AN APPLICATION OF GENETIC PROGRAMMING PARADIGM ON THE STOCK MARKET*

WEN-KUEI HSIEH

Department of Finance, De Lin Institute of Technology, No.1 Lane 380, Ching-Yuang Rd. Tu-Chen, Taipei 236, R.O.C.

SUNG-YI HSIEH

Department of Finance, De Lin Institute of Technology, No.1 Lane 380, Ching-Yuang Rd. Tu-Chen, Taipei 236, R.O.C.

Recent results indicate that the market timing approach may be a viable alternative to the buy-and-hold approach. In this paper, we propose genetic programming (GP) as a means to automatically generate such trading rules on the stock market. Based upon its performance, this paper gives a thorough analysis of the application of genetic programming to financial trading. Computational results, based on historical pricing, are reported for the Capitalization Weighted Stock Index (Taiwan).

1. Introduction

Technical analysis is aimed at devising trading rules capable of exploiting fluctuation on the financial markets. Recent results indicate that the market timing strategy may be a viable alternative to the buy-and-hold strategy, where the assets are kept over a relatively long time period. The market timing approach is more dynamic and focuses on market fluctuations. The trading rules, through technical analysis, are devised to generate appropriate buying and selling signals. The purpose of this paper is to demonstrate that genetic programming, in the field of evolutionary algorithms, can be exploited to automatically generate such trading rules.

The first results, using technical analysis in various financial domains, in the 1960s and 1970s supported the “efficient market hypothesis’, which implies that there should not be any exploitable pattern in the data (Fama [1], Jensen and Bennington [2]). Some recent results seem to indicate otherwise. Such as Brock et al [3], followed by Bessembinder and Chan [4], also demonstrated the simple trading rules could be profitable (but, without transaction costs).

Nevertheless, these developments are based on a priori rules determined through technical analysis. The emergence of new technology, in particular evolutionary algorithms, allows a system to automatically generate and adapt

---

* The authors are grateful to the AI-ECON Research Center of National Chengchi Univ., Taiwan, for their supporting of the “Genetic Programming Simulation System.”

trading rules to particular applications. Genetic algorithms (Holland [5]) have already been applied to a number of financial applications (also Bauer [6]). For learning trading rules, the genetic programming (GP) approach of Koza [7] looks more promising, as it provides a flexible framework for adjusting the trading rules. Although, the first attempts by Chen and Yeh [8] and Allen and Karjalainen [9] on the stock exchange markets did not show any excess returns with regard to the buy and hold approach, other recent applications of GP are more encouraging (Neely et al. [10], Neely and Weller [11], Marney et al. [12].

The remainder of the paper is the following. Section 2 gives a brief introduction to genetic programming (GP) and the experimental designs in terms of the setting of the control parameters. The fitness function is fully discussed in Section 3. The data used in this paper associated with its preprocessing is detailed in Section 4. Computational results obtained with the Capitalization Weighted Stock Index (Taiwan) are reported in section 5. Then conclusion follows.

2. Genetic Programming

2.1. *Genetic Algorithm and Genetic Programming*

Genetic algorithm (Holland [6], Michalewicz [13]) is a randomized search procedure working on a population of indivicuals or solutions encoded as linear bit strings. This population evolves over time through the application of operators which mimic those found in nature, namely, selection of the fittest, crossover and mutation.

Genetic programming (Koza [7], Koza et al. [14]) extends, a recent development in the field of evolutionary algorithms, classical genetic algorithms by allowing the processing of non-linear structures. Mean while, genetic programming extends the above paradigm by allowing the evolution of programs encoded as tree structures. These programs are constructed from a predefined set of functions and terminals.

2.2. *Generating trading rules with GP*

In financial trading, each computer program represents a trading program, and the society of computer programs represents a collection of trading programs. The population size, denoted by Pop, is a key control parameter of GP. The evolution of the population of the trading program proceeds in a cycle. Each cycle is counted as one generation. The maximum number of generations (Gen) combined with Pop gives the maximum search resources to be used in the discovery process. In this paper, Pop is set to 500, and Gen is set to 100 (Table 1).

The content of each trading program is determined by a list composed of atoms. The atoms can be functions (operators), variables or constants (operands), and can be a list as well. The functions and variables must be declared at the beginning, and are known as the function set and the terminal set. Our choice of the function set and the terminal set is in line with [9], [10], [11]. The idea is to satisfy the closure property [7]. A collection of these functions becomes our function set, and a collection of these variables or constants becomes our terminal set (see Table 1). These functions and terminals are sufficient enough to span both any simple technical trading rules and the potential complex ones. In addition to the price, the functions and terminals can be enriched in several directions; for example, we can include other variables in the terminal set such as volatility, volume, technical indexes, as well as some well-known trading rules.

With the function set and terminal set specified in Table 1, we start the evolution by initializing a population of Pop random trading programs, called the initial population. Each trading program in then applied to the market and to give the trading recommendation. Based on the recommendation and based on the data, we measure its fitness. We then rank all trading programs of the current generation by their fitness. The top-ranked k programs are selected, and are further tested over another unseen data set, called the validation set (see Section 4). Their fitness is calculated, and they are saved in a list of winners. The standard genetic operators, including reproduction, crossover and mutation, are then applied to the current generation of the programs and in order to generate a new generation.

After the emergence of the new generation, we then follow the same procedure to measure the fitness of each trading program of the new generation. The best k models are selected as the new list of winners. This finishes the evolution for one generation, and they cycle starts over again by generating the next new generation of trading programs, and this cycle goes on and on until it meets a termination criterion.

Table 1. Control parameters of GP.

Population size (Pop)	500
Initialization	Ramp-half-and-half
Offspring trees created By crossover By point mutation By tree mutation (grow method) By elite By reproduction	50% 20% 20% 0.2% 9.8%
Function set	+, -, x, / Norm, average, max, min, lag And, or, not, >, < If-then-else, true, false
Terminal set	Price, constants
Selection Scheme	Tournament selection (size=2)
Termination criterion Number of generation (Gen) Stagnation tolerance (g)	100 50
Validation Number of best models saved (k)	1
Fitness Fitness function Transaction cost (c)	Equation (2) (stock market) 0.5% (stock market)

There are two termination criteria. One is the maximum number of generations (Gen), and the other is the maximum number of consecutive stagnations (g). A stagnation is defined as a state in which none of the k incumbents in the winners list has been replaced by any from the new generation of programs. In this paper, k is set to one, the same as [9] and [10], where g is set to 50, which is twice the size of [9] and [10].

3. Fitness Function: Return of Investment in the Stock Markets

The calculation of the investment return, fitness function, is defined as follows. Let P_t be the Capitalization Weighted Stock Index (Taiwan) on day t, i_t be the interest rate on day t, and the return for day t is π_t:

$$\pi_t = \begin{cases} \ln(P_t) - \ln(P_{t-1}) & \text{in the marke} \\ \ln(1+i_t) & \text{out of the market} \end{cases} \qquad \text{Eq. (1)}$$

Let n denote the total number of transactions, i.e. the number of times a True (in the market) is followed by a False (out of the market) plus the number of times a False (out of the market) is followed by a True (in the market). Also, let c be the one-way transaction cost. The rate of return over the entire period of T days, as an arithmetic sum, is

$$II = \sum_{t=1}^{T} \pi_t + n * \log \frac{1-c}{1+c} \qquad \text{Eq. (2)}$$

However, it can be show that the total return, based on a continuously compounded return (geometric sum), is

$$R = \exp^{II} - 1 \qquad \text{Eq. (3)}$$

4. Data and Data Preprocessing

The Capitalization Weighted Stock Index (Taiwan) is used in this paper. We use the daily data from January 1989 to December 2006. As to the risk free interest rate, we mainly consider the rate of the inter-bank overnight loan from Central Bank of Taiwan. Data for stock index are made available from Info-Times databank.

Due to the non-stationary nature of the data, it is, therefore, desirable to transform the original series into a stationary series. Since we use the first year of data for normalization, the usable data starts from 1990, 1993, 1996, and 1998.

1.1990-1993, 1994-1995, 1996-1998	training	validation	test
2.1993-1996, 1997-1998, 1999-2001	training	validation	test
3.1996-1999, 2000-2001, 2002-2004	training	validation	test
4.1998-2001, 2002-2003, 2004-2006	training	validation	test

Figure 1. The use of stock data: training, validation, and testing

We adopt a very standard way of decomposing the whole dataset into three sections, namely, training set, the validation set, and the testing set. To guard against potential data snooping in the choice of time periods, we use four successive training periods, validation periods, and test periods, as shown in Figure 1. The four-year training and two-year validation periods start in 1990, 1993, 1996, and 1998, with the out-of-sample test periods starting in 1996, 1999, 2002, and 2004. For example, the first trail uses the years 1990 to 1993 as the training period, 1994 to 1995 as selection period, and 1996 to 1998 as the test period. For each of four training periods, we carry out fifty trials.

5. Computational Results

The results obtained with GP on the Capitalization Weighted Stock Index (Taiwan) as well as the bench mark return are shown in Tables 2, and 3. For each period, the numbers are the average of 50 different runs.

5.1. *The Return of Investment: GP and the Capitalization Weighted Stock Index (Taiwan)*

The return rates of GP and The Capitalization Weighted Stock Index (Taiwan) itself are 0.8661, 0.0167 during testing period 1996-1998; 0.0472, 0.0066 during testing period 1999-2001; 0.0104, 0.0038 during testing period 2002-2004; 0.0152, 0.0069 during testing period 2004-2006. Among these four test periods, the return rates of GP are significantly higher than those of The Capitalization Weighted Stock Index, Taiwan (Table 2).

5.2. *The Return of Investment: GP and Buy and Hold*

The return rates of GP and B&H are 0.8661, 0.0824 during testing period 1996-1998; 0.0472, -0.0326 during testing period 1999-2001; 0.0104, 0.0321 during testing period 2002-2004; 0.0152, 0.0983 during testing period 2004-2006.

Table 2. The return of investment: GP and the Capitalization Weighted Stock Index (Taiwan).

Testing	GP	CWSI (Taiwan)
1996-1998	0.866053368 **	0.016695833
1999-2001	0.047192432**	0.006599167
2002-2004	0.010355007**	0.003799167
2004-2006	0.015169966**	0.006855000

Table 3. The return of investment: GP and Buy and Hold.

Testing	GP	Buy and Hold
1996-1998	0.866053368 **	0.08241871
1999-2001	0.047192432**	-0.03257200
2002-2004	0.010355007	0.03212114
2004-2006	0.015169966	0.09832800*
total	**0.938770773**	**0.18029585**

As a result, among these four test periods, the return rates of GP are much higher than those of B&H, by two testing periods. In fact, the overall return of GP is significantly higher then that of B&H (Table 3).

6. Conclusion

This paper proposed an application of the Genetic Programming (GP) paradigm for automatically generating trading rules on the Capitalization Weighted Stock Index (Taiwan) and gave a thorough analysis of its application to financial trading. The results show that the overall return of GP is much higher then that of buy and hold when the market is in a slight fluctuating pattern or when it is stable.

References

1. E. F. Fama, Efficient capital markets: a review of theory and empirical work, *Journal of Finance* **25**, 383-417 (1970).
2. M. Jensen and G. Bennington, Random walks and technical theories: some additional evidences, Journal of Finance 25, 269-82 (1970).
3. W. Brock, J. Lakohishok and B. Lebaron, Simple technical trading rules and the stochastic properties of stock returns, *Journal of Finance* **47**, 1731-64 (1992).
4. H. Bessembinder, K. Chan, The profitability of technical trading rules in the Asian stock markets, *Pacific Basin Finance Journal* **7**, 257-84 (1995).
5. J. H. Holland, Adaptation in natural and artificial systems, Ann Arbor, *MI: The Univ. of Michigan Press*, (1975).
6. R. J. Bauer, Genetic algorithms and investment, *New York: Wiley*, (1994).
7. J. R. Koza, *Cambridge, MA: MIT Press*, (1992).
8. S. H. Chen and C. H. Yeh, Toward a computable approach to the efficient market hypothesis: an application of genetic programming, *Journal of Economic Dynamics & Contorl* 21, 1043-63 (1996).
9. F. Allen and R. Karjalainen, Using genetic algorithms to find technical trading rules, Journal of Financial Economics 51, 245-71(1999).
10. C. Neely, P. Weller and R. Dittmar, Is technical analysis in the foreign exchange market profitable? a genetic programming approach, *Journal of Financial and Quantitative Analysis* 32, 405-26 (1997).
11. C. Neely and P. Weller, Technical trading rules in the European monetary system, *Journal of International Money and Finance* 18, 429-58 (1999).
12. J. P. Marney, C. Fyfe, H. Tarbert and D. Miller, Risk adjusted returns to technical trading rules: a genetic programming approach. *Computing in Economics and Finance, Society for Computational Economics, Yale Univ., USA, paper #147* (2001).
13. Z. Michalewicz, Genetic algorithms+data structure =evolution programs, *3rd ed. Berlin: Springer*, (1996).
14. J. R. Koza, F. H. Bennett III, D. Andre and M. A. Keane, Genetic programming III: Darwinian invention and problem solving *San Francisco, CA: Morgan Kaufmann*, (1999).