold in the index and execute most of the trades in the futures market. This suggests that there are relatively few opportunities in the spot market for GP with the transaction costs considered. However, because the transaction cost in the futures market is much lower, GP may find strategies that improve on buy-and-hold strategy. When the objective was risk-adjusted return, GP recognized the need to reduce the risk and generated trading programs with lower day-to-day variation. Nearly half of the trading rules outperformed the buy-and-hold strategy in the index by the μ–σ^2 criterion specified. However, given the variation of performance over the years, using GP to find hedging strategies did not lead to consistent results. The market-timing abilities of these genetically generated trading rules were also studied. There was some evidence for superior timing abilities of the rules, with over 40% of the rules having significantly positive timing coefficients.

Just as White (1988) pointed out in his work on neural networks, finding evidence against efficient markets is not easy. Traders who want to use GP may be disappointed by my results, whereas believers of efficient markets may not be surprised by them at all. However, this article extends the study of GP techniques by applying it to a new market, by adopting a realistic trading process, and by learning to maximize pure returns and risk-adjusted returns. Additionally, the importance of sample selection is highlighted in evaluating technical trading strategies and trading mechanisms. Note that AK, NWD, and this study all focused on timing the market with technical trading rules. Future research may need to focus more on asset selection than on market timing, as most professional traders would argue that it is much easier to pick stocks than to time market highs and lows.[23] Applying the similar techniques to locate better performing financial assets would be an interesting direction for future research.

BIBLIOGRAPHY

Allen, F., & Karjalainen, R. (1995). Using genetic algorithms to find technical trading rules. Unpublished manuscript, Wharton School, University of Pennsylvania, Philadelphia.

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

Andrews, M., & Prager, R. (1994). Genetic programming for the acquisition of double auction market strategies. In K. E. Kinnear (Ed.), Advances in genetic programming. Cambridge, MA: MIT Press.

Arifovic, J. (1994). Genetic algorithm learning and the cobweb model. Journal of Economic Dynamics and Control, 18, 3–28.

Arifovic, J. (1996). The behavior of the exchange rate in the genetic algorithm and experimental economics. Journal of Political Economy, 104, 510–541.

Badrinath, S., Kale, J., & Noe, T. (1995). Of shepherds, sheep and the cross auto-correlations in equity returns. Review of Financial Studies, 8, 401–432.

Boucher, M. (1999). The hedge fund edge: Maximum profit/minimum risk global trend trading strategies. New York: Wiley.

Brock, W., Lakonishok, J., & LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. Journal of Finance, 47, 1731–1764.

Brown, D. P., & Jennings, R. H. (1989) On technical analysis. Review of Financial Studies, 2, 527–551.

Cooper, M. (1999). Filter rules based on price and volume in individual security overreaction. Review of Financial Studies, 12, 901–935.

Fama, E., & French, K. (1993). Common risk factors in the returns of stocks and bonds. Journal of Financial Economics, 33, 3–56.

Ferson, W. E., & Schadt, R. W. (1996). Measuring fund strategy and performance in changing economic conditions. Journal of Finance, 51, 425–461.

Fleming, J., Ostdiek, B., & Whaley, R. E. (1995). Trading costs and the relative rates of price discovery in stock, futures, and option markets. Unpublished manuscript, Department of Finance, Rice University, Houston, TX.

Froot, K. A., Scharfstein, D. S., & Stein, J. C. (1992). Herd on the street: Informational inefficiencies in a market with short-term speculation. Journal of Finance, 47, 1461–1484.

Grossman, S. J. (1995). Dynamic asset allocation and the informational efficiency of markets. Journal of Finance, 50, 773–787.

Grossman, S. J., & Stiglitz, J. (1980). On the impossibility of informationally efficient markets. American Economic Review, 70, 393–408.

[23]See, for example, Boucher (1999).

Ho, T.-H. (1996). Finite automata play repeated prisoner's dilemma with information processing costs. Journal of Economic Dynamics and Control, 20, 173–207.

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

Holland, J. H. (1975) Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Ann Arbor: University of Michigan Press.

Jorion, P., & Goetzmann, W. N. (1999). Global stock markets in the twentieth century. Journal of Finance, 54, 953–980.

Kawaller, I. G., Koch, P. D., & Koch, T. W. (1993). Intraday market behavior and the extent of feedback between S&P 500 futures prices and the S&P 500 index. Journal of Financial Research, 16, 107–121.

Keynes, J. M. (1936). The general theory of employment, interest and money. London: Macmillan.

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

Koza, J. R. (1994). Genetic programming: II: Automatic discovery of reusable programs. Cambridge, MA: MIT Press.

Lakonishok, J., & Smidt, S. (1988). Are seasonal anomalies real? A ninety-year perspective. Review of Financial Studies, 1, 403–425.

LeBaron, B., & Weigend, A. (1995). Evaluating neural network predictors by bootstrapping. Unpublished manuscript, Department of Economics, University of Wisconsin, Madison.

Loffler, A. (1996). Variance aversion implies μ–σ^2 criterion. Journal of Economic Theory, 69, 532–539.

Merton, R. C., & Henriksson, R. D. (1981). On market timing and investment performance: II. Statistical procedure for evaluating forecasting skills. Journal of Business, 54, 513–534.

Neely, C., Weller, P., & Dittmar, R. (1997). Is technical analysis in the foreign exchange market profitable? A genetic programming approach. Journal of Financial and Quantitative Analysis, 32, 405–426.

Noe, T. H., & Pi, L. (1995). Genetic algorithm, learning, and the dynamics of corporate takeovers. Unpublished manuscript, Department of Finance, Georgia State University, Atlanta, GA.

Pesaran, M., & Timmermann, A. (1995). Predictability of stock returns: Robustness and economic significance. Journal of Finance, 50, 1201–1228.

Peterson, R. L., & Peterson, T. L., Jr. (1996). Neural network forecasts of S&P 500 prices: Implications for academic research and futures trading systems. Unpublished manuscript, Department of Finance, Texas Tech University, Lubbock, TX.

Sullivan, R., Timmermann, A., & White, H. (1999). Data-snooping, technical trading rule performance, and the bootstrap. Journal of Finance, 54, 1647–1692.

Treynor, J., & Mazuy, K. (1966). Can mutual funds outguess the market? Harvard Business Review, 44, 131–136.

Trippi, R. R., & DeSieno, D. (1992). Trading equity index futures with a neural network, a machine learning-enhanced trading strategy. Journal of Portfolio Management, 3, 27–33.

White, H. (1988). Economic prediction using neural networks: The case of IBM daily stock returns. Proceedings of IEEE International Conference on Neural Networks, II, 451–458.