

Dynamic Generation of Investment Recommendations Using Grammatical Evolution

Carlos Martín, David Quintana*, Pedro Isasi

Department of Computer Science and Engineering, Universidad Carlos III de Madrid, Leganés (Spain)

Received 6 July 2020 | Accepted 20 March 2021 | Published 22 April 2021



ABSTRACT

The attainment of trading rules using Grammatical Evolution traditionally follows a static approach. A single rule is obtained and then used to generate investment recommendations over time. The main disadvantage of this approach is that it does not consider the need to adapt to the structural changes that are often associated with financial time series. We improve the canonical approach introducing an alternative that involves a dynamic selection mechanism that switches between an active rule and a candidate one optimized for the most recent market data available. The proposed solution seeks the flexibility required by structural changes while limiting the transaction costs commonly associated with constant model updates. The performance of the algorithm is compared with four alternatives: the standard static approach; a sliding window-based generation of trading rules that are used for a single time period, and two ensemble-based strategies. The experimental results, based on market data, show that the suggested approach beats the rest.

KEYWORDS

Dynamic Strategy,
Evolutionary
Computation, Finance,
Grammatical Evolution,
Structural Change,
Trading.

DOI: 10.9781/ijimai.2021.04.007

I. INTRODUCTION

FINANCIAL markets are subject to structural changes which cause investment rules that were profitable in the past to lose their effectiveness over time. This characteristic of market dynamics requires the implementation of mechanisms that detect structural changes and update investment strategies accordingly.

From Allen and Karjalainen [1] influential work on evolution of trading rules using Genetic Programming (GP) [2], many authors have followed with contributions based on the same technique or on the use of Grammatical Evolution (GE) [3]. Among them, [4]-[11].

The main disadvantage of the rules generated by the standard algorithm is the fact that they are static and, therefore, do not take into account the prevalent structural changes. Given the situation, the range of possibilities would have two extremes: either holding the same model through time or updating it with any new piece of information to keep up with the evolution of the price generation process.

Even though one might think that the constant update of the model is likely to be more appropriate than the static alternative, in practice it is often the case that the latter results in an increase in transaction costs that erodes completely the additional return obtained from better market timing.

This study improves the standard approach presenting an alternative that commutes between an active model and a candidate one. The algorithm has a hysteresis component that limits overtrading, while maintaining the ability to change the active model to cope with changes in the market price generation mechanism.

This research contributes to the state of the art improving the traditional use of Grammatical Evolution in this context. GE and GP-based approaches are subject to well-known limitations [12], such as the difficulties to outperform the market in the face of strong upward trends, but are popular due the advantages that they offer, such as flexible representation. While there are many potential trading algorithms based on a wide variety of techniques [13]-[15], benchmarking them would require a separate study that goes beyond the aim of this one.

The rest of the document is structured as follows: first, we provide a brief overview of the relevant literature on GE for algorithmic trading. That will be followed by a description of the canonical GE-based static approach and then the introduction of the proposed solution in section IV. The experimental analysis used to evaluate the approach will be reported in section V. Finally, we will devote the last section to summary and conclusions.

II. LITERATURE ON GE FOR TRADING

As we already mentioned, the efforts to obtain profitable trading rules using technical indicators with flexible representation is not new. The seminal contribution by Allen and Karjalainen [1] using GP paved the way for a substantial amount of related works. These extend the mentioned study suggesting new sets of functions and terminals, evaluation methods or investing universes.

In addition to these variations, other authors have explored the possibility of relying on a different core algorithm, GE. This approach, also under the framework of Evolutionary Computation, shares the main advantages of GP in this context and has been widely used in finance and economics [16]. According to a recent study [17], this technique seems to be more robust and to generate trading rules that are simpler and, therefore, easier to interpret. Hence, our decision to rely on GE as the core algorithm.

* Corresponding author.

E-mail address: dquintan@inf.uc3m.es

Please cite this article in press as:

C. Martín, D. Quintana, P. Isasi. Dynamic Generation of Investment Recommendations Using Grammatical Evolution, International Journal of Interactive Multimedia and Artificial Intelligence, (2021), <http://dx.doi.org/10.9781/ijimai.2021.04.007>

Among the earliest studies focused on GE for trading purposes we can highlight the pioneering one authored by Brabazon and O'Neill [18]. This work explores the possibility of generating investment rules for the money market using GE. To that end, these researchers relied on a limited set of indicators together with a risk management mechanism that offered encouraging results. The same year, Dempsey et al. [19] also used GE to evolve investment rules based on technical indicators for the Nikkei 225 and S&P 500 indices. This time, the authors highlight the importance of limiting transaction costs and present a strategy to reduce the number of trading signals by means of a decay constant that controls the investment size of the trades. Their results suggest that the approach is useful. The Nikkei 225 index was beat by a wide margin while the American benchmark seemed to be significantly more difficult to exploit.

Contreras et al. [20] tested the potential of GE to produce profitable trading rules in the Spanish stock market using technical indicators. These researchers compare the profitability of the resulting strategies vs a GA-based alternative that they introduced in a previous work [21] and report profits of 14% vs losses of 20% by the GA-based strategy. These same authors presented more recently two extended versions. The first one, a hybrid one that follows a multi-objective approach [22], includes multi-strategies to limit unforeseen losses and offered good performance in their experimental analysis. The second one [23], was used to explore the feasibility of using meta-GE approximation. The approach, which relies on two overlapped instances of grammatical evolution, uses a combination of macroeconomic, fundamental and technical indicators to generate trading rules. Once again, the system, which promoted more robust and lower-risk portfolios, resulted in promising results.

Schmidbauer et al. [24] introduce a GE-based trading rule selection framework that considers robustness. To that end, they developed a multi-objective fitness test that considers both observed series and synthetic ones generated using bootstrap. They tested their approach on five-minute EUR/USD exchange market data and came to the conclusion that the use of their a-priori robustness criterion improves both robustness and profitability. Despite of this, they did not get to find profitable strategies in their experiments.

We could mention a related study Oesch and Maringer [25] where these authors use GE to develop a high frequency trading system that exploits volume inefficiencies at the bid-ask spread. Their experimental results show that the system identifies strategies that are both profitable and robust.

Martín et al. [26] introduced a GE-based ensemble approach that includes a voting mechanism with an inertia component that balances a certain degree of adaptability to structural change while limiting the number of trading signals. The results, based on S&P500 data, suggest that this strategy beats the traditional static approach that relies on a single model for the whole period and ensembles that implement more standard voting systems.

The approach that we present in this work poses a number of advantages vs other alternatives discussed above. The most important of them are the ability to adapt automatically to structural changes, and the fact that it is done in a way that limits the overtrading that often drags down the performance of other algorithms.

Finally, it is worth mentioning that the applications of GE are by no means limited to this field. This technique might potentially be used to tackle other problems related to economics and management, and its potential should be explored in domains like blockchain in banking [27]-[28] or industrial diagnosis [29].

III. TRADITIONAL STATIC APPROACH

Grammatical evolution, a metaheuristic closely related to genetic programming developed by Ryan et al. [3], encodes individuals as strings of integers that are mapped to programs by means of context-free grammars. Like other techniques within the framework of evolutionary computation, it is a stochastic population-based approach that refines solutions in an iterative way by means of the application of a number of operators (selection, crossover, mutation etc.) according to a basic loop.

While genotypes take the form to strings of integers, phenotypes are structured as Lisp-style functional trees. The connection between these two elements is managed by user-specified grammars that describe the core elements of the programs: terminals, non-terminals and the associated lexis and syntax. It is worth noting that these descriptions, usually in Backus-Naur form (BNF), often incorporate domain knowledge and constrain the search space. The use of grammars offers an important advantage over standard genetic programming, like simultaneously enforcing closure and allowing different data types.

The process starts with the initialization of the population. This requires the generation of a as many vectors of integers as individuals. The vectors are then mapped to terminal and non-terminal elements according to the grammar. The initialization gradually generates functional trees (or their s-string equivalents) until either all the non-terminal nodes get the required inputs, or all the initial contents of the vectors get used, case in which these are expanded with as many additional elements as necessary.

The range of trading rules that can be generated, is determined by the selection of the set of terminal and non-terminal elements, hence its importance. For the purposes of this study we basically relied on a previous study by Lohpetch and Corne [7]. The only difference was that we used daily data instead of monthly information. This choice aligns the study with most of the ones mentioned in the previous section. The set of non-terminal nodes included two relational operators (>and <) and three logical ones (*And*, *Or* and *Not*).

Regarding the terminal elements, we considered a number of technical indicators and the most important basic daily index prices, including *Opening* and *Closing*, *Maximum* and *Minimum*). Simple moving averages, another popular indicator, were included for 2, 3, 5 and 10-months (*MA2*, *MA3*, *MA5* and *MA10*). We included the 3-month as a momentum oscillator, and 12-month Rate of Change indicator (*Roc3* and *Roc12*) and the 3-month rolling minima and maxima (*Mx1*, *Mx2*, *Min1* and *Min2*) as short-term price resistance indicators. Finally, we added two trend-line indicators, upper and lower resistance lines (*UR* and *LR*) to characterize the speed and direction of price changes.

Generating syntactically correct investment rules based on these elements requires the definition of grammars. As we mentioned, these specify which operators and terminals are acceptable as input arguments, together with their appropriate outputs. In this study we also follow the one introduced in [7], which is detailed in Table I in BNF-form. The application of this grammar results in the obtainment of conditional rules that were interpreted as recommendations for being invested, "1", or in cash, "0", depending on market conditions.

Fig. 1 illustrates the structure of these trading rules showing one that could be potentially generated using this approach. In this case, the rule would trigger a recommendation to be invested if either the maximum trading price is larger than the two-month rolling maximum, the two-month rolling maximum is larger than the lower trend line indicator or the 5-month moving average is larger than the 2-month one, or both.

TABLE I. GRAMMAR USED TO DEFINE TRADING RULES

N°	Modulus	Grammar Rule
1	1	<Rule> ::= <bool>
2	5	<Bool> ::= (And <bool><bool>) (Or <bool><bool>) <Bool> ::= (Not <bool>) <Bool> ::= (><exp><exp>) (<<exp><exp>)
3	16	<Exp> ::= (Opening) (Closing) (Maximun) (Minimum) <Exp> ::= (Me2) (Me3) (Me5) (Me10) <Exp> ::= (Roc3) (Roc12) <Exp> ::= (Max1) (Max2) (Min1) (Min2) <Exp> ::= (UR) (LR)

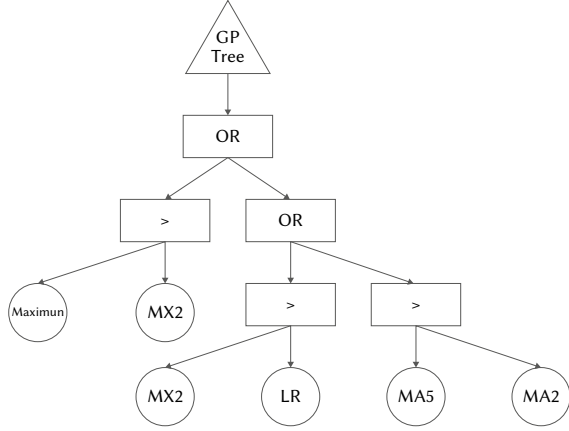


Fig. 1. Example of trading rule represented as a tree. This would be equivalent to the Common Lisp S-expression (Or (>Maximun Mx2) (Or (>Mx2 LR) (>MA5 MA2))).

Once we have discussed rule structure, we will consider evaluation. The quality of trading rules was assessed in terms of net profit or loss. Return, r , was characterized as the sum of returns minus transaction costs using continuous compounding like in [4] and [30]. Hence, the fitness function could be defined formally as

$$r = \sum_{t=1}^T r_t \cdot I_b(t) + \sum_{t=1}^T r_f(t) \cdot I_s(t) + n \cdot \ln\left(\frac{1-c}{1+c}\right) \quad (1)$$

where $r_t = \ln(P_t) - \ln(P_{t-1})$ represents the return on the index computed as its price difference between time $t - 1$ and t ; $I_b(t)$ is a dichotomous variable that is equal to one in the periods where the rule recommends being invested and zero in the rest; $I_s(t)$ is $1 - I_b(t)$, and $r_f(t)$ is the risk-free rate of return for one period of time prevailing at time t .

Regarding the last term of the expression, it is an estimate for the transaction costs resulting from the purchases and sales derived from moving from a recommendation to be invested in the market to hold a cash position, and vice-versa. Here n represents number of transactions and c the one-way transaction cost as a fraction of price (in the experimental analysis it was set at 0.25%).

This process is illustrated in Fig. 2, where we can see the investment positions recommended by an investment rule on the S&P 500 over 250 trading days. There, we show the behavior of the index together with an overlay using a thicker line that shows the return accumulation process. The strategy tracks the market, and therefore its performance, whenever the evaluation of the rule for the day returns “1”, and falls down to the bottom in case the recommendation for the day is being out of the market. Initially the rule recommends being in cash for about a month, accruing the risk-free interest rate, and then investing in the index for 4 days.

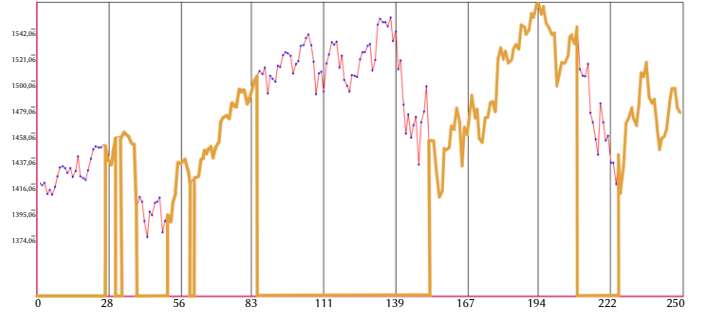


Fig. 2. Example of trading rule performance evaluation. Thicker line illustrates the recommended investment position and return accumulation for a trading rule over the period.

In this case, the evolved trading rule recommends moving in and out of the market several times. In the process, it anticipates two important market corrections, and prevents losses staying in cash over the period from approximately day 84 to 151 and between trading days 208 and 224.

The main loop of GE requires the use genetic operators to drive the iterative improvement process at the core of the metaheuristic. These are very similar to the ones used in the more popular GP, making the range of alternatives very wide. In this study we relied on standard ones: tournament selection, single-point crossover and uniform mutation. Given that the use of these operators often results in a significant number of malformed individuals, we implemented two standard repair mechanisms. The first one, duplication, was used whenever the new individual happened to be too short. This strategy replicates a portion of the individual, selected randomly, and extends the genome up to the required size. The second, truncation, was used in the opposite scenario. It disregards the final elements of the sequence of integers that are unnecessary.

Finally, the core setup used in this study implements non-parametric parsimony pressure, a mechanism that punishes complexity selecting the simplest rule whenever there is a tie in the selection operator. This limit bloating and, as a result, enhances performance, reducing overfitting. In addition to that, this offers the advantage of improving interpretability, a key aspect in this domain.

IV. DYNAMIC APPROACH

The adaptive approach we suggest involves a dynamic selection mechanism that commutes between an active rule and a candidate one, optimized for the most recent available market data.

The system relies on the use of sliding windows. Given a window size, defined by a constant w , the process starts selecting the w most recent data points to evolve a trading rule using GE. Once a rule is obtained, it can be used to generate investment recommendations for future periods. If we do this only once, and we use the resulting strategy over the whole test period, we obtain the standard non-adaptive approach often found in the literature. We will label it *Static*.

On the opposite end of the spectrum, we might repeat the process to generate rules that would only be used once for a single time-step. That is, if we moved the sliding window one time-step at a time, we would obtain overlapping training samples that would differ in a single element. The new one would add the most recent data point, and drop the oldest one. If we considered only the most recent rule to provide an investment recommendation for the next period, and we iterated, we would obtain a very adaptive strategy that we will call *Naïf*. These rules could also be combined into ensembles so that the recommendations of the e most recent rules for any specific period could be combined into a single one either using a simple majority voting approach, *Majority*, or a weighted voting mechanism, *Weighted*.

The solution that we introduce intends to achieve a balance between the need to adapt to structural change and to limit the impact of over-trading. The starting point would be the *Naif* approach, which evolves a trading rule for each time step using a sliding window with a fix window size w . Given a time t , a trading rule is evolved using GE in the period between $t - w$ and $t - 1$. Given that it would be the only alternative, it would be considered the best rule and therefore, it be used to generate the first investment recommendation. At $t+1$, a candidate rule would be evolved based on the period from $t-w+1$ to t . At that point, the current best rule and the new candidate rule would be compared. The best one would get the status of current best rule and therefore, would be used to generate the required investment recommendation for $t+1$.

The process of updating the best investment rule comparing the performance of the current one vs the candidate rule based on the most recent w data points is repeated at every time-step. As a result, we obtain a dynamic investment strategy. It is worth noting that once the current best rule is replaced by a new one, it is lost. The process can only move forward.

Rule comparison is a key aspect that requires clarification. It is made on the basis of investment performance on a common evaluation period that will always be the training period for the new candidate rule (the most recent w data points). This also means that, given period overlaps, from the point of view of current best rule, performance will always be based on the evaluation of investment recommendations on a period that includes test data to some degree. If the current best performing model had been updated in the previous period (and, therefore, had generated recommendations for one period only) and we use it as reference point, we would have $w - 1$ recommendations on training data and 1 recommendation on test data. On the other end of the spectrum, if the best current model had maintained its status for w or more consecutive time periods, we would consider the whole w recommendations on test data. Once again, we would like to emphasize that this explanation uses the current best-performing model as the reference point. As we mentioned, as long as the new candidate rule is concerned, the two rules are compared based on performance on its training sample.

This process is illustrated with an example in Fig. 3. Panel 2a describes the selection of a first inversion rule, obtained with GE, on a training sample of $w = 10$ periods, $t_5 - t_{14}$ (row 1, in dotted black and white). Being the first rule, it is chosen as an active rule and will generate the recommendation. Once the first recommendation is obtained, step T1, "0", t_{15} (in black and white), the sliding window moves one period to the right and the second rule is generated, panel 2b, based on the time period $t_6 - t_{15}$, (row 2, in dotted black and white). The performance of this candidate rule is compared with the performance of the best rule in progress, rule 1, in the same period (marked with black lines). That means we would consider 9 time periods that were part of the training sample for the active rule, $t_6 - t_{14}$ (row, 2, in dotted black and white), and one of the test sample, t_{15} (in black). Based on that, the candidate rule, 2, would take over the role of current best rule, and its recommendation, "0", would be used as the output of the system for T2, the second time step of the test period t_{16} (in black and white) and the sixteenth element of the sequence.

In T3, in 2c, the best current rule would be the second one, and the new candidate would be obtained based on the period from t_7 to t_{16} (row 3, in dotted black and white). In the example, the comparison in this period would favor rule 2 and, as a result, the candidate rule, 3, would be discarded and, once again, the output of the system would mirror the recommendation of the second rule, "0", to stay out of the market. The dynamics of the process in test periods 4-6, represented in 2d - 2f are very similar. As the sliding window moves, it generates new candidate rules that are compared with the best alternative at that time, updating them accordingly.

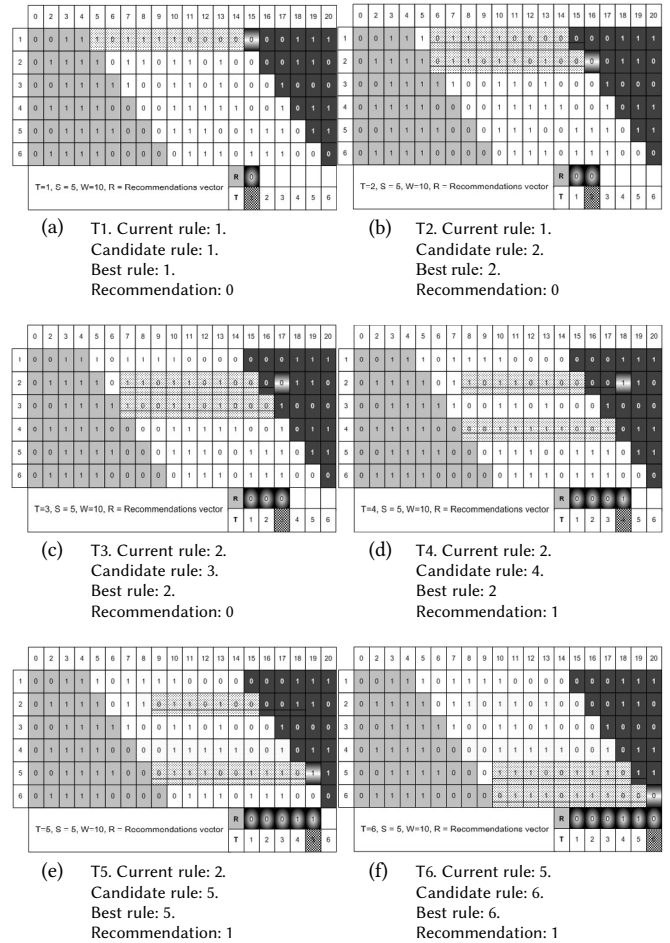


Fig. 3. Illustration of algorithm behavior. Rule selection for test periods 1 to 6. Training samples in darker color. Test samples in lighter color. Investment recommendations by rule: "1" stay in the market "0" stay in cash. Basis of rule comparison in rectangles.

V. EXPERIMENTAL ANALYSIS

This section describes the experimental work used to evaluate the dynamic approach introduced in the previous section. We start discussing the main aspects of the experimental setup, including data set, experimental protocol and parameterization, to then analyze the results.

A. Experimental Setup

The performance of the *Dynamic* approach was assessed comparing its returns vs. the four comparable strategies described in 4: *Static*, *Naif*, and the two ensemble-based alternatives, *Majority* and *Weighted*, to time the Standard & Poor's 500 index.

The sample covered 13 years of daily data starting from the beginning of 2005 to the end of 2017. In addition to the index information, we also needed the historical daily short-term risk-free interest rates of return over the same periods. The former was obtained from the commercial provider Datastream, while interest rates were downloaded from the Federal Reserve Bank of Atlanta.

The main data set was divided in two. The first sample, which covered from 2005 to the end of 2012 was used to run the exploratory experiments for parametrization purposes while the rest was kept for testing. Once the parameters were chosen, the final performance was evaluated on an annual basis in the period from the beginning of

2013 to the end of 2017. Given the evaluation period, we also used the last portion of the first sample to train some of the models that were compared in the test sample.

While the decision to break down the comparative analysis in 5 sub-periods instead of using only one makes no difference in the way the models are trained, we understand that this kind of evaluation provides relevant information on the evolution of reliability over time. Given that GE is a stochastic method we repeated the experiments 30 times.

The exploratory analysis resulted in the selection of population size of 500 individuals, which evolved over 50 generations carrying over the best individual of every generation to the next one. The population was initialized using geometric series with a minimum initial complexity of 5 and a probability of growth of 0.85.

Regarding the main operators, we used simple tournament with a size of 2, and applied the following to those selected: one-point crossover with probability of 0.85, duplication with a probability of 0.05, and uniform mutation with a probability of 0.1. The latter randomly modified genes within a specified range (-128, 127) with a probability of 0.05, allowing a circular wrapping of gene vector up to 16 times before discarding the individual as invalid.

In order to improve diversity and avoid premature convergence, both during the initialization process of the algorithm and during the genetic mutation operation, if the same individual appeared more than once in the population, we tried up to 100 attempts to replace it with a new one.

The size of the training window was set at three years (753 sessions), while the number of sessions in the test period was fixed at one year (251 sessions).

In relation to the ensembles, the number of rules, e , considered to generate the recommendations was set to 5 and the weights of the *Weighted* approach were set to [0.05,0.1,0.15,0.25,0.45]. Here, the last elements of the vector make reference to the emphasis given to the rules based on the most recent information. That is, the vote of the most recent rule would have significantly more importance than the rest.

B. Results

The main results of the experiments are summarized in Table II. There, we report the most important descriptive statistics for the net returns over 30 experiments. The table details the performance of the *Dynamic* approach plus the two benchmarks that represent the opposite extremes in terms of adaptation to structural change, *Static* and *Naif*, plus the two ensembles, *Majority* and *Weighted*, described before.

As we can see, the *Dynamic* strategy provided the best average performance. If we focus on average yearly return over the whole 5-year period, it offered 10.71% net return. Meanwhile, the performance of the most competitive alternative, the *Static* approach, was 2.54%. This result, though significantly poorer, was still better than the one obtained by the ensembles and, specially, the *Naif* one, which resulted in an average yearly net loss of 3.96%. If we consider reliability, the *dynamic* strategy also yielded more consistency, as the average of the yearly return variances was very low compared to the one obtained using the *Static* approach.

Once we analyze the results year by year, we can see that *Dynamic* clearly dominates *Static*, *Naif*, *Majority* and *Weighted* regardless of market conditions. Although the rank of the four alternatives changes between them over time, *Dynamic* consistently outperforms the four of them.

The significance of the reported mean performance differences vs. *Dynamic* was formally tested. The process followed to that end started with the assessment of the normality of the distribution of returns using Kolmogorov-Smirnov test with the Lilliefors correction [31]. Whenever the normality of the results was rejected, we relied on

non-parametric Wilcoxon' test [32]. Conversely, in case that it could not be rejected, we used Levene's homoskedasticity test [33]. At that point, depending on the result, we employed either a t-test [34] or Welch's [35]. According to this protocol, all the differences with the exception of one were significant at 1%. The observed performance differences might be explained by two main reasons: better market timing and better control of transaction costs. The latter aspect was analyzed tracking the number of purchase and sale orders generated by the three methods. This information is reported in Table III.

TABLE II. NET RETURN. MAIN DESCRIPTIVE STATISTICS OVER 30 RUNS. TEST RESULTS

	Strategy	Mean		Median	Var.	Max.	Min.
2013	Dynamic	0.2161		0.2318	0.0017	0.2543	0.1311
	Static	0.0381	*	0.0409	0.0001	0.0715	0.0051
	Naif	0.1318	**	0.1318	0.0002	0.1520	0.0921
	Majority	0.1367	**	0.1363	0.0002	0.1557	0.0964
	Weighted	0.1372	**	0.1387	0.0001	0.1582	0.1088
2014	Dynamic	0.0922		0.0994	0.0002	0.1042	0.0522
	Static	0.0606	**	0.0827	0.0015	0.1042	0.0079
	Naif	0.0559	**	0.0567	0.0002	0.0817	0.0257
	Majority	0.0852	**	0.0876	0.0001	0.0969	0.0471
	Weighted	0.0782	**	0.0784	0.0001	0.0961	0.0548
2015	Dynamic	-0.0152		-0.0123	0.0002	-0.0123	-0.0935
	Static	-0.0160	*	-0.0123	0.0001	-0.0123	-0.0601
	Majority	-0.0903	**	-0.0817	0.0017	-0.0303	-0.1738
	Weighted	-0.1266	**	-0.1270	0.0016	-0.0488	-0.1958
2016	Dynamic	0.0758		0.0889	0.0012	0.0889	-0.0217
	Static	0.0241	**	0.0043	0.0016	0.0889	-0.0223
	Naif	-0.1411	**	-0.1440	0.0014	-0.0335	-0.2193
	Majority	-0.0448	**	-0.0383	0.0012	0.0075	-0.1231
	Weighted	-0.0828	**	-0.0778	0.0023	0.0043	-0.1922
2017	Dynamic	0.1668		0.1668	0.0000	0.1668	0.1668
	Static	0.0204	**	0.0092	0.0016	0.1668	0.0092
	Naif	-0.0718	**	-0.0755	0.0011	0.0295	-0.1316
	Majority	0.0176	**	0.0294	0.0022	0.1094	-0.0753
	Weighted	0.0025	**	0.0057	0.0023	0.0943	-0.1136
Mean	Dynamic	0.1071		0.1149	0.00066	0.1204	0.0470
	Static	0.0254		0.0250	0.00099	0.0838	-0.0120
	Naif	-0.0396		-0.0419	0.00104	0.0311	-0.0985
	Majority	0.0209		0.0267	0.0011	0.0678	-0.0458
	Weighted	0.0017		0.0036	0.0013	0.0608	-0.0676

**Significant vs. Dynamic at 1%

While it is clear that the *Naif* strategy was, by far, the most active one, the rank of the rest is not stable. The *Dynamic* strategy traded less often than the rest 3 out of the five years. The performance of the *Naif* approach was therefore severely undermined, as it is apparent once we analyze performance in gross terms.

Table IV is similar to Table II. The difference is that it represents gross returns and, therefore, the performance has not been adjusted for transaction costs. If we consider average yearly performance over the 2013-2017 period, we see that the 11.56% return offered by the *Dynamic* approach beats both the *Naif* and the *Static* one, with 7.20% and 3.72% respectively. That is also the case for the ensembles, as the one based on simple majority obtained an average gross return of 6.66%, very similar to the one provided by the one based on weighted voting with 6.63%.

TABLE III. NUMBER OF TRANSACTIONS

	Strategy	Mean	Median	Var.	Min.	Max.
2013	Dynamic	4.93	4	6.82	2	10
	Static	7.20	4	48.17	4	32
	Naif	14.00	14	16.00	4	20
	Majority	5.80	6	3.96	4	10
	Weighted	5.07	4	1.86	4	10
2014	Dynamic	4.40	4	5.63	2	10
	Static	7.07	4	25.31	2	14
	Naif	14.73	14	11.72	8	22
	Majority	8.00	8	1.66	6	10
	Weighted	9.13	8	4.6	6	14
2015	Dynamic	2.40	2	3.42	2	12
	Static	2.40	2	1.21	2	6
	Naif	42.40	42	24.94	36	56
	Majority	17.00	18	10.69	10	26
	Weighted	24.33	24	21.26	18	40
2016	Dynamic	2.73	2	8.69	2	18
	Static	5.53	4	13.43	2	18
	Naif	72.47	72	58.40	62	92
	Majority	29.33	28	19.95	18	40
	Weighted	43.13	42	36.33	28	58
2017	Dynamic	2.00	2	0.00	2	2
	Static	1.40	0	13.28	0	12
	Naif	79.40	81	102.39	52	100
	Majority	31.33	32	35.40	20	44
	Weighted	45.93	48	111.86	30	78
Mean	Dynamic	3.29	2.80	4.91	2.00	10.40
	Static	4.72	2.80	20.28	2.00	16.40
	Naif	44.60	44.60	42.69	42.00	48.40
	Majority	18.29	18.40	17.50	11.60	26.00
	Weighted	25.52	25.20	35.18	17.20	40.00

The breakdown by year shows that *Naif* profited much than *Static* in the periods where there was more to be gained. Losses in bad years were also mitigated to a very large extent. It is worth mentioning that disregarding transaction costs turned the 2016 major losses for the *Naif* strategy into profits. Conversely, the *Static* approach evolved a large proportion of trading rules that did not provide any trading signals, hence making the difference between net a gross performance negligible. The ensembles generally performed in line with the *Naif* in gross terms, but they obtained very good results in 2014, where *Majority* got to beat *Dynamic*. Having said that, the difference was small and not statistically significant.

As we also observed when we analyzed net returns, the *Dynamic* approach introduced in this study seems to be the most reliable one in terms gross performance. The average of yearly return variances was around half of the of the second most stable approach.

The fact that the *Dynamic* approach also offered such good results in gross terms indicate that the dominance that we observed in net returns can be explained by a both combination a combination of adaptability to structural change and limited transaction costs. The *Naif* strategy identified and exploited small structural changes, but the excessive trading caused by constant replacements of investment rules increased transaction costs to the point of making flexibility counterproductive. This is likely to be caused by the fact that trading rules identified using GE are implicitly optimized to limit the number of signals, as they are penalized in the fitness function. Once there is

TABLE IV. GROSS RETURN. MAIN DESCRIPTIVE STATISTICS OVER 30 RUNS. TEST RESULTS

	Strategy	Mean		Median	Var.	Max.	Min.
2013	Dynamic	0.2297		0.2531	0.0014	0.2593	0.1561
	Static	0.0561	**	0.0509	0.0004	0.1215	0.0193
	Naif	0.1661	**	0.1654	0.0003	0.2215	0.1409
	Majority	0.1512	**	0.1519	0.0001	0.1676	0.1214
	Weighted	0.1540	**	0.1562	0.0001	0.1732	0.1338
2014	Dynamic	0.1032		0.1092	0.0001	0.1141	0.0721
	Static	0.0783	**	0.0927	0.0007	0.1092	0.0429
	Naif	0.0915	**	0.0890	0.0001	0.1165	0.0716
	Majority	0.1052		0.1074	0.0001	0.1184	0.0671
	Weighted	0.1010		0.0987	0.0001	0.1161	0.0798
2015	Dynamic	-0.0092		-0.0073	0.0001	-0.0073	-0.0635
	Static	-0.0100		-0.0073	0.0001	-0.0073	-0.0451
	Naif	-0.0627	**	-0.0627	0.0024	0.0576	-0.1379
	Majority	-0.0478	**	-0.0415	0.0014	0.0084	-0.1288
	Weighted	-0.0658	**	-0.0692	0.0014	0.0112	-0.1361
2016	Dynamic	0.0826		0.0939	0.0008	0.0939	-0.0042
	Static	0.0379	**	0.0293	0.0015	0.0952	-0.0073
	Naif	0.0357	**	0.0409	0.0018	0.1346	-0.0282
	Majority	0.0285	**	0.0355	0.0010	0.0775	0.0481
	Weighted	0.0250	**	0.0275	0.0019	0.1243	-0.0497
2017	Dynamic	0.1718		0.1718	0.0000	0.1718	0.1718
	Static	0.0239	**	0.0092	0.0017	0.1718	0.0092
	Naif	0.1296	**	0.1290	0.0013	0.1968	0.0591
	Majority	0.0959	**	0.0294	0.0014	0.1694	0.0323
	Weighted	0.1174	**	0.1110	0.0014	0.1922	0.0479
Mean	Dynamic	0.1156		0.1241	0.00048	0.1264	0.0665
	Static	0.0372		0.0350	0.00088	0.0981	0.0038
	Naif	0.0720		0.0723	0.00119	0.1454	0.0211
	Majority	0.0666		0.0565	0.0008	0.0280	0.1082
	Weighted	0.0663		0.0648	0.0010	0.0151	0.1234

** Significant vs. Dynamic at 1% * Significant vs. Dynamic at 5%

a constant change, the strategy that is used in practice is not none of the optimized ones and, therefore, the implicit control mechanism for transaction costs is likely to be affected. The *Static* approach does not offer the flexibility of the rest but, if structural change is limited over the period of use, the trade-off of limited transaction costs vs the loss of accuracy over time could still lead to good results.

The experimental results support the importance of using dynamic approaches like the described one. The advantages are not limited to the ability to generate relevant trading signals in a dynamic environment, which is clearly an important aspect, but also the possibility of doing it at the same time that it controls transaction costs. The proposed strategy *Dynamic* offers a good compromise between two often conflicting objectives: offering flexibility to adapt to structural changes and limiting the number of orders.

VI. SUMMARY AND CONCLUSIONS

The development of investment rules using grammatical evolution often entails obtaining a single rule based on a training period, which is then used to generate recommendations over time. The use of sliding windows improves the adaptability to the structural changes that prevail in financial series but tends to result in excessive transaction costs. Therefore, it is necessary to find alternatives that offer a balance

between flexibility and transaction expenses.

In this study we improve the standard approach introducing new solution that involves a dynamic selection mechanism, which switches between an active rule and a candidate one optimized for the most recent market data available. The process also includes a hysteresis component that reduces the risk of overtrading.

The approach was benchmarked against four alternatives based on the same core algorithm over a period of five years. The alternatives included: the standard static approach *Static*, a solution that updates constantly the decision rule, *Naïf*, and two ensemble-based solutions that differ in the voting mechanism that they implement, *Majority* and *Weighted*.

The results obtained support the superiority of the new solution both in terms of return and reliability, followed by the *Static* approach.

The analysis of the impact of transaction costs on profitability highlights the importance of limiting overtrading, since there is a clear inverse relationship between the number of purchase and sale orders and performance. The *Naïf* approach trades much more often than the rest, and commissions drag down its returns to a very large extent. It is worth noting that controlling this aspect seems to be a key success factor of the *Dynamic* alternative, but not the only one.

These findings bring out the importance of holding a balance between the importance to adapt to market structural changes and the risk of updating constantly recommendation models that are implicitly optimized for the longer term. The results support the new approach as a mechanism capable of maintaining the balance sought between these two contradictory goals.

Future lines of work might include replicating the study with other assets or financial indices; testing the approach with genetic programming; extending the grammar to analyze the impact on the results, or exploring the possibility of updating dynamically the set of terminals and nodes to make sure that the building blocks of the rules remain relevant over time.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support of the Spanish Ministry of Science, Innovation and Universities under grant PGC2018-096849-B-I00 (MCFin).

This work has been supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3MXX), and in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

REFERENCES

- [1] F. Allen, R. Karjalainen, "Using genetic algorithms to find technical trading rules", *Journal of Financial Economics* vol. 51, no. 2, pp. 245–271, 1999.
- [2] J.R. Koza, "Genetic programming as a means for programming computers by natural selection", *Statistics and Computing*, vol. 4, no. 2, pp. 87–112 1994.
- [3] C. Ryan, J. Collins, M. O'Neil, "Grammatical Evolution: Evolving Programs for an Arbitrary Language", in *Proc. of the First European Workshop on GP*. Springer Berlin Heidelberg pp. 83–96, 1998.
- [4] C. Setzkorn, L. Dipietro, R. Purshouse, "Evolving Rule-Based Trading Systems", Technical Report ULCS-02-005, Department of Computer Science, University of Liverpool, 2002.
- [5] C.J. Neely, "Risk-adjusted, ex ante, optimal technical trading rules in equity markets", *International Review of Economics & Finance*, vol. 12, pp. 69–87, 2003.
- [6] N. Navet, S.H. Chen, "On predictability and profitability: Would GP induced trading rules be sensitive to the observed entropy of time series?", *Studies in Computational Intelligence*, vol. 100, pp. 197–210, 2008.
- [7] D. Lohpetch, D. Corne, "Discovering effective technical trading rules with genetic programming: Towards robustly outperforming buy-and-hold" in *Proceedings of the World Congress on Nature and Biologically Inspired Computing, NABIC 2009*, pp. 439–444.
- [8] D. Lohpetch, D. Corne, "Discovering effective technical trading rules with genetic programming: Towards robustly outperforming buy-and-hold", in *Lecture Notes in Computer Science*, vol. 6025, 2010, pp. 171–181.
- [9] J. How, M. Ling, P. Verhoeven, "Does size matter? A genetic programming approach to technical trading" *Quantitative Finance*, vol. 10, no. 2, pp. 131–140, 2010.
- [10] A. Esfahanipour, S. Mousavi, "A genetic programming model to generate risk-adjusted technical trading rules in stock markets", *Expert Systems with Applications* vol. 38, no. 7, pp. 8438–8445, 2011.
- [11] V. Manahov, R. Hudson, H. Hoque, "Return predictability and the 'wisdom of crowds': Genetic Programming trading algorithms, the Marginal Trader Hypothesis and the Hayek Hypothesis", *Journal of International Financial Markets, Institutions and Money*, vol. 37, pp. 85–98, 2015.
- [12] S.H. Chen, T.W. Kuo, K.M. Hoi, Genetic Programming and Financial Trading: How Much About "What We Know" In: Zopounidis C., Doumpos M., Pardalos P.M. (eds) *Handbook of Financial Engineering. Springer Optimization and Its Applications*, vol. 18 (Springer US, Boston, MA, 2008), pp. 99–154.
- [13] E.A. Gerlein, M. McGinnity, A. Belatreche, S. Coleman, "Evaluating machine learning classification for financial trading: An empirical approach", *Expert Systems with Applications*, vol. 54, pp. 193–207, 2016.
- [14] R. Ray, P. Khandelwal, B. Baranidharan, "A survey on stock market prediction using artificial intelligence techniques", in *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 594–598.
- [15] T.L. Meng, M. Khushi, "Reinforcement learning in financial markets", *Data*, vol. 4, no. 3, 110, pp. 1–17, 2019.
- [16] A. Brabazon, "Grammatical Evolution in Finance and Economics: A Survey", in Ryan C., O'Neill M., Collins J. (eds) *Handbook of Grammatical Evolution*. Springer, pp. 263–288, 2018.
- [17] C. Martín, D. Quintana, P. Isasi, "Evolution of trading strategies with flexible structures: A configuration comparison". *Neurocomputing*, vol. 331, pp. 242–262, 2019.
- [18] A. Brabazon, M. O'Neill, "Evolving technical trading rules for spot foreign-exchange markets using grammatical evolution", *Computational Management Science*, vol. 1, pp. 311–327, 2004.
- [19] I. Dempsey, M. O'Neill, A. Brabazon, "Evolving technical trading rules for spot foreign-exchange markets using grammatical evolution", *GECCO 2004 Workshop Proceedings* pp. 9137–9142, 2004.
- [20] I. Contreras, J.I. Hidalgo, L. Nunez-Letamendia, "Evolving technical trading rules for spot foreign-exchange markets using grammatical evolution", in *Applications of Evolutionary Computing, EvoApplications 2013: EvoCOMNET, EvoCOMPLEX, EvoENERGY, EvoFIN, EvoGAMES, EvoIASP, EvoINDUSTRY, EvoNUM, EvoPAR, EvoRISK, EvoROBOT, EvoSTOC*, vol. 7835 (Springer, Berlin, Heidelberg, 2013), pp. 244–253.
- [21] I. Contreras, J.I. Hidalgo, L. Núñez-Letamendia, "A GA Combining Technical and Fundamental Analysis for Trading the Stock Market", in *A GA combining technical and fundamental analysis for trading the stock market* (Springer, Berlin, Heidelberg, 2012), pp. 174–183.
- [22] I. Contreras, J.I. Hidalgo, L. Núñez-Letamendia, "Evolving technical trading rules for spot foreign-exchange markets using grammatical evolution", *Journal of Intelligent and Fuzzy Systems*, vol. 32, no. 3, pp. 2461–2475, 2017.
- [23] I. Contreras, J.I. Hidalgo, L. Núñez-Letamendia, J.M. Velasco, "A metagrammatical evolutionary process for portfolio selection and trading", *Genetic Programming and Evolvable Machines*, vol. 18, no. 4, pp. 411–431, 2017.
- [24] H. Schmidbauer, A. Rösch, T. Sezer, V.S. Tunaliog'lu, "Robust trading rule selection and forecasting accuracy", *Journal of Systems Science and Complexity*, vol. 27, no. 1, pp. 169–180, 2014.
- [25] C. Oesch, D. Maringer, "Robust trading rule selection and forecasting accuracy", *Quantitative Finance*, vol. 17, no. 5, pp. 717–727, 2017.
- [26] C. Martín, D. Quintana, P. Isasi, "Grammatical evolution-based ensembles for algorithmic trading", *Applied Soft Computing*, vol. 84, 105713, pp. 1–10, 2019.

- [27] R. Arjun, K. R. Suprabha, "Innovation and Challenges of Blockchain in Banking: A Scientometric View". *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 3, pp. 7–14, 2020. 10.9781/ijimai.2020.03.004.
- [28] A.T. Chatfield, C. Reddick, "Blockchain Investment Decision Making in Central Banks: A Status Quo Bias Theory Perspective", in *Proceedings of Americas Conference on Information Systems AMICS 2019 Proceedings* pp. 1–10.
- [29] F.Z. Benkaddour, N. Taghezout, F.Z. Kaddour-Ahmed, I.-A. Hammadi, "An adapted approach for user profiling in a recommendation system: Application to industrial diagnosis", *International Journal of Interactive Multimedia and Artificial Intelligence*, vol 5, no. 3, pp. 118–130, 2018.
- [30] L. Becker, M. Seshadri, "GP-evolved technical trading rules can outperform buy and hold", in *Proceedings of the Sixth International Conference on Computational Intelligence and Natural Computing*, Embassy Suites Hotel and Conference Center, Cary, North Carolina USA, September 26–30, 2003, pp. 26–30.
- [31] H., Lilliefors, "On the Kolmogorov–Smirnov test for normality with mean and variance unknown", *Journal of the American Statistical Association*, vol. 62, pp. 399–402, 1967.
- [32] F., Wilcoxon, "Individual comparisons by ranking methods", *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
- [33] H., Levene, "Robust tests for equality of variances" In Ingram Olkin; Harold Hotelling; et al. (eds.). *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*. Stanford University Press. pp. 278–292, 1960.
- [34] "Student" W. S. Gosset, "The probable error of a mean", *Biometrika*, vol. 6, vol. 1, pp. 1–25, 1908.
- [35] B.L., Welch, "The generalization of 'Student's problem' when several different population variances are involved". *Biometrika*, vol. 34, no. 1–2, pp. 28–35, 1947.



Carlos Martín

Bachelor in Computer Science from UNED. He also holds an M.S. in Computer Science and Technology and Ph.D. in Computer Science from Universidad Carlos III de Madrid. His main interests in the field of Artificial Intelligence are focused on optimization techniques based on Evolutionary Computation. He is an engineer at the Security Operations Center of the Air Force JSTCIS Cyberdefense Directorate

in The Spanish Ministry of Defense, Officer in charge of the Forensics, Intrusion Detection, Malware Analysis and Mitigation and Recovery sections.



David Quintana

Visiting Professor with the Department of Computer Science at Universidad Carlos III de Madrid, Spain. There, he is part the bio-inspired algorithms group EVANNAL. He holds a Bachelor in Business Administration and a Ph.D. in Finance from Universidad Pontificia Comillas (ICADE), a Bachelor in Computer Science from UNED and an M.S. in Intelligent Systems from Universidad Carlos III

de Madrid. His current research interests are mainly focused on applications of evolutionary computation and artificial neural networks in finance and economics. David is former Chair of the Computational Finance and Economics Technical Committee of the IEEE Computational Intelligence Society.



Pedro Isasi

Graduate and Doctor in Computer Science by the Polytechnic University of Madrid since 1994. Currently he is University professor and head of the Evolutionary Computation and Neural Networks Laboratory in the Carlos III of Madrid University. Dr. Isasi has been Chair of the Computational Finance and Economics Technical Committee (CFETC) of the IEEE Computational

Intelligence Society (CIS), Head of the Computer Science Department and Vice-chancellor in the Carlos III University among others. His research is centered in the field of the artificial intelligence, focusing on problems of Classification, Optimization and Machine Learning, fundamentally in Evolutionary Systems, Metaheuristics and Artificial Neural Networks.