ROBUST TRADING RULE SELECTION AND FORECASTING ACCURACY*

Harald Schmidbauer[†]/ Angi Rösch[‡]/ Tolga Sezer[§]/ Vehbi Sinan Tunalıoğlu[¶]

Abstract

Trading rules performing well on a given data set seldom lead to promising out-of-sample results, a problem which is a consequence of the in-sample data snooping bias. Efforts to justify the selection of trading rules by assessing the out-of-sample performance will not really remedy this predicament either, because they are prone to be trapped in what is known as the out-of-sample data-snooping bias.

Our approach to curb the data-snooping bias consists of constructing a framework for trading rule selection using a-priori robustness strategies, where robustness is gauged on the basis of time-series bootstrap and multi-objective criteria. This approach focuses thus on building robustness into the process of trading rule selection at an early stage, rather than on an ex-post assessment of trading rule fitness. Intra-day FX market data constitute the empirical basis of our investigations. Trading rules are selected from a wide universe created by genetic algorithms. We show evidence of the benefit of this approach in terms of indirect forecasting accuracy when investing in FX markets.

Key words: Trading rule selection; genetic algorithm; data-snooping bias; a-priori robustness; time-series bootstrap; intra-day FX markets; efficient market hypothesis

1 Introduction

Technical trading rule performance in financial markets have been a widely discussed and investigated research field in literature. One of the main motivations for researchers to engage in this field has been to determine whether technical trading rules can be employed to provide superior financial performance. This motivation is of course directly related to the "Efficient Market Hypothesis" (EMH). Evidence that these trading rules provide significant profit opportunity when trading certain assets means that we succeed to reject the null hypothesis of efficiency of this particular market.

In one of the earlier studies — and perhaps the most extensive using 90 years of daily stock prices — Brock, Lakonishok, and LeBaron [2] find that a considerable number of technical trading rules applied to the Dow Jones Industrial Average (DJIA) provide significant financial performance. If these results were uncontested, this would have implied — against many researchers' belief — that the EMH even in its weak form did not hold.

*This research project was presented on the 31th International Symposium on Forecasting ISF2011, held in Prague, Czech Republic, June 26-29, 2011. The paper is published in: The International Institute of Forecasters (ed.), *Proceedings of the 31th International Symposium on Forecasting ISF2011*, Prague, Czech Republic, June 26-29, 2011. ISSN: 1997-4124.

[†]Bilgi University, Istanbul, Turkey, & Ideal Analytix, Singapore; e-mail: harald@hs-stat.com

[‡]FOM University of Applied Sciences, Munich, Germany, & Ideal Analytix, Singapore; e-mail: angi@angi-stat.com

[§]Ideal Analytix, Singapore; e-mail: tolga.sezer@ideal.sg

[¶]Ideal Analytix, Singapore; e-mail: sinan.tunalioglu@ideal.sg

However, an important drawback and rarely directly addressed issue at that stage of the research was the problem of data-snooping. To quote White [9]:

> "*Data-snooping occurs when a given set of data is used more than once for purposes of inference or model selection. When such data reuse occurs, there is always the possibility that any satisfactory results obtained may simply be due to chance rather than to any merit inherent in the method yielding the results.*"

For example, validation of in-sample results by means of out-of-sample test were also contested due to what is termed the "out-of-sample data-snooping". In fact the problem of data-snooping was an ubiquitous problem called by many researchers to be remedied and controlled for (Merton [4]). As acknowledged by Brock, Lakonishok, and LeBaron [2], they were not able to account fully for the data-snooping issue.

The second line of research involved the extension of the earlier research on technical trading rules by applying new procedures and methods to take account of the effect of data-snooping more accurately, thereby also catering to the demands raised in literature.

Sullivan, Timmerman, and White [8] use "White's Reality Check" bootstrap methodology to correct for the effects of data-snooping. In their view, this method would make it possible to evaluate whether the performance of technical trading rules is a result of superior economic content, or simply due to luck. They conclude that the superior performance of the best technical trading rule identified by Brock, Lakonishok, and LeBaron [2] is not repeated in the out-of-sample experiment covering the 10-year period 1987–1996. They go on to provide possible explanations of their findings. Qi and Wu [6], for instance, find evidence of profitability and significance even after applying White's Reality Check bootstrap methodology.

However, besides the inferences on the influence and effects of the data-snooping bias, Sullivan, Timmermann, and White's research represents a methodological novelty which is of particular value for investors who are searching for successful investment strategies. Consequently, robustness has emerged as an important criterion to gauge the validity of the results and to mitigate the data-snooping biases for both, researchers as well as investors.

In this paper, we construct a framework for technical trading rule selection using a-priori robustness strategies, where robustness is gauged on the basis of time-series bootstrap and multi-objective criteria. This approach focuses thus on building robustness into the process of trading rule selection at an early stage, rather than on an ex-post assessment of trading rule fitness.

The premise leading to this approach is that if data-snooping is a ubiquitous problem which needs to be accounted for, then the end result of a selection process which survives perturbations of the original data series will be robust and, hence, the out-of-sample test will be less spurious. The crucial assumption herein is of course that the permutation method chosen remedies data-snooping biases effectively (while preserving the predictable component of the original dataset).

The fundamental problem of any permutations in the context of economic time series is that most of them destroy the shape and structure of the original dataset by detrending or differencing in order to achieve stationarity. Stationary time series are integrated of order zero, I(0). However, irreversibility is an important property of most economic time series, making the assumption of a zero memory I(0) process quite unrealistic. A bootstrap method which has a particular practical appeal is the "Maximum Entropy Bootstrap" method by Vinod and López-de-Lacalle [3]. The permutations do not need to undergo all the shape-destroying transformations and retain the time-dependence structure of ACF and PACF.

A further implicit and often neglected assumption which one ought to be skeptic about, and which lends this research field its actual relevance, is that the agents are adequately represented by the individual technical trading rules. In real life, however, the actual agents are traders who

are not necessarily bound to using single rules at each point in time. For instance, trading rule strategies may consist of inter-temporal variations of technical trading rules or rule sets and may be further complicated by combinations depending on the dynamics of the markets. Although it is very difficult to fully represent an agent's reaction function, a better proxy is needed than the individual rules. Here, we make use of the advances in computational technology and deploy grammar-based genetic programming tools.

Grammatical Evolution (GE) is an "evolutionary automatic programming methodology" introduced by O'Neill and Ryan [5]. It can be used to evolve systems with large numbers of rule sets. The rule sets are formulated in terms of functional expressions which can be as general as the researcher determines. Ultimately, these expressions produce a mapping between a universe of provided input data vectors and the output vector(s) (Brabazon, and O'Neill [1]). A particular strength of the methodology in our context is that the functional form need not be specified a priori. The main benefit of suchlike context-independence emerges when the researcher has a theoretical or intuitive idea of the nature of the explanatory variables, but has no means to determine the functional relationship between the explanatory and the dependent variable(s). In a pioneering work, Brabazon, and O'Neill [1] test the grammatical evolution approach to evolve trading rules for FX markets. They use daily data from 1992 to 1997 and find superior out-of-sample returns. However, the caveat of data-snooping biases is only tackled by extending the out-of-sample data and by dividing them into two hold-out periods. This may reduce the data-snooping danger somewhat, but does not exclude the eventuality of it.

We aim to extend and enrich this field of research by introducing a new procedure which combines advances in bootstrap permutation methodology, the computational techniques applied by Brabazon and O'Neill and our own a-priori robustness criterion in an attempt to produce robust, non-spurious trading systems based on technical trading rules.

This paper is organized as follows. Section 2 sets the stage for our investigation by introducing the basic concept of algorithmic trading. Our idea of a-priori robustness and how it can be put into effect is the topic of Section 3. The experimental design of our investigation is explained in Section 4. Empirical results when investing in the intra-day EUR/USD market are given and discussed in Section 5. Finally, Section 6 summarizes and draws some conclusions.

2 Algorithmic Trading

In our context, a trading rule is a buy (or sell) signal derived from evidence concerning past process characteristics. For purposes of selectivity and discriminating power, we consider trading rules giving either buy (sell) or don't buy (don't sell, respectively) signals, rather than giving either a buy or a sell signal. A variety of trading rules, which were used in the present project with different alternative parameter settings, is implemented in the R package "TTR".[1] Signals are generated once every five minutes to derive a decision how to proceed. They require the input of four price series, summarizing events in a five-minute interval: "open" (the opening value), "high" and "low" (the maximum and minimum values), and "last" (the closing value).

Different trading rules may yield contradictory trading signals even when based on the same empirical evidence (the same past price series). Retrospectively (that is, using in-sample data), new and more complex trading rules ("programs") can be constructed by connecting together simple trading rules with logical operators (if, not, and, or). A complex trading rule is not obtained in a single step, but as an outcome of an evolutionary process. This is accomplished by using a variant of Grammatical Evolution (GE). The basic flow of evolution is displayed in the flowchart in Figure 1.

[1] Available online: cran.r-project.org/web/packages/TTR; accessed July 2011.

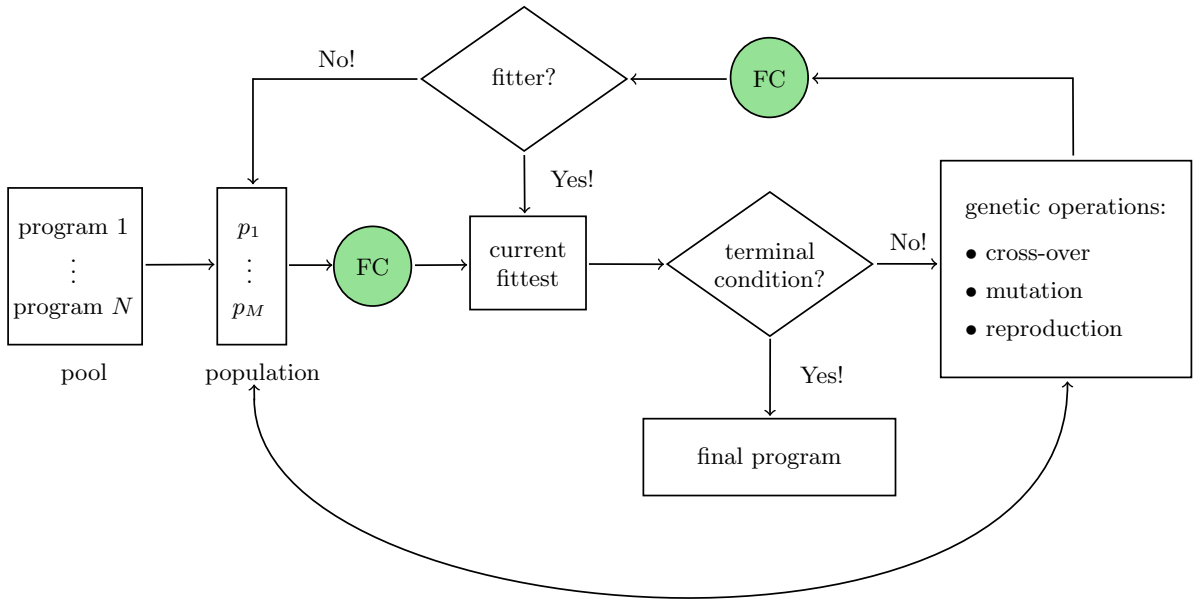

Figure 1: Creating a program

The starting point in the creation of composite trading rules is a pool of N randomly generated programs, from which an initial population of size M is randomly selected. This population is modified in evolutionary steps, repeating the cycle as indicated by the arrows in Figure 1, with the objective of obtaining fitter programs from generation to generation.

Within the evolution process, production rules are applied at each stage until a complete program is produced. We use steady state replacement, meaning M parents produce M children the best of which replace the worst individuals in the active population, if the offspring has better fitness. The standard genetic operators of mutation (probability x), crossover (probability y) and reproduction/duplication (probability z) are adopted.

At each stage of the evolution, the programs are passed to the evaluation module. The evaluation module applies the complete program to the in-sample data set ("open", "high", "low", "last" values and resulting individual indicator signals) and mimics trading to achieve the resulting positions ("Long", "Short", "Square"). It always trades 1 unit of the home currency (1 EUR in our case). This means the grammar is able to evolve complete trading strategies by deciding when to open and when to close a trade. For example, when a "Long" signal is produced, 1 unit of EUR is bought against USD. The consecutive "Square" signal implies closing a position, and hence 1 unit of EUR has to be sold against USD ending up at "Square". For each opening and closing of a trade, the module takes the corresponding side of the price (bid/ask). The spread is defined as 1.5bps.

To assess the success of the program, a fitness function must be defined. The fitness function defines the quality of a trading program candidate. There are many opinions as to what defines a "good" program. In our program, we define a "good" program as one which is profitable and has a high Hit-Miss ratio (ratio of number of positive net returns to number of negative net returns), while it satisfies the minimum requirement of number of trades. Thus, for example, we prefer 9/3 to 3/1.

3 A-Priori Robustness

The basic idea of our approach to a-priori robustness is: A trading rule performing well on the original time series (of prices or returns) should also perform well when exposed to an alternative scenario, exhibiting similar features without being identical to the original time series. The term "a-priori" refers to the fact that the performance of a trading rule is evaluated exclusively for the in-sample period (prior to trading). This procedure bypasses the risk of data-snooping bias and also provides a more manageable foundation for practical trading, since a trading rule which was found fit can be used for actual trading without having to undergo further tests using out-of-sample data.

As explained in the previous section, simple trading rules which constitute the ingredients of programs (see Figure 1) require the input of four price series, summarizing events in a five-minute interval: "open", "high" and "low", and "last". Hence, four series have to be created to provide an alternative scenario, on which a program can be evaluated. The four series are created as follows. Starting from a modified "last" series created by maximum entropy bootstrap, the "open" series is created using a procedure based on kernel density estimation of "open"/"last" ratios. Finally, "high" and "low" values result from an ARMA model for the span "high minus low", covering the heteroskedasticity typically found in the series in a smoother and more robust way than a GARCH could for this type of high-frequency data.

An example of modified "last" series is shown in Figure 2; Figures 3 and 4 display an example of original and modified four-series scenarios.

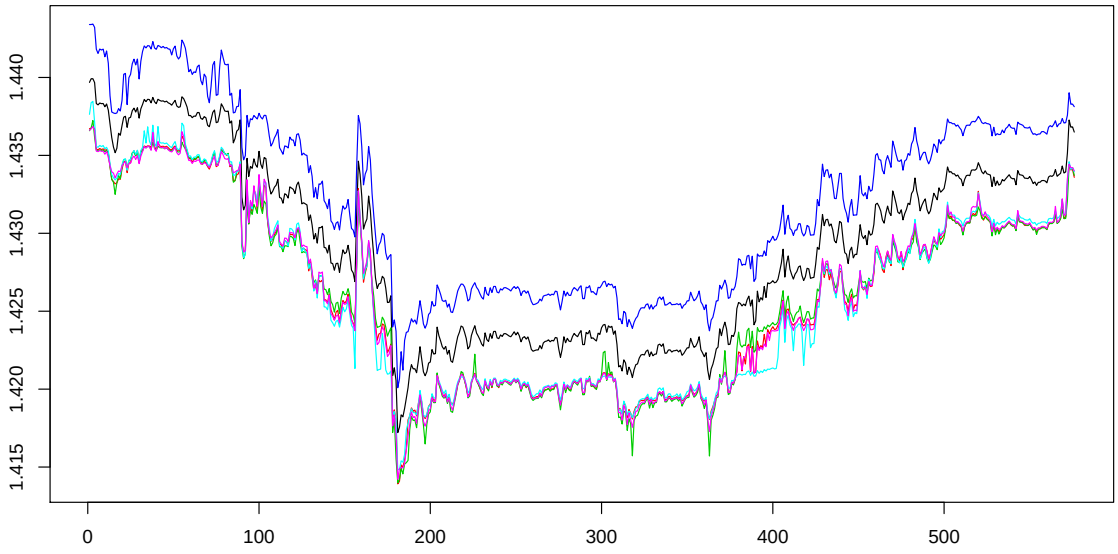

Figure 2: Last prices: original time series (black) and bootstrap simulations

This concept of a-priori robustness is then implemented in the program algorithm. Evaluating the fitness criterion (FC in Figure 1) involves modified price series, in order to attain robust programs. Once the terminal condition has been reached, a trading rule has been found which would have been successful in-sample, as well as with perturbed in-sample data, and which is hence robust in this sense, raising the prospect of fitness also when applied to out-of-sample data, unless the market is entirely efficient.

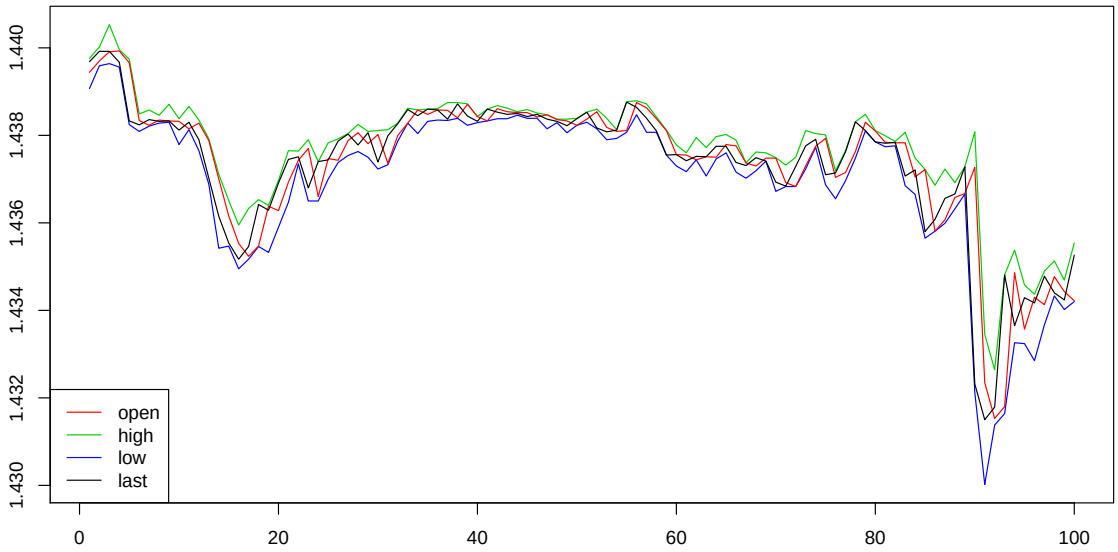

Figure 3: The four price sequences — original

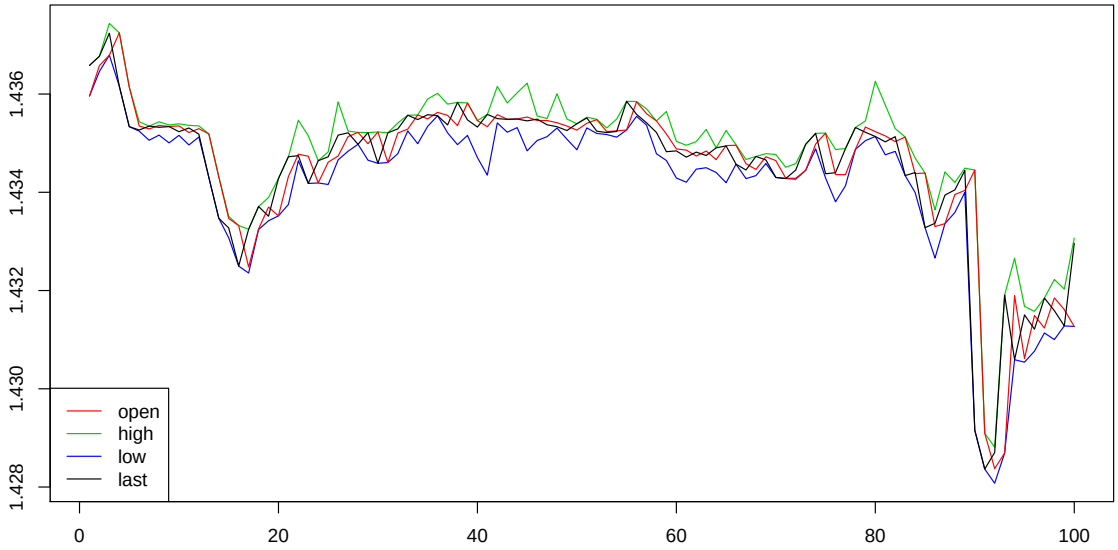

Figure 4: The four price sequences — simulation

4 Experimental Design

In order to investigate the impact of our a-priori robustness criterion on out-of-sample total return, and particularly, profit & loss (pnl) variability, we ran the genetic algorithm both with and without implementation of robustness on five-minute EUR/USD exchange market data: Twenty out-of-sample trading days between February and June 2011 were randomly selected, four examples of each type, Monday through Friday. The two preceding trading days constituted the in-sample period, respectively. For the study of interactions with the input population size to our algorithm, we considered two different population sizes, 1 000 ("low"), and 5 000 ("high"). Thus, our experimental design comprised $2 \cdot 20 \cdot 2 = 80$ ceteris paribus combinations of robustness,

period, and population size. On each of these combinations, we ran 6 replications of our genetic algorithm. Table 1 shows the details.

Source of variation	degrees of freedom		
robustness: yes/no	$2-1$	$=$	1
population size: low/high	$2-1$	$=$	1
period: 1/2/3/.../20	$20-1$	$=$	19
interactions 1: robustness/population size	$(2-1)\cdot(2-1)$	$=$	1
interactions 2: robustness/period	$(2-1)\cdot(20-1)$	$=$	19
interactions 3: population size/period	$(2-1)\cdot(20-1)$	$=$	19
error	$2\cdot2\cdot20\cdot(6-1)+ (2-1)\cdot(2-1)\cdot(20-1)$	$=$	419
total	$2\cdot2\cdot20\cdot6-1$	$=$	479

Table 1: Potential sources accounting for out-of-sample profit variability

5 Results

Figure 5 gives a first impression of our results in terms of total return. Implementation of a-priori robustness appears to weaken the in-sample fitness of the trading program performing best, but to improve its performance in the out-of-sample period.

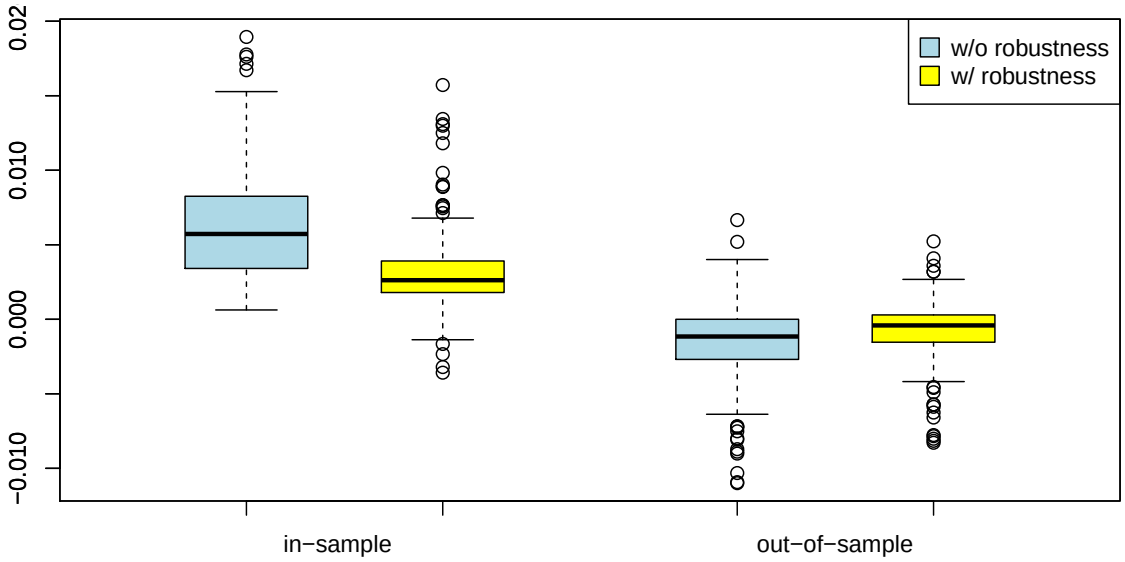

Figure 5: Results: total return

However, the display in Figure 5 ignores the different conditions under which the data were produced. In addition, what is even more important from an investor's perspective, it disregards the existence of transaction fees.

Table 2 gives the results of an ANOVA of the out-of-sample profit & loss (pnl) produced by the fittest trading program found under the respective conditions. The results indicate that population size can be neglected as a source of variation. However, the implementation of a-priori robustness has a significant effect, and the selected time period as well. These findings

were confirmed when applying an ANOVA methodology allowing for heteroskedasticity in the data.

```
Response: pnl
                               Df     Sum Sq    Mean Sq F value    Pr(>F)
robust                          1 0.00016633 1.6633e-04 23.1137 2.132e-06 ***
population                      1 0.00000000 0.0000e+00  0.0000  0.997601
as.factor(period)              19 0.00174162 9.1664e-05 12.7383 < 2.2e-16 ***
robust:population               1 0.00000004 4.4000e-08  0.0061  0.937986
robust:as.factor(period)       19 0.00028345 1.4918e-05  2.0731  0.005274 **
population:as.factor(period)   19 0.00006786 3.5710e-06  0.4963  0.963951
Residuals                     419 0.00301511 7.1960e-06
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Table 2: ANOVA results

Figure 6 sheds light on out-of-sample accounting results, when an investor is trading \$100,000 (the EUR/USD bid-ask spread was set to be 1.5bps). A-priori robustness appears to have an enhancing effect on pnl. This effect may be partly attributed to lower transaction fees, which can be traced back to more cautious trading under the a-priori robustness criterion; see Figure 7. However, fees cannot explain everything. Average results are given in Table 3.

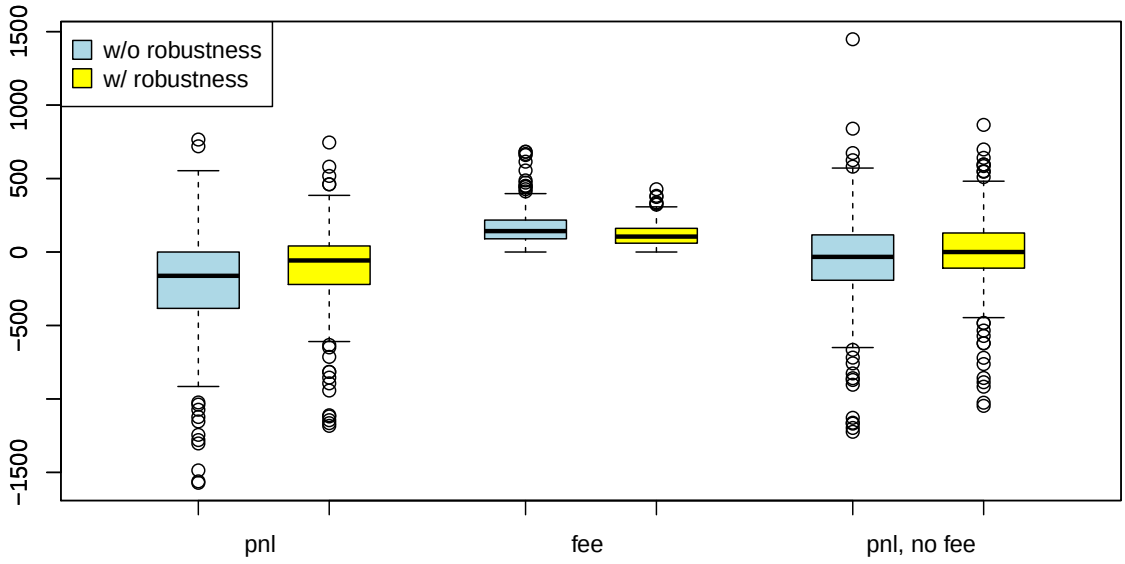

Figure 6: When trading \$100,000...: accounting results

Population sizes can be neglected as source of variability in results, however periods, and in particular the position of the out-of-sample day in the week, cannot. Figure 8 gives an impression of the impact on pnl. The means of out-of-sample pnl are consistently higher under the a-priori robustness criterion. It seems that Friday is the out-of-sample day which is affected the most by a-priori robustness. On Fridays, algorithmic trading uses to be interrupted; market uncertainty is graded the highest on this day. There is evidence that our a-priori robustness criterion can cope with this lack of certainty particularly well.

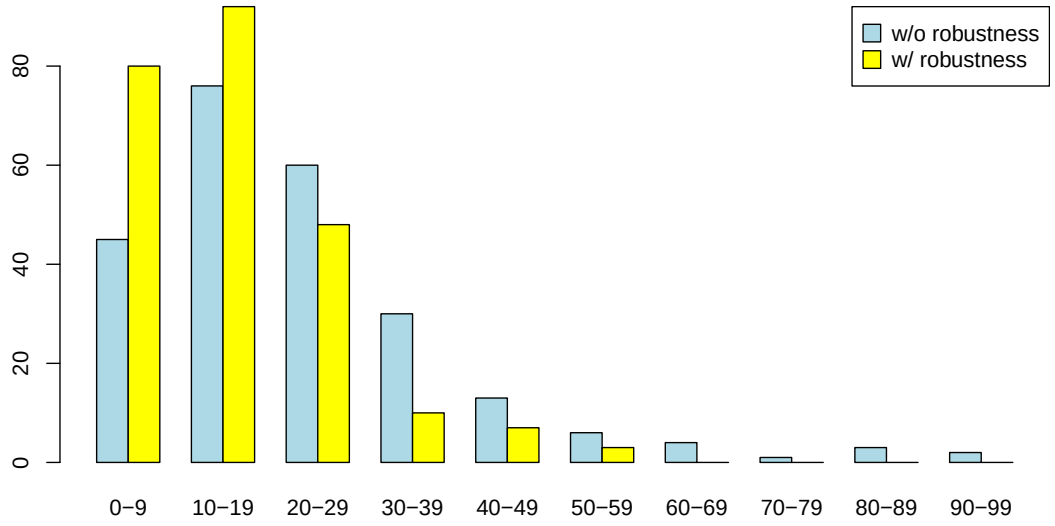

Figure 7: Distribution of times traded

When trading \$100,000...	w/o robustness	w/ robustness
pnl:	−230.72	−112.99
fee:	169.22	111.03
pnl, no fee:	−61.50	−1.96
times traded:	22.56	14.80

Table 3: When trading \$100,000...: average results

6 Summary and Conclusions

Our goal was to construct and test a new framework for trading rule selection which might be able to curb the data snooping bias impending on both the in-sample and out-of-sample stage of performance evaluation. At the core of our approach is the concept of a-priori robustness, which means that a trading rule performing well on the original time series of prices should also perform well when exposed to an alternative scenario, exhibiting similar features without being identical to the original time series. We deploy evolutionary programming tools (a genetic algorithm) for the selection process, and a multi-object fitness criterion which involves the original as well as modified time series. Intra-day EUR/USD market data from the first half of the year 2011 constitute the empirical basis of our study. We carried out an experimental design with the robustness criterion, size of the population of trading programs as input to the genetic algorithm, and in-sample/out-of-sample period as sources of variation.

Our findings suggest that our a-priori robustness criterion gives less spurious results, and that it prevents in-sample overfitting. We observed a consistent pattern of robustness impact on in-sample and out-of-sample trading program fitness, namely curbing the former, but enhancing the latter. There is evidence that out-of-sample profit can be increased, though the point of profitability is not yet achieved. Only parts of this effect can be traced back to higher trading reluctance of programs under the a-priori robustness check, and then to lower transaction fees. While the profit variability does not appear to depend on the size of the input parent population of trading programs, we did find period effects. In particular, there is one out-of-sample day which is most affected by a-priori robustness, namely Friday. An explanation of this finding may be provided from within the debate on the "Efficient Market Hypothesis". As most of

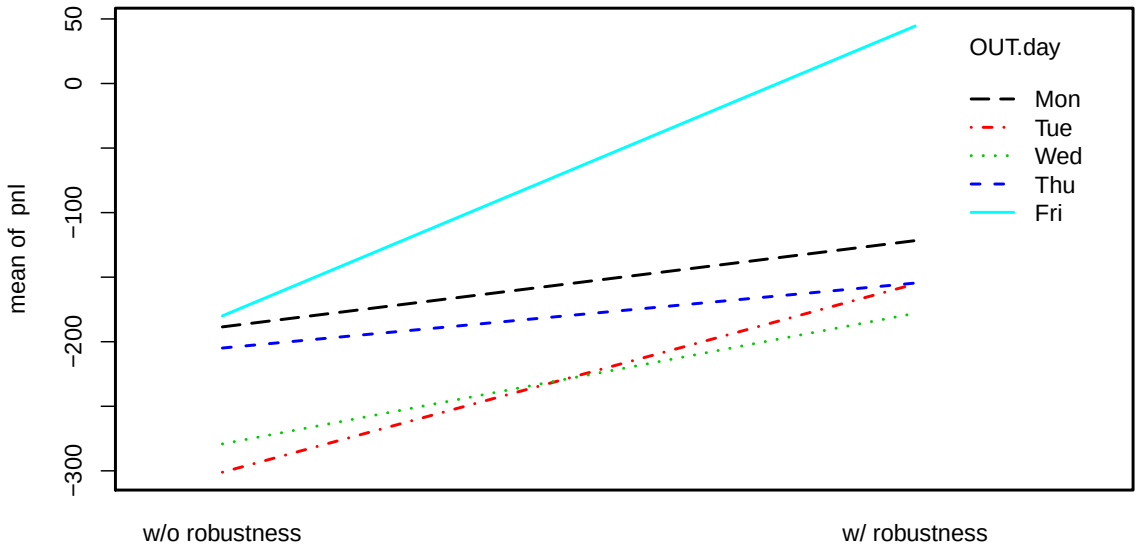

Figure 8: When trading \$100,000...: interaction between robustness and out-of-sample day

algorithmic trading uses to be interrupted for reasons of protection against higher uncertainty on Fridays, the market's efficiency may be questioned on this weekday. It seems that our a-priori robustness criterion can cope with lack of certainty particularly well.

References

[1] Brabazon A., O'Neill, M. (2004): Evolving technical trading rules for spot foreign-exchange markets using grammatical evolution, *Computational Management Science* 1, 311–332.

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

[3] Vinod, H.D., and López-de-Lacalle, J. (2009): Maximum Entropy Bootstrap for Time Series: The meboot R Package, *Journal of Statistical Software* 29, Issue 5.

[4] Merton, R. (1987): On the state of the efficient market hypothesis in financial economics, in: Rudiger Dornbusch, Stanley Fischer, and John Bossons, eds.: *Macroeconomics and Finance: Essays in Honor of Franco Modigliani*, MIT Press, Cambridge, Mass.

[5] O'Neill, M., and Ryan, C. (2001): Grammatical evolution, *Evolutionary Computation* 5, 349–358.

[6] Qi M., and Wu, Y. (2005): Technical Trading-Rule Profitability, Data Snooping, and Reality Check: Evidence from the Foreign Exchange Market, *Journal of Money, Credit, and Banking* 38, 2135–2158.

[7] R Development Core team (2011): *R: A language and environment for statistical computing.* R foundation for statistical computing, Vienna, Austria. URL http://www.R-project.org.

[8] Sullivan, R., Timmermann, A., and White, H. (1999): Data-Snooping, Technical Trading Rule Performance, and the Bootstrap, *The Journal of Finance* 54, 1647–1691.

[9] White, H. (2000): A reality check for data snooping, *Econometrica* 68, 1097–1126.