STATISTICAL ANALYSIS OF GENETIC ALGORITHMS IN DISCOVERING TECHNICAL TRADING STRATEGIES

Shu-Heng Chen and Chueh-Yung Tsao

ABSTRACT

In this study, the performance of ordinal GA-based trading strategies is evaluated under six classes of time series model, namely, the linear ARMA model, *the* bilinear model, *the* ARCH model, *the* GARCH model, *the* threshold model *and the* chaotic model. *The performance criteria employed are the winning probability, accumulated returns, Sharpe ratio and luck coefficient. Asymptotic test statistics for these criteria are derived. The hypothesis as to the superiority of GA over a benchmark, say, buy-and-hold, can then be tested using Monte Carlo simulation. From this rigorously-established evaluation process, we find that simple genetic algorithms can work very well in linear stochastic environments, and that they also work very well in nonlinear deterministic (chaotic) environments. However, they may perform much worse in pure nonlinear stochastic cases. These results shed light on the superior performance of GA when it is applied to the two* tick-by-tick *time series of foreign exchange rates:* EUR/USD *and* USD/JPY.

Applications of Artificial Intelligence in Finance and Economics
Advances in Econometrics, Volume 19, 1–43

ISSN: 0731-9053/doi:10.1016/S0731-9053(04)19001-4

1. INTRODUCTION

Genetic algorithms (GAs) have been developed by Holland (1975) to mimic some of the processes observed in natural evolution. They are based on the genetic processes of natural selection which have become widely known as the "survival of the fittest" since Darwin's celebrated work. In recent years, GAs have been successfully applied to find good solutions to real-world problems whose search space is complex, such as the traveling salesman problem, the knapsack problem, large scheduling problems, graph partitioning problems, and engineering problems, too.[1]

In finance, Bauer (1994) provides the first application of GAs to discover trading strategies. Since then, GAs have gradually become a standard tool for enhancing investment decisions.[2] While many studies have supported the effectiveness of GAs in investment decisions; however, the foundation of these applications has not been well established. The thing that concerns us, therefore, is the *robustness* of these empirical results. For example, if GAs are effective for the investment in one market at one time, would the same result apply to the same market or different markets at different times? It is for the purpose of pursuing this generality, that we see the necessity of building a solid foundation upon which a rigorous evaluation can be made.

In this paper, a statistical approach to testing the performance of GA-based trading strategies is proposed. Instead of testing the performance of GAs in specific markets as a number of conventional studies already have, we are interested in a market-independence issue: *what makes GAs successful and what makes them not*? Since the data to which GAs are applied consist of financial time series, the question can be rephrased as follows: what are the *statistical properties* which distinguish a successful application of GA from an unsuccessful one? One way to think of the question is to consider two markets following different stochastic processes. One market follows stochastic process A, and the other stochastic process B. If GAs can work well with stochastic process A, but not B, then the successful experience of GAs in the first market is certainly not anticipated in the second market.

Having said that, this paper follows the following research methodology. First, some financially-related stochastic processes are singled out as the standard scenarios (testbeds) to test the performance of GA. Second, appropriate *performance criteria* are used to evaluate the performance of the GA over these testbeds. Third, the associated *asymptotic statistical tests* are applied to examine whether the GAs perform significantly differently as opposed to a familiar *benchmark*. By this procedure, we may be able to distinguish the processes in which the GA has competence from others in which it does not. Once the critical properties are grasped, we can then apply the GA to the financial time series whose stochastic properties are well-known, and test whether the GA behaves consistently with what we have learned from the previous statistical analysis.

By means of the procedure established in this paper, we hope to push forward the current applications of GAs or, more generally, computational intelligence (CI), toward a more mature status. After all, whether GA will work has been asked too intensely in the literature. The very mixed results seem to suggest that we look at the same question at a finer level and start to inquire why it works or why it doesn't. We believe that there are other ways to do something similar to what we propose in this paper.[3] We do not exclude these possibilities. In fact, little by little, these efforts will eventually enable GA or CI tools to rid themselves of their notoriety for being *blackboxes*.

The rest of the paper is organized as follows. Section 2 introduces a specific version of GA, referred as to the ordinary GA (OGA), used in this paper. Section 3 will detail the classes of stochastic processes considered in this paper and the reasons for this choice. Section 4 reviews the four performance criteria and establishes their associated asymptotic test. Section 5 sets up the Monte Carlo simulation procedure. Section 6 summarizes and discusses the actual performance of the GA over the artificial data, whereas the counterpart over the real data is given in Section 7. Section 8 concludes this paper.

2. TRADING WITH GAS

A *trading strategy* g can be formally defined as a mapping:

$$g\colon \Omega \rightarrow \{0, 1\}. \tag{1}$$

In this paper, is assumed to be a collection of *finite-length binary strings*. This simplification can be justified by the *data-preprocessing procedure* which *transforms* the raw data into *binary strings*. The *range* of the mapping g is simplified as a 0–1 action space. In terms of simple *market-timing* strategy, "1" means to "*act*" and "0" means to "*wait*." Here, for simplicity, we are only interested in *day trading*. So, "*act*" means to buy it at the opening time and sell it at the closing time.

Like all financial applications of GA, the start-off question is the *representation* issue. In our case, it is about how to effectively characterize the mapping g by a *finite-length binary string*, also known as a *chromosome* in GA. Research on this issue is very much motivated by the format of existing trading strategies, and there are generally two approaches to this issue. The first approach, called the *decision tree* approach, was pioneered by Bauer (1994). In this approach each trading strategy is represented by a decision tree. Bauer used bit strings to encode these decision tress, and generated and evolved them with genetic algorithms. The second approach, called the *combinatoric* approach, was first seen in Palmer et al. (1994). The combinatoric approach treats each trading strategy as one realization from $\binom{n}{k}$ combinations, where $l \leq k \leq n$, and n is the total number of given trading rules. Using GAs, one can encode the *inclusion* or *exclusion* of a specific trading rule as a bit and the whole trading strategy as a bit string (chromosome).

Both approaches have very limited expression power. While various enhancements are possible, they all lead to non-standard GAs in the sense that their representations are not based on finite-length binary strings. Since the main focus of this paper is to illustrate a statistical foundation of the GA, we try to avoid all unnecessary complications, including the use of those non-standard representations. In other words, at this initial stage, we only make the illustration with the ordinary genetic algorithm (OGA), and, for that reason, Bauer's simple decision- tree representation is employed. However, it is clear that the statistical foundation presented in this paper is also applicable to GAs with different representations.

Bauer's decision-tree representation corresponds to the following general form of *trading strategies*

(**IF** (CONDS)
THEN (BUY AND SELL [**DAY TRADING**])
ELSE (WAIT)).

The CONDS appearing in the trading strategy is a *predicate*. CONDS itself is a logical composition of several primitive predicates. In this paper, all CONDSs are composed of three primitive predicates. Each primitive predicate can be represented as:

$$\text{Cond}(Z) = \begin{cases} 1(\text{True}), & \text{if } Z \oplus a, \\ 0(\text{False}), & \text{if } Z \oplus a \end{cases} \tag{2}$$

where Z, in our application, can be considered as a time series of returns indexed by t, e.g. r_{t-1}, r_{t-2}, etc., and a can be regarded as a *threshold* or *critical value* ($a \in \aleph$, a set of integers).$\oplus \in \{\geq, <\}$ and $\ominus = \{\geq, <\} - \oplus$. An example of CONDS with three primitive predicates is

$$\text{CONDS}(r_{t-1}, r_{t-2}, r_{t-3}) = \text{Cond}(r_{t-1}) \vee (\text{Cond}(r_{t-2}) \wedge \text{Cond}(r_{t-3})), \tag{3}$$

where "$\vee$" refers to the logic operator "OR," and "$\wedge$" refers to "AND."

Following Bauer, we use a 21-bit string to encode a trading strategy of this kind. Details can be found in the Appendix (Section A.1). Let G be the collection of all trading strategies encoded as above. Then the cardinality of G is 2^{21}

($\#(G) = 2^{21}$), which is more than 2 million. The search over the space G can be interpreted as a *numerical* algorithm as well as a *machine learning* algorithm for solving a mathematical optimization problem. Without losing generality, consider the trading strategy with only *one* primitive predicate,

$$\text{Cond}(Z) = \begin{cases} 1(\text{True}), & \text{if } r_{t-1} \geq a, \\ 0(\text{False}), & \text{if } r_{t-1} < a. \end{cases} \tag{4}$$

Suppose the stochastic process of r_t is *strictly stationary* and denote the joint density of r_{t-1} and r_t by $f(r_{t-1}, r_t)$. In this simplest case, a trading strategy is parameterized by a single parameter a. Denote it by g_a. Then the optimal strategy g_{a^*} can be regarded as a solution to the optimization problem

$$\max_a E\,(\ln(\pi_n)), \tag{5}$$

where

$$\pi_n = \prod_{t=1}^{n}(1 + r_t) \tag{6}$$

is the accumulated returns of g_a over n consecutive periods. It can be shown that the solution to the problem (5) is

$$a^* = F^{-1}(0), \quad \text{if } F^{-1}(0) \text{ exists.} \tag{7}$$

where

$$F(a) = \int_{-\infty}^{\infty} \ln(1 + r_t) f(a, r_t)\,\mathrm{d}r_t \tag{8}$$

To solve Eq. (7), one has to know the density function of $f(r_{t-1}, r_t)$, which can only be inferred from the historical data. In this case, GAs are used as a *machine learning* tool to obtain an estimate of this joint density. Also, to arrive at a value for a^*, we have to know the inverse function of $F(a)$, which in general can only be solved numerically. In this case, GAs are used as a *numerical technique* to solve this problem. Therefore, in the trading-strategy problem, GAs are used simultaneously as a *numerical* technique and a *machine learning* tool to determine the critical parameter a^*. In the general case when CONDS has more than one predicate, the mathematical formulation of the problem can become very complicated, but the dual role of GAs remains unchanged. This interpretation justifies the mathematical significance of using GAs to discover the trading strategies.

The GA employed in this paper is a very basic version, which we shall call the ordinary genetic algorithm (*OGA*). In this study, we only focus on the OGA. Nonetheless, in a further study, it will be interesting to see whether a better result can be expected from advanced versions of GAs. The technical details of the OGA are given in the Appendix (Section A.2).

3. TESTBEDS

There are six stochastic processes used to evaluate the performance of GAs. They are:

(1) the linear stationary time series (also known as the Auto-Regressive and Moving-Average (*ARMA*) processes),
(2) the bilinear processes,
(3) the Auto-Regressive Conditional Heteroskedasticity (*ARCH*) processes,
(4) the Generalized ARCH (*GARCH*) processes,
(5) the threshold bilinear processes, and
(6) the chaotic processes.

All of the six classes have been frequently applied to modeling financial time series. Linear ARMA processes are found to be quite useful in *high-frequency financial data* (Campbell et al., 1997; Roll, 1984). Bilinear processes are often used to model the nonlinear dependence in both low- and high-frequency data (Drunat et al., 1998; Granger & Andersen, 1978). The ARCH processes are the most popular econometric tools for capturing the nonlinear dependence in the form of the second moment (Bollerslev et al., 1992). The threshold processes are good for asymmetric series and bursts (Tong, 1990). Finally, chaotic time series have been a topic of interest in finance over the last decade (Brock et al., 1991). Some details of these classes of processes are briefly reviewed from Sections 3.1 to 3.6.

These six processes are general enough to cover three important classes of dynamic processes, namely, linear stochastic processes, nonlinear stochastic processes, and nonlinear deterministic processes. This enables us to analyze the GA's performance in terms of some generic properties. For example, would it be easier for the GA to perform better with the linear (stochastic) process than with the nonlinear (stochastic) process, and with the deterministic (nonlinear) processes than with the stochastic (nonlinear) processes? The answers to these questions can certainly help us to delineate the effectiveness of GAs.

3.1. Linear Time Series

The linear time series model, also known as the *Auto-Regressive and Moving-Average* (*ARMA*(p,q)) model, was initiated by Box and Jenkings (1976). It has the

Table 1. Data Generating Processes – ARMA

Code	Model	Parameters			
		ϕ_1	ϕ_2	θ_1	θ_2
L-1	ARMA(1,0)	0.3	0	0	0
L-2	ARMA(1,0)	0.6	0	0	0
L-3	ARMA(2,0)	0.3	−0.6	0	0
L-4	ARMA(2,0)	0.6	−0.3	0	0
L-5	ARMA(0,1)	0	0	0.3	0
L-6	ARMA(0,1)	0	0	0.6	0
L-7	ARMA(0,2)	0	0	0.3	−0.6
L-8	ARMA(0,2)	0	0	0.6	−0.3
L-9	ARMA(1,1)	0.3	0	−0.6	0
L-10	ARMA(1,1)	0.6	0	−0.3	0
L-11	ARMA(2,2)	0.4	−0.4	0.4	0.4
L-12	ARMA(2,2)	0.6	−0.3	−0.3	−0.6
L-13	White Noise	*Gaussian* (0, 0.1)			

following general form:

$$r_t = \mu + \sum_{i=1}^{p} \phi_i r_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t, \tag{9}$$

where $\varepsilon_t \overset{iid}{\sim} N(0, \sigma^2)$. In all Monte Carlo simulations conducted in this paper, μ is set to 0 and σ^2 is set to 0.01. Thirteen ARMA(p,q) models were tested. The parameters of these thirteen ARMA(p,q) models are detailed in Table 1. Among these thirteen models, there are four pure AR models (L1–L4), four pure MA models (L5–L8), and four mixtures (L9–L12). The last one is simply *Gaussian white noise*.

3.2. Bilinear Process

The second class of stochastic processes considered in this paper is the *bilinear process* (BL), which was first studied by Granger and Anderson (1978), and subsequently by Subba-Rao (1981) and Subba-Rao and Gabr (1980). The BL process is constructed simply by adding the cross-product terms of r_{t-i} and ε_{t-j} to a linear ARMA process so it can be regarded as a second-order nonlinear time series model. In other words, if the parameters of all cross-product terms are zero, then the BL process can be reduced to the ARMA process.

Table 2. Data Generating Processes – Bilinear.

Code	Model	Parameters					
		ϕ_1	θ_1	ψ_{11}	ψ_{12}	ψ_{21}	ψ_{22}
BL-1	BL(0,0,1,1)	0	0	0.6	0	0	0
BL-2	BL(0,0,1,1)	0	0	0.3	0	0	0
BL-3	BL(0,1,1,2)	0	0.3	0	0.6	0	0
BL-4	BL(0,1,1,2)	0	0.6	0	0.3	0	0
BL-5	BL(1,0,2,1)	0.3	0	0	0	0.6	0
BL-6	BL(1,0,2,1)	0.6	0	0	0	0.3	0
BL-7	BL(1,1,2,2)	0.3	0.3	0	0	0	0.3
BL-8	BL(1,1,2,2)	0.3	0.3	0	0	0	0.6

The general form of a bilinear process, BL(p, q, u, v) is:

$$r_t = \mu + \sum_{i=1}^{p} \phi_i r_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \sum_{m=1}^{u}\sum_{n=1}^{v} \psi_{mn} r_{t-m} \varepsilon_{t-n} + \varepsilon_t, \tag{10}$$

where $\varepsilon_t \overset{iid}{\sim} N(0, \sigma^2)$. Eight specific bilinear processes are employed for our Monte-Carlo simulation. In all of these processes, $\mu = 0$ and $\sigma^2 = 0.01$. Other parameters are given in Table 2. Notice that the first two (BL-1, BL-2) do not have the linear component, and only the nonlinear cross-product terms are presented.

3.3. ARCH Processes

The third class of models considered is the *Auto-Regressive Conditional Heteroskedasticity* (ARCH) *process* introduced by Engle (1982), which has played a dominant role in the field of financial econometrics. The ARCH process is mainly used to replicate the three stylized facts of financial time series, namely, the fat-tailed marginal distribution of returns, the time-variant volatility of the returns, and clustering outliers. Consequently, unlike the ARMA process, ARCH mainly works only on the second moment, rather than the first moment. Nonetheless, by combining the two, one can attach an ARMA(p, q) process with an ARCH (q') process, called the ARMA(p, q)-ARCH(q') process. Its general form is

$$r_t = \mu + \sum_{i=1}^{p} \phi_i r_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{11}$$

$$\varepsilon_t | \Omega_{t-1} \sim N(0, \sigma_t^2) \tag{12}$$

Table 3. Data Generating Processes – ARCH.

Code	Model	Parameters				
		ω	α_1	α_2	ϕ_1	θ_1
AH-1	AR(0)-ARCH(1)	0.005	0.3	0	0	0
AH-2	AR(0)-ARCH(1)	0.005	0.6	0	0	0
AH-3	AR(0)-ARCH(2)	0.001	0.3	0.5	0	0
AH-4	AR(0)-ARCH(2)	0.001	0.5	0.3	0	0
AH-5	AR(1)-ARCH(1)	0.005	0.6	0	0.6	0
AH-6	AR(1)-ARCH(2)	0.001	0.5	0.3	0.6	0
AH-7	MA(1)-ARCH(1)	0.005	0.3	0	0	−0.6

$$\sigma_t^2 = \omega + \sum_{m=1}^{q'} \alpha_m \varepsilon_{t-m}^2 \tag{13}$$

where $\omega > 0$, $\sigma_m \geq 0$, $m = 1, \ldots, q'$ and Ω_t denotes the information set available at time t.

Seven ARCH processes are included in this study. They share a common value of μ, which is 0. Values of other parameters are detailed in Table 3. Notice that the first four processes do not have linear signals ($\phi_1 = 0$, $\theta_1 = 0$), whereas the fifth and the sixth processes are associated with an AR(1) linear signal ($\phi_1 = 0.6$), and the last process has a MA(1) linear signal ($\theta_1 = -0.6$).

3.4. GARCH Processes

A generalized version of the ARCH process, known as the *generalized ARCH* (GARCH) process, was introduced by Bollerslev (1986). GARCH generalizes Engle's ARCH process by adding additional conditional autoregressive terms. An ARMA(p, q) process with a GARCH error term of order(p', q'), ARMA(p, q)-GARCH(p', q'), can be written as

$$r_t = \mu + \sum_{i=1}^{p} \phi_i r_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{14}$$

$$\varepsilon_t | \Omega_{t-1} \sim N(0, \sigma_t^2) \tag{15}$$

$$\sigma_t^2 = \omega + \sum_{m=1}^{q'} \alpha_m \varepsilon_{t-m}^2 + \sum_{n=1}^{p'} \beta_n \sigma_{t-n}^2 \tag{16}$$

Table 4. Data Generating Processes – GARCH.

Code	Model	Parameters					
		β_1	β_2	α_1	α_2	ϕ_1	θ_1
GH-1	AR(0)-GARCH(1,1)	0.3	0	0.5	0	0	0
GH-2	AR(0)-GARCH(1,1)	0.5	0	0.3	0	0	0
GH-3	AR(0)-GARCH(1,2)	0.2	0	0.2	0.4	0	0
GH-4	AR(0)-GARCH(1,2)	0.2	0	0.4	0.2	0	0
GH-5	AR(0)-GARCH(2,1)	0.2	0.4	0.2	0	0	0
GH-6	AR(0)-GARCH(2,1)	0.4	0.2	0.2	0	0	0
GH-7	AR(1)-GARCH(1,1)	0.5	0	0.3	0	0.6	0
GH-8	AR(1)-GARCH(1,2)	0.2	0	0.4	0.2	0.6	0
GH-9	AR(1)-GARCH(2,1)	0.4	0.2	0.2	0	0.6	0
GH-10	MA(1)-GARCH(1,1)	0.3	0	0.5	0	0	–0.6

with $\omega > 0, \alpha_m = 0$ and $\beta_n \geq 0, m = 1, \ldots, q', n = 1, \ldots, p'$. Again, Ω_t denotes the information set available at time t.

Nine GARCH processes are attempted. In all cases, $u = 0$ and $\omega = 0.001$. Specifications of other parameters are given in Table 4. The 7th, 8th and 9th models (GH-7, GH-8, GH-9) are AR(1) processes combined with a GARCH error term, whereas the last model (GH-10) is a MA(1) process plus a GARCH error term. For the remaining six, there are no linear signals but just pure GARCH processes.

3.5. Threshold Processes

Tong (1983) proposed a *threshold autoregressive* (TAR) model which is of the form,

$$r_t = \mu^{(l)} + \sum_{i=1}^{p} \phi_i^{(l)} r_{t-i} + \varepsilon_t \tag{17}$$

if $r_{t-d} \in \Omega_1$ $(l = 1, 2, \ldots, k)$, where $\Omega_i \cap \Omega_j = \varnothing$ $(i, j = 1, \ldots, k)$ if $i \cdot j$ and $\cup_{l=1}^{k} \Omega_l = \Re$. The parameter k represents the number of thresholds and d is called the threshold lag (or delay parameter). Producing various limit cycles is one of the important features of the threshold models, and the TAR process can be applied to the time series which has an asymmetric cyclical form.

The threshold idea can be used as a module to add and to extend other processes. Here, we apply the threshold idea to the bilinear process (10), and extend it to a threshold bilinear (TBL) process. Let us denote a bilinear process (BL(p, q, u, v))

Table 5. Data Generating Processes – Threshold Processes.

Code	Model	Parameters						
		$\phi_1^{(1)}\ \phi_1^{(2)}$	$\phi_2^{(1)}\ \phi_2^{(2)}$	$\theta_1^{(1)}\ \theta_1^{(2)}$	$\psi_{11}^{(1)}\ \psi_{11}^{(2)}$	$\psi_{12}^{(1)}\ \psi_{12}^{(2)}$	$\psi_{21}^{(1)}\ \psi_{21}^{(2)}$	$\psi_{22}^{(1)}\ \psi_{22}^{(2)}$
TH-1	TBL(2;1,0,0,0)	0.3	0	0	0	0	0	0
		0.6	0	0	0	0	0	0
TH-2	TBL(2;1,1,0,0)	0.3	0	0.6	0	0	0	0
		0.6	0	0.3	0	0	0	0
TH-3	TBL(2;0,0,1,1)	0	0	0	0.3	0	0	0
		0	0	0	0.6	0	0	0
TH-4	TBL(2;1,1,2,2)	0.3	0	0	0	0	0.6	0
		0	0	0.3	0	0.6	0	0
TH-5	TBL(2;2,0,2,2)	0	0	0	0.3	0	0	–0.6
		0.3	–0.6	0	0	0	0	0

Note: The lag period d is set to 1 and $\mu^{(1)} = \mu^{(2)} = 0$ in all of the models. In addition, $\Omega_1 \equiv \{r_{t-d} \,|r_{t-d} \geq 0\}$ and $\Omega_2 \equiv \{r_{t-d} \,|r_{t-d} < 0\}$.

with k-thresholds by TBL(k, p, q, u, v), which can be written as

$$r_t = \mu^{(l)} + \sum_{i=1}^{p} \phi_i^{(l)} r_{t-i} + \sum_{j=1}^{q} \theta_j^{(l)} \varepsilon_{t-j} + \sum_{m=1}^{u}\sum_{n=1}^{v} \psi_{mn}^{(l)} r_{t-m}\varepsilon_{t-n} + \varepsilon_t \qquad (18)$$

It is trivial to show that TBL can be reduced to a threshold ARMA if $\psi_{mm}^{(l)} = 0$ for all m, n and l. Table 5 lists the five TBL processes considered in this paper. The motives for choosing these five series will become clear when we come to Section 6.4.

3.6. Chaotic Processes

All of the above-mentioned processes are stochastic. However, the time series that appear to be random does not necessary imply that they are generated from a stochastic process. Chaotic time series as an alternative description of this seemingly random phenomenon was a popular econometrics topic in the 1990s. While it is hard to believe that a financial time series is just a deterministic chaotic time series, the chaotic process can still be an important module for the working of a nonlinear time series. Five chaotic processes are employed in this study.

C-1: Logistic Map

$$r_t = 4r_{t-1}(1 - r_{t-1}), \quad r_t \in [0, 1] \quad \forall t \qquad (19)$$

C-2: Henon Map

$$r_t = 1 + 0.3r_{t-2} - 1.4r_{t-1}^2, \quad r_{-1}, r_0 \in [-1, 1] \tag{20}$$

C-3: Tent Map

$$\begin{cases} r_t = 2r_{t-1}, & \text{if } 0 \leq r_{t-1} < 0.5, \\ r_t = 2(1 - r_{t-1}), & \text{if } 0.5 \leq r_{t-1} \leq 1. \end{cases} \tag{21}$$

C-4: Poly. 3

$$r_t = 4r_{t-1}^3 - 3r_{t-1}, \quad r_t \in [-1, 1] \quad \forall t \tag{22}$$

C-5: Poly. 4

$$r_t = 8r_{t-1}^4 - 8r_{t-1}^2 + 1, \quad r_t \in [-1, 1] \quad \forall t \tag{23}$$

The series generated by all these stochastic processes (from Sections 3.1 to 3.6) may have a range which does not fit the range of the normal return series. For example, the process (19) is always positive. As a result, a contracting or a dilating map is needed. We, therefore, contract or dilate all series linearly and monotonically into an acceptable range, which is (−0.3, 0.3) in this paper.

4. PERFORMANCE CRITERIA AND STATISTICAL TESTS

Basic performance metrics to evaluate the performance of trading strategies have long existed in the literature. Following Refenes (1995), we consider the following four main criteria: *returns*, the *winning probability*, the *Sharpe ratio* and the *luck coefficient*. In this paper, the performance of the trading strategies generated by the ordinal genetic algorithm (OGA) is compared with that using a benchmark based on these four criteria. To make the evaluation process rigorous, performance differences between the OGA-based trading strategies and the benchmark are tested *statistically*. Tests for returns and winning probability are straightforward. Tests for the Sharpe ratio are available in the literature (see, for example, Jobson and Korkie (1981) and Arnold (1990)). However, tests for the luck coefficient are more demanding, and it has not been derived in the literature. In this paper, we develop asymptotic tests for the luck coefficient.

4.1. Returns

Let X and Y be the accumulated returns of an one-dollar investment by applying OGA-based trading strategies and the benchmark strategy, say, the buy-and-hold

(B&H) strategy, respectively. Assume that $E(X) = \mu$ and $E(Y) = \nu$. Let us estimate the μ and ν by the respective *sample average* $\bar{\pi}^2$ and $\bar{\pi}^1$ via the Monte Carlo simulation. Then one can test the null

$$H_0\colon \mu - n \leq 0, \tag{24}$$

with the following test statistic

$$Z_\pi = \frac{\sqrt{n}(\bar{\pi}^2 - \bar{\pi}^1)}{(\hat{\sigma}^2 + \hat{\tau}^2 - 2\hat{\rho}\hat{\sigma}\hat{\tau})^{1/2}}, \tag{25}$$

where $\hat{\sigma}^2$ and $\hat{\tau}^2$ are the sample variances of X and Y, $\hat{\rho}$ is the sample correlation coefficient of X and Y, and n is the sample size (the number of ensembles generated during the Monte Carlo simulation). By using the central limit theorem, it is straightforward to show that Z_π is an asymptotically standard normal test.

While testing the difference between $\bar{\pi}^2$ and $\bar{\pi}^1$ can tell us the performance of the GA as opposed to a benchmark, it provides us with nothing more than a point evaluation. In some cases, we may also wish to know whether the superiority, if shown, can extend to a large class of trading strategies. A common way to address this question is to introduce an *omniscient* trader. Let us denote the respective accumulated returns earned by this omniscient trader as $\bar{\pi}^*$.[4] Now, subtracting $\bar{\pi}^1$ from $\bar{\pi}^*$ gives us the total unrealized gain, if we only know the benchmark. Then, the ratio, also called the *exploitation ratio*,

$$\tilde{\pi} \equiv \frac{\bar{\pi}^2 - \bar{\pi}^1}{\bar{\pi}^* - \bar{\pi}^1} \tag{26}$$

is a measure of the size of those unrealized gains which can be exploited by using a GA. Based on its formulation, $\tilde{\pi}$ may be positive, negative or zero, but has one as its maximum. If $\tilde{\pi}$ is not only positive, but is also close to one, then its superiority is not just restricted to the benchmark, but may also have global significance.

In addition to the accumulated gross returns, one can also base the comparison on the excess return by simply subtracting one from the accumulated gross returns. A relative superiority measure of the GA as opposed to the benchmark can be defined accordingly as

$$\dot{\pi} \equiv \frac{(\bar{\pi}^2 - 1) - (\bar{\pi}^1 - 1)}{|\bar{\pi}^1 - 1|} = \frac{\bar{\pi}^2 - \bar{\pi}^1}{|\bar{\pi}^1 - 1|}. \tag{27}$$

4.2. *Winning Probability*

The mean return can sometimes be sensitive to outliers. Therefore, it is also desirable to base our performance criterion on some robust statistics, and the *winning probability* is one of this kind. The winning probability basically tells us, by randomly picking up an ensemble from one stochastic process, the probability that the GA will win. Formally, let (X, Y) be a random vector with the joint density function $h(x, y)$. Then p_w, defined as follows, is called the *winning probability*.

$$p_w = \Pr(X > Y) = \int \int_{x>y} h(x, y)\, \mathrm{d}x\, \mathrm{d}y \tag{28}$$

Based on the winning probability, we can say that X is *superior* to Y if $p_w > 0.5$, and *inferior* to Y if $p_w < 0.5$, and *equivalent* to Y if $p_w = 0.5$. The null hypothesis to test is

$$H_0\colon p_w \leq 0.5 \tag{29}$$

The rejection of (29) shows the superiority of the GA over the benchmark. An asymptotic standard normal test of (29) can be derived as

$$Z_w = \frac{\sqrt{n}(\hat{p}_w - 0.5)}{\sqrt{\hat{p}_w(1 - \hat{p}_w)}} \tag{30}$$

where $\hat{p}_w$ is the sample counterpart of p_w.

4.3. Sharpe Ratio

One criterion which has been frequently ignored by machine learning people in finance is the *risk* associated with a trading rule. Normally, a higher profit known as the *risk premium* is expected when the associated risk is higher. Without taking the risk into account, we might exaggerate the profit performance of a highly risky trading rule. Therefore, to evaluate the performance of our GA-based trading rule on a risk-adjusted basis, we have employed the well-known *Sharpe ratio* as the third performance criterion (Sharpe, 1966). The Sharpe ratio s is defined as the excess return divided by a risk measure. The higher the Sharpe ratio, the higher the risk-adjusted return.

Formally, let $X \sim f(x)$ with $E(X) = \mu$ and $\mathrm{Var}(X) = \sigma^2$. Then the value

$$s = \frac{\mu - c}{\sigma} \tag{31}$$

is called the *Sharpe ratio* of X where c is one plus a risk-free rate. Furthermore, to compare the performance of two trading strategies in the Sharpe ratio, let $X \sim f(x)$ and $Y \sim g(y)$ with $E(X) = \mu$, $E(Y) = \nu$, $\mathrm{Var}(X) = \sigma^2$ and $\mathrm{Var}(Y) = \tau^2$. Then the difference

$$d = \frac{\mu - c}{\sigma} - \frac{\nu - c}{\tau} \tag{32}$$

is called the *Sharpe-ratio differential* between X and Y. Accordingly, X is said to have a *higher* (*lower*) Sharpe ratio relative to Y if $d > 0$ ($d < 0$). Otherwise, X and Y are said to be *identical* in terms of the Sharpe ratio.

Jobson and Korkie (1981) derive an asymptotic standard normal test for the Sharpe-ratio differential. However, we do not follow their Taylor expansion formulation. Instead, by applying *Slutzky's theorem*, the *Cramer*δ *theorem*, and the *multivariate central limit theorem*, a standard normal test for the null

$$H_0\colon d \le 0 \tag{33}$$

can be derived as follows:

$$Z_d = \frac{\sqrt{n}(\hat{d} - d)}{\hat{\omega}_1}, \tag{34}$$

where

$$\hat{d} = \frac{\bar{\pi}^2 - c}{\hat{\sigma}} - \frac{\bar{\pi}^1 - c}{\hat{\tau}}, \tag{35}$$

and

$$\hat{\omega}_1^2 = 2(1 - \hat{\rho}) + \frac{(\bar{\pi}^2 - c)}{\hat{\sigma}}(\bar{\theta} - \hat{\delta}) + \frac{(\bar{\pi}^1 - c)}{\hat{\tau}}(\hat{\psi} - \hat{\xi}) - \frac{(\bar{\pi}^2 - c)(\bar{\pi}^1 - c)}{\hat{\sigma}\hat{\tau}}\frac{(\hat{\phi} - 1)}{2} + \frac{(\bar{\pi}^2 - c)^2}{\hat{\sigma}^2}\frac{(\hat{\gamma} - 1)}{4} + \frac{(\bar{\pi}^1 - c)^2}{\hat{\tau}}\frac{(\hat{\eta} - 1)}{4} \tag{36}$$

$\hat{\delta}$, $\hat{\gamma}$, $\hat{\xi}$ and $\hat{\eta}$ are the corresponding sample third and fourth moments of X and Y, whereas $\hat{\rho}$, $\hat{\theta}$, $\hat{\psi}$, $\hat{\phi}$ are the corresponding sample mixed moments between X and Y (also expressed as Eq. (37)).

$$\begin{bmatrix} \frac{E(X-u)^3}{\sigma^3} \\ \frac{E(X-u)^4}{\sigma^4} \\ \frac{E(Y-v)^3}{\tau^3} \\ \frac{E(Y-v)^4}{\tau^4} \end{bmatrix} = \begin{bmatrix} \delta \\ \gamma \\ \xi \\ \eta \end{bmatrix}, \begin{bmatrix} \frac{E(X-u)E(X-v)}{\sigma\tau} \\ \frac{E(X-u)^2(Y-v)}{\sigma^2\tau} \\ \frac{E(X-u)(Y-v)^2}{\sigma\tau^2} \\ \frac{E(x-u)^2(Y-v)^2}{\sigma^2\tau^2} \end{bmatrix} = \begin{bmatrix} \rho \\ \theta \\ \psi \\ \phi \end{bmatrix} \tag{37}$$

4.4. Luck Coefficient

The largest positive trade can be very important if it makes a significant contribution towards skewing the average profit dramatically. When this happens, people can be severely misled by the sample mean. As a solution to this problem, the *trimmed mean* is often used in statistics. A similar idea in finance is known as the *luck coefficient*. The luck coefficient l_ε is defined as the sum of the largest 100% returns, $\varepsilon \in (0, 1)$, divided by the sum of total returns. In a sense, the larger the luck coefficient, the weaker the reliability of the performance. The luck coefficient, as a performance statistic, is formally described below.

Let $\{X_1, X_2, \ldots, X_m\}$ be a random sample from $f(x)$ with $E(X) = \mu$. The order statistic of this random sample can be enumerated as $X_{(1)}, X_{(2)}, \ldots, X_{(m)}$, where $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(m)}$. Then, from the *order statistics*, it is well known that

$$X_{(m)} \sim g(x_{(m)}) = m[F(x_{(m)})]^{m-1} f(x_{(m)}) \tag{38}$$

where F is the *distribution function* of X. Furthermore, let $X_i \overset{iid}{\sim} f(x)$, $i = 1, 2, \ldots, m$ and $X_{(m)} \sim g(x_{(m)})$ as described above with $E(X_{(m)}) = \mu$. Then the ratio

$$l_\varepsilon = \frac{\varepsilon \mu_\varepsilon}{\mu} \tag{39}$$

is called the *luck coefficient* of X where $\varepsilon = \frac{1}{m}$. In this study, is set to 0.05. Here we want to see how much of the contribution to mean returns comes from the largest 5% of trades.

For making a comparison between strategies, the *luck-coefficient ratio* is defined as follows. Let $X_i \overset{iid}{\sim} f_x(x)$ with $E(X) = \mu$, $Y_i \overset{iid}{\sim} f_y(y)$ with $E(Y) = \nu$, $i = 1, 2, \ldots, m$ and $X_{(m)} \sim g_x(x_{(m)})$ with $E(X_{(m)}) = \mu$, $Y_{(m)} \sim g_y(y_{(m)})$ with $E(Y_{(m)}) = \nu$. Then the ratio

$$r_\varepsilon = \frac{\varepsilon \nu_\varepsilon / \nu}{\varepsilon \mu_\varepsilon / \mu} = \frac{\mu \nu_\varepsilon}{\nu \mu_\varepsilon} \tag{40}$$

is called the *luck-coefficient ratio* of X relative to Y where $\varepsilon = \frac{1}{m}$. Based on this definition, X is said to have a *lower* (*higher*) luck coefficient relative to Y if $r > 1$ ($r < 1$). Otherwise, X and Y are said to be *identical* in terms of the luck coefficient. However, to the best of our knowledge, the respective asymptotic standard normal test for the null

$$H_0: r \leq 1 \tag{41}$$

is not available in the literature. Nevertheless, similar to the derivation of the test of the Sharpe ratio (34), it is not hard to cook up such a test by using *Slutzky's theorem*, the *Cramer* δ *theorem*, and the *multivariate central limit theorem*, which is given in Eq. (42)

$$Z_r = \frac{\sqrt{n}(\hat{r}_\varepsilon - r_\varepsilon)}{\hat{\omega}_2}, \tag{42}$$

where

$$\hat{r}_\varepsilon = \frac{\bar{\pi}^2 \bar{\pi}^1_m}{\bar{\pi}^1 \bar{\pi}^2_m}, \tag{43}$$

and

$$\begin{aligned}
\hat{\omega}_2^2 = {} & \frac{\varepsilon(\bar{\pi}^1_m)^2}{(\bar{\pi}^1)^2(\bar{\pi}^2_m)^2}\left(\hat{\sigma}^2 + \frac{(\bar{\pi}^2)^2\hat{\tau}^2}{(\bar{\pi}^2)^2}\right) + \frac{(\bar{\pi}^2)^2}{(\bar{\pi}^1)^2(\bar{\pi}^2_m)^2}\left(\hat{\tau}^2_\varepsilon + \frac{(\bar{\pi}^1_m)^2\hat{\sigma}^2_\varepsilon}{(\bar{\pi}^2_m)^2}\right) \\
& - \frac{2\bar{\pi}^2\bar{\pi}^1_m\hat{\tau}}{(\bar{\pi}^1)^3(\bar{\pi}^2_m)^2}(\varepsilon\bar{\pi}^1_m\hat{\rho}\hat{\sigma} + \bar{\pi}^2\hat{\lambda}\hat{\tau}_\varepsilon) - \frac{2\bar{\pi}^2\bar{\pi}^1_m\hat{\sigma}_\varepsilon}{(\bar{\pi}^1)^2(\bar{\pi}^2_m)^3}(\bar{\pi}^1_m\hat{\sigma}\hat{\iota} + \bar{\pi}^2\hat{\tau}_\varepsilon\hat{\zeta}) \\
& + \frac{2\bar{\pi}^2\bar{\pi}^1_m}{(\bar{\pi}^1)^2(\bar{\pi}^2_m)^2}\left(\hat{\sigma}\hat{\kappa}\hat{\tau}_\varepsilon + \frac{\bar{\pi}^2\bar{\pi}^1_m\hat{\sigma}_\varepsilon\hat{o}\hat{\tau}}{\bar{\pi}^1\bar{\pi}^2_m}\right).
\end{aligned} \tag{44}$$

$\bar{\pi}^1_m$ and $\bar{\pi}^2_m$ are the corresponding sample means of $Y_{(m)}$ and $X_{(m)}$. $\hat{\tau}^2_\varepsilon$ and $\hat{\sigma}^2_\varepsilon$ are the corresponding sample variances of $Y_{(m)}$ and $X_{(m)}$, and $\hat{\rho}$, $\hat{\zeta}$, $\hat{\kappa}$, $\hat{\iota}$, $\hat{\lambda}$, and $\hat{o}$ are the corresponding sample correlation coefficients as indicated in Eq. (45).

$$\begin{bmatrix} \mathrm{corr}(X_i, Y_i) \\ \mathrm{corr}(X_{(m)}, Y_{(m)}) \\ \mathrm{corr}(X_i, Y_{(m)}) \end{bmatrix} = \begin{bmatrix} \rho \\ \zeta \\ \kappa \end{bmatrix}, \begin{bmatrix} \mathrm{corr}(X_i, X_{(m)}) \\ \mathrm{corr}(Y_i, Y_{(m)}) \\ \mathrm{corr}(Y_i, X_{(m)}) \end{bmatrix} = \begin{bmatrix} \iota \\ \lambda \\ o \end{bmatrix} \tag{45}$$

5. MONTE CARLO SIMULATION

Since it is hard to obtain analytical results of the performance of the GA in relation to various stochastic processes, Monte Carlo simulation methodology is used in this study. Each stochastic process listed in Tables 1–5 and Eqs (19) to (23) is used to generate 1000 *independent* time series, each with 105 observations ($\{r_t\}_{t=1}^{105}$).[5] For each series, the first 70 observations ($\{r_t\}_{t=1}^{70}$) are taken as the training sample, and the last 35 observations ($\{r_t\}_{t=76}^{105}$) are used as the testing sample. The OGA are then employed to extract trading strategies from these training samples. These strategies are further tested by the testing samples, and the resulting accumulated returns (p) are calculated, i.e.

$$\pi = \prod_{t=76}^{105}(1 + r_t) \tag{46}$$

In the meantime, the accumulated returns of the benchmark are also calculated. In following convention, our choice of the benchmark is simply the buy-and-hold (B&H) strategy.

Let π_i^1 ($i = 1, 2, \ldots, 1000$) be the accumulated returns of the B&H strategy when tested on the ith ensemble of a stochastic process, and π_i^2 be the accumulated returns of the OGA when tested on the same ensemble. The issue which we shall address, given the set of observations $S(\equiv \{\pi_i^1, \pi_i^2\}_{i=1}^{1000})$, is to decide *whether the OGA-based trading strategies can statistically significantly outperform the B&H strategy under the stochastic process in question*. The answers are given in the next section.

6. TEST RESULTS

6.1. ARMA Processes

We start our analysis from the linear stochastic processes. Table 6 summarizes the statistics defined in Section 4. Several interesting features stand out. First, from the statistics $\hat{p}_w$ and z_w, it can be inferred that, in accumulated returns, the probability that the OGA-based trading strategies can beat the B&H strategy *is significantly greater than 0.5*. For the stochastic processes with linear signals (L-1–L-12), the winning probability $\hat{p}_w$ ranges from 0.713 (L-5) to 0.991 (L-12). What, however, seems a little puzzling is that, even in the case of *white noise* (L-13), the GA can also beat B&H statistically significantly, while with much lower winning probabilities p_w (0.606). This seemingly puzzling finding may be due to the fact that a pseudo-random generator can actually generate a series with signals *when the sample size is small*. For example, Chen and Tan (1999) show that, when the sample size is 50, the probability of having signals in a series generated from a pseudo-random generator is about 5%, while that probability can go to zero when the sample size is 1000. Therefore, by supposing that the OGA-based trading strategies can win in all these atypical ensembles and get even with the B&H strategy in other normal ensembles, then $\hat{p}_w$ can still be significantly greater than 0.5.

Second, by directly comparing $\bar{\pi}^1$ with $\bar{\pi}^2$, we can see that, except for the case of white noise, the OGA-based trading strategies unanimously outperform the B&H strategy *numerically* in all linear ARMA*(p, q)* processes. From the $\dot{\pi}$ statistic (27), we see that the triumph of GA over B&H extends from a low of 19% (L-10) to a high of 916% (L-3). The z_p statistic, ranging from 2.12 to 47.39, signifies the statistical significance of these differences. Third, to see how the GA effectively exploited the excess potential returns earned by the omniscient trader, $\tilde{\pi}$ is also included in Table 6. There it is observed that the GA exploited 2–31%

Table 6. Performance Statistics of the OGA and B&H – ARMA.

Code	Model	$\bar{\pi}^1$	$\bar{\pi}^2$	$\bar{\pi}^*$	z_π	$\tilde{\pi}$ (%)	$\dot{\pi}$ (%)	$\hat{p}_w$	z_w
L-1	ARMA(1,0)	1.198	1.355	4.388	6.33	4	20	0.732	16.56
L-2	ARMA(1,0)	1.992	2.868	6.658	13.67	19	88	0.859	32.62
L-3	ARMA(2,0)	0.845	2.265	5.480	42.98	31	916	0.976	98.35
L-4	ARMA(2,0)	1.123	1.185	5.170	27.08	2	50	0.896	41.02
L-5	ARMA(0,1)	1.103	1.269	4.241	7.63	5	161	0.713	14.89
L-6	ARMA(0,1)	1.199	1.775	5.166	20.61	15	289	0.861	32.99
L-7	ARMA(0,2)	0.853	1.633	5.104	39.97	18	531	0.926	51.46
L-8	ARMA(0,2)	1.065	1.522	5.285	21.58	11	703	0.848	30.65
L-9	ARMA(1,1)	0.898	1.229	4.128	24.55	10	325	0.812	25.25
L-10	ARMA(1,1)	1.452	1.538	4.783	2.12	3	19	0.721	15.58
L-11	ARMA(2,2)	1.306	2.588	6.957	30.43	23	419	0.927	51.90
L-12	ARMA(2,2)	0.721	2.167	6.189	47.39	26	518	0.991	164.40
L-13	ARMA(0,0)	0.983	0.993	3.881	0.67	0	59	0.606	6.85
Code	Model	$\hat{s}_1$	$\hat{s}_2$	$\hat{d}$	z_d	$\hat{l}^1_{0.05}$	$\hat{l}^2_{0.05}$	$\hat{r}_{0.05}$	z_r
L-1	ARMA(1,0)	0.166	0.438	0.272	11.74	0.179	0.126	1.416	3.32
L-2	ARMA(1,0)	0.236	0.526	0.290	8.40	0.310	0.214	1.450	1.75
L-3	ARMA(2,0)	−0.342	1.181	1.523	32.14	0.115	0.106	1.087	1.68
L-4	ARMA(2,0)	0.111	0.877	0.767	24.53	0.182	0.114	1.594	4.45
L-5	ARMA(0,1)	0.110	0.419	0.309	13.40	0.169	0.117	1.449	4.23
L-6	ARMA(0,1)	0.135	0.602	0.467	5.02	0.216	0.138	1.563	2.48
L-7	ARMA(0,2)	−0.353	0.948	1.301	27.67	0.108	0.099	1.092	1.68
L-8	ARMA(0,2)	0.065	0.624	0.559	18.18	0.181	0.120	1.509	4.18
L-9	ARMA(1,1)	−0.307	0.524	0.831	22.43	0.093	0.092	1.007	0.16
L-10	ARMA(1,1)	0.214	0.392	0.177	5.39	0.263	0.171	1.534	2.50
L-11	ARMA(2,2)	0.170	0.854	0.684	11.19	0.240	0.141	1.708	3.34
L-12	ARMA(2,2)	−1.363	1.224	2.587	36.46	0.083	0.105	0.795	−6.21
L-13	ARMA(0,0)	−0.025	−0.016	0.010	0.37	0.130	0.096	1.353	3.90

Note: $\bar{\pi}^1$, $\bar{\pi}^2$ and $\bar{\pi}^*$ are the respective sample mean return of OGA, B&H and the omniscient trader. $\tilde{\pi}$ is the exploitation ratio (Eq. 26), and $\dot{\pi}$ is the relative superiority index (Eq. 27). $\hat{p}_w$ is the sample winning probability of OGA over B&H (Eq. 28). $\hat{s}_1$ and $\hat{s}_2$ are the corresponding sample Sharpe ratio of OGA and B&H (Eq. (31)). Their sample difference is $\hat{d}$ (Eq. (32)). $\hat{l}^1_{0.05}$ and $\hat{l}^2_{0.05}$ are the sample luck coefficient of OGA and B&H (Eq. (39)), and $\hat{r}_{0.05}$ is the sample luck coefficient ratio between the two (Eq. (40)). The z_π, z_w, z_d and z_r are the test statistics of the mean return difference, winning probability, Sharpe ratio differential, and luck coefficient ratio, respectively. The critical value of them is 1.28 at the 10% significance level, and is 1.64 at the 5% significance level.

of the potential excess returns. However, as we expect, it was to no avail when the scenario changed to white noise.

As mentioned earlier, we should not judge the performance of the GA solely by the profitability criterion. The risk is a major concern in business practice.

We, therefore, have also calculated the Sharpe ratio, a risk-adjusted profitability criterion. It is interesting to notice that in all cases the Sharpe-ratio differential ($\hat{d}$) is positive. In other words, the GA still outperforms B&H even after taking into account the risk. The test of this differential also lends support to its statistical significance.

Finally, we examine whether the GA wins just by *luck* in the sense that its return performance depends heavily on its best 5% trades. Based on the statistic of luck coefficient $\hat{r}_{0.05}$, it is found that in only one of the 13 cases, i.e. the case L-12, dose the GA have a higher luck coefficient; in the other 12 cases, the luck-coefficient ratios are larger than 1, meaning that the dominance of the GA over B&H cannot be attributed to the presence of a few abnormally large returns. From the test z_r, this result is again significant except for the case L-9. All in all, we can conclude that if the return follows a simple linear ARMA process, then the superior performance of the GA compared to B&H is expected.

6.2. Bilinear Processes

By moving into the bilinear processes, we are testing the effectiveness of the GA when the return series is *nonlinear*. Table 7 summarizes all the key statistics. Obviously, the performance of the GA is not as glamorous as before. Out of the eight battles, it loses twice (cases BL-1 and BL-2) to B&H (see z_p and z_w). Taking the risk into account would not help reverse the situation (see z_d). It is, however, interesting to notice a unique feature shared by BL-1 and BL-2. As mentioned in Section 3.2, the two stochastic processes do not have any linear component (all ϕ_i and θ_j in Eq. (10) or Table 2 are zero). In other words, these two cases are *pure nonlinear* (pure bilinear). If some linear components are added back to the series, then the significant dominance of the GA does come back. This is exactly what happens in the other six cases (BL-3 to BL-8), which all have the ARMA component as a part (Table 2).

Even for the six cases where the GA wins, we can still observe some adverse impacts of nonlinearity on the GA. Roughly speaking, Table 7 shows that the distribution of both $\dot{\pi}$ and $\tilde{\pi}$ becomes lower as opposed to those items observed in the linear stochastic processes. So, not only does the advantage of the GA relative to B&H shrink, but its disadvantage relative to the omniscient also becomes larger.

However, nonlinearity does not change many of the results in relation to the luck coefficients. The luck-coefficient ratios are all higher than 1, and most of the results are statistically significant, indicating the relative stability of the GA.

Table 7. Performance Statistics of the OGA and B&H – Bilinear.

Code	Model	$\bar{\pi}^1$	$\bar{\pi}^2$	$\bar{\pi}^*$	z_π	$\tilde{\pi}$ (%)	$\dot{\pi}$ (%)	$\hat{p}_w$	z_w
BL-1	BL(0,0,1,1)	1.253	1.126	4.398	−6.78	−4	−50	0.491	–0.57
BL-2	BL(0,0,1,1)	1.151	1.064	4.228	−4.66	−3	−58	0.517	1.08
BL-3	BL(0,1,1,2)	1.302	1.830	5.341	11.50	13	175	0.861	17.78
BL-4	BL(0,1,1,2)	1.186	1.356	4.449	6.95	5	91	0.745	17.78
BL-5	BL(1,0,2,1)	1.260	1.419	4.539	5.07	5	61	0.747	17.97
BL-6	BL(1,0,2,1)	2.292	3.143	7.226	9.89	17	66	0.877	36.30
BL-7	BL(1,1,2,2)	1.841	2.471	6.448	8.83	14	75	0.848	30.65
BL-8	BL(1,1,2,2)	1.602	2.287	5.894	19.57	16	114	0.870	34.79
Code	**Model**	$\hat{s}_1$	$\hat{s}_2$	$\hat{d}$	z_d	$\hat{l}^1_{0.05}$	$\hat{l}^2_{0.05}$	$\hat{r}_{0.05}$	z_r
BL-1	BL(0,0,1,1)	0.316	0.251	–0.065	–3.29	0.132	0.105	1.256	3.30
BL-2	BL(0,0,1,1)	0.190	0.144	–0.046	–2.21	0.144	0.101	1.427	4.14
BL-3	BL(0,1,1,2)	0.167	0.425	0.259	7.31	0.182	0.124	1.793	3.08
BL-4	BL(0,1,1,2)	0.162	0.724	0.562	16.32	0.232	0.129	1.465	3.22
BL-5	BL(1,0,2,1)	0.178	0.465	0.287	13.53	0.211	0.138	1.531	3.54
BL-6	BL(1,0,2,1)	0.251	0.539	0.289	10.38	0.346	0.226	1.534	2.05
BL-7	BL(1,1,2,2)	0.285	0.711	0.426	9.29	0.270	0.168	1.603	2.67
BL-8	BL(1,1,2,2)	0.179	0.386	0.207	2.52	0.272	0.182	1.494	1.14

Note: $\bar{\pi}^1$, $\bar{\pi}^2$ and $\bar{\pi}^*$ are the respective sample mean return of OGA, B&H and the omniscient trader. $\tilde{\pi}$ is the exploitation ratio (Eq. (26)), and $\dot{\pi}$ is the relative superiority index (Eq. (27)). $\hat{p}_w$ is the sample winning probability of OGA over B&H (Eq. (28)). $\hat{s}_1$ and $\hat{s}_2$ are the corresponding sample Sharpe ratio of OGA and B&H (Eq. (31)). Their sample difference is $\hat{d}$ (Eq. (32)). $\hat{l}^1_{0.05}$ and $\hat{l}^2_{0.05}$ are the sample luck coefficient of OGA and B&H (Eq. (39)), and $\hat{r}_{0.05}$ is the sample luck coefficient ratio between the two (Eq. (40)). The z_π, z_w, z_d and z_r are the test statistics of the mean return difference, winning probability, Sharpe ratio differential, and luck coefficient ratio, respectively. The critical value of them is 1.28 at the 10% significance level, and is 1.64 at the 5% significance level.

6.3. ARCH and GARCH Processes

As we have already seen from the bilinear processes, nonlinearity can have some adverse effects on the performance of the GA. It would be imperative to know whether this finding is just restricted to a specific class of nonlinear processes or can be generalized to other nonlinear processes. In this and the next two sections, we shall focus on this question, and briefly mention other details when we see the necessity.

Let us first take a look at the results of the other two nonlinear stochastic processes, namely, ARCH and GARCH. Just like what we saw in the bilinear processes, these two classes of processes can become pure nonlinear stochastic if some specific coefficient values are set to zero. This is basically what we do

Table 8. Performance Statistics of the OGA and B&H – ARCH.

Code	Model	$\bar{\pi}^1$	$\bar{\pi}^2$	$\bar{\pi}^*$	z_π	$\tilde{\pi}$ (%)	$\dot{\pi}$ (%)	$\hat{p}_w$	z_w
AH-1	AR(0)-ARCH(1)	1.038	1.013	3.195	−1.99	−1	−66	0.546	2.92
AH-2	AR(0)-ARCH(1)	1.001	1.005	4.251	0.19	0	400	0.592	5.92
AH-3	AR(0)-ARCH(2)	0.985	0.991	2.307	0.67	0	40	0.562	3.95
AH-4	AR(0)-ARCH(2)	1.007	0.997	2.268	−1.09	−1	−143	0.529	1.84
AH-5	AR(1)-ARCH(1)	1.175	1.509	2.187	22.88	33	191	0.862	33.19
AH-6	AR(1)-ARCH(2)	1.300	1.705	3.061	17.64	23	135	0.838	29.01
AH-7	MA(1)-ARCH(1)	0.869	1.551	3.602	44.12	25	521	0.959	73.20
Code	**Model**	$\hat{s}_1$	$\hat{s}_2$	$\hat{d}$	z_d	$\hat{l}^1_{0.05}$	$\hat{l}^2_{0.05}$	$\hat{r}_{0.05}$	z_r
AH-1	AR(0)-ARCH(1)	0.170	0.038	−0.032	−1.33	0.117	0.091	1.285	4.53
AH-2	AR(0)-ARCH(1)	0.001	0.010	0.009	0.34	0.149	0.105	1.411	3.19
AH-3	AR(0)-ARCH(2)	−0.038	−0.035	0.002	0.09	0.100	0.079	1.269	4.03
AH-4	AR(0)-ARCH(2)	0.017	−0.012	−0.030	−1.22	0.099	0.080	1.246	3.24
AH-5	AR(1)-ARCH(1)	0.211	0.774	0.563	15.42	0.145	0.109	1.331	3.43
AH-6	AR(1)-ARCH(2)	0.221	0.605	0.384	10.79	0.187	0.140	1.332	2.15
AH-7	MA(1)-ARCH(1)	−0.641	1.126	1.766	35.75	0.076	0.086	0.889	−3.44

Note: $\bar{\pi}^1$, $\bar{\pi}^2$ and $\bar{\pi}^*$ are the respective sample mean return of OGA, B&H and the omniscient trader. $\tilde{\pi}$ is the exploitation ratio (Eq. (26)), and $\dot{\pi}$ is the relative superiority index (Eq. 27). $\hat{p}_w$ is the sample winning probability of OGA over B&H (Eq. (28)). $\hat{s}_1$ and $\hat{s}_2$ are the corresponding sample Sharpe ratio of OGA and B&H (Eq. (31)). Their sample difference is $\hat{d}$ (Eq. (32)). $\hat{l}^1_{0.05}$ and $\hat{l}^2_{0.05}$ are the sample luck coefficient of OGA and B&H (Eq. (39)), and $\hat{r}_{0.05}$ is the sample luck coefficient ratio between the two (Eq. (40)). The z_π, z_w, z_d and z_r are the test statistics of the mean return difference, winning probability, Sharpe ratio differential, and luck coefficient ratio, respectively. The critical value of them is 1.28 at the 10% significance level, and is 1.64 at the 5% significance level.

in Tables 3 and 4. Notice that, based on these settings, AH-1 to AH-4 (ARCH) and GH-1 to GH-6 (GARCH) are all pure nonlinear stochastic processes, i.e. pure ARCH or pure GARCH without linear ARMA components. For the rest, they are a mixture of pure ARCH (GARCH) and linear ARMA processes. Tables 8 and 9 summarize the results of the two stochastic processes. A striking feature is that, in contrast to its performance in mixed processes, the GA performed dramatically worse in pure nonlinear ARCH and GARCH scenarios.

Let us take the ARCH processes as an illustration. In the mixed processes AH-5, AH-6 and AH-7, the GA has a probability of up to 80% or higher of beating B&H, and earned 135–521% more than B&H. The fact that these excess returns are not compensation for risk is further confirmed by the Sharpe-ratio differentials which are significantly positive. In addition, the GA exploited 23% to 33% of the potential returns earned by the omniscient trader. However, when coming to the pure

Table 9. Performance Statistics of the OGA and B&H – GARCH.

Code	Model	$\bar{\pi}^1$	$\bar{\pi}^2$	$\bar{\pi}^*$	z_π	$\tilde{\pi}$ (%)	$\dot{\pi}$ (%)	$\hat{p}_w$	z_w
GH-1	AR(0)-GARCH(1,1)	0.987	0.983	2.457	−0.42	0	−31	0.539	2.47
GH-2	AR(0)-GARCH(1,1)	0.968	0.979	2.580	1.19	1	34	0.554	3.44
GH-3	AR(0)-GARCH(1,2)	1.008	1.007	2.474	−0.04	0	−13	0.544	2.79
GH-4	AR(0)-GARCH(1,2)	0.998	1.007	2.434	0.90	1	450	0.572	4.60
GH-5	AR(0)-GARCH(2,1)	0.978	1.001	2.637	2.24	1	105	0.584	5.39
GH-6	AR(0)-GARCH(2,1)	0.982	0.997	2.595	1.50	1	83	0.563	4.02
GH-7	AR(1)-GARCH(1,1)	1.428	1.926	3.511	18.40	24	116	0.856	32.07
GH-8	AR(1)-GARCH(1,2)	1.356	1.747	3.298	12.58	20	110	0.841	29.49
GH-9	AR(1)-GARCH(2,1)	1.378	1.934	3.616	19.20	25	147	0.872	35.21
GH-10	MA(1)-GARCH(1,1)	0.911	1.376	2.769	36.44	25	521	0.949	64.54
Code	Model	$\hat{s}_1$	$\hat{s}_2$	$\hat{d}$	z_d	$\hat{l}^1_{0.05}$	$\hat{l}^2_{0.05}$	$\hat{r}_{0.05}$	z_r
GH-1	AR(0)-GARCH(1,1)	−0.030	−0.652	−0.035	−1.19	0.101	0.079	1.282	4.30
GH-2	AR(0)-GARCH(1,1)	−0.080	−0.076	0.004	0.17	0.098	0.081	1.202	4.08
GH-3	AR(0)-GARCH(1,2)	−0.005	0.020	0.024	1.05	0.094	0.081	1.166	3.32
GH-4	AR(0)-GARCH(1,2)	0.020	0.026	0.007	0.27	0.108	0.093	1.151	1.68
GH-5	AR(0)-GARCH(2,1)	−0.051	0.005	0.056	2.04	0.103	0.083	1.233	4.10
GH-6	AR(0)-GARCH(2,1)	−0.044	−0.012	0.032	1.23	0.097	0.083	1.178	3.50
GH-7	AR(1)-GARCH(1,1)	0.244	0.620	0.375	11.06	0.225	0.158	1.426	2.72
GH-8	AR(1)-GARCH(1,2)	0.231	0.614	0.383	14.52	0.201	0.143	1.405	2.59
GH-9	AR(1)-GARCH(2,1)	0.703	0.239	0.465	13.47	0.213	0.147	1.454	3.13
GH-10	MA(1)-GARCH(1,1)	−0.476	1.034	1.509	29.43	0.070	0.081	0.867	−3.90

Note: $\bar{\pi}^1$, $\bar{\pi}^2$ and $\bar{\pi}^*$ are the respective sample mean return of OGA, B&H and the omniscient trader. $\tilde{\pi}$ is the exploitation ratio (Eq. (26)), and $\dot{\pi}$ is the relative superiority index (Eq. (27)). $\hat{p}_w$ is the sample winning probability of OGA over B&H (Eq. (28)). $\hat{s}_1$ and $\hat{s}_2$ are the corresponding sample Sharpe ratio of OGA and B&H (Eq. (31)). Their sample difference is $\hat{d}$ (Eq. (32)). $\hat{l}^1_{0.05}$ and $\hat{l}^2_{0.05}$ are the sample luck coefficient of OGA and B&H (Eq. (39)), and $\hat{r}_{0.05}$ is the sample luck coefficient ratio between the two (Eq. (40)). The z_π, z_w, z_d and z_r are the test statistics of the mean return difference, winning probability, Sharpe ratio differential, and luck coefficient ratio, respectively. The critical value of them is 1.28 at the 10% significance level, and is 1.64 at the 5% significance level.

nonlinear processes AH-1 to AH-4, this dominance either disappears or becomes weaker. This can be easily shown by the sharp decline in the statistics z_p, z_w and z_d in Table 8 with an almost 0% exploitation ($\tilde{\pi}$) of the maximum potential returns.

This discernible pattern also extends to Table 9. The double-digit z_p, z_w, and z_d of the mixed processes (GH-7 to GH-10) distinguish themselves from the low, or even negative, single-digit ones of the pure nonlinear processes (GH-1 to GH-6). For the former, the GA has 84–95% chance of beating B&H and earned 110–521% more than B&H. Again, from z_d, we know that the high returns are more than compensation for risk. Very similar to the case of ARCH, 20–25% of the maximum potential returns can be exploited by the GA, but that value $\tilde{\pi}$ drops near to 0% when the underlying processes change to pure GARCH.

Despite the fact that pure nonlinear processes continue to deal the GA a hard blow, as far as the winning probability is concerned, its relative performance to B&H is overwhelmingly good. This can be reflected by the z_w statistics which are consistently significantly positive in all cases. A similar property holds for the luck coefficient (see z_r). The only two exceptions are the cases AH-7 and GH-10, which, however, are not pure nonlinear. In fact, they both have MA(1) as their linear component.

6.4. Threshold Processes

The threshold process leads to a different kind of nonlinear process. While its global behavior is nonlinear, within each local territory, characterized by Ω_i, it can be linear. TH-1 and TH-2 in Table 5 are exactly processes of this kind. The former is switching between two AR(1) processes, whereas the latter is switching between two ARMA(1,1) processes. Since the GA can work well with linear processes, it would be interesting to know whether its effectiveness will extend to these local linear processes. Our results are shown in Table 10. The four statistics z_π, z_w, z_d, and z_r all give positive results. The GA is seen to exploit 20–30% of the maximum potential returns, and the winning probabilities are greater than 90%.

TH-4 and TH-5 are another kind of complication. TH-4 switches between two mixed processes, while TH-5 switches between a pure nonlinear process and a linear process. From previous experiences, we already knew that the GA can work well with the mixed process. Now, from Table 10, it seems clear that it can survive these two complications as well.

Finally, we come to the most difficult one TH-5, i.e the one which switches between two pure nonlinear (bilinear) processes. Since the GA did not show its competence in the pure nonlinear process, at least from the perspective of the return criteria, one may conjecture that TH-5 will deal another hard blow to the

Table 10. Performance Statistics of the OGA and B&H – Threshold.

Code	Model	$\bar{\pi}^1$	$\bar{\pi}^2$	$\bar{\pi}^*$	z_π	$\tilde{\pi}$ (%)	$\dot{\pi}$ (%)	$\hat{p}_w$	z_w
TH-1	TBL(2;1,0,0,0)	0.612	1.233	3.372	24.89	23	160	0.910	45.30
TH-2	TBL(2;1,1,0,0)	1.262	2.743	6.361	21.15	29	565	0.931	53.77
TH-3	TBL(2;0,0,1,1)	1.161	1.074	4.207	–4.38	–3	–54	0.502	0.13
TH-4	TBL(2;1,1,2,2)	1.271	1.406	4.497	5.41	4	50	0.717	15.23
TH-5	TBL(2;2,0,2,2)	0.654	1.236	3.890	37.38	18	168	0.919	48.56

Code	Model	$\hat{s}_1$	$\hat{s}_2$	$\hat{d}$	z_d	$\hat{l}^1_{0.05}$	$\hat{l}^2_{0.05}$	$\hat{r}_{0.05}$	z_r
TH-1	TBL(2;1,0,0,0)	–0.398	0.374	0.772	9.33	0.267	0.119	2.252	4.30
TH-2	TBL(2;1,1,0,0)	0.093	0.727	0.634	11.86	0.329	0.163	2.012	2.95
TH-3	TBL(2;0,0,1,1)	0.208	0.176	–0.032	–1.42	0.136	0.098	1.394	3.72
TH-4	TBL(2;1,1,2,2)	0.208	0.426	0.219	10.41	0.192	0.140	1.379	2.97
TH-5	TBL(2;2,0,2,2)	–0.813	0.484	1.297	16.88	0.130	0.097	1.343	3.54

Note: $\bar{\pi}^1$, $\bar{\pi}^2$ and $\bar{\pi}^*$ are the respective sample mean return of OGA, B&H and the omniscient trader. $\tilde{\pi}$ is the exploitation ratio (Eq. (26)), and $\dot{\pi}$ is the relative superiority index (Eq. (27)). $\hat{p}_w$ is the sample winning probability of OGA over B&H (Eq. (28)). $\hat{s}_1$ and $\hat{s}_2$ are the corresponding sample Sharpe ratio of OGA and B&H (Eq. (31)). Their sample difference is $\hat{d}$ (Eq. (32)). $\hat{l}^1_{0.05}$ and $\hat{l}^2_{0.05}$ are the sample luck coefficient of OGA and B&H (Eq. (39)), and $\hat{r}_{0.05}$ is the sample luck coefficient ratio between the two (Eq. (40)). The z_π, z_w, z_d and z_r are the test statistics of the mean return difference, winning probability, Sharpe ratio differential, and luck coefficient ratio, respectively. The critical value of them is 1.28 at the 10% significance level, and is 1.64 at the 5% significance level.

GA. Both z_p and z_d in Table 10 confirm this conjecture. Not just the returns, but z_w shows that the winning probability is also not good, which is similar to what we experienced in BL-1 and BL-2. The only criterion that remains unaffected by this complication is the luck coefficient. Furthermore, it turns out that z_r seems to give the most stable performance across all kinds of processes considered so far, except the MA process.

6.5. Chaotic Processes

Chaotic processes are also nonlinear, but they differ from the previous four nonlinear processes in that they are *deterministic* rather than *stochastic*. These processes can behave quite erratically without any discernible pattern. Can the GA survive well with this type of nonlinear process? The answer is a resounding *yes*. All the statistics in Table 11 are sending us this message.

The winning probabilities are all higher than 85%. In the case of the Henon map (C-2), the GA even beats B&H in all of the 1000 trials. In addition, in this

Table 11. Performance Statistics of the OGA and B&H – Chaos.

Code	$\bar{\pi}^1$	$\bar{\pi}^2$	$\bar{\pi}^*$	z_π	$\tilde{\pi}$ (%)	$\dot{\pi}$ (%)	$\hat{p}_w$	z_w
C-1	1.019	5.664	21.876	31.15	22	24447	0.993	186.99
C-2	5.387	23.235	33.452	85.62	64	407	1.000	*
C-3	0.937	4.124	11.374	44.65	31	5059	0.990	352.49
C-4	1.188	3.066	25.563	22.91	8	999	0.950	65.29
C-5	0.928	1.790	23.172	17.18	4	1197	0.876	36.08
Code	$\hat{s}_1$	$\hat{s}_2$	$\hat{d}$	z_d	$\hat{l}^1_{0.05}$	$\hat{l}^2_{0.05}$	$\hat{r}_{0.05}$	z_r
C-1	0.009	0.832	0.824	16.59	0.297	0.184	1.615	2.28
C-2	1.600	2.502	0.901	23.56	0.112	0.090	1.252	4.39
C-3	−0.075	1.160	1.235	28.92	0.153	0.127	1.207	2.75
C-4	0.074	0.627	0.554	10.39	0.348	0.200	1.739	2.66
C-5	−0.045	0.518	0.563	14.45	0.279	0.169	1.649	2.88

Note: $\bar{\pi}^1$, $\bar{\pi}^2$ and $\bar{\pi}^*$ are the respective sample mean return of OGA, B&H and the omniscient trader. $\tilde{\pi}$ is the exploitation ratio (Eq. (26)), and $\dot{\pi}$ is the relative superiority index (Eq. (27)). $\hat{p}_w$ is the sample winning probability of OGA over B&H (Eq. (28)). $\hat{s}_1$ and $\hat{s}_2$ are the corresponding sample Sharpe ratio of OGA and B&H (Eq. (31)). Their sample difference is $\hat{d}$ (Eq. (32)). $\hat{l}^1_{0.05}$ and $\hat{l}^2_{0.05}$ are the sample luck coefficient of OGA and B&H (Eq. (39)), and $\hat{r}_{0.05}$ is the sample luck coefficient ratio between the two (Eq. (40)). The z_π, z_w, z_d and z_r are the test statistics of the mean return difference, winning probability, Sharpe ratio differential, and luck coefficient ratio, respectively. The critical value of them is 1.28 at the 10% significance level, and is 1.64 at the 5% significance level.

map, the GA is seen to exploited 64% of the potential excess returns earned by the omniscient trader, which is the highest of all the processes tested in this paper. One of the possible reasons why the GA can work well with these nonlinear deterministic processes is that they are not pure nonlinear. C-1, C-2 and C-4 have linear AR(1) or AR(2) components. C-3, like the threshold processes, switches between two linear processes. As already evidenced in Section 6.4, the GA can handle these types of processes effectively. So, the success is not totally unanticipated.

However, the explanation above does not apply to C-5, which has no linear component. Nonetheless, statistics such as z_π, $\tilde{\pi}$ and $\hat{p}_w$ all indicate that this process is not as easy as the other four. For example, only 4% of the potential excess returns are exploited in this process. Regardless of these weaknesses, the fact that the GA can dominate B&H in this case motivates us to ask the following question: *Can the GA work better for the pure nonlinear deterministic processes than the respective stochastic ones, and hence can it help distinguish the chaotic processes from the stochastic processes*? This is a question to pursue in the future.

6.6. Summary

The Monte Carlo simulation analysis conducted above provides us with an underpinning of the practical financial applications of the GA. It pinpoints the kinds of stochastic processes which we may like to see fruitful results. We have found that the GA can perform well with all kinds of stochastic processes which have a linear process (signal) as a part of them. Preliminary studies also suggest that it may also work well with chaotic processes. However, the class of nonlinear stochastic processes presents a severe limitation for the GA. In the next section, we shall see the empirical relevance of these results by actually applying OGA-based trading strategies to financial data.

7. EMPIRICAL ANALYSIS

7.1. Data Description and Analysis

The empirical counterpart of this paper is based on two sets of high-frequency time series data regarding foreign exchange rates, namely, the Euro dollar vs. the U.S. dollar *EUR/USD* and the U.S. dollar vs. the Japanese yen *USD/JPY*.[6] The data is from January 11, 1999 to April 17, 1999. Data within this period are further divided into 12 sub-periods with roughly equal numbers of observations. Table 12 gives the details.

Let $P^U_{i,t}(P^P_{i,t})$ denote the t-th ($t = 1, 2, \ldots, n_i$) observation of the ith sub-period ($i = A, B, \ldots, L$) of the EUR/USD (USD/JPY) forex series. The price series is transformed into the return series by the usual logarithmic formulation,

$$r^j_{i,t} = \ln(P^j_{i,t}) - \ln(P^j_{i,t-1}) \tag{47}$$

where $j = U, P$. Tables 13 and 14 give some basic statistics of the returns of each sub-period.

Both return series share some common features. From Tables 13 and 14, the mean, median and skewness of these two return series are all close to zero. The kurtosis is much higher than 3, featuring the well-known *fat-tail* property. The Jarque-Bera (1980) test further confirms that these forex returns do not follow the normal distribution, and that is true for each sub-period. In addition, the series is not independent due to its significant negative first-order serial correlation ρ_1. However, there is no evidence of serial correlation in higher orders.[7]

To apply what we learned from the Monte Carlo simulation to predict the effectiveness of the GA over these series, we must first gauge their likely stochastic processes. Here we follow a standard procedure frequently used in econometrics

Table 12. Data Quotations – EUR/USD and USD/JPY.

Sub-Period	A	B	C	D	E	F
EUR/USD						
Number	12000	12000	12000	12000	12000	12000
From (GMT)	2/25 7:59	3/1 0:59	3/3 15:36	3/8 6:43	3/10 6:53	3/12 7:26
To (GMT)	2/26 8:22	3/2 7:17	3/5 3:04	3/9 1:08	3/11 7:12	3/15 1:16
Sub-Period	G	H	I	J	K	L
Number	12000	12000	12000	12000	12000	12000
From (GMT)	3/17 7:36	3/19 0:19	3/24 15:06	3/26 15:46	3/31 7:32	4/15 6:14
To (GMT)	3/18 6:12	3/22 2:01	3/26 2:12	3/30 6:23	4/02 1:14	4/17 0:37
Sub-Period	A	B	C	D	E	F
USD/JPY						
Number	12000	12000	12000	12000	12000	10808
From (GMT)	1/11 6:11	1/15 0:00	1/27 15:14	2/04 8:47	2/17 7:20	2/23 6:10
To (GMT)	1/14 8:11	1/21 0:00	2/03 3:24	2/11 2:43	2/23 6:09	2/26 21:48
Sub-Period	G	H	I	J	K	L
Number	12000	12000	11026	12000	12000	12000
From (GMT)	2/28 18:15	3/04 10:02	3/09 21:52	3/15 5:25	3/18 6:07	3/24 13:00
To (GMT)	3/04 10:01	3/09 21:52	3/15 1:21	3/18 6:06	3/24 13:00	3/30 10:41

Note: GMT: Greenwich Mean Time.

(Chen & Lu, 1999). First, notice that all series used in our Monte Carlo simulation are *stationary*. To make sure that the forex returns are stationary, the *Augmented Dickey-Fuller* (*ADF*) test is applied (Dickey & Fuller, 1979). From Table 15, the null hypothesis that $r^j_{i,t}$ contains a unit root is rejected at the 1% significance level, meaning that the $r^j_{i,t}$ are stationary.

Second, since our Monte Carlo simulations demonstrate the effectiveness of the GA over the linear stochastic processes, it is important to know whether the forex returns have a linear component. To do so, the famous Rissanen's predictive stochastic complexity (*PSC*) as a linear filter is taken.[8] Table 15 gives the ARMA(p, q) process extracted from the forex return series. A MA(1) linear process is founded for both forex returns in each sub-period. In fact, it re-confirms the early finding that the high-frequency forex returns follow a MA(1) process (Moody & Wu, 1997; Zhou, 1996).

Third, it should be not surprising if none of these series is just linear. To see whether nonlinear dependence exists, one of the most frequently used statistics, the *BDS* test, is applied to the residuals filtered through the PSC filter.[9] There

Table 13. Basic Statistics of the Return Series – EUR/USD.

Sub-Period	A	B	C	D	E	F
Mean	−2.56E−07	−8.13E−07	−7.37E−07	5.39E−07	5.63E−07	−7.49E−07
Median	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Std. Dev.	0.000252	0.000252	0.000213	0.000191	0.000238	0.000264
Skewness	−0.015831	0.007214	−0.034436	0.002017	−0.001071	−0.009908
Kurtosis	5.606484	5.558600	5.636056	5.976148	6.136196	5.757020
Jarque-Bera	3397.10	3273.05	3476.48	4428.37	4917.45	3800.46
P-value	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
ρ_1	−0.513935	−0.503725	−0.494695	−0.504014	−0.486925	−0.509612
Sub-Period	**G**	**H**	**I**	**J**	**K**	**L**
Mean	3.81E−07	−8.00E−07	−7.48E−07	−5.64E−08	2.37E−07	−1.13E−06
Median	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Std. Dev.	0.000225	0.000217	0.000184	0.000241	0.000292	0.000219
Skewness	0.011155	−0.050369	−0.119412	0.007646	−0.021431	−0.203838
Kurtosis	6.512019	5.435495	6.226714	5.337107	8.780986	10.97326
Jarque-Bera	6166.88	2970.40	5233.92	2730.92	16708.03	31861.55
P-value	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
ρ_1	−0.493223	−0.505528	−0.480500	−0.498232	−0.475452	−0.464571

Note: ρ_1 is the first-order autocorrelation coefficient. Jarque-Bera statistic converges to a chi-square distribution with two degrees of freedom under the normality assumption.

Table 14. Basic Statistics of the Return Series – USD/JPY.

Sub-Period	A	B	C	D	E	F
Mean	3.97E−07	−5.16E−07	−2.01E−06	2.54E−07	1.69E−06	−1.44E−06
Median	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Std. Dev.	0.000413	0.002108	0.001853	0.000332	0.000311	0.000363
Skewness	0.008135	0.080038	−0.018340	−0.057694	0.022959	−0.003358
Kurtosis	6.769064	6.711594	6.854310	7.170642	6.757800	6.374525
Jarque-Bera	7091.806	6898.478	7426.049	8700.883	7059.230	5123.885
P-value	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
ρ_1	−0.343317	−0.338790	−0.370748	−0.362052	−0.360786	−0.335953
Sub-Period	**G**	**H**	**I**	**J**	**K**	**L**
Mean	2.53E−06	−1.09E−06	−2.54E−06	−2.75E−07	−7.87E−07	1.90E−06
Median	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Std. Dev.	0.000301	0.000279	0.000322	0.000287	0.000265	0.000247
Skewness	0.080100	0.019734	0.079313	0.002414	−0.019244	0.213584
Kurtosis	5.597214	6.763973	6.747828	8.198238	7.650768	6.701801
Jarque-Bera	3385.029	7083.936	6459.934	13508.60	10811.96	6941.746
P-value	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
ρ_1	−0.436860	−0.396329	−0.344660	−0.348622	−0.361993	−0.364189

Note: ρ_1 is the first-order autocorrelation coefficient. Jarque-Bera statistic converges to a chi-square distribution with two degrees of freedom under the normality assumption.

Table 15. Basic Econometric Properties of the Return Series – EUR/USD and USD/JPY.

Sub-Period	A	B	C	D	E	F
EUR/USD						
ADF	−74.9502	−76.4264	−74.0755	−76.6226	−77.4292	−79.1714
Critical Value	−3.4341	−3.4341	−3.4341	−3.4341	−3.4341	−3.4341
PSC	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)
	G	H	I	J	K	L
ADF	−74.7427	−74.7053	−68.8254	−73.4958	−72.3726	−67.6148
Critical Value	−3.4341	−3.4341	−3.4341	−3.4341	−3.4341	−3.4341
PSC	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0.1)
	A	B	C	D	E	F
USD/JPY						
ADF	−57.1573	−55.2394	−56.0518	−56.8433	−55.0202	−51.1507
Critical Value	−2.5660	−2.5660	−2.5660	−3.4341	−3.4341	−3.4342
PSC	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)
	G	H	I	J	K	L
ADF	−59.3422	−57.4123	−55.5809	−58.0822	−57.5485	−59.5623
Critical Value	−3.4341	−3.4341	−3.4341	−3.4341	−3.4341	−3.4341
PSC	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0.1)

Note: The "Critical Value" indicates the critical value of the ADF test that is taken from the table provided by Dickey and Fuller at the 1% significance level.

are two parameters used to conduct the BDS test. One is the distance measure (standard deviations), and the other is the embedding dimension. The parameter "ε" considered here is equal to one standard deviation. (In fact, other are also tried, but the results are not sensitive to the choice of ε.) The embedding dimensions considered range from 2 to 5. Following Barnett et al. (1997), if the absolute values of all BDS statistics under various embedding dimensions are greater than 1.96, the null hypothesis of an identical independent distribution (IID) is rejected. From Table 16, the BDS statistics for the EUR/USD and USD/JPY are all large enough to reject the null hypothesis, i.e. nonlinear dependence is detected.

Fourth, given the existence of the nonlinear dependence, the next step is to identify its possible form, i.e. by modeling nonlinearity. While there is no standard answer as to how this can be done, the voluminous (G)ARCH literature over the past two decades has proposed a second-moment connection (Bollerslev et al., 1992). In order to see whether (G)ARCH can successfully capture nonlinear signals, we

Table 16. The BDS Test of the PSC-filtered Return Series – EUR/USD and USD/JPY.

Sub-Period	A		B		C		D		E		F	
Part	I	II	I	II	I	II	I	II	I	II	I	II
EUR/USD												
DIM = 2	20.47	26.82	22.58	26.56	13.60	20.25	17.15	14.66	18.23	18.09	18.03	19.37
DIM = 3	27.57	34.17	30.61	34.72	19.44	26.84	22.50	20.12	22.78	23.48	24.63	26.43
DIM = 4	33.60	40.03	37.25	40.81	23.80	31.27	26.80	24.22	25.68	27.63	30.21	32.09
DIM = 5	38.50	45.80	43.40	46.75	27.43	35.23	30.38	27.40	28.54	31.23	35.26	37.94
	G		H		I		J		K		L	
	I	II	I	II	I	II	I	II	I	II	I	II
DIM = 2	12.04	16.97	23.90	19.45	13.06	12.40	20.13	13.41	35.69	19.74	8.18	22.23
DIM = 3	17.84	22.20	30.02	25.59	17.30	17.31	26.84	18.79	46.83	24.39	10.98	27.08
DIM = 4	21.09	26.34	34.39	30.41	20.35	20.57	31.24	22.98	56.42	27.22	12.97	30.22
DIM = 5	24.08	30.18	39.31	35.47	23.29	23.40	35.39	26.48	66.58	29.79	14.20	33.13
	A		B		C		D		E		F	
	I	II	I	II	I	II	I	II	I	II	I	II
USD/JPY												
DIM = 2	15.36	23.15	15.68	13.41	12.00	16.63	14.76	20.44	12.98	17.84	17.88	16.61
DIM = 3	17.89	28.38	18.83	16.04	14.54	20.02	17.11	23.15	16.08	20.87	21.35	18.94
DIM = 4	20.03	31.37	20.17	17.89	15.32	22.24	18.72	24.27	17.49	22.82	23.35	20.44
DIM = 5	22.30	34.58	21.57	19.13	16.07	24.42	20.28	25.43	18.52	24.56	24.43	22.16
	G		H		I		J		K		L	
	I	II	I	II	I	II	I	II	I	II	I	II
DIM = 2	15.65	11.34	15.56	16.84	16.44	15.51	20.98	17.79	19.41	15.51	15.28	15.61
DIM = 3	17.64	13.92	18.57	18.91	18.50	18.68	25.07	21.84	21.94	16.84	16.32	17.87
DIM = 4	19.30	15.35	20.86	19.45	19.78	21.02	27.72	24.43	23.23	17.52	17.21	19.34
DIM = 5	20.82	16.49	23.10	19.73	20.95	22.76	30.10	26.45	24.15	18.56	18.14	20.62

Note: Due to the size of the data which is beyond the affordable limit of the software computing the BDS statistics, each sub-period was divided into two parts before the BDS test was applied. The BDS statistic follows an asymptotically standard normal distribution.

carry out the *Lagrange Multiplier* (*LM*) test for the presence of ARCH effects. The LM test for ARCH effects is a test based on the following model:

$$\sigma_t^2 = h(\alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \ldots + \alpha_p \varepsilon_{t-p}^2), \tag{48}$$

where h is a differential function. The null hypothesis that the ARCH effect does not exist is

$$\alpha_1 = \ldots = \alpha_p = 0. \tag{49}$$

By taking $p = 1, 2, \ldots, 4$, the LM test results are given in Table 17. It is found that the ARCH effect does exist in both return series.

Table 17. The LM Test of the ARCH Effect in the Return Series – EUR/USD and USD/JPY.

Sub-Period	A	B	C	D	E	F
EUR/USD						
$p=1$	1029.94	821.665	681.92	560.27	463.98	401.08
$p=2$	1572.34	1191.26	998.22	1094.72	960.83	585.88
$p=3$	2030.32	1501.74	1202.15	1320.58	1052.54	705.17
$p=4$	2169.98	1731.33	1295.77	1471.40	1195.93	871.73
Sub-Period	G	H	I	J	K	L
$p=1$	275.07	797.26	411.61	390.94	1584.30	1571.04
$p=2$	423.33	1168.19	689.02	553.11	1668.88	1587.53
$p=3$	493.11	1262.87	1001.22	678.90	1714.39	1640.60
$p=4$	551.99	1354.28	1050.53	715.68	2036.42	1641.41
Sub-Period	A	B	C	D	E	F
USD/JPY						
$p=1$	533.15	411.35	479.80	769.49	550.15	685.34
$p=2$	639.75	490.58	6018.02	849.31	604.18	752.71
$p=3$	677.49	531.78	667.50	854.11	614.26	821.85
$p=4$	709.00	559.97	687.09	923.01	636.99	854.71
Sub-Period	G	H	I	J	K	L
$p=1$	600.528	545.791	696.185	749.650	883.107	795.762
$p=2$	648.101	656.653	758.918	1094.82	926.127	929.618
$p=3$	695.639	727.043	811.000	1101.78	939.221	1059.00
$p=4$	726.942	764.836	844.766	1103.08	951.489	1109.23

Note: The LM test is asymptotically distributed as χ^2 with p degrees of freedom when the null hypothesis is true. There is no need to report the p values here because they are all 0.0000.

After these series of statistical tests, we may conclude that basically both *the EUR/USD and the USD/JPY return series have MA(1) as a linear component and ARCH as a part of its nonlinear components*. In Section 6.3, the Monte Carlo simulation analysis already indicated that the GA can work well with MA(1) plus (G)ARCH processes. To see the empirical relevance of the simulation study, in the next sections, the GA is applied to the two return series.

7.2. Experimental Design

In order to compare the empirical results with our earlier simulation analysis, the experiments are designed in a similar fashion to the one which our Monte Carlo simulation follows. Specifically, many "*ensembles*" are generated from the original series to evaluate the performance of the GA. Of course, rigorously speaking, they are not the "ensembles" defined in the stochastic process. They are just subseries taken from the original return series. Each subseries has 105 observations. The first 70 observations are treated as the training sample, and the last 35 observations are used as the testing sample.

Nonetheless, to make the tests we developed in Section 4 applicable, we cannot just continuously chop the return series into subseries, because doing so will not make the sampling process independent, and hence will violate the fundamental assumption required for the central limit theorem. One solution to this problem is to leave an interval between any two consecutive subseries so that they are not immediately connected. The purpose in doing this is hopefully to make them independent of each other as if they were sampled independently. However, how large an interval would suffice? To answer this question, we take a subsequence with a fixed number of lags, say, $\{r^j_{i,t}, r^j_{i,t+k}, r^j_{i,t+2k}, \ldots\}$ from the original return series, where k varies from 40, 60, . . . , to 300. We then apply the BDS test to each of these subsequences.

Table 18 summarizes the BDS test results. For the EUR/USD case, it is found that when k is greater than 100, the null hypothesis that the subsequence $\{r^j_{i,t}, r^j_{i,t+k}, r^j_{i,t+2k}, \ldots\}$ is IID is not rejected. In other words, leaving an interval of 100 observations between each of two consecutive subseries would suffice. For the EUR/USD case, k can even be smaller than 60. To ensure the quality of the sampling process, we, however, take an even larger number of lags, i.e. $k = 200$. This choice leaves us with a total of 720 subseries from the EUR/USD and 709 subseries from the USD/JPY.

The GA is then employed to extract trading strategies from the training samples of these subseries, and the strategies extracted are further applied to the respective testing samples. The resulting accumulated returns (p) are then compared with that of the B&H strategy.

7.3. Results of the Experiments

Since the analysis of the data shows that the two forex returns are mixtures of MA(1) and (G)ARCH processes, our previous results of Monte Carlo simulations may provide a good reference for what one can expect from such empirical applications. Both Tables 8 and 9 indicate the superior performance of the GA over B&H, except in relation to the criterion for the luck coefficient, when the underlying stochastic processes are MA plus (G)ARCH. Will the dominance carry over?

Table 18. The BDS Test of the Lag Period in the Return Series – EUR/USD and USD/JPY.

Lag	DIM = 2	DIM = 3	DIM = 4	DIM = 5
EUR/USD				
40	2.94	3.45	3.86	4.18
60	0.72	1.20	1.27	1.38
80	1.11	1.21	1.38	1.50
100	0.66	0.66	0.69	0.69
120	0.61	0.66	0.79	0.88
140	0.45	0.52	0.54	0.58
160	0.30	0.43	0.46	0.54
180	0.21	0.30	0.42	0.49
200	−0.01	0.08	0.12	0.11
220	0.11	0.14	0.13	0.13
240	0.25	0.24	0.27	0.24
260	−0.02	−0.04	−0.04	−0.01
280	0.10	0.11	0.14	0.14
300	0.06	0.07	0.05	0.01
USD/JPY				
40	1.39	1.50	1.50	1.57
60	0.53	0.69	0.75	0.89
80	0.56	0.63	0.72	0.80
100	−0.08	−0.12	−0.12	−0.16
120	0.13	0.22	0.19	0.20
140	0.01	−0.13	−0.14	−0.09
160	0.05	0.09	0.09	0.12
180	−0.01	−0.07	0.01	0.06
200	−0.04	−0.08	−0.08	−0.06
220	0.21	0.29	0.30	0.32
240	0.15	0.13	0.11	0.12
260	0.05	0.12	0.09	0.07
280	−0.14	−0.09	−0.11	−0.10
300	0.06	0.02	0.05	0.04

Note: The BDS statistic follows an asymptotically standard normal distribution.

Table 19 is the kind of table which we have presented many times in Section 6. All the key statistics z_p, z_w, and z_d are consistent with those of AH-7 (Table 8) and GH-10 (Table 9). So, in both forex return series, the dominance of the GA over B&H is statistically significant. The consistency continues even to a finer level of the results: $\bar{\pi}^1 < 1$ and $\bar{\pi}^2 > 1$. As already seen, B&H earned negative profits in both of the cases AH-7 and GH-10, while the GA earned positive profits in both cases. In addition, both the winning probability and the exploitation ratio are also

Table 19. Performance Statistics of the OGA and B&H – EUR/USD and USD/JPY.

	$\overline{\pi}^1$	$\overline{\pi}^2$	$\overline{\pi}^*$	z_π	$\tilde{\pi}$ (%)	$\dot{\pi}$ (%)	$\hat{p}_w$	z_w
EUR/USD	0.9999	1.0012	1.0028	38.58	43	9257	0.972	77.10
USD/JPY	0.9999	1.0010	1.0039	23.70	27	11462	0.850	26.17
	$\hat{s}_1$	$\hat{s}_2$	$\hat{d}$	z_d	$\hat{l}^1_{0.05}$	$\hat{l}^2_{0.05}$	$\hat{r}_{0.05}$	z_r
EUR/USD	−0.0338	1.4193	1.4532	18.32	0.0812	0.0933	0.8710	−1.69
USD/JPY	−0.0086	0.8786	0.8873	20.64	0.0826	0.0948	0.8713	−1.66

Note: $\overline{\pi}^1$, $\overline{\pi}^2$ and $\overline{\pi}^*$ are the respective sample mean return of OGA, B&H and the omniscient trader. $\tilde{\pi}$ is the exploitation ratio (Eq. (26)), and $\dot{\pi}$ is the relative superiority index (Eq. (27)). $\hat{p}_w$ is the sample winning probability of OGA over B&H (Eq. (28)). $\hat{s}_1$ and $\hat{s}_2$ are the corresponding sample Sharpe ratio of OGA and B&H (Eq. (31)). Their sample difference is $\hat{d}$ (Eq. (32)). $\hat{l}^1_{0.05}$ and $\hat{l}^2_{0.05}$ are the sample luck coefficient of OGA and B&H (Eq. (39)), and $\hat{r}_{0.05}$ is the sample luck coefficient ratio between the two (Eq. (40)). The z_π, z_w, z_d and z_r are the test statistics of the mean return difference, winning probability, Sharpe ratio differential, and luck coefficient ratio, respectively. The critical value of them is 1.28 at the 10% significance level, and is 1.64 at the 5% significance level.

comparable. $\hat{p}_w$ is around 95% for both AH-7 and GH-10, and $\tilde{\pi}$ is about 25%. The value of $\hat{p}_w$ remains as high for the EUR/USD series, while it drops a little to 85% for the USD/JPY series. As to $\tilde{\pi}$, it is also about 25% for the USD/JPY series, but is greater than 40% for the EUR/USD series.

Notice that our earlier simulation result already indicated that, for some reason unknown to us, the MA component when combined with the ARCH or GARCH component may bring a negative impact to the luck coefficient. This has been already shown in the cases AH-7 and GH-10. What interests us here is that this observation repeats itself in our empirical results. The statistic z_r is statistically negative in both return series. As a result, to a large extent, what we have found from the early Monte Carlo simulations applies quite well to the real data. Hence, the GA can be useful in extracting information to develop trading strategies involving these high-frequency financial data because the underlying stochastic process, based on the Monte Carlo simulation analysis, is not a hard one for the GA.

8. CONCLUDING REMARKS

The literature on financial data mining, driven by the rapid development and applications of computational intelligence tools, are frequently clothed with a "magic house" notoriety. Unlike in mainstream econometrics, users are usually not well informed of the stochastic properties of these tools, which in turn makes it difficult to grasp the significance of the result obtained from one specific application, be it positive or negative. An essential question is how we can know that what happens in one specific application can or cannot extend to the other one. Will we still be so “lucky” next time?

By using the Monte Carlo simulation methodology, a statistical foundation for using the GA in market-timing strategies is *initiated*. This foundation would allow us to evaluate how likely the GA will work given a time series whose underlying stochastic process is known. This helps us to distinguish the *luck* from *normal expectations*. We believe that this is a major step toward lightening the black box. We emphasize that this work provides *a* statistical foundation, not *the* statistical foundation, because there are many other ways of enriching the current framework and of making it more empirically relevant.

First, different benchmarks may replace the B&H strategy. This is particularly so given a series of articles showing that simple technical analysis can beat B&H. However, since we can never run out of interesting benchmarks, the exploitation ratio $\tilde{\pi}$ introduced in this paper will always be a good reference. For example, in this paper, we can hardly have a $\tilde{\pi}$ of 30% or higher. Consequently, the 70% left there may motivate us to try more advanced version of the GA or different computational intelligence algorithms.

Second, financial time series are not just restricted to the six stochastic processes considered in this paper, but introducing new stochastic processes causes no problems for the current framework. Third, different motivations may define different evaluation criteria. The four criteria used in this paper are by no means exhausted. For example, the downside risk or VaR (Value at Risk) frequently used in current risk management can be another interesting criterion. However, again, it is straightforward to add more criteria to the current framework as long as one is not bothered by deriving the corresponding statistical tests. Fourth, the focus of this paper is to initiate a statistical foundation. Little has been addressed regarding the practical trading behavior or constraints. Things like transaction costs, non-synchronous trading, etc., can be introduced to this framework quite easily. Fifth, our framework is also not restricted to just the ordinary GA, for the general methodology applies to other machine learning tools, including the more advanced versions of the GA.

Finally, while, in this paper, we are only interested in the statistical foundation, we do not exclude the possibilities of having other foundations. As a matter of fact, we believe that a firm statistical foundation can show us where to ask the crucial questions, and that will help build a more general mathematical foundation. For example, in this paper, we have been already well motivated by the question as to why the GA performed quite poorly in the pure nonlinear stochastic processes, but performed well in the chaotic processes. Of course, this statistical finding alone may need more work before coming to its maturity. However, the point here is that theoretical questions regarding the GA's performance cannot be meaningfully answered unless we have firmly grasped their behavior in a statistical way.

NOTES

1. The interested reader can obtain more spread applications in the fields of research from Goldberg (1989).

2. A bibliographic list of financial applications of genetic algorithms and genetic programming can be found in Chen and Kuo (2002) and Chen and Kuo (2003). For a general coverage of this subject, interested readers are referred to Chen (1998a), Chen (2002) and Chen and Wang (2003). As opposed to the conventional technical analysis, the advantages of using GAs and GP are well discussed in Allen and Karjalainen (1999), and is also briefly reviewed in another paper of this special issue. (Yu et al., 2004).

3. For example, Chen (1998b) sorted out three *stochastic properties* which may impinge upon the performance of GAs in financial data mining. These are the *no-free-lunch property*, the *well-ordered property* and the *existence of temporal correlation*. Several tests of these properties are then proposed and an a *priori* evaluation of the potential of GAs can be made based on these proposed tests.

4. $\bar{\pi}^*$ is a sample average of π_i^*, which is the accumulated return earned by the omniscient trader in the *i*th ensemble of the Monte Carlo simulation.

5. Doing this enables us to apply the central limit theorem to derive the asymptotic distribution of the various test statistics mentioned in Section 4.

6. The main source of this dataset is the interbank spot prices published by Dow Jones in a multiple contributors page (the TELERATE page). This covers markets worldwide 24 hours a day. These prices are quotations of the average prices of bid and ask and not actual trading prices. Furthermore, they are irregularly sampled and therefore termed as *tick-by-tick* prices.

7. The clear cut-off pattern appearing at the first lag suggests that these series involve a MA(1) process. Later on, from more rigorous statistics, we will see that indeed it is the case.

8. The detailed description can be found in Chen and Tan (1996).

9. Once the linear signals are filtered out, any signals left in the residual series must be nonlinear. "BDS" stands for "Brock, Dechert and Scheinkman" see Brock et al. (1996).

ACKNOWLEDGMENTS

An earlier version of this paper has been presented at the 2003 International Conference on Artificial Intelligence (IC-AI03). The first author is grateful for the research support from NSC grant No. NSC 91-2415-H-004-005. We also thank the two anonymous referees for their comments and suggestions and Bruce Stewart for proof reading this paper.

REFERENCES

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

Arnold, S. F. (1990). *Mathematical statistics*. New Jersey: Prentice-Hall.

Barnett, W. A., Gallant, A. R., Hinich, M. J., Jungeilges, J. A., Kaplan, D. T., & Jensen, M. J. (1997). A single-blind controlled competition among tests for nonlinearity and chaos. Paper presented at the 1997 Far Eastern Meeting of the Econometric Society (FEMES'97), Hong Kong, July 24–26, 1997 (Session 4A).

Bauer, R. J. (1994). *Genetic algorithms and investment strategies*. Wiley.

Bollerslev, T. P. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*, 307–327.

Bollerslev, T., Chou, R. Y., & Kroner, K. F. (1992). ARCH modeling on finance: A review of the theory and empirical evidence. *Journal of Econometrics*, *52*, 5–59.

Box, G. E. P., & Jenkings, G. M. (1976). *Time series analysis: Forecasting and control*. San Fransisco: Holden-Day.

Brock, W. A., Dechert, W. D., Scheinkman, J., & LeBaron, B. (1996). A test for independence based on the correlation dimension. *Econometric Reviews*, *15*, 197–235.

Brock, W. A., Hsieh, D., & LeBaron, B. (1991). *Nonlinear dynamics, chaos and instability*. Cambridge, MA: MIT Press.

Campbell, J. Y., Lo, A. W., & MacKinlay, A. C. (1997). *The econometrics of financial markets*. Princeton University Press.

Chen, S.-H. (1998a). Evolutionary computation in financial engineering: A road map of GAs and GP. Financial Engineering News 2, No. 4. Also available from the website: http://www.fenews.com/1998/v2n4/chen.pdf.

Chen, S.-H. (1998b). Can we believe that genetic algorithms would help without actually seeing them work in financial data mining? In: L. Xu, L. W. Chan, I. King & A. Fu (Eds), *Intelligent Data Engineering and Learning: Perspectives on Financial Engineering and Data Mining* (Part I, The Foundations, pp. 81–87). Singapore: Springer-Verlag.

Chen, S.-H. (Ed.) (2002). *Genetic algorithms and genetic programming in computational finance*. Kluwer.

Chen, S.-H., & Kuo, T.-W. (2002). Evolutionary computation in economics and finance: A bibliography. In: S.-H. Chen (Ed.), *Evolutionary Computation in Economics and Finance* (pp. 419–455). Physica-Verlag.

Chen, S.-H., & Kuo, T.-W. (2003). Discovering hidden patterns with genetic programming. In: S.-H. Chen & P. P. Wang (Eds), *Computational Intelligence in Economics and Finance* (pp. 329–347). Springer-Verlag.

Chen, S.-H., & Lu, C.-F. (1999). Would evolutionary computation help for designs of artificial neural nets in financial applications? In: Proceedings of 1999 Congress on Evolutionary Computation. IEEE Press.

Chen, S.-H., & Tan, C.-W. (1996). Measuring randomness by rissanen's stochastic complexity: Applications to the financial data. In: D. L. Dowe, K. B. Korb & J. J. Oliver (Eds), *ISIS: Information, Statistics and Induction in Science* (pp. 200–211). Singapore: World Scientific.

Chen, S.-H., & Tan, C.-W. (1999). Estimating the complexity function of financial time series: An estimation based on predictive stochastic complexity. *Journal of Management and Economics*, *3*.

Chen, S.-H., & Wang, P. P. (2003). Computational intelligence in economics and finance. In: S.-H. Chen & P. P. Wang (Eds), *Computational Intelligence in Economics and Finance* (pp. 3–55). Springer-Verlag.

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, *74*, 427–431.

Drunat, J., Dufrénot, G., & Mathieu, L. (1998). Modelling burst phenomena: Bilinear and autogressive exponential models. In: C. Dunis & B. Zhou (Eds), *Nonlinear Modelling of High Frequency Financial Time Series* (pp. 201–221). Wiley.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of U.K. inflation. *Econometrica*, *50*, 987–1008.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley.

Granger, D. W. J., & Anderson, A. P. (1978). *An introduction to bilinear time series models*. Gottingen and Zurich: Vandenhoech & Ruprecht.

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

Jarque, C. M., & Bera, A. K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economic Letters*, *6*, 255–259.

Jobson, J. D., & Korkie, B. M. (1981). Performance hypothesis testing with the Sharpe and Treynor measures. *Journal of Finance*, *36*(4), 889–908.

Moody, J., & Wu, L. (1997). What is the “true price”? – state space models for high frequency FX data. Proceedings of the Conference on Computational Intelligence for Financial Engineering. IEEE Press.

Palmer, R. G., Arthur, W. B., Holland, J. H., LeBaron, B., & Taylor, P. (1994). Artificial economic life: A simple model of a stockmarket. *Physica D*, *75*, 264–274.

Refenes, A.-P. (1995). Testing strategies and metrics. In: A.-P. Refenes (Ed.), *Neural Networks in the Capital Markets* (pp. 67–76). New York: Wiley.

Roll, R. (1984). A simple implicit measure of the effective bid-ask spread in an efficient market. *Journal of Finance*, *39*, 1127–1139.

Sharpe, W. F. (1966). Mutual fund performance. *Journal of Business*, *39*(1), 119–138.

Subba-Rao, T. (1981). On the theory of bilinear time series models. *Journal of the Royal Statistical Society, Series B*, *43*, 244–255.

Subba-Rao, T., & Gabr, M. M. (1980). An introduction to bispectral analysis and bilinear time series models. In: *Lecture Notes in Statistics* (Vol. 24). New York: Springer-Verlag.

Tong, H. (1983). Threshold models in nonlinear time series analysis. In: *Lecture Notes in Statistics* (Vol. 21). Heidelberg: Springer-Verlag.

Tong, H. (1990). *Non-linear time series: A dynamical system approach*. New York: Oxford University Press.

Yu, T., Chen, S.-H., & Kuo, T.-W. (2004). A genetic programming approach to model international short-term capital flow. *Advances in Econometrics* (special issue of ‘Applications of AI in Finance & Economics’).

Zhou, B. (1996). High-frequency data and volatility in foreign-exchange rates. *Journal of Business and Economic Statistics*, *14*(1), 45–52.

APPENDIX A

A.1. Coding Trading Strategies

Based on the trading formulation (3), to encode a trading strategy, we only need to encode the CONDS with three primitive predicates, which means the following three parts:

- $\vec{a} = (a_1, a_2, a_3)$,
- $\vec{\oplus} = (\oplus_1, \oplus_2, \oplus_3)$,
- the logical combination of the three predicates Cond(r_{t-i}) $(i = 1, 2, 3)$.

To encode $\vec{a}$, we first transform the range of the variable Z $[Z_{\min}, Z_{\max}]$ into a fixed interval, say [0, 31].

$$Z^* = \frac{Z - Z_{\min}}{Z_{\max} - Z_{\min}} \times 32 \tag{A.1}$$

Then Z^* will be further transformed by Eq. (A.2).

$$Z^{**} = \begin{cases} n, & \text{if } n \le Z^* < n+1 \\ 31 & \text{if } Z^* = 32 \end{cases} \tag{A.2}$$

Since there are only 32 cutoff values, each a_i can be encoded by a 5-bit string. Hence the vector $\vec{a}$ can be encoded by a 15-bit binary string. To encode $\vec{\oplus}$, notice that each $\oplus$ has only two possibilities: $\ge$ or $<$. Therefore, a $\vec{\oplus}$ can be encoded by a 3-bit binary string (Table A.1). Finally, there are a total of totally 8 logical combinations for three predicates and they can be encoded by 3-bit strings (Table A.2).

In sum, a CONDS can be encoded by a 21-bit string (3 for logical combinations, 3 for inequalities, and 15 for the three thresholds). Therefore, each trading strategy can be represented by a 21-bit string.

Table A.1. Binary Codes for Inequality Relation.

Code	$\oplus_1$	$\oplus_2$	$\oplus_3$
0(000)	$\ge$	$\ge$	$\ge$
1(001)	$<$	$\ge$	$\ge$
2(010)	$\ge$	$<$	$\ge$
3(011)	$\ge$	$\ge$	$<$
4(100)	$<$	$<$	$\ge$
5(101)	$<$	$\ge$	$<$
6(110)	$\ge$	$<$	$<$
7(111)	$<$	$<$	$<$

Table A.2. Binary Codes for Logical Combinations.

Logic Code	Logical Combination of Predicates
0(000)	Cond 1 OR (Cond 2 AND Cond 3)
1(001)	Cond 1 AND (Cond 2 OR Cond 3)
2(010)	(Cond 1 OR Cond 2) AND Cond 3
3(011)	(Cond 1 AND Cond 2) OR Cond 3
4(100)	(Cond 1 OR Cond 3) AND Cond 2
5(101)	(Cond 1 AND Cond 3) OR Cond 2
6(110)	Cond 1 OR Cond 2 OR Cond 3
7(111)	Cond 1 AND Cond 2 AND Cond 3

A.2. Ordinary Genetic Algorithms

The GA described below is a very basic version of a GA, and is referred to as the ordinary genetic algorithm (OGA). More precisely, it is very similar to the GA employed in Bauer (1994).

- The genetic algorithm maintains *a population of individuals*,

$$P_i = \{g_1^i, \ldots, g_n^i\} \tag{A.3}$$

  for iteration i, where n is *population size*. Usually, n is treated as fixed during the whole evolution. Clearly, $P_i \subset G$.
- *Evaluation step:* Each individual g_j^i represents a trading strategy at the ith iteration (population). It can be implemented with the *historical data* r_{t-1}, r_{t-2}, and r_{t-3} by means of Eq. (2). A specific example is given in Eq. (3). Each trading strategy g_j^i is evaluated by a *fitness* function, say Eq. (6).
- *Selection step:* Then, a new generation of population (iteration $i+1$) is formed by randomly selecting individuals from P_i in accordance with a *selection scheme*, which, in this paper, is the *roulette-wheel selection scheme*.

$$M_i = P_s(P_i) = (s_1(P_i), s_2(P_i), \ldots, s_n(P_i)) \tag{A.4}$$

  where

$$s_k : \left\{\binom{G}{n}\right\} \to G, \tag{A.5}$$

  $k = 1, 2, \ldots, n$, and $\left\{\binom{G}{n}\right\}$ is the set of all populations whose population size is n. The set M_i is also called the *mating pool*.
- *Alteration step:* Some members of the new population undergo transformations by means of *genetic operators* to form new solutions.

- *Crossover:* We use *two-point crossover* c_k, which create new individuals by combining parts from two individuals.

$$O_i = P_c(M_i) = (c_1(M_i), c_2(M_i), \ldots, c_{n/2}(M_i)) \tag{A.6}$$

where

$$c_k\colon \left\{ \binom{G}{n} \right\} \to G \times G, \tag{A.7}$$

$k = 1, 2, \ldots, n/2$. O_i is known as the set of *offspring* in the GA.

- *Mutation:* We use *bit-by-bit mutation* m_k, which creates new individuals by flipping, with a small probability, each bit of each individual of O_i.

$$P_{i+1} = P_m(O_i) = (m_1(O_i), m_2(O_i), \ldots, m_n(O_i)) \tag{A.8}$$

where

$$m_k\colon \left\{ \binom{G}{n} \right\} \to G \tag{A.9}$$

$k = 1, 2, \ldots, n$.

- After the evaluation, selection and alteration steps, the new population P_{i+1} is generated. Then we proceed with the three steps with P_{i+1}, and the loop goes over and over again until a termination criterion is met. The control parameters employed to run the OGA are given in Table A.3.

Table A.3. Control Parameters of OGA.

Number of generations	100
Population size (n)	100
Selection scheme	Roulette-wheel
Fitness function	Accumulated returns
Elitist strategy	Yes
Rank min	0.75
Crossover style	Two-Point
Crossover rate	0.6
Mutation rate	0.001