Genetic optimization of a trading algorithm based on pattern recognition

Riemann Ruiz-Cruz, Chelsi Sedano and Oscar Flores
Research Laboratory on Optimal Design, Devices and Advanced Materials -OPTIMA-, Department of Mathematics and Physics
ITESO, Periférico Sur Manuel Gómez Morín 8585
Tlaquepaque, Jalisco, México
riemannruiz@iteso.mx, if714509@iteso.mx, if715029@iteso.mx

***Abstract*—In the present paper, a trading strategy based on pattern recognition is optimized by means of a genetic algorithm. The genetic algorithm is used to determine decisions of buy/sell based on the patterns found through time for a portfolio in the stock market. The predominant algorithms used in this work were the K-means clustering algorithm to find the patterns in different time lapses, and the genetic algorithm for optimization. The results are supported by simulations using a selected shares of the Mexican stock market.**

I. Introduction

The stock market and trading operations have changed significantly in the last decades. The trading process has been automatized by the means of electronic systems and it keeps evolving. This has made it possible for trading operations to be carried out with more precision and taking advantage of market opportunities [1].

For a long time, the fuzzy logic modeling, the mathematical modeling and the artificial intelligence are a common part of decision making in a portfolio optimization approach for financial engineering processes. In [2], a forecasting model is proposed which is used to estimate the dynamics of international trade. The proposed algorithm is used to estimate, but still depend on an expert to perform operations in the stock market. In [3], a genetic algorithm based fuzzy neural network to formulate the knowledge base of fuzzy inference rules which can measure the qualitative effect on the stock market. The neural network is used to determine the buying-selling points. In [4], the support vector machine algorithm is used to estimate the the upper bound and lower bound of the $S\&P500$ index. The estimation was used to make trades when different stock movement patterns appears, and the proposed algorithm show a promising performance compared to the buy-and-hold strategy.

The high-frequency trading has become a dominant for in the U.S. capital market, accounting for over 70% of the dollar trading volume [5]. The implementation of this technique is complicated for investors who do not have the ability to access high-performance technology.

In [6], a portfolio model is proposed which allows simulations of the purchase and sale operations taking into account the operating commissions. In order to simplify the algorithm design, the portfolio model in [6] is used in this work taking into account that our portfolio is formed by a single asset.

It is well known that clustering algorithms have shown a very good capacity for pattern detection in data sets. The K-means clustering algorithm for pattern finding in any time series is a good approach for getting information about any stock at any time. In [7], the K-Means algorithm is used to detect the most representative patterns in a time series making comparisons between time intervals within the same series. Since getting deeper information of the shares is the basis for getting a functional trading algorithm, K-Means clusters must give sufficient information for the algorithm to know when to trade. Markov chain and fuzzy inference system was part of the modeling proposed by [7], where the Markov process estimates the price for future time, and a decision is made by expert trader provided rules. This model provides a way to make the inference of the action that must be performed based on the most likely price pattern in the future. The main disadvantage is that it is required to have knowledge of an trader who provides the rules to carry out the buying and selling operations. Different from the above perspective, since an expert trader may be wrong sometimes as well, the main focus of this research is based on searching the best decision once there it is a pattern identified by the algorithm. Markov chains could be useful for decision making, but since; it is based on historical data, it can be only estimated a short time in the future, and the portfolio value can be extended indeterminately; then it is decided not to use them.

In this paper, a trading strategy is proposed for a portfolio composed of shares in the stock exchange. The proposed strategy is based on the K-Means algorithm to determine patterns in a time series of stock market prices. In order to avoid the dependence on an expert trader that defines the best actions to be taken in each given situation, a heuristic optimization algorithm is proposed to determine the best action in presence of an investment opportunity determined by the clustering algorithm. The solutions obtained from the heuristic algorithm can be considered as recommendations from an expert trader. The performance of the trading algorithm with real prices of the Mexican stock exchange is validated through simulations.

The present paper is organized as follows: in section II, the mathematical preliminaries are included. In section III, the trading algorithm is obtained using a clustering algorithm optimized with a genetic algorithm. The simulation results

of the proposed algorithm are shown in section IV. Finally, section V presents the conclusions of the paper.

II. Mathematical preliminaries

In this section the mathematical preliminaries are presented. The proposed trading algorithm is built based on a clustering algorithm (K-means) which is used to detect patterns in a time interval into a time series; additionally, a trading operation model is used to simulate the buy and sell transactions in function of the pattern detected. The trading decisions in a situation determined are obtained by means a heuristic optimization algorithm.

A. K-Means

The clustering algorithm K-Means is a method widely used to automatically find clusters by means the data set partition into K groups[8]. The clusters created can be considered as the predominant patterns in the data set. The algorithm consists of grouping the elements of a data set based on their proximity to a cluster centroid that is initially chosen randomly.

Given a data set with n observations as m-dimensional real vectors, the K-means clustering groups the n observations in K partitions to minimize the within-cluster sum of square distances. Then, the objective is to find

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{K} \sum_{x \in S_i} \|x - \mu_i\|^2 \tag{1}$$

where, S_i is a partition of the K partitions ($\mathbf{S} = \{S_1, S_2, \ldots, S_K\}$), μ_i is the centroid of the observations in S_i[9].

Given an initial set of K means or centroids $\{\mu_1, \mu_2, \ldots, \mu_K\}$, the algorithm proceeds iteratively in two steps[10]:

- Assignment step: Assign each observation to the cluster whose mean has the least squared Euclidean distance, this is intuitively the "nearest" mean. $S_i^{(t)} = \left\{x_p : \|x_p - \mu_i^{(t)}\|^2 \leq \|x_p - \mu_j^{(t)}\|^2 \forall j, 1 \leq j \leq k\right\}$, where each x_p is assigned to exactly one $S^{(t)}$, even if it could be assigned to two or more of them.
- Update step:Calculate the new centroids of the observations in the new clusters. $\mu_i = \bar{x}, x \in S_i$.

When a certain number of iterations have occurred, the clusters are formed by the observations that belong to the last partitions obtained. The algorithm must be iterated a sufficient number of times until the obtained clusters do not change between iterations.

B. Trading model

A investment portfolio is a collection of company shares and other investments that are owned by a particular person or organization[11]. Managed by professionals, portfolios use to have certain assets depending on the main objective of the client; there might be arbitrageurs (which are defined as those who have returns without any risk), speculators (those who pretend to earn money based on the held assets) and hedgers (who are focused on reducing risks rather than earning money by the financial assets). The resulting strategies are meant to be for speculators, therefore risk may be assumed expecting to have interests in return.

There is a proposed model for each asset in a portfolio and the money invested in it, this trading model is defined as follows [6]:

$$\begin{aligned} v_{p,k} &= p_k n_{a,k} + m_k, \\ m_{k+1} &= m_k - p_k u_k - r_{com} p_k \|u_k\|, \\ n_{a,k+1} &= n_{a,k} + u_k, \end{aligned} \tag{2}$$

where $v_{p,k}$ is the present value, m_k is the available money, $n_{a,k}$ is the number of held assets (shares), r_{com} is the charged commission for every transaction, u_k is the number of shares to buy/sell in any given step k, and p_k is the stock price of the corresponding asset.

The model considers that short sales can not be made and leveraged transactions are not possible either, then the following limitations for control signal u_k must be considered:

$$u_{min} \leq u_k \leq u_{max}, \tag{3}$$

where, the control signal limits are defined as $u_{max} = floor\left(\frac{m_k}{(1+r_{com})p_k}\right)$ and $u_{min} = -n_{a,k}$. Thus, you can not sell more than you have and you can not buy more than what money allows.

C. Genetic algorithm

As is well known, the genetic algorithms (GAs) is an robust and efficient optimization technique which is inspired by the way the genetics of the species works. These algorithms are very effective for complex optimization problems with a large search space[12], [13], [14].

Essentially, a genetic algorithm proposes a number n of possible solutions x_i, with $i = 1, 2, \ldots, n$, and the population is composed for all these solutions. The entire population is evaluated to determine which of these possible solutions are the best based on a desired objective function or fitness function $F(x_i)$. The best solutions are considered the *Parents* who will move on to the next generation and these will generate new possible solutions known as *children*.This process is iterative until the optimization objective has been achieved.

The GA algorithm is composed by the following steps or modules[12]:

- An encoding strategy that determines the way in which potential solutions will be represented to form the chromosomes.
- A population of chromosomes or individuals.
- Mechanism (fitness function) for evaluating each chromosome.
- Selection/reproduction procedure.
- Genetic operators like crossover and mutation.
- Probabilities of performing genetic operators.
- Some termination criterion.

The flowchart of a general GA algorithm is presented in the Fig. 1.

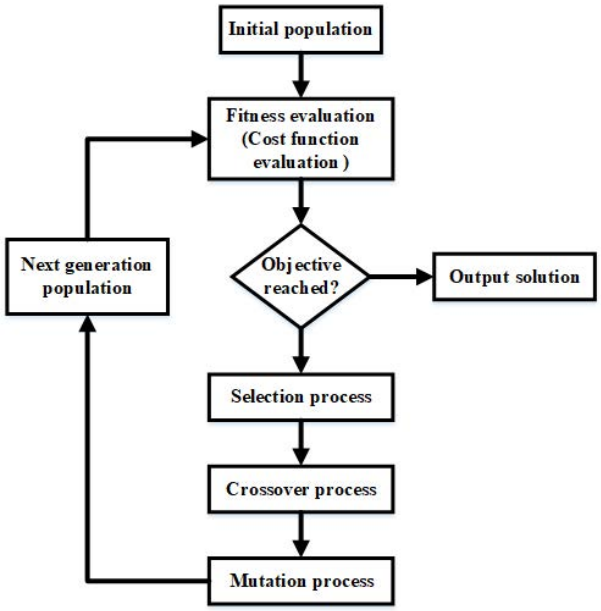

Fig. 1. Flowchart of the GA algorithm

III. Trading algorithm design

The development of the proposed algorithm in this paper is composed of 4 main phases, which are described in detail in this section.

A. Data selection

The initial part of this project was to identify a set of data information that lets us visualize its behavior through time. Due to the situation, it was determined that the BMV's (Mexican Stock Exchange, by its Spanish acronym) information could be useful in this investigation. By examining the principal BMV indexes it was found that the $S\&P/BMV\ IPC2$ (most commonly known as "IPC") was the most dominant for it represents the top 35 enterprises in the Mexican market .

It was decided to take an arbitrary sample from the IPC that represents the index itself. The chosen enterprises were:

- AC (Arca Continental)
- ALFAA (Alfa)
- ALPEKA (Alpek)
- ALSEA (Alsea)
- ELEKTRA (Grupo Elektra)
- IENOVA (Infraestructura Energética NOVA)
- MEXCHEM (Mexichem)
- PE&OLES (Industrias Peñoles)
- PIMFRA (Promotora y operadora de infraestructura)
- WALMEX (Wal-mart de México)

B. K-means clustering

In order to identify the 'common' patterns in any given time, it was needed to have certain groups (clusters) predefined. K-means works by clustering elements based on their euclidean distance (every single element must be quantitative). For pattern recognition it was required to standardize the elements.

In financial technical analysis is common to take a time lapses of one week, one and two months, and half-year. Then, in the K-means clustering there were used four nubs of time lapse; 5 days, 20 days, 40 days and 125 days. Considering that the BMV is only open at business days; the 5 day nub gets in representation of a week, 20 days as a month, 40 days as two months, and 125 days as a half-year.

Four clusters were used for every single nub of time. Each of these clusters representing the immediate patterns; up, down, down-up, and up-down movements.

C. Decision Making

There are 4 possible patterns for 5 days, 4 for 20 days, 4 for 40 days, and 4 for 125 days (fig. 2). Let us say a 'case' is set of circumstances formed by one pattern of 5 days, 20 days, 40 days and 125 days, therefore, there exist 4^4 possible cases, which equals to 256 possible cases.

Given any case, there are three 'decisions' that can be executed; buy, sell or do-nothing. A set of 'choices' would contain 256 decisions (one per case), every single decision is assumed to be independent from the others. Thus, there exist $3^{256} \approx 1.39 \times 10^{122}$ possible sets of choices that cannot be simulated and tested by brute force due to computational constraints. Then, a heuristic optimization is proposed to explore the set of choices to find the best decisions' set.

D. Genetic Algorithm

If every case is recognized, then, any set of choices can be simulated. Initially, having 64 random sets of choices, each of them is simulated and qualified by the cost function

$$J_{max} = \sum_{i=1}^{n} (r_i) \tag{4}$$

$$r_i = \frac{p_i}{p_{i-1}} - 1 \tag{5}$$

where r is the daily return of each simulation i, which in turn defined as (5). The 16 best qualifications are chosen to be the *Parents* of the Genetic Algorithm (Selection Process).

For the creation of every child of the next generation 256 decisions must be made . Each decision includes another random decision from any of the 16 available *Parents*. The result of this process (Crossover process) is a new child which may contain information of all the 16 *Parents*. 64 new children are generated with the 16 *Parents* in order to reach the number of the population.

In order to have not-decaying generations, mutation process may appear. For mutation, 64 random decisions out of the 256 of each child are chosen to be modified. Each one of these chosen decisions will mutate and be overwritten by a new random decision. The outcome of the mutation process is a set of 64 children, where every child is expected to be different to any *Parent*.

By the time 64 new-crossovered-mutated children are generated, they have to be simulated and qualified by the previously defined cost function (4). If a child gets a better qualification than a *Parent* it will substitute it, having as

Parents the 16 best qualified out of the 80 family members (*Parents* + *children*). Selection, crossover and mutation processes are repeated until 300 iterations are completed.

IV. Results

A. K-means

In the clustering process there were four 'common' results for each period of time (5 days, 20 days, 40 days and 125 days);

- constantly upwards
- constantly downwards
- Up and down
- Down and up

With any new price, a new time series can be formed in many different periods of time with the previously existing prices and the one added. The trained K-means model is used to classify time series, so; the new short periods of 5-days will belong to one of the 5 day clusters listed in figure 2. The 20 days time series will belong to any of the 20 day clusters, and so until any added time series belongs to one of the clusters mentioned before.

B. Genetic

In Figure 3, the evolution of the 16 *Parents* are shown. The 16 *Parents* are qualified by their daily average return (equation 4). At the beginning, it can be observed that the evolution process is fast, but at a certain point the qualification of the *Parents* begins to converge in a logarithmic-like shape. After 300 generations, the daily performance reached by the best trained *Parent* was 0.000367 and the worst reached 0.000327, so the difference between them is about 12.23%.

The 16 best *Parents* and an *Average Parent* were simulated to see how much each one earned in the unknown time, Figure 5. The *Average Parent* is made out of the rounded average for each decision from the sixteen *Parents*. In Figure 4, in the simulations of the WALMEX can be appreciated ; that the ones before the vertical black line are tested in the known prices' period and after the line, the unknown prices'. For this share in the best case the earnings are 13% and in the worst case, losses can be up to 8.8% for the unknown time. In this share there is a benefit behavior of the *Average Parents* but in some other shares there is a poor performance.

The figure 6 represents the results of the simulations of the outcome *Parents* in the time where the algorithm was not trained (last 62 business days). The column *MAX* refers to the highest return of all *Parents*, the columns *MIN* contains the lowest return of all the *Parents*, the *AVG* column is the average of the returns of the 16 *Parents*, and the *MAX_F* column represents the result of the simulated *Average Parent*. Note that the decisions in the enterprises **ALPEKA** and **ELEKTRA** over-performs with almost all of the *Parents*; the average of the returns is above the 10%, the maximum for both is above the 15%, and the minimum is near to 0% for both cases. In the other hand, the enterprises **ALSEA** and **MEXCHEM** highly under-performs with the decisions taken. Another interesting result is **PE&OLES**; The max is highly over the mean, and the min is highly under the mean .The rest of the enterprises simulations could be considered as 'normal', for the average, min and max are similar to each other.

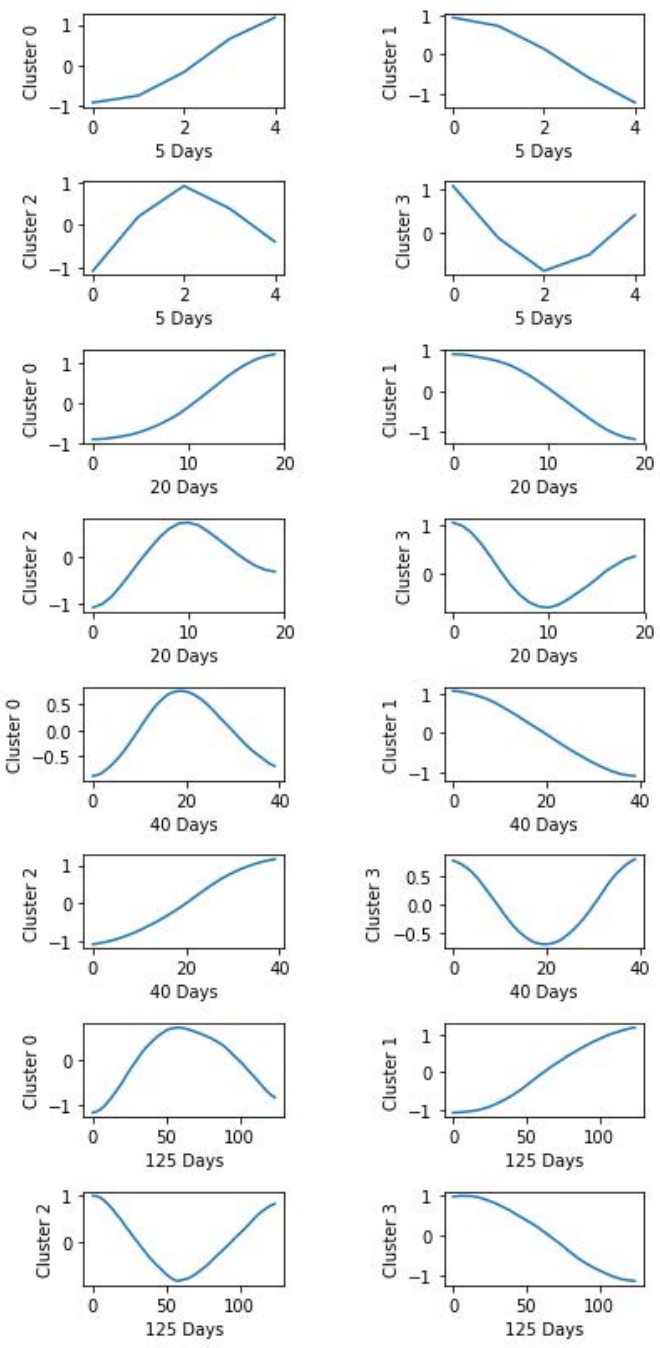

Fig. 2. K-means clusters

Considering that any quantity of money is invested in this *Parent's* strategy, the return is -0.15%. If there were considered just the best *Parents* there it is a return of 9.47% in three months, but if there where considered just the worst *Parents* a return of -8.94% is obtained. For the whole strategy the *Average Parent* decisions were not the best.

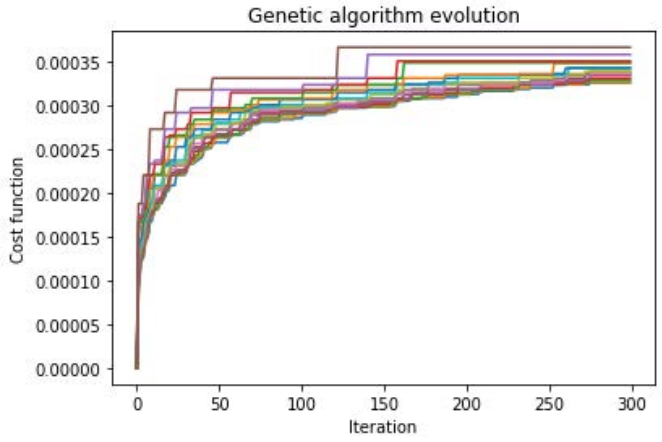

Fig. 3. Evolution of the *Parents* during the optimization process

V. Conclusions

The K-means clustering is useful for identifying patterns in time series. Clusters combined with other mathematician and computing engineering gets to promising results. There are good trading strategies (*Parents*), but there are also losing strategies. The problem is to identify which are the good working *Parents*, so that for every time series there is an appropriate decision that leads to good returns.

The trading strategies consider the commission per transaction, which is closer to the real trading conditions. It is assumed that the last known price is the spot price, therefore, every buy/sell will be executed at the last known price. There are many heuristic assumptions such as; number of *Parents*, iterations, crossover, mutation, nubs of days, time periods, number of clusters, centroids per cluster, etc.

The following research topic should consider different nubs and time periods in order to find better pattern identification and therefore, better results. The fact that the given price is not the same as the buy/sell price must be considered . For future research the possibility of 'shorting' in actions should be added. The invested amount of money in each share is another important factor to consider, due to the different situations of any single share and their possibilities to earn money. It could also be considered a penalizing variable for the volatility of the daily returns.

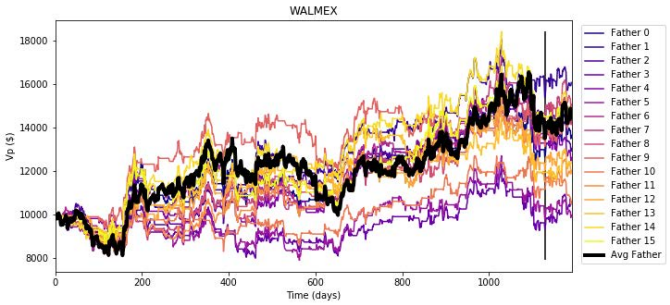

Fig. 4. Trading simulation of the *Parents* with real prices stock market

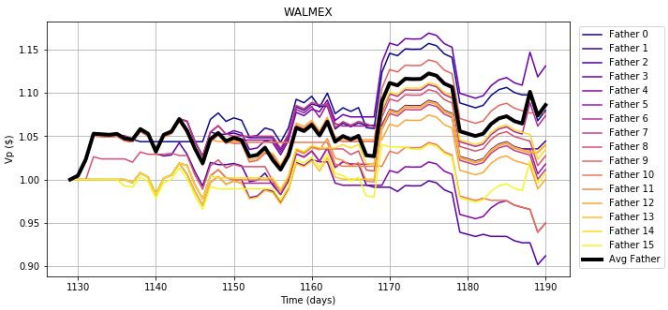

Fig. 5. Trading simulation algorithm with unknown prices

	MAX	MIN	AVG	AVG_F
AC	0.07094	-0.08694	-0.00558	-0.02375
ALFAA	0.06892	-0.08064	-0.02084	-0.07005
ALPEKA	0.15691	0.05474	0.11490	0.10684
ALSEA	-0.03953	-0.20984	-0.14434	-0.15447
ELEKTRA	0.17329	-0.03416	0.11584	0.15216
IENOVA	0.12991	-0.06378	0.05017	0.07411
MEXCHEM	-0.00750	-0.15465	-0.09331	-0.07351
PE&OLES	0.17820	-0.13444	-0.03110	-0.12091
PINFRA	0.08481	-0.09624	-0.02576	-0.00465
WALMEX	0.13101	-0.08800	0.02481	0.06248
Total Average	0.094695300	-0.089395465	-0.001521206	-0.005175555

Fig. 6. Returns simulation with unknown data of the all *Parents*

References

[1] M. Gsell, *Essays on Algorithmic Trading*. Columbia University Press, 2010. [Online]. Available: https://books.google.com.mx/books?id=IVoZBQAAQBAJ

[2] O. Castillo and P. Melin, "A new fuzzy-genetic approach for the simulation and forecasting of international trade non-linear dynamics," in *Proceedings of the IEEE/IAFE/INFORMS 1998 Conference on Computational Intelligence for Financial Engineering (CIFEr) (Cat. No.98TH8367)*, June 1998, pp. 189–196.

[3] C. C. R.J. Kuo and Y. Hwang, "An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network," *Fuzzy Sets and Systems*, vol. 118, no. 1, pp. 21–45, February 2001.

[4] Q. Wen, Z. Yang, Y. Song, and P. Jia, "Automatic stock decision support system based on box theory and SVM algorithm," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1015 – 1022, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417409005107

[5] F. Zhang, "High-frequency trading, stock volatility, and price discovery," *Available at SSRN 1691679*, 2010.

[6] R. Ruiz-Cruz, "Portfolio modeling for an algorithmic trading based on control theory," in *Second Conference on Modelling, Identification and Control of Nonlinear Systems IFAC MICNON 2018*, Guadalajara, Jal, Mexico, Jul 2018, pp. 1–6.

[7] R. Ruiz-Cruz and A. D. Diaz-Gonzalez, "Investment portfolio trading based on Markov chain and fuzzy logic," in *2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, Nov 2018, pp. 1–6.

[8] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281 – 297.

[9] E. S. Hans-Peter Kriegel and A. Zimek, "The (black) art of runtime evaluation: Are we comparing algorithms or implementations?" *Knowledge and Information Systems*, vol. 52, pp. 341 – 378, 2017.

[10] D. J. C. MacKay, *Information Theory, Inference adn Learning Algorithms*, C. U. Press, Ed. Cambridge University Press, 2003.

[11] C. U. Press. (2018) Portfolio. Cambridge University. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/portfolio

[12] U. Maulik, S. Bandyopadhyay, and A. Mukhopadhyay, *Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics*. Springer Berlin Heidelberg, 2011. [Online]. Available: https://books.google.com.mx/books?id=bzMzUHyYEBQC

[13] L. Davis, *Handbook of genetic algorithms*, ser. VNR computer library. Van Nostrand Reinhold, 1991. [Online]. Available: https://books.google.com.mx/books?id=Kl7vAAAAMAAJ

[14] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, ser. Artificial Intelligence. Addison-Wesley Publishing Company, 1989. [Online]. Available: https://books.google.com.mx/books?id=3_RQAAAAMAAJ