Investment Portfolio Optimization using GA

Maikel Sousa (m20200735@novaims.unl.pt), David Sotto-Mayor Machado (m20201023@novaims.unl.pt), Catarina Moreira (m20201034@novaims.unl.pt)

Abstract: Of all human endeavors, stock market trading is one of complexity because its volatility. The purpose of this work is to apply genetic algorithms to optimize, within a budget, the stocks to select in a portfolio to obtain a profit in 15 days. Disregarding taxes, certain regulations, and fees, by means of genetic algorithms and an Arima forecasting model (Moving Average 15) we build candidates portfolios to maximize return, a task for which genetic algorithms are well suited. When assessing against the real value of the stocks selected, the return remains positive.

Keywords: Genetic Algorithms, Stocks, Portfolio, Arima, Population.

I. Introduction

The stock market is tool at disposal of certain companies and is known for its capacity for raising capital. Several investors, whether as particulars or organizations, around world use it to generate profit from their capital. Since its beginnings, several people have claimed to have the "magic wand" for investing in these financial assets, and although the present work is far from finding that "magic wand", it proposes a way to develop investment strategies using genetic algorithms for maximization of profit with a limited capital.

The investment strategy to obtain profit from the optimization would be through the buying and selling a group of stocks, which we call portfolio, and this investment strategy is usually known as trading. Important to mention that the present work is assuming no fees, no taxes, and disregarding any regulation requirements.

Several configurations of Genetic Algorithms are tested to find the configurations that are worth to assess and that have enough statistical significance. On a second level, the performance of these configurations are tested against their real market value to find no loss from the investment strategy proposed by the algorithms.

II. Methodology

The pool of stocks for investment is quite huge for this endeavor if we look at the whole world, to make the problem more relatable, we selected stocks from the United States which is a referent within the field, and where some of the most important global companies are settled. Mainly any deviation on this market correlates in changes for the rest of the markets around the globe, especially in negative terms.

To approach the optimization of the investment portfolio, we took inspiration from the Knapsack problem, by means of the Python library *Insvestpy*, defining a budget of $100.000 for investment and trying to maximize profit.

Using the *Investpy* python library we extracted all the stocks with price available between the dates 01-01-2021 to 30-04-2021. Leaving us with a pool of 4070 stocks, each indexed for their subsequent representation and search.

The individuals (Portfolios) were defined as a list of size 50 using the indexes mentioned above. We wanted to allow the algorithm to find portfolios that may include a stock more than once. This created some particularities, for example, portfolios [1,2] and [2,1] are the same in terms of fitness (which we will define shortly) and, the alternative, to define them as sets would deprive the algorithm to search solutions such as [1,1] or [2,2].

The fitness function was defined on 2 steps.

First step, we wanted to, somehow, predict the value of a stock in 15 days to assess if it was worth selecting it as investment. For this, we used a fast-running forecasting model like the *Arima (0,0,15)* (Moving Average 15), applied on the difference between the closing and opening price per day of a stock. From this, the model would give us an output of the expected changes in price for the following 15 days, those values were added giving the expected price change of the stock (Pred). This would account for a short-term change of prices of the stock.

On a second step, we add this predicted price change of a stock to the current price (closing price at the 30-04-2021) as the value in 15 days of that stock. Subsequently, all these individual values are added for all the stocks in the portfolio.

Before outputting the fitness value obtained as the predicted price of the portfolio, we penalize those that in their current price go above the defined budget by transforming it into its additive inverse. Allowing the algorithm to improve by itself. Taking a particular consideration to not apply this detriment on those portfolios that may have already a negative predicted price, which can be induced from the "pred" variable.

When defining this fitness function, we found a bottleneck in performance if the Moving average 15 was computed at the same time as running the optimization, as well as redundancy because the algorithm was retrieving the price several times for the same stocks and making repeated calculations. Therefore, we stored in memory the value of each stock at closing on the day 30/04/2021 as well as the predicted value for that stock.

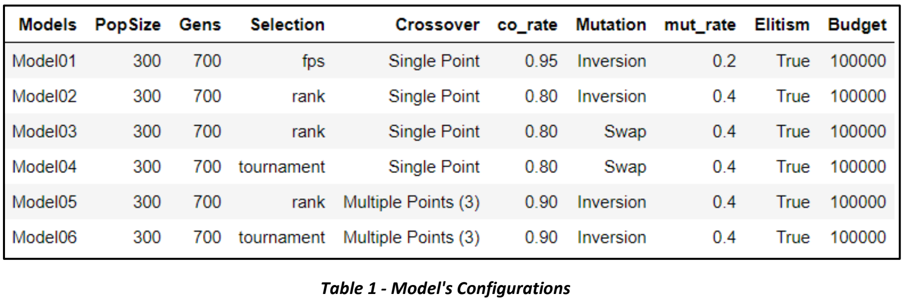
This fitness allowed us to look at the short-term trend of daily price changes of the stock to assess its future behavior and volatility on the short term.

The neighborhood was defined by taking each of the elements (stocks) in the individuals (portfolios) and filling up the remainder elements by means of a random choice.

Now that we have defined the fitness, population and neighborhood from the above parameters, the optimization was implemented using the Genetic Algorithms from the Charles Library developed at the CIFO practical sessions at Nova IMS in 2021. Several parameter configurations between: selection, crossover, crossover rate, mutation and mutation rate were tested to choose the ones worth exploring deeper. At this stage, is important to mention that the crossover operators Cycle and PMX were not suited for this problem setup because, by definition, all elements in belonging to one individual should also belong to the other, but in this setup, we may have one stock that is present on one but not in the other and its position does not imply a change in terms of fitness.

The limitation in crossover operators leaved us with no other option than to use the single point implementation, but we decided to overcome this limitation and increase the crossover possibilities, therefore, we implemented a multiple point crossover operator, finding that 3 random points in the parents allowed enough diversity in the offspring population to grow.

The different model's configurations are described in Table 1 below.



Models	PopSize	Gens	Selection	Crossover	co_rate	Mutation	mut_rate	Elitism	Budget
Model01	300	700	fps	Single Point	0.95	Inversion	0.2	True	100000
Model02	300	700	rank	Single Point	0.80	Inversion	0.4	True	100000
Model03	300	700	rank	Single Point	0.80	Swap	0.4	True	100000
Model04	300	700	tournament	Single Point	0.80	Swap	0.4	True	100000
Model05	300	700	rank	Multiple Points (3)	0.90	Inversion	0.4	True	100000
Model06	300	700	tournament	Multiple Points (3)	0.90	Inversion	0.4	True	100000

Table 1 - Model's Configurations

The configurations above were the result of the tunning of all the parameters, we found the above the configurations worth to pursue. The implementation of elitism, although is not vital for the algorithms it does produce better results.

III. Results

The models configurations above were executed 50 times each to obtain enough statistical significance on their performance, and the average best fitness per generation were as:

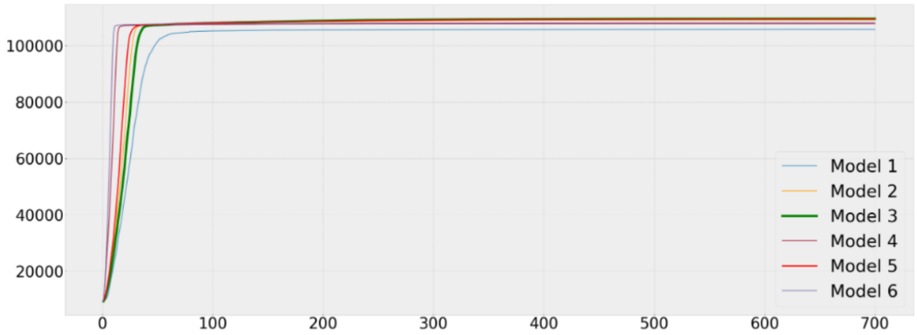


Figure 1 – Average best Fitness per generation for each model.

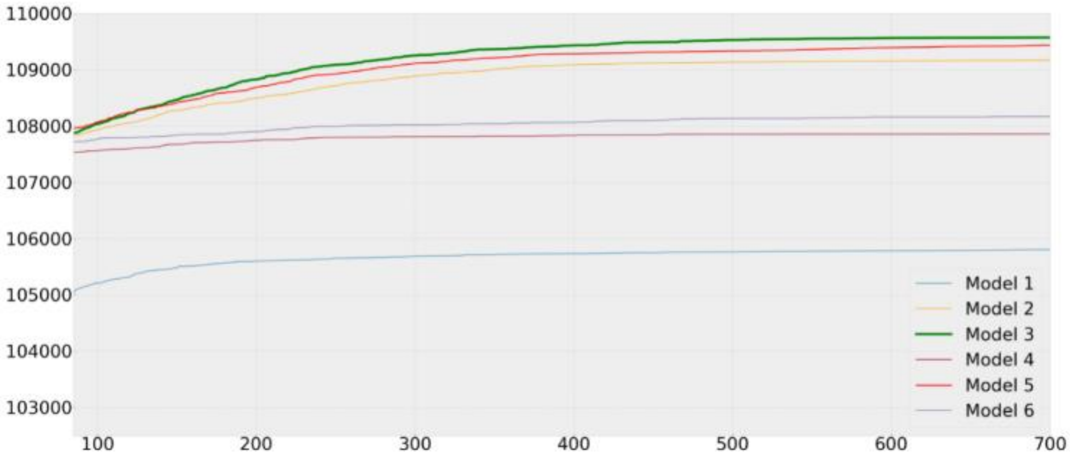


Figure 2 - (Zoomed) Average best Fitness per generation for each model.

Feel free to explore the results on this Plotly interactive graph.

By looking the results in terms of best fitness average through the 50 iterations of each model we can list the best performing from better to worse:

1. Model 03.
2. Model 05.
3. Model 02.

Although each of the algorithms reach a considerable good performance by generation 200 their average best fitness keeps improving on each generation and we can say that model 03, in average, finds the best portfolio for investment.

Now that we have assessed the Model 03 as out best configuration in terms of Fitness, we take a deeper look into Model 03, and find that for the 50 iterations the local and global optimal are reached several times and the algorithm does not get stock on local optima:

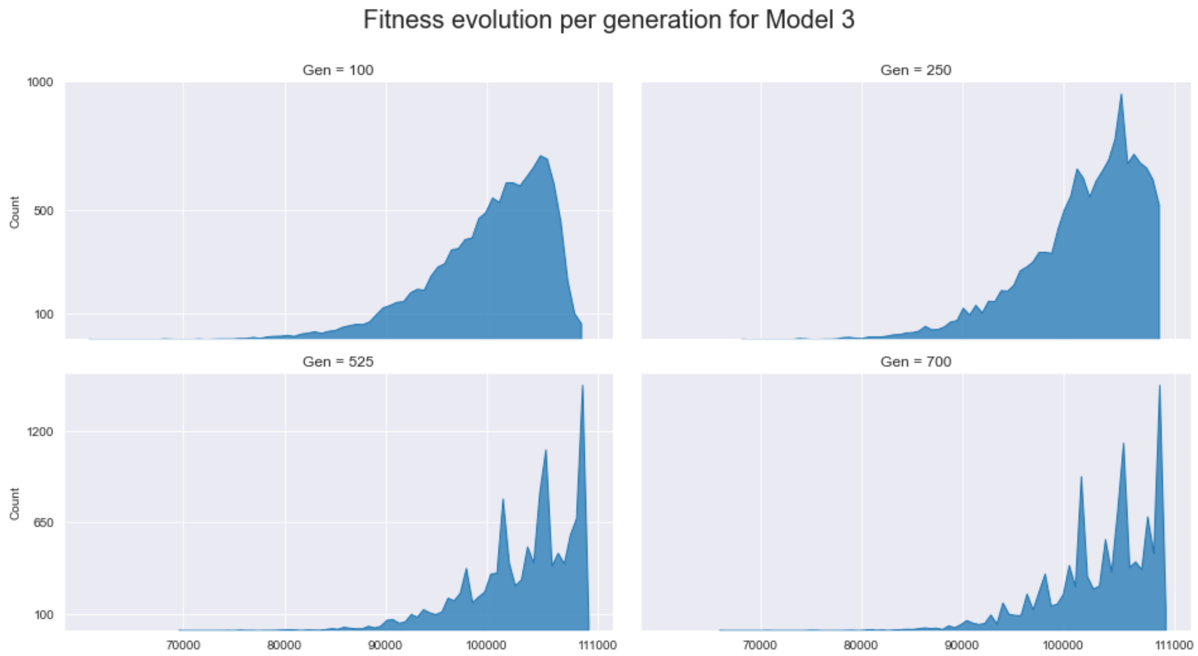


Figure 3 – Histogram per generation.

In this animation you can see how the histogram is normally distributed until generation 30, something expected, since we are doing more than 31 iterations of the model. At that moment, the algorithm reaches the maximum budget, and in the following generations the results are refined to find the best solution possible, finding optimal solutions quite early and is the reason we can spot several peaks in the histogram as the generations move forward.

On a second level, we took the best portfolios (individuals) found by each of the models and compared their expectations against the real prices of the stocks 15 days later. Although each model chooses different types of stocks for the best performing portfolio (you can visit this document if you wish to have details on that) we obtain the following results in terms expected return vs real return:

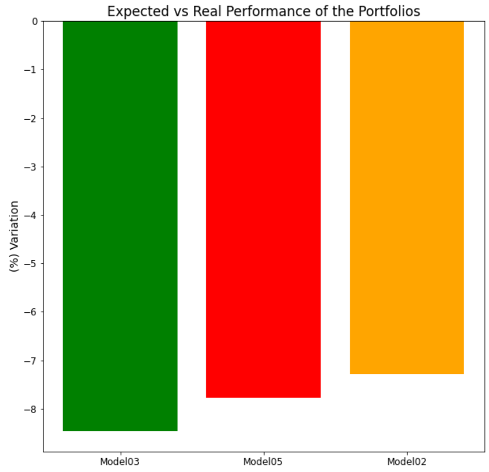
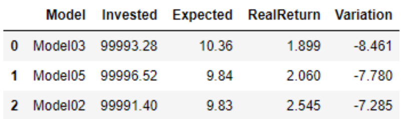


Figure 4 – Variation of the best portfolios found by each model.



	Model	Invested	Expected	RealReturn	Variation
0	Model03	99993.28	10.36	1.899	-8.461
1	Model05	99996.52	9.84	2.060	-7.780
2	Model02	99991.40	9.83	2.545	-7.285

Table 2 - Expected Return vs Real

The deviation above has a lot to do with the real performance of each stock inside the portfolio, we weighted the return of each to its portfolio, considering how much importance has from the investment stage:

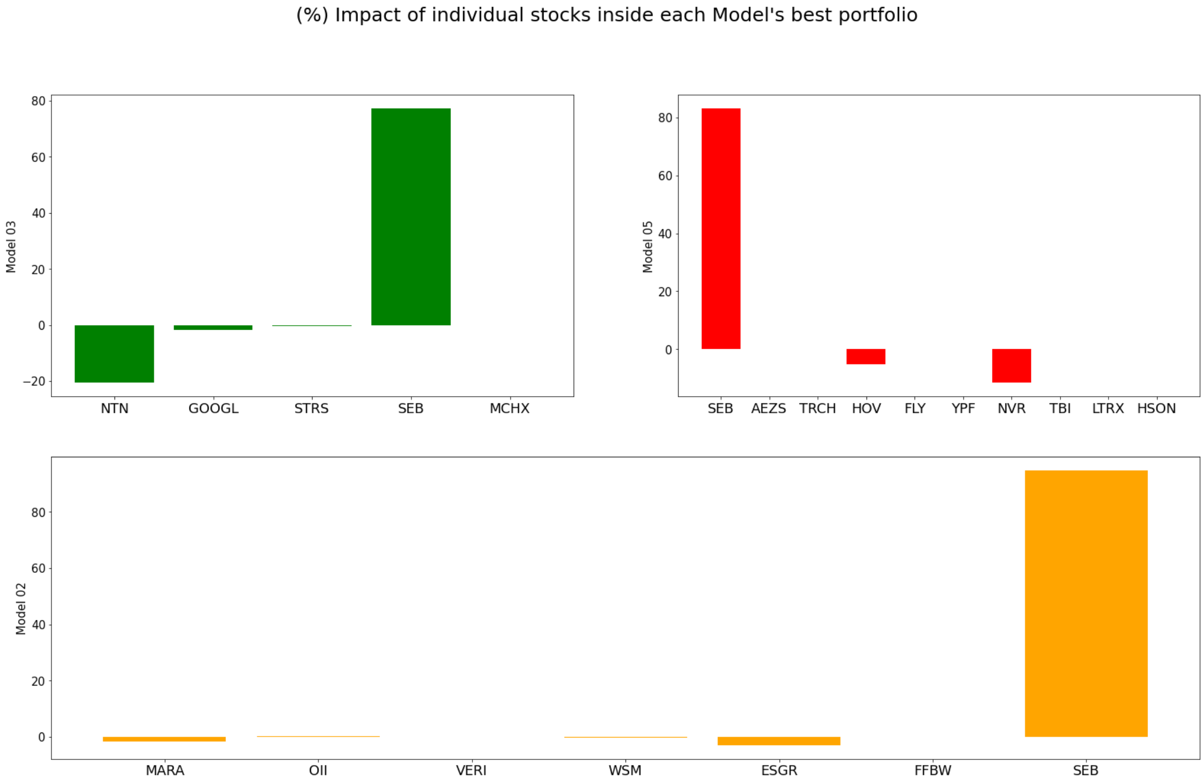


Figure 5

IV Conclusions

We found for this problem that a high rate of mutation is beneficial to find solutions with better fitness. The Model 03 parametrization is very good at finding optimal solutions on the population, which are found without the need to go to a high number of generations and with enough statistical evidence that supports its capacity to find the optimal solutions.

On the other side, when comparing the best individuals on models 03, 05 and 02, none of those portfolios found gave us a negative performance when compared to the real prices, the authors, before embarking on this quest, were expecting a negative return on the investment choices of the algorithms, given the volatility of the field and its current May 2021 context. Under this scope, we find a better portfolio on model 02.

This current work poses an opportunity to highlight an improvement on the first stage of the fitness function calculation, where the forecast of changes in prices daily for an individual stock is computed, because the deviation between the expected and real value of the portfolios is given by individual deviations within the changes of price of the stocks.

The fact that no loss was generated by the application of GAs when trying to optimize portfolio investment was a pleasant surprise for the authors of this work leaving them eager to dig into this subject deeper on future work.

Project hosted on: https://github.com/maikelps/InvestmentPortfolioOptimization

The outputs and a notebook were stored on one drive given GitHub file size limitation