Value investment strategy optimized by distributed genetic algorithms

Helder de Oliveira Mota

Supervisors: Prof. Rui Fuentecilla Maia Ferreira Neves
Prof. Nuno Cavaco Gomes Horta
Instituto Superior Técnico

***Abstract*—This document presents a solution that predicts the stock market evolution, using a fundamental approach. The application main goal is optimizing the value strategy with two objectives, the investment return and the investment risk. The chosen approach combines elements from genetic algorithms, multi objective optimization and a parallelization of those elements. To validate the solution, it was tested in a period between July 2013 and July 2015. The simulations show that a selection based on financial ratios can be used to evaluate the best companies, obtaining above market returns. The parallel version improves the sequential version, both in solution quality and runtime execution.**

***Index Terms*—Value, NSGA2, parallel genetic algorithm, stock exchange, multi objective optimization.**

I. Introduction

Stock markets are an important part of most countries, particularly developed ones, involving millions of dollars in transactions every day. Since the market has a very large number of stocks, there is the question of which ones are a good investment? There are many strategies to answer this question, but i am focusing on the value strategy. I will try to optimize this strategy, using a variety of techniques, choosing good companies in an automatic fashion.

II. Related Work

A. Market Analysis

Regarding the stock market there are two main approaches which to analyse it, they are the fundamental analysis and the technical analysis.

The fundamental analysis is based on data released by the governments, regulators and companies. With such amount of data the fundamental analysis is divided in three parts, the macroeconomic analysis, the industry analysis and the company analysis. Macroeconomic analysis focuses on economic factors of a country or a group of countries, such factors include gross domestic product, unemployment rate, interest rates, inflation among other factors[1]. Industry analysis evaluates an industry, obtaining data regarding the number of clients, industrial growth and number of competitors[2]. Finally there is the company analysis that takes a deep look into the company. There are 3 main documents that fundamental investors take a look[3], the income statement, balance sheet and cash flow statement. The income statement shows a company performance in a given time period. The balance sheet presents a company assets and liabilities at a given moment. The cash flow statement reveals how much money enters and leaves the company in a given time period. All these reports are released once every 3 months.

Technical analysis[4] focus on the behaviour of a stock in the market and takes little consideration of the company that the stock represents. There are two approaches to technical analysis, the technical indicators and the graphic analysis. Technical indicators centrers around stock prices and transaction volumes, using indicators derived from those to predict stock price movements. Graphical analysis checks for graphical patterns formed by prices and volumes to forecast the stock markets movements.

The value strategy[5] defends that a investor should only to buy stocks where the respective company presents great economics, using primarily a fundamental analysis.

B. Optimization Techniques

In the artificial intelligence area were developed many techniques to help researchers solve hard problems, one example is the genetic algorithm.

Genetic Algorithms (GA)[6] draw its inspiration from the evolution of species and natural selection. It starts with a random population, it evaluates every individual and the best ones are used to create a new population for the next generation. By repeating this cycle the the populations evolves with better individuals. There are three main operations in a genetic algorithm, the selection, the crossover and the mutation.

Selection chose the elements for the other two operations. There are many ways for individual selection, like only the best ones, a mixture of the best and random individuals and others. The crossover uses two previously chosen individuals, called parents, and creates the descendants that have characteristics from both parents. The main goal of this operation is to create better individuals.

At last, the mutation changes randomly some of the individuals, creating new solutions and introducing some diversification in the population.

GAs need a degree of diversification to better explore the solution space for better results, so the population size and the way one defines the genetic operators should have some consideration. One advantage of this algorithm is the easy interpretation of its results.

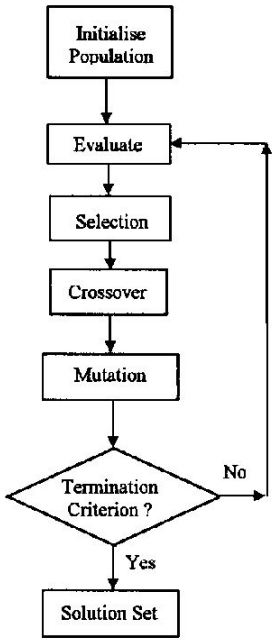

Figure 1 - GA flowchart

C. *Multi-objective Optimization*

Many real world problems have more than one objective to optimize and many times those objectives conflict with each other. Like the stock market investment, the goal is to obtain good returns with the least risk involved in that investment[7].

The pareto dominance states that a solution dominates other solution if improves it improves one objective without degrading other objectives[8]. If a solution isn't dominated by any other solution it is called optimal. With all optimal solutions the pareto front is created.

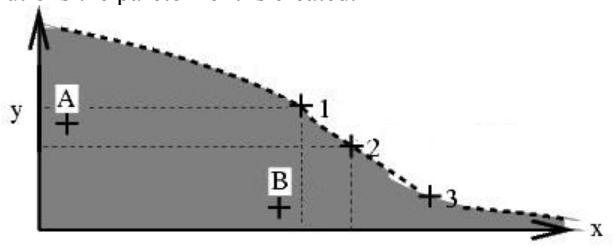

Figure 2 - Pareto front

The non-dominated sorting genetic algorithm 2 (NSGA2)[9] combines successfully the genetic algorithms and the multi-objective approach. It uses non-dominated solutions and organizes them by a metric called crowding distance. This metric measures the distance between an individual ant its closest neighbours, giving priority to individuals with greater distance, increasing population diversity. The NSGA2 is also an elitist GA, therefore it keeps the best solutions from the previous population.

D. *Parallelism*

Genetic algorithms can be used in a parallel fashion, achieving better execution times and sometimes better results that sequential ones[10][11]. There are five main types of parallel genetic algorithms (PGA) which are the independent runs, master-slave, distributed, cellular and hybrid approaches[12].

The independent runs consists simply on running the sequential algorithm on many processes with different parameters. This approach serves mainly to collect statistical data in a shorter time.

The master-slave[13] is identical to sequential GA, only distributing the evolution of individuals to other processes. Such model can be useful in situations where the evolution function is complex.

The distributed model[14] consists of running various GA with smaller populations in different processes, exchanging a small number of individuals at certain times. With many smaller populations, those tend to explore different solutions at a faster pace. The migration helps with the diversity of smaller populations and helps them obtaining better solutions. This model can achieve better results and execution times.

The cellular model[15] restricts crossover to the neighbour individuals. As such different solutions tend to appear at different points in the population and are spread slowly trough the generations. Parallelization is achieved by dividing the individual by many processes. This model can achieve better results and execution times.

Lastly, the hybrid model[16] consists of combining two or more of the other models, combining each model strengths.

III. System Architecture

For this work it was chosen the value strategy for stock market investment, optimizing it with a sequential NSGA2 and parallel version. The S&P500 was the chosen market.

The application uses data from yahoo finance and EDGAR database to obtain information about companies quotes and fundamentals. Parsing data from EDGAR requires user supervision, since there are many inconsistencies present.

I chose several fundamental ratios to evaluate each stock, these ratios are the operating margin, income before taxes to operating income, receivable to revenue ratio, debt to equity ratio, return on equity, net income growth and treasury growth. With this ratios the system determines if a company is good or not. It is used on technical ratio, which i called simple moving average ratio, and it used to buy a stock at a local minim.

Weight OP	Weight IBTtOIR	Weight RtRR	Weight DER	Weight ROE
Weight NIG	Weight TG	Weight SMAR	*n*days	Stoploss

Table 1 - Chromosome

The chromosome of each individual is the weight of each ratio, raging from 0.1 to 0.6, plus two additional parameters, the ndays and stoploss. The parameter ndays is number of days to be used by the moving average and is between 5 days to 21 days.

The stoploss indicates the maximum amount of loss that the system is willing to take, and rages between 0.05 and 0.2.

As for the objectives, obviously they are the investment return and risk. The return is calculated using return on investment (ROI), which measures the amount of return relative to the investment cost. To measure risk i use a combination of the value at risk (VAR) and the stoploss. VAR is statistical measure that determines the amount of money one investor can lose in a single day or month with certain confidence level. Stoploss restricts the amount of money the system can lose, therefore if the stoploss is lower than the VAR it lowers the overall risk. Risk is calculated according the equation 1.

$$Risk = VAR * 0.7 + Stoploss * 0.3 \quad (1)$$

Besides the standard NSGA2, it was developed and used 2 parallel versions. The first version changes the NSGA2 into a distributed model. The distributed version uses a doubly linked ring topology. Its processes communicate with other processes every 5 generations to send and receive 4 individuals. There are two ways to chose individuals for migration, the four best individuals and choosing four random individuals.

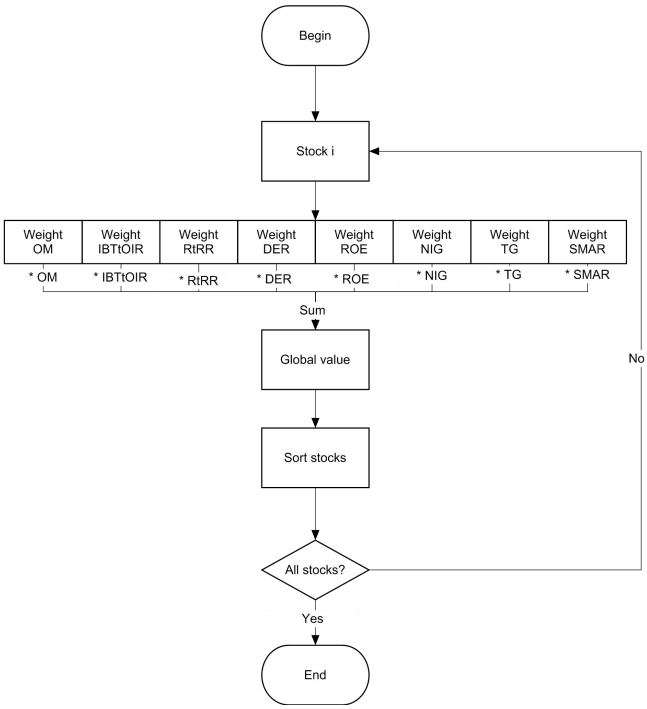

Figure 3 - Stock evaluation

The second parallel version is a hybrid model. At first level of parallelization it uses the distributed model described above. In the second level it uses a cellular model, creating a neighbourhood for the crossover operation. At this second level there isn't any kind of parallelization, so i only try to measure how this solution affects final results and not its execution time.

Finally there is the investment system. The system starts with initial amount that is distributed by 4 trimesters. At the start of each trimester it invests in 5 companies evenly and in a one year time these stocks are sold. The only way for for a stock to be sold before the one year time is for its stock price reach the stoploss, in this case the stock is sold automatically. Since fundamental data is released at each trimester, the stock evaluation occurs once every three months. The only data that is updated daily are the stock quotes and checking the stoploss.

Every trimester the system buys the best five stocks, with two restrictions. First it can't buy the same stock in two consecutive trimesters. Second it can only buy stocks that are available in the market for at least one year. Such restrictions diversify the portfolio and minimize risk.

IV. Results

The results obtained are from a test period raging from July 2013 and July 2015. It was used 479 stocks from the S&P500, with the benchmark being the S&P500 return in the same time period. The individuals selects from the training stage to the test stage are the best for each level of risk. This means the best individual with risk equal to 14% is selected, the best with 15% risk is selected and so fourth.

A. Sequential GA

In this simulation the system uses the standard NSGA2, with a population of 128 individuals. At the training stage, the system is able to detect a stable pareto front. From this stage it was selected 5 individuals, ranging from a risk of 14% to 18%.

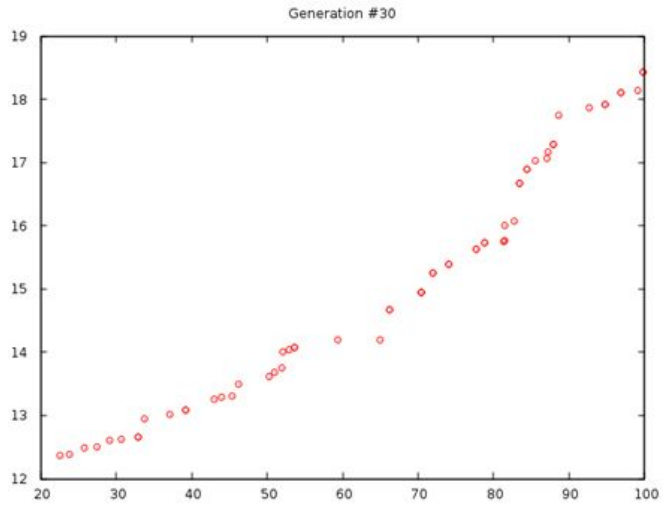

Figure 4 - Sequential training results

The S&P500 return for the test period was 24.81%. Looking at table 2, only on individual is below the benchmark, meaning that our strategy is correct.

Individual	Return(%)
1	25,44

2	24,20
3	31,36
4	30,36
5	32,24

Table 2 - Sequential version returns

B. *Distributed NSGA2*

In this simulation it is used the created distributed version of NSGA2. The simulations use 2, 4 and 8 processes, with a population of 64, 32, 16 respectively. For all simulations the global number of individuals is 128, maintaining the problem size. It also compares the migrant policy for each simulation.

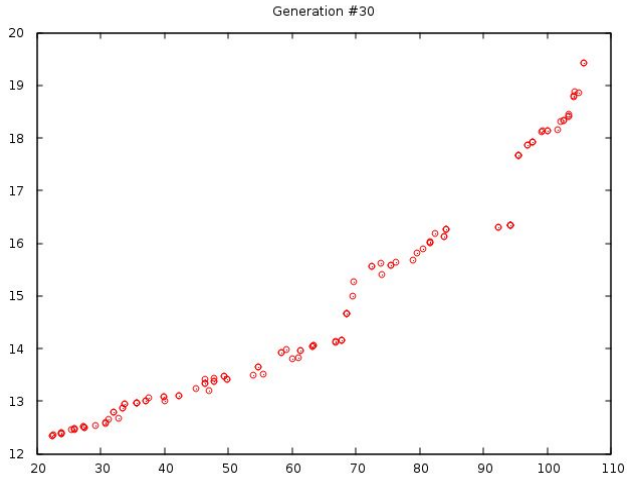

Figure 5 - Distributed training results, 2 processes, best migrants

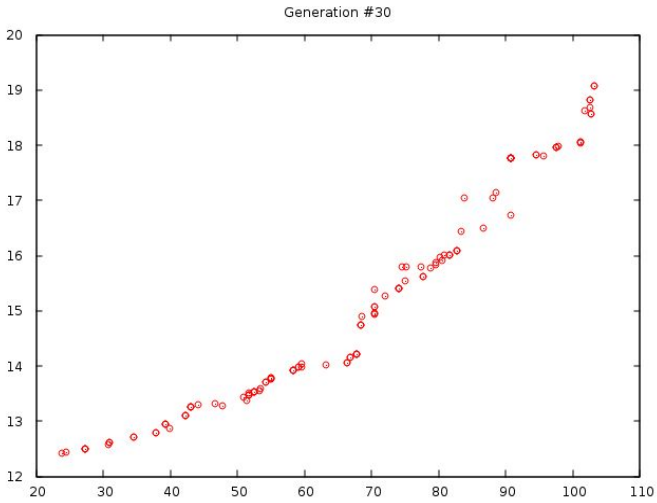

Figure 6 - Distributed training results, 2 processes, random migrants

Starting with the simulation with 2 processes the chosen individuals from the training phase shows better results in the test phase than the sequential simulation. Also it was able to discover one extra individual, raging from 14% risk to 19% risk. The migrant policy influences the results, with the random migrants obtaining better returns.

Individual	Best migrants returns (%)	Random migrants returns (%)
1	25,86	25,84
2	25,98	25,98
3	22,73	32,13
4	22,41	31,51
5	32,05	32,18
6	31,68	30,82

Table 3 - Distributed version returns, 2 processes

Now for the simulation with 4 processes, the training phase is able to produce 7 individuals to the test phase, one more than the 2 processes simulation and 2 more than the sequential simulation.

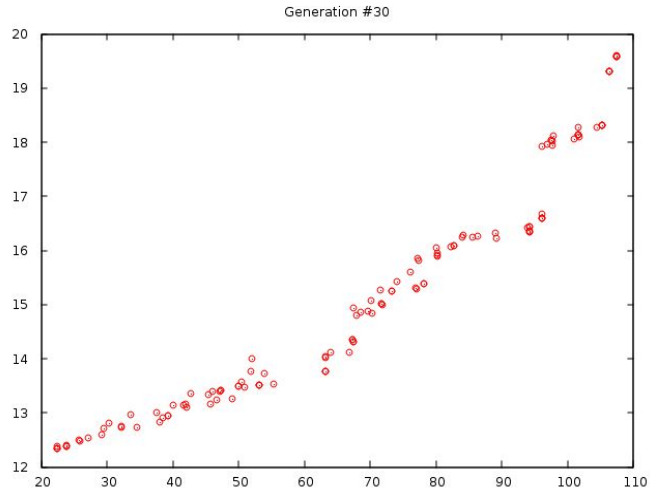

Figure 7 - Distributed training results, 4 processes, best migrants

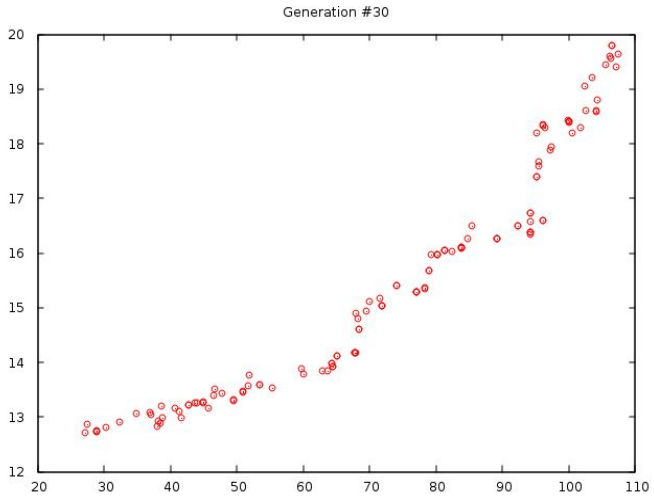

Figure 8 - Distributed training results, 4 processes, random migrants

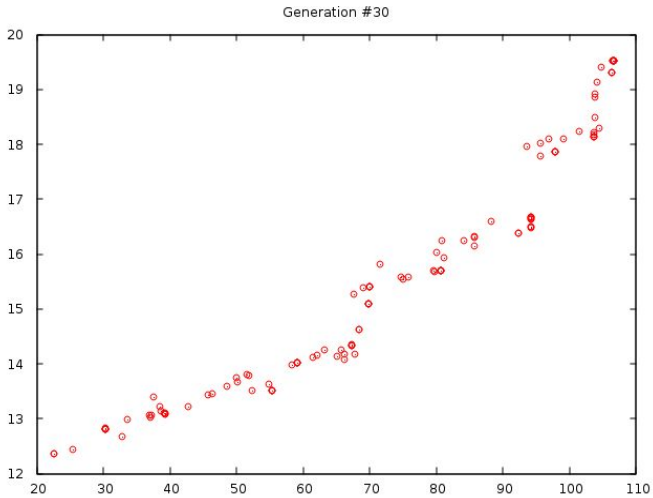

Figure 9 - Distributed training results, 8 processes, best migrants

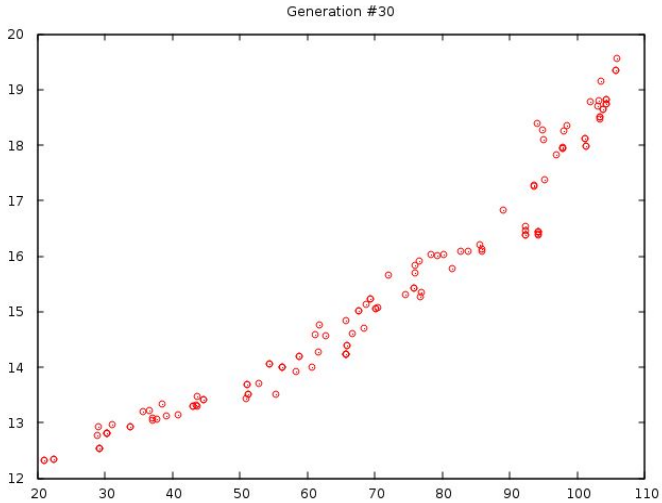

Figure 10 - Distributed training results, 8 processes, random migrants

Individual	Best migrants returns (%)	Random migrants returns (%)
1	25,84	25,86
2	23,88	23,88
3	23,58	23,58
4	23,5	31,25
5	32,03	33,18
6	29,72	30,34
7	28,88	28,7

Table 4 - Distributed version returns, 4 processes

Like the 2 processes simulation, the random migrants obtain better results, achieving a new maximum return of 33,18%, more than 8% than the benchmark, 24,81%.

As for the 8 processes simulation, it starts to decrease in quality, because the population size starts to be to small for the diversity required.

Individual	Best migrants returns (%)	Random migrants returns (%)
1	25,86	25,86
2	31,83	23,88
3	22,73	22,73
4	31,94	31,46

5	28,5	31,11
6	29,89	30,64

Table 5 - Distributed version returns, 8 processes

C. *Hybrid NSGA2*

In this simulation it is used the created hybrid version of NSGA2. It maintains the same parameters of the distributed version, the only difference being the migration policy. It was only used the random migrants, because it showed better results than the best migrants.

Individual	2 processes return (%)	4 processes return (%)	8 processes return (%)
1	25,86	25,79	25,86
2	25,96	31,83	25,92
3	23,58	22,73	23,58
4	31,15	32,51	31,41
5	31,27	32,51	32,03
6	30,72	30,98	29,96

Table 6 - Hybrid version returns

As the table shows, the 4 processes simulation presents better results, but are similar to the distributed version. The hybrid does not improve the solutions global quality it loses an individual in 4 processes simulation.

D. *Parallelization*

One of the main advantages of parallelism is the ability to reduce the execution time. In table 7 are resumed the comparison between the sequential algorithm and distributed version. The hybrid version was ignored, because it has the same parallelization than the distributed version.

Nº processes	Population size	Time(s)
1	128	167
2	64	89
3	44	64
4	32	49
5	24	40
6	20	35
7	20	36
8	16	29

Table 7 - Population size and execution times

Nº processes	Lei de Amdahl	*Speedup*	Efficiency
1	-	-	-
2	1,99	1,88	0,94
3	2,97	2,61	0,87
4	3,94	3,40	0,85
5	4,90	4,18	0,83
6	5,85	4,77	0,80

7	6,67	4,64	0,66
8	7,73	5,75	0,71

Table 8 - Parallelization Results

Table 8 shows that the distributed version reduces the execution time in comparison with the sequential algorithm. However this reduction happens at a slower rate than the one given by the Amdahl law. This means that the communication overhead grows proportionally with the number of processes, thus reducing the application scalability.

V. CONCLUSIONS

The value strategy is a sound strategy, as our system was able to obtain better results than the market, in all simulations. The distributed version was able to improve the results when compared with the sequential version. One thing that was unexpected is the effects of the migration policy, because i believed that the best migrants would present better results than the random policy. It turns that is the other way around, with random migrants offering better diversification and better results. The problem size was the same for the various simulations, but the 8 processes simulation started to show degradation in the results, showing that there is a minimum threshold for the population size. The distributed version is a valid improvement over the sequential NSGA2 both in terms of results a execution times.

The hybrid version was also a negative surprise, because i thought that it would obtain better results than the distributed version, but it obtained similar results. Such thing happens because the NSGA2 sorts all individuals every generation, nullifying the established neighbourhood effects. The cellular model to be used with the NSGA2 implies alterations and added complexity, which may defeat the purpose of the parallelization.

REFERENCES

[1] Bernard Baumohl, The Secrets of Economic Indicators: Hidden Clues to Future Economic Trends and Investment Opportunities, 2nd edition, Wharton School Publishing, 2007.

[2] Zvi Bodie, Alex Kane and Alan Marcus, Investments, 9th edition, McGraw-Hill/Irwin, 2010.

[3] Marry Buffet and David Clark, Warren Buffet and the Interpretation of Financial Statements: The search for the company with a durable competitive advantage.

[4] Fernando Braga de Matos, Ganhar em Bolsa, Dom Quixote, 2007.

[5] Lawrence A. Cunningham, The Essays of Warren Buffet: Lessons for Corporate America, 3rd edition, Carolina Academic Press, 2013.

[6] Tom M. Mitchell, Machine Learning, McGraw-Hill Science/ Engineering/Math, 1997.

[7] Magda B. Fayek, Hatem M. El-Boghdadi e Sherin M. Omran, Multi-objetive Optimization of Technical Stock Market Indicators using GAs, International Journal of Computer Aplications, Vol 68 - nº 20, 2013.

[8] E. Zitler, M. Laumanns and S.Bleuler, A Tutorial on Evolutionary Multiobjective. In Metaheuristics for multiobjective optimization, Springer Berlin Heidelberg, 2004.

[9] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, e T. Meyarivan, A Fast Elitist Multi-objective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation, April 2002.

[10] V. S. Gordon e D. Whitley, "Serial and Parallel Genetic Algorithms as Function Optimizers". Procs. of the 5th ICGA, S. Forrest (ed.), Morgan Kaufmann, 1993.

[11] F. Herrera e M. Lozano, Gradual Distributed Real-Coded Genetic Algorithms, Technical Report #DECSAI-97-01-03, February 1997 (revised version 98).

[12] Gabriel Luque e Enrique Alba, Parallel Genetic Algorithms: Theory and Real World Aplications, Studies in Computational Intelligence, Springer-Verlag Berlin Heidelberg, 2011.

[13] Masoumeh Vali, New Optimization Approach Using Clustering-Based Parallel Genetic Algorithm.

[14] Alfons Balmann, Applying Parallel Genetic Algorithms to Economic Problems: The case of Agricultural Land Markets, IIFET, 2000.

[15] Heinz Muhlenbein, Evolution in Time and Space - The Parallel Genetic Algorithm.

[16] Dudy Lim, Yew-Soon Ong, Yaochu Jin, Bernhard Sendhoff e Bu-Sung Lee, Efficient Hierarchical Parallel Genetic Algorithms Using Grid Computing, Elsevier, 2006.