Algorithmic Trading Using Genetic Algorithms in the Brazilian Stock Exchange

Jean Pierre Jarrier Conti[1,2], Heitor Silvério Lopes[1]

[1] *Bioinformatics and Computational Intelligence Laboratory (LABIC)*

Federal University of Technology - Parana (UTFPR)

[2]Technology Division - Telefonica Brazil

jpjconti@gmail.com, hslopes@utfpr.edu.br

***Abstract*—The evolution of the computational capacity has been helping financial markets to increase the success in their operational running strategies on its investment portfolios. After stock market evolved to make all its operations electronically a new approach called algorithmic trading has gained attention from academic researches. This paper presents a novel method of the dynamic optimization to improve the profit of the algorithmic trading. Combining two genetic algorithms, the proposed approach seek to finding the best optimization and trading window for a trading strategy. The performance of this approach was evaluated with data of the last five years of two stocks traded at the Brazilian Stock Exchange. Comparing the results obtained with classical moving averages indicators, the proposed method performed better in all cases using the complete dataset and using year by year, all experiments using shares of PETR4. These results suggest that the discovery of the optimal trading and optimization window we can improve the system trading strategy and lead to increased profits.**

***Index Terms*—Finance, optimization, evolutionary computation, genetic algorithm, algorithmic trading**

I. Introduction

The buy and sell movements in stock markets move billions of dollars in transactions each day. Around the world many business opportunities occur when shares moves up and down. Many researches have studied the financial market and, in most cases, the focus is to understand the methods and algorithms to support the decision-making in different financial market segments [1] [2].

In recent years, the computational systems have helped trading companies to increase their operational success in the management of investment portfolios. Along years, the increasing use of information technology improved the stock markets such that nowadays all their operations are electronically and online. Then, algorithmic trading (also known as AT, "algo", or "black-box") emerged as new generation of decision-support systems that gained attention from the academia [3]. In the early 2012 algorithmic trading accounted for at least 50% of the total US equity trading [4]. An algorithmic trading can be defined as a system trader where the buy/sell decisions are based on computer algorithms [5] and information of the financial market. In this way, the evolution of the Internet has become a place with several opportunities to make trades. There are a lot of strategies running on the Internet and trades automating your strategies using specific softwares or complex frameworks. A full algorithmic trading system has a various components, such as: pretrader analysis, trading signal generation, trade execution and post-trade analysis [3].

As shown in Fig. 1, pretrader analysis in an algorithmic trading system involves analyzing financial information and the generation of trading signals. Broadly, there are two fronts used in stock market to study the tendency of the prices and the generation of trading signals:

- **Fundamentalist analysis**: in this case, the decision of buying and selling is provided from fundamental information of companies [6]
- **Technical analysis**: it considers that current price oscillations can be related to past historical prices. It used statistical analysis to create technical indicators that help traders to identify tendencies and opportunities in the market [7].

Many of the strategies created on algorithmic trading systems use technical analysis indicators to create trading signals. There are several input parameters in the technical indicators and several different types of indicators are used in a trading system. Although strategies begin by the basic concepts and, along time they continue being improved, backtesting can be used to evaluating the performance of a strategy. The idea behind the backtesting process is the use of historical data

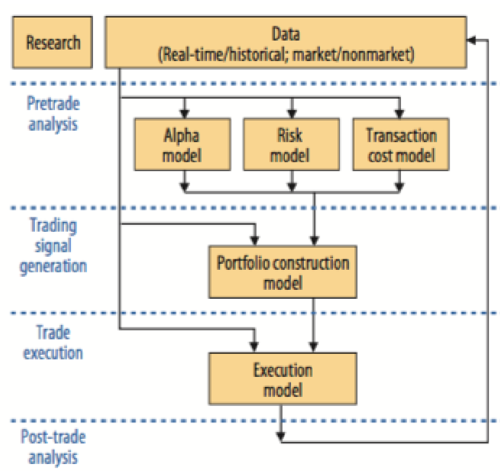

Fig. 1. A complete algorithmic trading system. The pretrade analysis includes three mathematical models. After, a portfolio construction model is generated. Finally a model executes the trade. (Adapted from [3])

and to evaluate the behavior of the strategy.

In this way, an approach often used is search the better indicators parameters for the system. However, in some cases the range of the variables and the quantity of the parameters makes it impossible. For this problem, genetic algorithms can be used to find the best parameters of the system at a specific time. Additionally, another problem is to find the time after the optimization that the system must be optimized again. In this perspective, this paper seeks to answer these questions:

- What is the better optimization window for tuning the parameters of an algorithmic trade system?
- How long to wait before it is needed to tune these parameters again?

Therefore, this work aims to find optimization and trading window to maximize the profit of the trading system. Considering that the market can change from time to time, this proposal aims at dynamically tuning the parameters and verify if they turned out to be better to an algorithmic trading system.

The the paper is structured as follows: Section II presents an introduction about stock market technical indicators, as well as some related work and optimization techniques in the financial markets. Next, in Section III explain the data used, the methods and experiments. Experimental results and their analysis are shown in Section IV. Section V presents the conclusions and future work.

II. Background and Related Work

A. Technical Indicators

Technical Indicators can be defined as mathematical formulae that use as input, price, volume (or both) of a stock, in a given time window. There are hundreds of technical indicators. However, they can be classified into three main groups: trend, momentum, and volatility-based indicators [8].

The trend indicators are used to get the direction of the market, whilst momentum indicators can help the investor by providing buy and sell points. Volatility indicators show in what extent the prices are volatile. Table I summarizes the most common technical indicators.

TABLE I
COMMON TECHNICAL INDICATORS

Indicator	Group
Moving Average (SMA)	Trend
MACD	Trend
ADX	Trend
Average True Range	Volatility
Bollinger Bands	Volatility
Relative Strength Index (RSI)	Momentum
Stochastic Oscillator (STC)	Momentum
Williams %R	Momentum

B. Moving Average Indicator and Trading Rules

A moving average is trend indicator and is used to follow the trend. It computes the value of the average from the last n days and the result can be exponential or linear [8]. The Simple Moving Average (SMA) was used in this work, and it is defined in Equation 1 as a simple average of closing prices ($Close$) for the last N days.

$$SMA = SUM(Close, N)/N \tag{1}$$

Exponential moving average (EMA) is similar to the SMA. $Close$ is the close price at the current time, EMA(prev) is the previous periods of the exponential moving average value and P is the percentage of using the close price value. Equation 2 define EMA:

$$EMA = (Close * P) + (EMA(prev) * (1/P)) \tag{2}$$

Fig. 2. Moving Average Indicator on PETR4 stock from the Brazilian Stock Exchange.

In general, trading rules are the building blocks of a trading system and those rules are based on the values of indicators. It generates buy and sell signals according to the steps defined in [8]. When the system receives the suggestion there are 3 possibilities: buy, sell or hold the position. Figure 2 shows a 21-days moving average plotted for PETR4 stock, in the Brazilian Stock Exchange (BOVESPA[1]). The up arrow shows a suggestion to open a position (buy) in the market when the closing price is crossing up the 21 days moving average indicator. Conversely, the down arrow shows a suggestion to close (sell) the position when the closing price is crossing down the SMA indicator. This paper will use this trading rule to build its algorithmic trading.

C. Evolutionary Computation

Evolutionary Computation (EC) is defined as a family of algorithms inspired by the evolution of living beings and can solve complex engineering optimization problems [9].

The most widely known EC method is the Genetic Algorithm (GA). They are a simplified version of Darwinian evolution and using genetic operators like selection, crossover, and mutation to deliver the best solution to optimization problems [10].

In genetic algorithms, the variables of the problem are encoded in chromosome, usually in the binary format. A population of chromosomes (or individuals) is randomly created and submitted to the evolutionary process. At each generation,

[1]http://www.bmfbovespa.com.br

a new population is created after choosing the best individuals of the preceding population and applying genetic operators (crossover and mutation) to create new individuals. each individual is evaluated by a fitness function [10], [11] that measures how good/efficient is the corresponding solution for the problem in hand. The key point in GA is the selection procedure, since it is where the principle of natural selection takes place. There are some classic selection methods: (i) Fitness proportionate selection – it is similar to a roulette wheel in a casino, with slots of size proportional to the fitness of each individual, (ii) Tournament selection – several "tournaments" are run with a few of randomly chosen individuals, and the best of each tournament is, (iii) Rank-based selection: all the population is ranked according the fitness and individuals are selected sequentially. For more details about EC algorithms and applications, see [12].

D. Related Work

Several papers describe techniques used to build trading strategies in the financial markets. In general, from the strategy perspective, these papers can be divided into: (i) fundamental analysis, (ii) technical analysis, and (iii) blending analysis. On the other hand, from the evolutionary computation perspective we have: evolutionary algorithm, swarm intelligence, and hybrid EC techniques [4].

In [13] a trading system based on indicators of the technical analysis and GA was proposed. The system created used five popular technical indicators (EMA, MACD, RSI and Williams) to build fifteen different trading systems. The model consisted of two stages: elimination of unacceptable stocks and stock trading construction. The proposed expert system used data of 15 stocks in the Thai Stock Exchange from 2011-2014. A similar approach was found in [14], who proposed a new genotypic encoding method named allele-based indirect coding. Technical indicators like MA, EMA, and Bollinger bands were used in the trading rules. A similar concept of optimization and trading sliding window was built. However, the time was fixed at years and months. The indirect coding was superior to the direct coding in computational costs.

[15] proposed a different approach using two GAs. First, a GA was used to find a set of optimized parameters of different technical indicators. Then, another GA optimized the distribution of stock weighting in the portfolio. In this research, seven technical indicators are used (SMA, MACD, STC, RSI, Williams, MFI, MTM). Results suggested that the use of several indicators can lead to good results.

In [16] a GA was applied to find optimal mix of technical trading rules and decide which rules should be applied to particular Egyptian stocks. A similar approach was done by [17], but applied to the Madrid Stock Exchange. A mix of GA and Genetic Programming (GP) was proposed by [18] for the simulation of stock markets, including the process of creating decisions and the simulation of behavior of trading agents. Recently, [19] proposed an GA to predict the price movement of stocks in high-speed trading, using price data of stocks on the microscopic level.

For a more detailed survey of the application of GA and other computational intelligence methods in the financial markets, see [1] and [11].

III. Methods and Experiments

This section describes the methods and experiments performed. Initially, an overview of the proposed method is detailed. Next, the experimental environment will be shown. Details from the dataset and trading rules will be addressed in the next subsection. Finally, the details of the experiments will be presented.

A. Overview of Proposed Method

The proposed approach seeks to maximize profit, by finding the best optimization and trading windows for a trading system. Using a dynamic trading window, parameters of some technical indicators are adjusted to find the best values for trading in a specific period of time. Similarly, the proposed approach also searches the best optimization window to optimize the system.

The proposed framework is shown in Figure 3. Two genetic algorithms, namely AG1 and AG2, are combined sequentially to optimize the proposed solution. AG1 is used to improve the optimization window, whilst AG2 optimizes the parameters of the technical indicators. The processing flow of the method is summarized as follows:

1) Each new generation AG1 creates an optimization and trading window.
2) AG2 finds the best moving average for the given window.
3) AG1 computes the final accumulated profit for a trading window.
4) A new generation of optimization values and trading window is created.

Figure 4 complements the explanation showing how a dynamic training window works. At each generation of AG1, an optimization and trading window is created and it slides for all the dataset. This approach creates a dynamic training window since each new training window will optimize the parameters of the technical indicator.

In terms of internal structure both AG1 and AG2 have a simple, but scalable, chromosome structure. This was done since aiming at future work to improve our approach and integrate several other technical indicators. Figure 5 shows in details the chromosome representation of variables. Moreover, Table II consolidates all parameters used by AG1 and AG2.

B. Experimental environment

Python [2] environment with Anaconda [3] data science packets were used to development and run the experiments. Together, Python and Anaconda are the most popular data science platform and high-level programming language for general-purpose programming. Inside the Anaconda package, we have

[2]http:www.python.org

[3]https://www.anaconda.com

TABLE II
MAIN PARAMETERS OF AG1 AND AG2

Parameters	Value
Min/Max value to moving average	3/90
Min/Max value to days of optimization	15/180
Min/Max value to days of trading	15/180
AG1 Generations	80
AG1 Population	8
AG1 Mutation	0.25
AG1 Crossover	0.70
AG2 Generations	60
AG2 Population	8
AG2 Mutation	0.4
AG2 Crossover	0.70

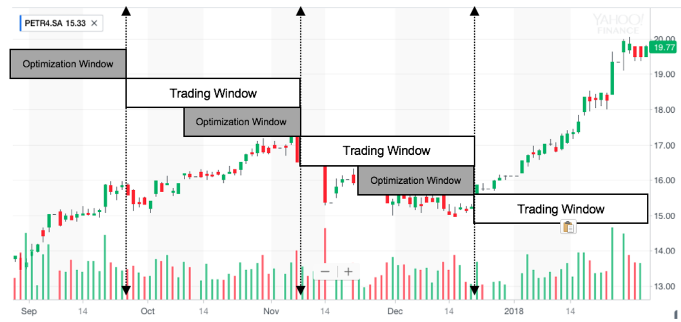

Fig. 4. For each AG1 generation, an optimization and trading window is created and it slides for all dataset. This approach creates a dynamic training window since each new training window will optimize the parameters of the technical indicator. This picture represents the simulation of the one specific training window and optimization window. Naturally, the algorithm will use AG1 to find the best training and optimization window of the period.

also used Jupiter Notebooks, a web-based, interactive computing notebook environments. For each experiment, a different Python notebook was created.

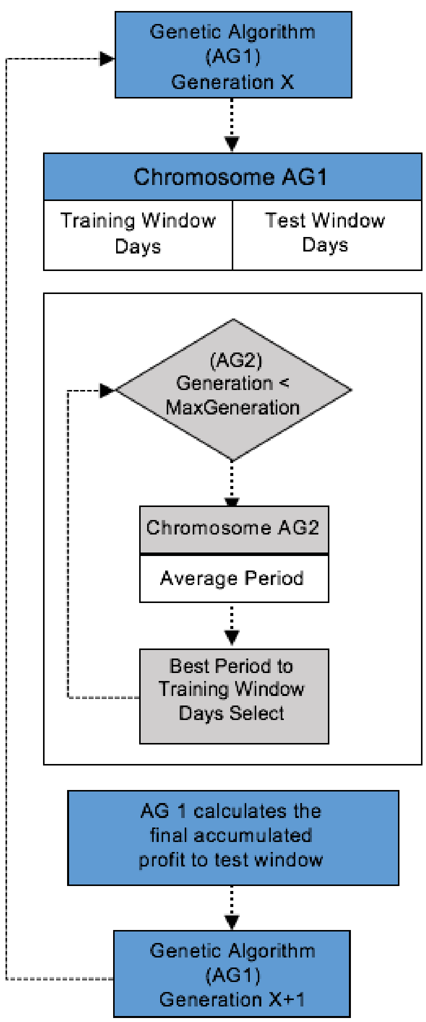

Fig. 3. The complete framework proposal where two genetic algorithms (AG1 and AG2) are combined.

Two more frameworks were used to create a complete experimental environment. They are Inspyred[4] and PyAlgoTrade[5]. The first is a library of bio-inspired algorithms Python, containing the most important evolutionary algorithms. PyAlgoTrade is a Python algorithmic trading library with focus on backtesting and support to live-trading. This framework is scalable, free and full documented. Among its main functions, we can highlight: supports the market, limit, stop and stop Limit orders, supports Yahoo finance, Google finance and CSV files. It includes a set of examples that help to create simulated strategies. We used Inspyred version 1.0.1 and PyalgoTrading version 0.18.

C. Datasets and trading rules

The data used for the experiments include information of the Brazilian Stock Exchange (BM&F Bovespa). These data were collected from Yahoo Financial and exported via comma-separated values (.csv). The dataset contains data extracted between 2013/01/01 until 2018/01/01. Two Brazilian companies were selected for this particular study: Petrobras (PETR4), Petróleo Brasileiro S.A and *Companhia Energética de Minas*

[4]http://pythonhosted.org/inspyred/
[5]http://gbeced.github.io/pyalgotrade/

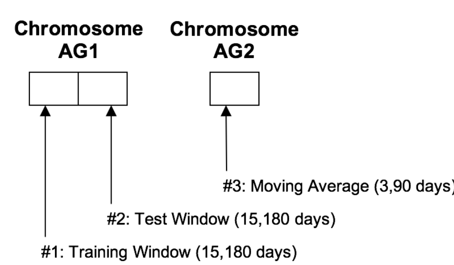

Fig. 5. Details about the chromosome AG1 and AG2 with its variables. In terms of the structure, both AG1 and AG2 has a simple chromosome structure. This structure can scale once we add severals trading rules variables.

Gerais – CEMIG (CMIG4). These companies are part of the Bovespa Index.

To create optimization and trading window, the datasets were split into two windows during the simulation. The size of these windows depends on the AG1 population value and are variable between 15 and 180 days. Inside the dataset there are 4 columns:

- Open - the first trade price in the period;
- Close - the last trade price in the period;
- High - the higher trade price in the period;
- Low- the lower trade price in the period.

As previously discussed, trading rules are building blocks of our trading system that generates buy and sell signals according to the specific rules definition. In this work, a simple trading rule was defined in our algorithmic trading. The system will open a position in the market when the closing price is crossing up the moving average indicator. At the same time, the system will close the position when the closing price is crossing down the SMA indicator. The use of rules based on moving averages is one of the simplest techniques easily programmable to generate buying and selling signals [20].

D. Experiments

Several experiments were done to adjust the control parameters of AG1 and AG2, including: population, mutation and crossover probability, selection method, evaluation, elitism, replacement and Gaussian mutation rate. Those last two are predefined variations for EC methods available in the Python scientific library. The Gaussian mutation variable returns the mutants created by a Gaussian mutation on the candidates. In the generational replacement a large portion of the population (frequently, the whole population) is substituted by its descendants. Optionally, a weak elitism can be done, maintaining some of the best individuals for the next generation. The choice of the optimal parameters was experimental. However, the adjustment of parameters took into consideration the evolution of the best, worst, and median fitness of the population along generations. This was done to guarantee slow convergence and avoid overfitting. Stocks PETR4 and CMIG4 were used in all experiments.

Once the control parameters were defined, two different experiments were performed. (i) AG1 and AG2 were run using all dataset segmented year by year. (ii) AG1 and AG2 were run using complete dataset. Using the trading rules previously discussed, these experiments were compared to classical values of MA (Moving Average) indicators: they are 7, 13 and 21. Table III shows all experiments done.

IV. Results and Analysis

For each round of experiments, different processing times were observed. For small test and optimization window sizes, the processing time was significantly shorter than those for larger windows. On the average, the processing time spent by a simulation of one year was around 1h:15min in a workstation with Xeon processor E5-2450-v2 2.50GHz and 32GB RAM. Therefore, the time spent to simulate all the dataset for each stock was about 6 hours. To simulate the second option (the complete dataset) a little more than 3 hours were needed.

Table IV presents the results of all experiments divided by stocks (PETR4 and CMIG4), using the complete dataset (refer to 2013/2018 experiments) and, year by year. At line six, where we see a dash ”-”, we can find the sum of the results above - this occurs to PETR4 and CMIG4. The last line of each stock we can check the result of the complete dataset.

Complimentarily, Figures 6 and 7 shows the results as a bar chart. It is possible to observe in both Figures that the proposed method performs better than the classic approach using (7,13,21) MAs. In most cases, the percentage of the accumulative returns on equity overcame the traditional method. In the experiments using PETR4, the proposed method performed better than the classic values. However, for the CMIG4 stock, the proposed method had a zero-to-zero result in one case (2013-2014) and worse in another (2016-2017). However, the proposed method performs better in all the cases using the complete dataset, as shown in Figure 8.

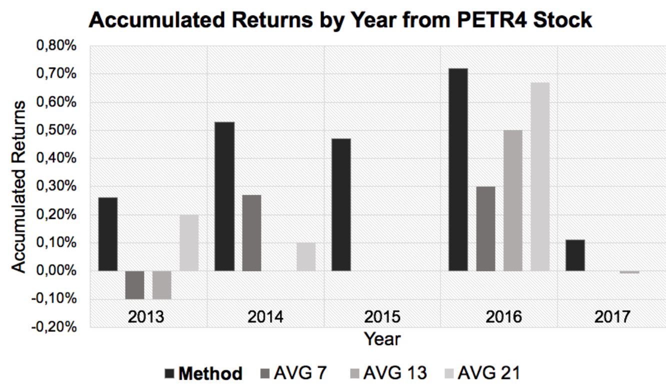

Fig. 6. Detailed results of the experiments performed with PETR4 stocks, shown year by year, compared with classic moving averages.

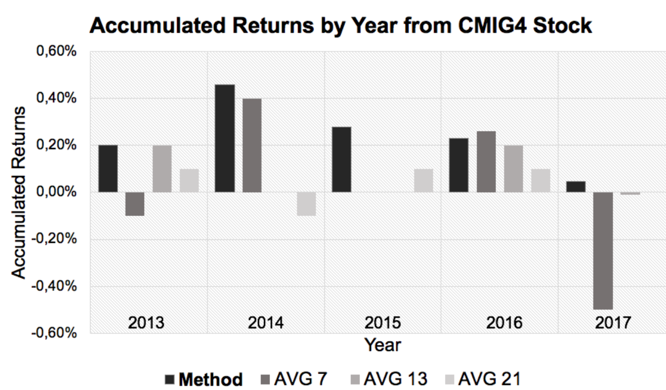

Fig. 7. Detailed results of the experiments performed with CMIG4 stocks, shown year by year, compared with classic moving averages.

It is important to explain that the focus of the paper is to understand if a dynamic training window can improve the algorithmic trading. Therefore we do not compare with the

TABLE III
DETAILS OF THE EXPERIMENTS DONE WITH AG1 AND AG2 FOR FINDING OPTIMIZED CONTROL PARAMETERS.

Experiment	Population	Mutation	Crossover	Selection	Evaluation	Elitism	GReplace	Gaussian
1	8	-	0.7	proportionate selection	80	-	False	False
2	8	-	0.8	rank selection	80	-	False	False
3	8	-	0.9	proportionate selection	80	-	False	False
4	8	0.3	0.7	rank selection	80	-	False	False
5	8	0.3	0.7	tournament selection	80	1	True	False
6	24	0.3	0.7	tournament selection	240	-	False	False
7	24	0.3	0.7	tournament selection	240	1	False	True
8	48	0.3	0.7	tournament selection	600	1	False	False
9	60	0.3	0.7	tournament selection	600	1	True	False
10	8	0.25	0.7	tournament selection	80	1	False	False

TABLE IV
RESULTS OF THE EXPERIMENTS USING STOCKS CMIG4 AND PETR4. *Dataset* IS THE BEGINNING AND ENDING YEAR OF THE DATA USED IN THE EXPERIMENT; *Optimization Win* AND *Trading Win* ARE THE OPTIMIZATION AND TRADING WINDOWS, RESPECTIVELY, CHOSEN BY THE ALGORITHM FOR THE PERIOD SELECTED; IS THE TRADING WINDOW CHOSEN BY THE ALGORITHM FOR THE THE SELECTED PERIOD; *Results* IS THE PERCENT OF THE CUMULATIVE RETURNS ON EQUITY OBTAINED BY THE APPLICATION OF THE ALGORITHM; *Avg 7*, *Avg 13* AND *Avg 21* ARE THE RESULTS OF THE CLASSIC MOVING AVERAGE USING 7, 13 AND 21 PERIODS, RESPECTIVELY.

Stock	Dataset	Opt Win	Trading Win	Results	Avg 7	Avg 13	Avg 21
CMIG4	2013-2014	99	62	0,20%	-0,10%	0,20%	0,10%
CMIG4	2014-2015	57	159	0,46%	0,40%	0,00%	-0,10%
CMIG4	2015-2016	51	84	0,28%	0,00%	0,00%	0,00%
CMIG4	2016-2017	21	108	0,23%	0,26%	0,20%	0,10%
CMIG4	2017-2018	49	172	0,05%	-0,50%	-0,01%	0,00%
-	-	-	-	1,22%	0,06%	0,39%	0,20%
CMIG4	2013-2018	167	159	0,260%	0,00%	0,20%	0,20%
PETR4	2013-2014	147	159	0,26%	-0,10%	-0,10%	0,20%
PETR4	2014-2015	53	54	0,53%	0,27%	0,00%	0,10%
PETR4	2015-2016	34	102	0,47%	0,00%	0,00%	0,00%
PETR4	2016-2017	163	134	0,72%	0,30%	0,50%	0,67%
PETR4	2017-2018	140	171	0,11%	0,00%	-0,01%	0,00%
-	-	-	-	2,09%	0,47%	0,39%	0,97%
PETR4	2013-2018	81	112	1,44%	0,67%	0,59%	0,68%

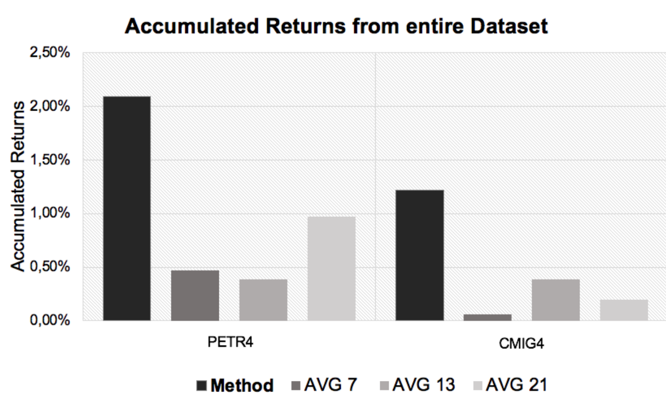

Fig. 8. Detailed results of the accumulated returns for both stocks, PETR4 and CMIG4, regarding the sum of all years, compared with classic moving averages.

"buy and hold" procedure because the proposal is compare with the same method or same trading rules. On the other hand, the low performance of the algorithmic trading build is based on the hypothesis that the strategy chosen is a very basic and poor strategy. According to [20], rules based on moving averages are among the simplest techniques to generate buying and selling signals, and, also, moving average strategies are not recognized by good profits.

Although moving average based trading rules have poor results, the method proposed improved the results of the trading strategy in most cases. A discussion point of this approach is the possibility of the overfitting of the system. However, the method proposed keep the system constantly dynamic. Considering that an optimization window was found, the system always be adapting.

In terms of a practical application, the method could be used for optimizing a real-world strategy with a massive amount of technical indicators. Considering that, currently, some trading systems optimize the parameters of their technical indicators empirically, and others do not optimize parameters at all, this method may be helpful in both cases to find optimal values for their parameters.

V. CONCLUSIONS

In this work, two genetic algorithms working sequentially were used improve the profit of an algorithmic trading system. The proposed method optimizes continuously the parameters of the technical indicators and moving averages, to as to constantly adapt to the market trend and improve the final profit. The performance of the proposed system was evaluated with two stocks of the Brazilian Stock Exchange (BM&F BOVESPA) using a 6-year long database (from 2013 to 2018). After that, a simple trading rule was defined and experiments were conducted using the dataset separated year by year and using the complete dataset.

The proposed method performed better all the cases considering the accumulated returns for all years. The architecture presented is a novel model in terms of the trading optimization. It is important to notice that in all recently published reviews and surveys keep fixed the training window. In some cases, the optimization and trading window are defined as specific number of days or using some empirical approach. In general, we conclude that the proposed method is quite promising and it can be used to improve trading strategies. In addition to the work done, the following items suggest directions for the extension of current work:

- To evaluate the proposed method in more stocks of the Brazilian Stock Exchange.
- To evaluate the proposed method in other markets like, futures market, cryptocurrency and Forex markets.
- To include more advanced indicators and parameters in trading rules to evaluate their impact in profit.

Acknowledgements

Author H.S.Lopes thanks to CNPq for the research grants no. 311778/2016-0 and 423872/2016-8, and Fundao Araucria for the financial support by means of PRONEX 042/2018.

References

[1] R. C. Cavalcante, R. C. Brasileiro, V. L. Souza, J. P. Nobrega, and A. L. Oliveira, "Computational intelligence and financial markets: A survey and future directions," *Expert Systems with Applications*, vol. 55, pp. 194–211, 2016.

[2] C. Slamka, B. Skiera, and M. Spann, "Prediction market performance and market liquidity: A comparison of automated market makers," *IEEE Transactions on Engineering Management*, vol. 60, no. 1, pp. 169–185, Feb 2013.

[3] G. Nuti, M. Mirghaemi, P. Treleaven, and C. Yingsaeree, "Algorithmic trading," *Computer*, vol. 44, no. 11, pp. 61–69, Nov 2011.

[4] Y. Hu, K. Liu, X. Zhang, L. Su, E. Ngai, and M. Liu, "Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review," *Applied Soft Computing*, vol. 36, pp. 534–551, 2015.

[13] M. Radeerom, "Automatic trading system based on genetic algorithm and technical analysis for stock index," *International Journal of Information Processing and Management*, vol. 5, no. 4, p. 124, 2014.

[5] E. Chan, *Quantitative trading: how to build your own algorithmic trading business*. New York: John Wiley & Sons, 2009, vol. 430.

[6] B. Graham, *O Investidor Inteligente*. Rio de Janeiro: Nova Fronteira, 2007.

[7] G. F. Santana, "O poder de previsão da análise técnica: uma aplicação para o mercado futuro de IBOVESPA," Master's thesis, 1997.

[8] M. Ozturk, I. H. Toroslu, and G. Fidan, "Heuristic based trading system on forex data using technical indicator rules," *Applied Soft Computing*, vol. 43, pp. 170–186, 2016.

[9] D. B. Fogel, "What is evolutionary computation?" *IEEE Spectrum*, vol. 37, no. 2, pp. 26, 28–32, 2000.

[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[11] R. Aguilar-Rivera, M. Valenzuela-Rendón, and J. Rodríguez-Ortiz, "Genetic algorithms and darwinian approaches in financial applications: A survey," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7684 – 7697, 2015.

[12] H. S. Lopes, L. C. de Abreu Rodrigues, and M. T. A. Steiner, Eds., *Meta-Heursticas em Pesquisa Operacional*, 1st ed. Curitiba, PR: Omnipax, 2013.

[14] K. Matsui and H. Sato, "A comparison of genotype representations to acquire stock trading strategy using genetic algorithms," in *Proc. of International Conference on Adaptive and Intelligent Systems*, Sept 2009, pp. 129–134.

[15] T.-C. Fu, C.-P. Chung, and F.-L. Chung, "Adopting genetic algorithms for technical analysis and portfolio management," *Computers & Mathematics with Applications*, vol. 66, no. 10, pp. 1743–1757, 2013.

[16] F. A. Badawy, H. Y. Abdelazim, and M. Darwish, "Genetic algorithms for predicting the egyptian stock market," in *Proc. International Conference on Information and Communication Technology*. IEEE, 2005.

[17] J. Straßburg, C. González-Martel, and V. Alexandrov, "Parallel genetic algorithms for stock market trading rules," *Procedia Computer Science*, vol. 9, pp. 1306–1313, 2012.

[18] J. Štěpánek, J. Šťovíček, and R. Cimler, "Application of genetic algorithms in stock market simulation," *Procedia – Social and Behavioral Sciences*, vol. 47, pp. 93–97, 2012.

[19] C.-F. Huang and H.-C. Li, "An evolutionary method for financial forecasting in microscopic high-speed trading environment," *Computational Intelligence and Neuroscience*, no. Article ID 9580815, 2017.

[20] D. J. Bodas-Sagi, P. Fernández-Blanco, J. I. Hidalgo, and F. J. Soltero-Domingo, "A parallel evolutionary algorithm for technical market indicators optimization," *Natural Computing*, vol. 12, no. 2, pp. 195–207, 2013.