A New SAX-GA Methodology Applied to Investment Strategies Optimization

António Canelas
Instituto de Telecomunicações
IST–Torre Norte, AV. Rovisco Pais
1049-001 Lisboa, Portugal
antonio.canelas@ist.utl.pt

Rui Neves
Instituto de Telecomunicações
IST–Torre Norte, AV. Rovisco Pais
1049-001 Lisboa, Portugal
rui.neves@ist.utl.pt

Nuno Horta
Instituto de Telecomunicações
IST–Torre Norte, AV. Rovisco Pais
1049-001 Lisboa, Portugal
nuno.horta@lx.it.pt

ABSTRACT
This paper presents a new computational finance approach, combining a Symbolic Aggregate approXimation (SAX) technique together with an optimization kernel based on genetic algorithms (GA). The SAX representation is used to describe the financial time series, so that, relevant patterns can be efficiently identified. The evolutionary optimization kernel is here used to identify the most relevant patterns and generate investment rules. The proposed approach was tested using real data from S&P500. The achieved results show that the proposed approach outperforms both B&H and other state-of-the-art solutions.

Categories and Subject Descriptors
I.2.M [Artificial Intelligence]: Miscellaneous

General Terms
Algorithms, Performance, Economics, Experimentation.

Keywords
Pattern discovery, frequent patterns, pattern recognition, financial market, time series, genetic algorithm, SAX representation

1. INTRODUCTION
The domain of computational finance has received an increasing attention by people from both finance and intelligent computation domains. The main driving force in the field of computational finance, with application to financial markets, is to define highly profitable and less risky trading strategies. In order to accomplish this main objective, the defined strategies must process large amounts of data which include financial markets time series, fundamental analysis data, technical analysis data, etc. and produce appropriate buy and sell signals for the selected financial market securities. What may appear, at a first glance, as an easy problem is, in fact, a huge and highly complex optimization problem, which cannot be solved analytically. Therefore, this makes the soft computing and in general the intelligent computation domains specially appropriate for addressing the problem.

Recently, several works [8][14][15][16] have been published in the field of computational finance where soft computing methods are used for stock market forecasting, however, due to the complexity of the problem and the lack of generalized solutions this is undoubtedly an open research domain. The use of chart patterns is widely spread among traders as an additional tool for decision making, however, the problem in this case is to say how close enough should the market match a specified chart pattern to make a buy or sell decision.

In this paper a new approach combining a Symbolic Aggregate approXimation (SAX) technique together with an optimization kernel based on genetic algorithms (GA) is presented. The SAX representation is used to describe the financial time series, so that, relevant patterns can be efficiently identified. The evolutionary optimization kernel is here used identify the most relevant patterns and generate investment rules. The proposed approach was tested using real data from S&P500. Finally, the achieved results outperform the existing state-of-the-art solutions.

This paper is organized as follows; In Section 2 the related work is discussed. Section 3 describes the method of dimensional reduction of the time series used in the paper, SAX. Section 4 the proposed approach that puts together the GA and SAX is explained. Section 5 describes the experiments and results. Section 6 draws the conclusions.

2. RELATED WORK
First of all a distinction between pattern recognition and pattern discovery should be made. Recognition identifies known patterns on the time series, this case is a supervised approach, where a library of patterns [3], is created and a search made on the market data, trying to identify them [15]. In pattern discovery the quest is to find new patterns that occur in the time series, in this case, typically, some data segments or windows are compared with others. This case is an unsupervised approach, which is also the case presented on this paper.

Prediction of financial markets has been subject of many studies. In this last few years a combination of algorithms and methods have been used, Table 1. Many of the applications use GA, proving the good results of this type of optimization tool in the financial market world.

In order to create an efficient method of search, the time series should suffer some dimensionality reduction, the method used in this transformation must preserve the key essence of the data. Some of the methods to achieve this goal are the more commonly Discrete Fourier Transform (DFT) [1], Perceptually Important Points (PIP) [4], Piecewise Aggregate Approximation (PAA) [7].
More recently methods of symbolic representation of data and dimensional reduction began to appear, one of this methods is Symbolic Aggregate approxXimation (SAX) [10], which is based on PAA. This algorithm begins to divide the time series in windows, then each window in segments and reduces a set of points in each segment to their arithmetic mean and then converts this value to a symbol. To search patterns the sequences of symbols must be compared with each other to find similarities in the data, in the next section this method will be described in detail.

### 3. SAX METHOD

In order to find patterns, large time series of dimension \( m \) will be break into smaller time series windows of size \( n \ll m \). These windows must be compared with each other, so the characteristics of these time series must be similar, same magnitudes and base line. Therefore to apply this transformation to the windows, data has to be normalized (Eq. 1), this normalization doesn’t affect the original shape [5] and scales the data to the same relative magnitude Figure 1.

\[
x'_i = \frac{x_i - \mu_x}{\sigma_x} \quad (E. 1)
\]

Where \( x_i \) are the points in window \( W_k \), \( \mu_x \) is the mean of of the points in \( W_k \) and \( \sigma_x \) is the standard deviation of all the \( x_i \).

![Figure 1. Normalization process of the stock quote time series](image)

In PAA the time series windows are divided in \( w \) equal size segments and each segment is represented by the arithmetic mean of the points in it, according to Eq. 2.

\[
\bar{x}_j = \bar{x}_{j-1} + \frac{1}{n} \sum_{i=\lfloor w(j-1) + 1 \rfloor}^{\lfloor wj + 1 \rfloor} x'_i, \text{where} \quad \begin{cases} 
    w \rightarrow \text{is the number of segments} \\
    n \rightarrow \text{size of the window} \\
    x'_i \rightarrow \text{the points in the window}
\end{cases} \quad (E. 2)
\]

This equation (Eq.2) is valid if \( n/w \) has an integer result, in this case each point contribute entirely to the frame where is inserted, Figure 2.

![Figure 2. Size 12 window divided in 3 segments, each point contributes to one segment only](image)

In the case of a non integer relation, the point in the frontier between segments must contribute with some part to each of the segments, this method was developed by Li Wey\(^1\), as shown in Figure 3.

![Figure 3. Size 12 window divided in 5 segments, the points between segments contribute to the neighbours segments](image)

---

\(^1\) http://alumni.cs.ucr.edu/~wli/
Based on this method, the time series can now be represented by a smaller dimension set of numbers, Figure 4, where a set of points is now represented by their mean.

![Normalized Stock Quote and PAA](image)

**Figure 4. PAA representation**

After getting the PAA transformation, the amplitude of the time series must be divided into intervals, and to each of them is assigned a symbol. In order to produce equiprobable intervals and since the data has been normalized, a normal distribution curve will be applied to the vertical axis and breakpoints are calculated to produce equal areas under the curve, Figure 5.

![Normalized Stock Quote and PAA](image)

**Figure 5. SAX representation**

Then, each segment is evaluated to determine to which interval belongs. To each PAA level a symbol is assigned to represent that segment. Applying this method to all the segments, and all the windows will generate sequences of symbols, which now represents the time series.

The β’s breakpoints can be obtained from statistical books or like in Table 2 from the Matlab® code obtained in the SAX official web site.

<table>
<thead>
<tr>
<th>a</th>
<th>β₁</th>
<th>β₂</th>
<th>β₃</th>
<th>β₄</th>
<th>β₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>-0.43</td>
<td>0.43</td>
<td>0.67</td>
<td>0.84</td>
<td>0.97</td>
</tr>
<tr>
<td>4</td>
<td>-0.67</td>
<td>0</td>
<td>0.25</td>
<td>0.43</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>-0.84</td>
<td>-0.25</td>
<td>0</td>
<td>0.43</td>
<td>0.97</td>
</tr>
<tr>
<td>6</td>
<td>-0.97</td>
<td>-0.43</td>
<td>0</td>
<td>0.43</td>
<td>0.97</td>
</tr>
</tbody>
</table>

*Table 2. Breakpoints vs. a divisions*

Now, to discover new patterns the SAX sequences of symbols must be compared with each other or compared to a known sequence to find some wanted pattern. For the match between sequences it will be used Eq. 3 [10], to evaluate the distance between sequences P and Q, and reveal the degree of similitude between them, Figure 6.

\[
\text{MINDIST}(P, Q) = \sqrt{\frac{1}{T} \sum_{i=1}^{W} (\text{dist}(p_i, q_i))^2} \quad (\text{Eq. 3})
\]

Where \(\text{dist}()\) is a function defined as:

\[
\text{dist}(p_i, q_j) = \begin{cases} 
0 & |i - j| \leq 1 \\
\beta_{j+1} - \beta_i & i < j - 1 \\
\beta_{j-1} - \beta_j & i > j + 1 
\end{cases}
\]

The β’s are the breakpoints defined in Table 2.

![Distance between sequence P and Q](image)

**Figure 6. Distance between sequence P and Q**

In summary, the parameters that affect the SAX discretization of a time series, are the number of intervals that divides the normal curve and corresponds to the alphabet size of symbols, the window size and the word size that represents the time series in the window and generates the sequences.

4. APPROACH

The discovery of meaningful patterns was the objective of this work, our aim was not only to identify patterns or sequences that repeat over time, but also to create application rules of those patterns.

Trying to identify patterns in SAX, could be compared to make a search on a space of solutions, considering that this space can be rather large, the use of genetic algorithm was an obvious choice.

4.1 Estimation of SAX Parameters

In the final of section 3 the parameters that affect SAX were identified as the windows size \(n\), word size \(w\) and the alphabet size \(a\). To get the best values for these parameters it was selected a financial time series from an S&P500 stock with almost 3,100 points in the period of 1998 to 2010, and made an exhaustive search of patterns with several combinations of values. In our tests the parameters take integer values in the following intervals:

\[
\begin{align*}
n &\in [10 \ldots 150] \\
w &\in [2 \ldots n/2] \\
a &\in [2 \ldots 20]
\end{align*}
\]

2 http://www.cs.ucr.edu/~eamonn/SAX.htm
For the test it was developed a new application on C++ that converts the time series into SAX sequences using the combinations of values defined previously. For each combination, the number of different patterns detected and the number of occurrences of those patterns in the time series was evaluated. In this stage, the patterns are sequences of symbols exactly equal, by the definition of distance given in the SAX section, the distance between sequences could be zero even if they aren’t exactly the same, but here only patterns with the same sequence of symbols were considered. This will simplify and optimize the application, since the patterns are save as the key on a map associative container, this makes it easy to identify patterns without having to calculate distances between them.

After running the several combinations, the data was analyzed in Matlab®.

In Figure 7 is a representation of the number of different patterns found, it is possible to identify (inside the ellipse/red color) areas where the parameters tested reveal a larger number of patterns. The ellipse/red color indicates a higher number of patterns present. It is possible to verify, by the scatter points in dark blue, that patterns with bigger window size and word size only exists with small alphabet size, this makes sense since complex and longer patterns should be more difficult to find.

After running the several combinations, the data was analyzed in Matlab®.

In Figure 7 is a representation of the number of different patterns found, it is possible to identify (inside the ellipse/red color) areas where the parameters tested reveal a larger number of patterns. The ellipse/red color indicates a higher number of patterns present. It is possible to verify, by the scatter points in dark blue, that patterns with bigger window size and word size only exists with small alphabet size, this makes sense since complex and longer patterns should be more difficult to find.

The maximum number of different patterns identified in Figure 7 was 597, the point where the value is obtained:

\[ p_1 = \begin{cases} n = 17 \\ w = 4 \\ \alpha = 13 \end{cases} \]

This point will be tested at the search for patterns with the GA, it is reasonable to think that in an area with a large number of different patterns some of them will be important solutions to our problem.

Another analysis made to the data is the total number of occurrences of the patterns found for each combination of parameters, in Figure 8 a surface representing this fact is shown, for this analysis the maximum values was 2898, at the point:

\[ p_2 = \begin{cases} n = 10 \\ w = 3 \\ \alpha = 18 \end{cases} \]

The point \( p_2 \) needs to be explored, since areas where a large number of patterns occur are probable that relevant solutions appear.

A third analysis was made, this one combines the first two, here is studied the “importance” of the patterns for some combination of parameters, if a low number of patterns appear for a large number of occurrences is probably more significant than having many patterns with a small number of occurrences, the surface for this analysis is on Figure 9.

In this last figure was identified the value 239.4, for the next parameters:

\[ p_3 = \begin{cases} n = 10 \\ w = 2 \\ \alpha = 7 \end{cases} \]

This last point doesn’t appear to be very relevant as patterns parameters, because patterns with a two letter word probably will not be very important.

Several more time series were subject to the same exhaustive search, and the results were quite similar, the same areas were identified. From the values of this three points is possible to conclude that areas with small words and large alphabets, are...
probably important for pattern discovery. This doesn’t exclude the test and exploration of other areas in the solution space, because this test only identifies areas of high intensity and it may exists important patterns in areas of fewer density patterns.

4.2 Pattern Discovery

The goal is to identify patterns and application rules for those patterns. As the algorithm analyzes the financial time series, with the use of a sliding window of size defined by the SAX parameter, it will generate buying orders when the pattern is present and sell orders when it’s not or if the number of days in the buying position exceeds some threshold. So based on this description, the genetic algorithm should produce patterns and detect if they are present on the time series. Since the SAX representation is used, the patterns are sequences of symbols and the distance from the pattern to the time series must be calculated to identify their presence. This fact brings the need to find how close the time series should be to the pattern, in order to justify the buying decision, as well the algorithm must identify how far should be to issues a sell order. Our GA uses two distance measures, the first is the one presented in section 3, the second takes advantage of the discrete symbolic representation and how far the symbols are from each other, and is expected that avoids the trivial matches identified in [11], as can be seen in Eq.4

\[
\text{dist} = \sum_{i=1}^{w} (T_i - P_i)^2 \quad \text{(Eq. 4)}
\]

\[
\begin{align*}
\w & \rightarrow \text{Word size} \\
T_i & \rightarrow \text{Symbol i of the time series} \\
P_i & \rightarrow \text{Symbol i of the pattern}
\end{align*}
\]

An example of this measure is in Figure 10, and takes advantage of the possibility that C++ can subtract char data type. This method is faster than the standard MINDIST (Eq.3), since the operation doesn’t need to check the breakpoints table to do the calculations and the GA will adapt to this type of measure.

![Figure 10. Example of a distance calculation, based on the discrete symbolic representation](image1)

Based on the previous definition, of how the GA should behave, the chromosome presented in Figure 11 will be used in our population.

![Figure 11. Chromosome used in the GA](image2)

The chromosome is divided in two major parts, the first one are the parameters that support the decisions of buy and sell, here are the two distances from the time series to the pattern, that permits to evaluate if the pattern is present in order to buy (\(\text{dist. buy}\)) or if its effect is no longer present and it’s time to sell (\(\text{dist. sell}\)), another gene defines after how many days should the algorithm sell if it is in a buying position (\(\text{days sell}\)), the final gene of this part is a bit that identifies which of the measures should be used to evaluate the distances (\(\text{measure type}\)). The last part of the chromosome, are the symbols that constitute the pattern sequence (\(P_{1\ldots w}\)).

The selection process used is a random selection applied to the best half of the population and then uses a two point crossover to generate the offspring’s. The option for two points instead of a single point was made because of the structure of the chromosome, where the first point cuts the chromosome in the section of the rule parameters and the second in the pattern symbols area. A multi point crossover was also studied but the chromosome length is small, the term that could increase this measure is the word size, but as has been seen in section 4.1 this parameter has values around 3, so the chromosome has a total length of 8 genes. The generation of new population will be elitist, the best chromosomes will be preserved, and these elements will be randomly chosen to generate the new population. The mutation rate is of 10\%, in tests made in another study on pattern match in financial markets [15] proved good results with this value.

The fitness function that the GA will optimize is the total earnings produced by the investment strategy defined by the pattern and application rule associated with it.

The program will slide a window along the time series and converts it to a SAX sequence. The patterns in the chromosomes will them be compared with each window sequence to calculate the distance and apply the rules defined in the chromosome, the distance to buy or sell, Figure 12.
If the distance is less than the ‘Distance to buy’ defined in the chromosome the application will buy the stock. In case the stock has already been bought, the application sells the stock if the distance is bigger than the ‘Distance to Sell’ gene or if the stock was bought more days ago than the specified by the ‘Days to Sell’ gene. At the end of the time series a new epoch begins and the process restarts with a new population that includes the best individuals and the new offspring’s. The stop criteria used is the end of improvement in the fitness function for several generations.

5. EXPERIMENTS & RESULTS
In this section two case studies are presented, the first analysing all the stocks independently, the second trying to find the most important pattern for all the stocks. The application was tested in real market conditions. Data was retrieved from 1998 to 2010 for 99 stocks from the S&P500 index and the index itself. The chosen time span is large enough to embark a bull and a bear market. Particularly, the training and testing period were, respectively, from 1998 to 2004 and from 2005 to April, 21 of 2010. The reason to test the mentioned period was to include the instabilities and crisis of the year 2008 and beyond, to test our algorithm on the worst market conditions of the last years. The transactions costs were consider in all financial transactions to provide a result as close to a real scenario.

5.1 Case Study I
In this case study the stocks are being considered individually in order to identify patterns that occur on each financial time series. The tool will treat each stock separately and discover patterns and investment rules, the GA parameters are a population of 500 elements and the number of generations used in the stop criteria is 50. The tests are repeated for 10 runs and the SAX parameters used are the ones identified in section 4.1, other combination of SAX parameters were also tested in order to explore other areas of the solution space.

All the results are compared to the Buy and Hold (B&H) investment strategy, which is widely used as reference, based on the efficiency of the markets [2]. The three points detected as regions of interest in the solution space, see section 4.1, were evaluated for the index S&P 500, which is an important indicator of the market behavior. The results of all runs and of the best one, on this financial asset are presented in Table 3.

Table 3. Results for S&P500

<table>
<thead>
<tr>
<th>Point</th>
<th>GA+SAX Return (%)</th>
<th>Buy&amp;Hold Return (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Averaged 10 runs</td>
<td>Best Run</td>
</tr>
<tr>
<td></td>
<td>Period</td>
<td>Annual</td>
</tr>
<tr>
<td>P1</td>
<td>3.6</td>
<td>0.67</td>
</tr>
<tr>
<td>P2</td>
<td>-0.48</td>
<td>-0.09</td>
</tr>
<tr>
<td>P3</td>
<td>37.6</td>
<td>6.21</td>
</tr>
</tbody>
</table>

By the previous results, it is interesting to see that the combination of SAX parameters that give best results is for a pattern of only two word symbol. In all the runs for this point P3, the patterns the GA identifies are descendent like FB, FA, EB or DB, and the Days to Sell term is 11 for all the runs. So the algorithm tries to find minimum prices to buy and shortly sells them, after 11 days or if the stock price goes quickly apart from the pattern and this occurs when rapid up movement of the price manifests. As can be seen in Figure 13, in the magnified area this fact manifests. The problem is that strategy only works well on uptrends.

Figure 13. S&P500 investment return compared with B&H

In order to identify other strategies that can work well in downtrends and uptrends the rest of stocks were analyzed. It was also explored other areas of the solution space with different and more complex patterns. For example with a “Window Size” of 92, an “Alphabet Size” of 6 and a “Word Size” of 9, the GA found a pattern that gives good results in the “Alcoa Inc.” stock, Figure 14.

Figure 14. Pattern and Investment strategy found in Alcoa stock by our GA

In Figure 14 it is possible to observe how the algorithm avoids big drops, and saves the investment from the fall in 2008. The pattern found and used by GA is ‘DBDCEFBD’ and could be represented by Figure 15.

Figure 15. Pattern found by GA in the Alcoa investment test
The results presented in Table 4 are the average earnings of the investment strategies for the best chromosomes of all the stocks, compared to the B&H.

Table 4. Average return of all stocks investment strategies.

<table>
<thead>
<tr>
<th>SAX Parameters</th>
<th>GA+SAX Return (%)</th>
<th>Period</th>
<th>Annual</th>
<th>Buy&amp;Hold Return (%)</th>
<th>Period</th>
<th>Annual</th>
</tr>
</thead>
<tbody>
<tr>
<td>n a w</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 7 2</td>
<td>69.30</td>
<td>10.44</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 18 3</td>
<td>117.82</td>
<td>15.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17 13 4</td>
<td>115.18</td>
<td>15.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60 13 4</td>
<td>113.28</td>
<td>15.36</td>
<td></td>
<td>48.80</td>
<td>7.79</td>
<td></td>
</tr>
<tr>
<td>92 6 9</td>
<td>120.49</td>
<td>16.09</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>104 6 8</td>
<td>115.24</td>
<td>15.56</td>
<td></td>
<td>1061</td>
<td></td>
<td></td>
</tr>
<tr>
<td>128 7 8</td>
<td>111.43</td>
<td>15.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>136 5 12</td>
<td>122.39</td>
<td>16.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From Table 4 it is possible to verify that areas of SAX parameters, others than the detected in section 4.1, are valid and prove to get better results. As expected the pattern search for a two symbol word gave the worst results on average.

Another study, concerning the measure types used to determine the degree of similitude between the time series and the pattern, was made. To find which distance measure is used with better result in the investment strategies were analyzed only the best runs of the GA, the results are presented in Table 5.

Table 5. Measure type analysis

<table>
<thead>
<tr>
<th>Measure Type</th>
<th>Occurrences</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINDIST (Eq.3)</td>
<td>97</td>
<td>69.8%</td>
</tr>
<tr>
<td>ALPHAB. DIST. (Eq.4)</td>
<td>42</td>
<td>30.2%</td>
</tr>
</tbody>
</table>

From Table 5 is possible to verify that the method chosen by the GA to find the best investment strategy was the distance measure defined by the SAX method, which is a lower bounding of the Euclidian distance of the time series and the pattern. The sum of occurrences in the result table is bigger than 100, the number of stocks, because more than one run has the same result.

5.2 Case Study II

As was said in the beginning of the section 5, the first study treats the stock separately and tries to identify patterns and investment rules for each of them. In order to identify a global pattern and rule, it was feed to the GA all the stocks at once for the algorithm training. The objective was to identify a pattern and rule that could be applied to any stock.

This case study begins by the analysis of the alternative distance measure used by the GA to find the best solutions. The results are presented in Table 6

Table 6. Measure type analysis

<table>
<thead>
<tr>
<th>Measure Type</th>
<th>Occurrences</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINDIST (Eq.3)</td>
<td>94</td>
<td>77%</td>
</tr>
<tr>
<td>ALPHAB. DIST. (Eq.4)</td>
<td>28</td>
<td>23%</td>
</tr>
</tbody>
</table>

In Table 6 is possible to verify that the measure select by the GA was, once more, the MINDIST (Eq.3), which gives better results.

After the training period the best pattern found to invest on all the stock is in Figure 16.

Figure 16. Representation of the best global pattern found

This pattern provides the best returns on average for all the stocks, the SAX parameters associated with it are “Window Size” of 128, an “Alphabet Size” of 5 and a “Word Size” of 8.

For the presented pattern a portfolio of all the stocks was made, and the investment strategy was applied to this set of stocks. The results are presented in Figure 17. In this graphic is possible to see that the pattern takes profit by avoid the big dropdowns, in the more stable uptrend the results are below the B&H. This fact is explained by the fact that the algorithm will sell its position every 88 days, “Days to Sell” gene, and the transaction costs will reduce the profit.

Figure 17. Investment return of the portfolio

Table 7 presents the average best results of this investment strategy for the points of section 4.1.

Table 7. Results of global investment strategy

<table>
<thead>
<tr>
<th>SAX Points</th>
<th>GA+SAX Return(%)</th>
<th>Buy&amp;Hold Return (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Annual</td>
<td>Period</td>
</tr>
<tr>
<td>P1</td>
<td>51.44</td>
<td>8.15</td>
</tr>
<tr>
<td>P2</td>
<td>52.84</td>
<td>8.33</td>
</tr>
<tr>
<td>P3</td>
<td>52.09</td>
<td>8.23</td>
</tr>
</tbody>
</table>

The results from this case study are not as good as the ones from the previous case, this was expected because, here, the goal was to find a global rule and pattern that will suite to all the stocks, but the method still finds better solutions than B&H.
6. CONCLUSIONS
The methodology that was presented here has great potential on investment markets. The GA proves its adaptability to find good solutions to the problem at hand, this was possible thanks to the use of the SAX representation that is capable of dimensional reduction of the time series and still maintains the principal characteristics of the financial data.

From the two case studies was identified that the investment strategy to apply is the one that analyses the stocks independently, this was in part expected since the stocks behave in different ways to the conditions of the market and it is very difficult to find a global investment strategy.

A future evolution of this study is to include the SAX parameters in the GA, to identify the best ones to represent financial data. From the results of Case Study I is possible to conclude that areas with parameter values different from the ones found in section 4.1, produce better results. The limitation of the parameters to areas of high density of patterns reduces the search space, and the fact that better solutions have been found outside these areas proves that more frequent patterns aren’t as meaningful as expected.

7. REFERENCES


