Title	Evolutionary learning of technical trading rules without data-mining bias
Author(s)	Agapitos, Alexandros; O'Neill, Michael; Brabazon, Anthony
Publication date	2010-09
Publication information	Schaefer, R. ...et al. (eds.). Parallel Problem Solving from Nature – PPSN XI 11th International Conference, Kraków, Poland, September 11-15, 2010 : proceedings, part I
Conference details	11th International Conference on Parallel Problem Solving from Nature (PPSN 2010), Krakow, Poland, September 11-15, 2010
Publisher	Springer
Link to online version	http://springerlink.com/content/3443w87r12777233/
Item record/more information	http://hdl.handle.net/10197/2735
Publisher's statement	The final publication is available at springerlink.com.
Publisher's version (DOI)	http://dx.doi.org/10.1007/978-3-642-15844-5_30

Evolutionary Learning of Technical Trading Rules without Data-mining Bias

Alexandros Agapitos, Michael O'Neill, Anthony Brabazon

Financial Mathematics and Computation Research Cluster
Natural Computing Research and Applications Group
Complex and Adaptive Systems Laboratory
University College Dublin, Ireland
alexandros.agapitos@ucd.ie, m.oneill@ucd.ie, anthony.brabazon@ucd.ie

Abstract. In this paper we investigate the profitability of evolved technical trading rules when controlling for data-mining bias. For the first time in the evolutionary computation literature, a comprehensive test for a rule's statistical significance using Hansen's Superior Predictive Ability is explicitly taken into account in the fitness function, and multi-objective evolutionary optimisation is employed to drive the search towards individual rules with better generalisation abilities. Empirical results on a spot foreign-exchange market index suggest that increased out-of-sample performance can be obtained after accounting for data-mining bias effects in a multi-objective fitness function, as compared to a single-criterion fitness measure that considers solely the average return.

1 Introduction

The goal of objective technical analysis is the discovery of rules that will be profitable in the future. The research method is back-testing, which produces an observable measure of performance. On the basis of this test statistic an inference is made about a population parameter, the rule's expected performance out of sample. In evolutionary rule data-mining, many rules are back-tested in each generation, and the selection of rules that form the basis of the subsequent rules to-be-sampled is stochastically based on their observed performance. This refinement cycle eventually results in a rule with the best performance being designated as the output of the run. That is to say, this form of data mining involves a performance competition that leads to a winning rule being picked. The problem is that the winning rule's observed performance that allowed it to be picked over all other rules systematically overstates how well the rule is likely to perform in the future. This systematic error is the *data-mining bias*.

Out-of-sample rule performance deterioration is a well-known problem [1, 2]. A dominant explanation of the out-of-sample performance break-down is data-mining bias. This has two constituents: (1) randomness, which is a relatively large component of observed performance. It is reasoned that a portion of a rule's back-tested performance was merely luck - a coincidental correspondence between

the rule's signals and the market's non-recurring noise. Because this random component is a non-recurring phenomenon that will manifest differently in each sample of data, the rule's expected performance always falls below observed performance on the training data. (2) the logic of data mining, in which the best-performing rule is selected after a repertoire of candidate rules have been evaluated (the observed average return of the best-performing rule is a positively-biased statistic [1]).

Previous research on the induction of data-driven models by means of evolutionary computation accounted for the problem of data-mining bias using out-of-sample testing [3, 4], methods to restrict the model complexity [5], and ensemble learning [6]. The most prevalent of these, *out-of-sample* testing, is based on the valid notion that the performance of a data-mined rule, on out-of-sample data, provides an unbiased estimate of the rule's expected performance. However, out-of-sample testing suffers from several deficiencies. First and foremost, the unused status of the data reserved for out-of-sample testing has a short lifespan. Once it has been used, it is no longer able to provide unbiased estimates of rule performance. Secondly, it eliminates certain portions of the data from mining operations, thus, it reduces the amount of data available to detect patterns. When noise is high and information is low, the bigger the amount of training data the better the chance of mining something useful. Third, the decision about how to apportion the data between in- and out-of-sample subsets is arbitrary, and hence lacks desired objectivity.

In this paper we propose a new approach to account for the data-mining bias inherent in an iterative modelling technique, such as grammar-based Genetic Programming (GP) [7], where a single dataset is used more than once in the model construction process. Our approach is based on multi-objective evolutionary learning of technical trading rules, where the optimisation criterion is a weighted amalgamation of the rule's observed daily return and the statistical significance (p-value) of this test statistic. This permits data-mining bias to be undertaken with some degree of confidence, and incorporated in the objective function that drives the evolutionary search. Our expectation is that such a fitness measure will create an evolutionary pressure towards parts of the search space that contain individuals with true predictive abilities, and thus lead to better generalisation to unseen data. The rest of the paper is organised as follows. First we introduce two dominant methods for testing the null hypothesis that the best rule encountered during data mining has no predictive superiority over a benchmark model devoid of predictive power. The grammar-based GP system is then described, giving details of the grammar employed, the technical indicators serving as the building blocks for constructing trading rules, the evolutionary algorithm, the trading methodology, and the multi-objective fitness function. The description of the experimental approach comes next, followed by an analysis of the experimental results. Finally, a concluding section summarises our findings, and sketches future directions of this research.

2 Data-mining-adjusted Statistical Hypothesis Tests

In the context of evaluating the data mining of technical analysis rules, it is conceivable that by repeatedly examining different trading rules against the same dataset, some rules would appear to be profitable simply due to chance (data mining/data snooping bias). White's (2000) Reality Check (WRC) test [8] and Hansen's (2005) Superior Predictive Ability (SPA) test [9] provide comprehensive tests across all trading rules considered, and directly quantify the effect of data-mining bias by testing the null hypothesis that the performance of the best trading rule is no better than the performance of the benchmark set to a trading model totally devoid of predictive power according to a performance statistic (i.e. observed average return in a back-test). The best rule is identified by applying the performance measure to the full universe of trading rules, and a desired p-value is obtained by comparing the best rule's sample statistic to approximations of the sampling distribution of the test statistic. In both methods bootstrapping is used to approximate the sampling distribution of the test statistic.

Given M models (trading rules), let $\varphi_{k,t}$ ($k = 1, 2, \ldots, M$ and $t = 1, 2, \ldots, N$) denote their performance measures relative to the benchmark model over time t. The null hypothesis is that there does not exist a superior rule in the universe of M rules (joint test).

$$H_0 : \max_{k=1,\ldots,M} \varphi_k \leq 0. \tag{1}$$

Rejecting H_0 implies that there exist at least one rule that outperforms the benchmark. Setting the performance measure to the rule's mean return obtained by back-testing it in a historical sample of data, the benchmark becomes a rule that has an expected return of zero or less, thus, $\varphi_k = E(f_k)$, where f_k is the return of the k-th trading rule. It is then natural to base the test statistic of hypothesis test to the maximum of the normalised average of $f_{k,t}$:

$$\overline{V}_n = \max_{k=1,\ldots,M} \sqrt{n}\bar{f}_k \tag{2}$$

where $\bar{f}_k = \frac{1}{n}\sum_{t=1}^{n} f_{k.t}$, with $f_{k,t}$ the t-th observation of f_k.

White suggested using the stationary bootstrap method [10] to approximate the p-values of $\overline{V}_n$. In general, a bootstrap method derives the sampling distribution of a test statistic by resampling with replacement from an original sample. The reason that White decided to use a block-bootstrap method is to maintain some of the statistical properties of the bootstrapped time-series such as heteroskedasticity [8]. To describe the bootstrap algorithm, let X_n be a strictly stationary time-series. Suppose μ is a parameter of the whole joint distribution of the sequence (i.e mean). Given data $X_1, \ldots, X_N$ the goal is to make inferences about μ. Suppose $B_{i,b} = \{X_i, X_{i+1}, \ldots X_{i+b-1}\}$ be a block of b observations starting from X_i. In the case where $j > N$, X_j is defined to be X_i, where $i = (j \bmod N)$ and $X_0 = X_N$. Let p be a fixed number in $\{0, \ldots, 1\}$. independent of $X_1, \ldots, X_N$, let $L_1, L_2, \ldots$ be a sequence of independent and identically

distributed random variables having the geometric distribution, so that the probability of the event $L_i = m$ is $(1-p)^{(m-1)}p$ for $m = 1, 2, \ldots$. independent of X_i and the L_i, let $I_1, I_2, \ldots$ be a sequence of independent and identically distributed variables which have the discrete uniform distribution on $\{1, \ldots, N\}$. A pseudo-time-series $X_1^*, \ldots, X_N^*$ is generated in the following way: Sample a sequence of blocks of random length by the prescription $B_{I_1,L_1}, B_{I_2,L_2}, \ldots$. The first L_1 observations in the pseudo-time-series $X_1^*, \ldots, X_N^*$ are determined by the first block B_{I_1,L_1} of observations $X_{I_1}, \ldots, X_{I_1+L_1-1}$, the next L_2 observations are the observations in the second sampled block B_{I_2,L_2} , namely $X_{I_2}, \ldots, X_{I_2+L_2-1}$. This process is stopped once N observations in the pseudo-time-series have been generated.

Now, back to White's Reality Check, let f_k^*(b) denote the b-th bootstrapped sample of f_k and $\bar{f}_k^*(b) = \frac{1}{n}\sum_{t=1}^{n} f_{k.t}^*(b)$ its sample average. A bootstrapped sampling distribution $\overline{V}_n^*$ is obtained with the realisations:

$$\overline{V}_n^*(b) = \max_{k=1,\ldots,M} \sqrt{n}(\bar{f}_k^*(b) - \bar{f}_k), b = 1, \ldots, B. \tag{3}$$

The WRC p-value is obtained by comparing V_n the quantiles of the sampling distribution of $\overline{V}_n^*$. The null hypothesis is rejected whenever p-value is less than a given significance level.

Hansen pointed out two potential inefficiencies with White's Reality Check. First, the average returns $\bar{f}_k$ are not standardised. Second, despite that H_0 is composite, the sampling distribution of WRC is based on the "least favourable configuration" (the configuration that is least favourable to the alternative), that is all of the back-tested rules have expected returns of zero. Therefore, the WRD test may lose power dramatically when poor rules with very negative $E(f_k)$ are included in the test. The proposed SPA test is based on studentised returns:

$$\overline{V}_n = max\left(\max_{k=1,\ldots,M} \frac{\sqrt{n}\bar{f}_k}{\sigma_k}, 0\right) \tag{4}$$

where σ_k is a consistent estimator of the standard deviation of $\sqrt{n}\bar{f}_k$.

To avoid using the least favourable configuration and increase the power of the test, Hansen suggested a different way to bootstrap the distribution of $\overline{V}_n$. For the k-th rule, let $\bar{Z}_n^*(b)$ denote the sample average of the b-th bootstrapped sample of the centered returns:

$$Z_{k,t}^*(b) = f_{k,t}^*(b) - \bar{f}_k \mathbf{1}_{\{\bar{f}_k \geq -A_k\}} \tag{5}$$

where $\mathbf{1}$(G) denotes the indicator function of the event G, and $A_k = -\frac{\sigma_k}{4n^{1/4}}$. The p-value is obtained by the bootstrapped sampling distribution whose realisations are:

$$\overline{V}_n^*(b) = max\left(\max_{k=1,\ldots,M} \frac{\sqrt{n}\bar{Z}_k^*(b)}{\sigma_k}, 0\right), b = 1, \ldots, B. \tag{6}$$

3 Grammar-based Genetic Programming for Rule Induction

We employ a grammar-based GP system to evolve technical trading rules. The method used is outlined in the following sections.

3.1 Grammar

A context-free grammar is employed to type the language used for program representation. It is presented below:

```
<prog> ::= <if>
<if> ::= <predicate> <expr> <expr>
<expr> ::= <if> | <signal>
<signal> ::= golong | goshort
<predicate> ::= <ti> <op> <constant> | <ti> <op> <ti>
<op> ::= < | >
<ti> ::= MACD | RSI | SM | ADX
<constant> ::= -0.5 | -0.49 | ... | 0.49 | 0.5 | 1.0 | 2.0 ... | 100.0
```

Using the grammar above, a technical trading rule is represented as a disjunction of conjunctions of constraints on the values of technical indicators, taking the classical form of *oblique decision tree* learning for approximating discrete-valued target functions, however, here we also allow comparisons between technical indicators. The space of technical trading rules is formed using the following indicators: **Relative Strength Index** (RSI), **Moving Average Convergence Divergence** (MACD), **Stochastics Momentum** (SM), **Average Directional Movement Index** (ADX) [11]. MACD will usually oscillate around zero with unknown upper and lower bounds, whereas RSI, SM, and ADX oscillate in the range of $\{0, \ldots, 100\}$. The constants that form part of the predicates that test a real-valued technical indicator against some value come from the union of two sets, $\{-0.5, -0.49, \ldots, 0.49, 0.5\}$ and $\{1.0, \ldots 100.0\}$. The reason of choice of this representation is to enhance human understandability of the conditions that trigger certain trading signals, and treat the outcome of the evolutionary process as a decision-support system rather than merely as a black-box method for trading.

3.2 Trading Methodology

Each evolved rule outputs two values, 1 and -1, interpreted a long and short position respectively. The average return of a rule is generated as follows. Let r_t be the daily return of the index at time t, calculated using $(v_t - v_{t-1})/v_{t-1}$, where v_t and v_{t-1} are the values of the time-series at time t and $t-1$ respectively. Also, let s_{t-1} be the trading signal generated by the rule at time $t-1$. Then $d_t = s_{t-1}r_t$ is the realised return at time t. Using a back-test period, an average of d_t can be induced. We are not considering trading, slippage or interest costs.

3.3 Evolutionary Algorithm

For our evolutionary algorithm, we used a panmictic, generational, elitist genetic algorithm. The algorithm uses tournament selection with a tournament size of 7. Evolution proceeds for 50 generations, and the population size is set to $1,000$ individuals. Ramped-half-and-half tree creation with a maximum depth of 5 is used to perform a random sampling of rules during run initialisation. Throughout evolution, expression-trees are allowed to grow up to depth of 10. The evolutionary search employs a mutation-based variation scheme, where subtree mutation is combined with point-mutation; a probability governing the application of each, set to 0.6 in favour of subtree mutation. Neither recombination, nor reproduction were used.

3.4 Fitness Function

In the case of a single-objective fitness function, this takes the form of average daily return generated by the rule's trading signals over a back-test period specified by the training set. On the other hand, the multi-objective fitness function is defined as a weighted sum of average daily return and the p-value of this statistic that is generated from SPA test. For the weighting scheme to be effective the two objectives need to be similarly scaled. In the case of p-value, this is naturally defined within the $\{0, \ldots, 1\}$ interval. We employed a simple normalisation technique to make the average daily return of each individual in a population fall into the same $\{0, \ldots, 1\}$ interval, by taking into account the minimum and maximum average daily returns produced by individuals in each population.

3.5 Adapting the Data-mining bias Tests to a Population of Rules

SPA tests a composite null hypothesis whose test statistic is defined as the *maximum standardised mean of N means*, where N is the number of rules in our universe. It is therefore analogous to treat each evolving population as a universe of N individual rules. Under this formulation the following sequence of steps are involved in calculating the p-values:

1. Calculate the standardised mean daily return for each rule in a population on de-trended daily returns of a back-test period (note - this is not to be confused with de-trending of the time-series of an index). De-trended daily returns have an average daily return of zero, thus the expected return of a rule with no predictive power will be zero if its returns are computed from de-trended data. De-trending is a simple operation, where the average daily return over a period is subtracted from each daily return. Sort the population in descending order based on standardised mean daily return.
2. Using the the bootstrap method described in Section 2 create a bootstrapped sample of trading dates (days).
3. Using the dates obtained in the bootstrapped sample, a pseudo-track record based on the actual daily returns associated with these dates is created for each rule in the population.

4. For every rule in the population, each daily return in the pseudo-track record is adjusted according to Equation 5 (Section 2). The adjusted pseudo-track record of returns is then averaged and standardised.
5. The larger of these standardised values and zero is designated as the first value to form the sampling distribution of the test statistic of Equation 4 (Section 2).
6. Steps 2 to 5 are repeated many times (i.e B times). In this way, the sampling distribution of the test statistic is approximated from these B values.
7. The p-value of the first rule (in the sorted list of rules of step 1) is calculated as a fraction of B values that exceed the standardised mean daily return of the tested rule.
8. Remove the tested rule from the sorted list, and repeat steps 2 to 7 using the remaining rules.

4 Experimental Approach

This empirical study aims to reveal whether there is an advantage accruing from using the statistical significance of average daily returns acquired in a back-test as an additional objective, in order to drive the evolutionary search towards better-generalising technical trading rules (out-of-sample testing). For this, we are considering an exhaustive set of combinations (with a step of 0.1) for coefficients that weight the average daily return and the p-value in the multi-objective fitness function, in order to manually set the trade-off between the two. An obvious benchmark to compare against our methodology is the single-objective evolution of technical rules, using solely the average daily return as the fitness function.

This study uses daily closing foreign exchange rate of USD/GBP for the period of 01/01/1990 to 31/03/2010. The first $4,000$ trading days are used as the training-set, whereas the remaining $1,229$ as the test-set. The parameters of the technical indicators are set as follows: $n = 21$ for RSI; $n = 28$, $r = 30$, $s = 2$ for SM; $n = 14$ for ADX. For the stationary bootstrap procedure we set the number of bootstrapped samples to $2,000$, and the probability of success in each trial to 0.9 in order to generate the probability mass function of the geometric distribution. We perform 50 independent runs for each experimental setup allowing either for a multi-objective fitness function (with variable weighting coefficients), or a single-objective fitness function.

5 Results

A summary of the experimental results is depicted in Table 1. A statistically significant difference (unpaired t-test, $p < 0.05$, degrees of freedom $df = 98$) is found between the average daily return of single-objective fitness function and multi-objective one (weighting coefficients of 0.2, 0.8) during out-of-sample back-testing, suggesting a better generalisation ability of the trading rules that were

encouraged to take account of the data-mining bias during the induction process. Results on the out-of-sample data also show that the best evolved technical rule, using the multi-criterion fitness, outperforms its single-objective-evolution counterpart, obtaining an annualised return of 13.75% (average daily return of 0.055) as opposed to 8% (average daily return of 0.032). In addition, it is interesting to note that most combinations of coefficients for weighting the impact of different objectives yielded similar generalisation performance, suggesting that the (0.2, 0.8) interplay in favour of the p-value creates the tradeoff required to drive the evolutionary search towards the discovery of better-generalising trading rules. Nevertheless, optimising in favour of the p-value evidently leads to inferior in-sample performance; a statistically significance difference (unpaired t-test, $p < 0.0001$, $df = 98$) is found in the average daily return between single- and multi-objective fitness functions (0.2, 0.8) for the training period. This is intuitive, indicative of the closer fit to the training data that is obtained by a model that is evolved unconstrained for sole profitability maximisation.

Table 1. Performance summary. Average daily return has been abbreviated to AR. Means are based on best-of-run individuals from 50 evolutionary runs. The case where the weighting coefficients for AR and p-value are set to 1.0 and 0.0 respectively refers to the single-objective fitness function. Std. deviation in parentheses for mean. Best out-of-sample performance indicated in bold.

AR coeff.	p-value coeff.	Mean Train AR	Min Test AR	Mean Test AR	Max Test AR
1.0	0.0	0.037 (0.002)	−0.027	0.007 (0.013)	0.032
0.9	0.1	0.038 (0.003)	−0.014	0.008 (0.014)	0.042
0.8	0.2	0.037 (0.003)	−0.021	0.009 (0.015)	0.041
0.7	0.3	0.038 (0.003)	−0.015	0.006 (0.013)	0.030
0.6	0.4	0.037 (0.002)	−0.012	0.007 (0.014)	0.045
0.5	0.5	0.037 (0.003)	−0.014	0.007 (0.012)	0.026
0.4	0.6	0.037 (0.003)	−0.015	0.007 (0.015)	0.040
0.3	0.7	0.038 (0.003)	−0.022	0.006 (0.014)	0.031
0.2	0.8	0.033 (0.002)	−0.020	**0.014** (0.018)	**0.055**
0.1	0.9	0.038 (0.003)	−0.013	0.007 (0.013)	0.028

The graphs depicted in Figure 1 show the evolution of best-of-generation average daily return for the cases of single- and multi-objective (0.2, 0.8) fitness functions, for both in- and out-of-sample data-sets. Figures 1(a), 1(b) show that single-objective evolution learns faster, and the trading rules fit more closely to the training data, achieving bigger daily returns compared to the multi-objective case. Figure 1(c) illustrates the learning curve for the out-of-sample data, depicting an inherent difficulty in the generalisation ability of the best-of-generation trading rule; a phenomenon that has been widely documented in previous studies. An interesting result is depicted in Figure 1(d), where a good generalisation to unseen data is observed by relatively random rules in the initial generations, and then followed by a rapid decrease in performance up to approx. generation 7, before learning starts again. This learning behaviour is explained by the nature of the time-series in the training and testing data-sets, which allows relatively

random rules to better model the time-series fluctuations in the testing-set. However, as learning is dictated on the information provided by the training-set, the adaptive expression-tree structures are gradually fitting to in-sample data, drifting away from the genotypes that exhibited an initial out-of-sample superiority.

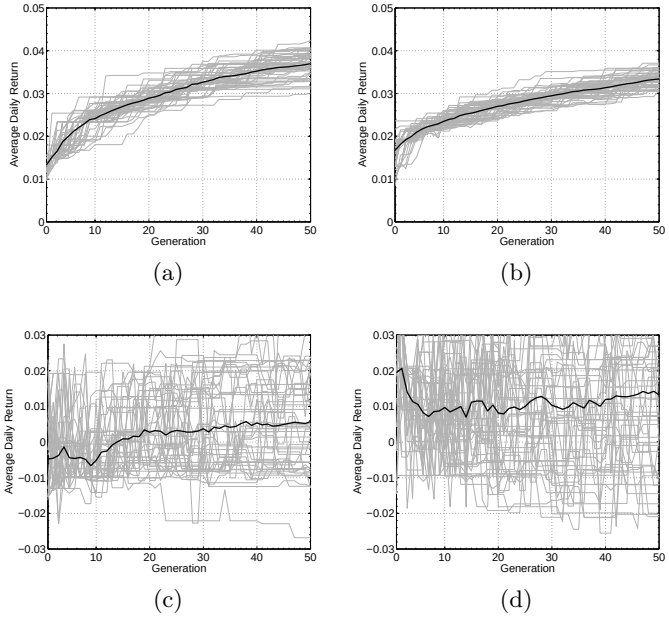

Fig. 1. Evolution of best-of-generation average daily return. (a) and (b) show in-sample evolution using sets of weighting coefficients represented by the tuples (1.0, 0.0) and (0.2, 0.8) respectively; (c) and (d) show the out-of-sample evolution for the same weighting coefficient setups. Each graph presents 50 evolutionary runs; average in bold.

6 Conclusion

Multiple studies on the evolutionary search for profitable technical trading rules have been conducted in the past, suggesting that this form of rule induction technique does have merit. However, the effects of data-mining bias in the generalisation ability of the trading rules have not been accounted for. We proposed a method to encourage the evolution of technical rules with statistically significant returns during a back-testing training period, in an expectation to increase their out-of-sample performance. This relies on a multi-criterion fitness function that in addition to a measure of profitability, takes into account Hansen's Superior Predictive Ability test, which can directly quantify the effect of data-mining bias, by testing the performance of the best mined rule in

the context of the full universe of technical trading rules. Initial experiments, using an index from a foreign-exchange market, are encouraging, resulting in human-understandable trading rules with better generalisation to unseen data after accounting for data-mining bias. Future work includes the application of this methodology to a wider range of market indices in order to corroborate its breadth of efficiency, and the employmdent of Pareto-based evolutionary optimisation. The grammar employed in our experiments was deliberately kept as simple as possible in an attempt to minimise exogenous factors affecting performance, in order to objectively assess the potential of the newly introduced method. Subsequent versions will rely on dynamically setting the constraints on technical indicators, as well as a richer repertoire of such primitive constructs.

Acknowledgement

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant Number 08/SRC/FM1389.

References

1. David Aronson, *Evidence-Based Technical Analysis*, John Wiley and Sons, Inc., 2007.
2. Robert Pardo, *Design, testing and optimisation of trading systems*, John Wiley and Sons, Inc., 1992.
3. Michael O'Neill, Anthony Brabazon, Conor Ryan, and J. J. Collins, "Evolving market index trading rules using grammatical evolution", in *Applications of Evolutionary Computing*, Lake Como, Italy, 18-19 April 2001, vol. 2037 of *LNCS*, pp. 343–352, Springer-Verlag.
4. James D Thomas and Katia Sycara, "The importance of simplicity and validation in genetic programming for data mining in financial data", in *Data Mining with Evolutionary Algorithms: Research Directions*, Alex Alves Freitas, Ed., Orlando, Florida, 18 July 1999, pp. 7–11, AAAI Press, Technical Report WS-99-06.
5. Hitoshi Iba, Hugo De Garis, and Taisuke Sato, "Genetic programming using a minimum description length principle", in *Advances in Genetic Programming*. 1994, pp. 265–284, MIT Press.
6. Leo Breiman and Leo Breiman, "Bagging predictors", in *Machine Learning*, 1996, pp. 123–140.
7. Michael O'Neill and Conor Ryan, *Grammatical Evolution: Evolutionary Automatic Programming in a Arbitrary Language*, vol. 4 of *Genetic programming*, Kluwer Academic Publishers, 2003.
8. Halbert White, "A reality check for data snooping", *Econometrica*, vol. 68, no. 5, pp. 1097–1126, September 2000.
9. Peter Reinhard Hansen, "A test for superior predictive ability", *Journal of Business & Economic Statistics*, vol. 23, pp. 365–380, October 2005.
10. Joseph Romano Dimitris Politis, "The stationary bootstrap", *Journal of American Statistical Association*, vol. 89, no. 428, pp. 1303–1313, 1994.
11. Perry Kaufman, *New Trading Systems and Methods*, Willey, 4th edition edition, 2005.