Parallel MOEAs for combinatorial multiobjective optimization model of financial portfolio selection

Fernando G. D. C. Ferreira, Gustavo P. Hanaoka, Felipe D. Paiva, Rodrigo T. N. Cardoso
Centro Federal de Educação Tecnológica de Minas Gerais,
7675 Amazonas Av., Nova Gameleira, Belo Horizonte - Brazil 30510–000
Email: rodrigoc@des.cefetmg.br

***Abstract*—This paper proposes a multiobjective optimization model with rebalancing and cardinality constraint to obtain portfolios consistent with practical aspects of the financial investment process. To perform this optimization, parallel versions of the evolutionary algorithms PDEA, SPEA2 and NSGA-II are proposed, aiming a low execution time in accordance with the rapid oscillations in the stock market. In-sample analysis compares sequential and parallel versions of evolutionary algorithms while out-of-sample analysis performs simulations using different algorithms and models, seeking to identify the effect of the rebalancing process for the model. Results point out the importance of the model with rebalancing and the efficiency of the parallel algorithms, mainly in relation to the execution time.**

I. Introduction

In stock market, investors looking for a high financial gain are also prone to get large losses. In 1952, Markowitz [8] defended the idea of diversifying investments, seeking maximum expected return (gain) and minimum variance (risk).

Based on this work, many other different models have been developed to obtain reasonable portfolios, for example, considering multiobjective optimizaton models [3, 17, 1]. Additionally, the variance can be replaced by other more coherent risk measures, such as CVaR [10] and real-world constraints can be incorporated in these models, to get closer to the market practices, such as cardinality and transaction costs [11, 12, 13]. Meanwhile, the inclusion of cardinality constraint leads the optimization of portfolios problem to be NP-complete [4]. And the inclusion of transaction costs in the model leads to another practical aspect that should be taken into account when investing: the value of these costs depends on the investor's previous portfolio. To model this aspect, some work includes the rebalancing process in the optimization model [22, 25, 17].

In this scenario, the use of computational techniques of artificial intelligence, such as evolutionary algorithms, can be very useful, providing optimal investments in a time compatible with the oscillations of market prices [6, 9, 7]. To further reduce the execution time of evolutionary algorithms and to try to adapt them to the rapid price oscillations that exist in the stock market, some papers use parallel versions of these algorithms, achieving significant time reductions in their executions [21, 23, 24].

Recently, a multiobjective optimization model with cardinality restriction has been proposed using integer decision variables, consistent with the way in which assets are sold in practice, in stock lots [18]. That paper also proposes two evolutionary algorithms to perform the optimization and a realistic simulation of operations in the market during five years. This paper intends to improve the model, adding a rebalancing process to the integer optimization process, and to propose a better otimization machinery, creating parallel versions of some evolutionary algorithms, compatible with the rapid oscillations in the stock market. In this regard, this paper proposes:

- (i) A multiobjective integer-optimization mean-CVaR model with cardinality constraint, with rebalancing process.
- (ii) Three multiobjective evolutionary algorithms, based on PDEA (Pareto Differential Evolutionary Algorithm) ([20]), NSGA-II (Non-dominated Sorting Genetic Algorithm II) ([5]) and SPEA2 (Strength Pareto Evolutionary Algorithm 2) ([19]), to cope with the proposed model. Specific repair operators were created for these algorithms, according to the nature of the problem.
- (iii) Parallel versions of the evolutionary algorithms in two different ways. The first uses island model parallelization that perform separately improvements in individuals blocks at the same time while the second uses parallel sorting methods on GPU instead of sequential ones.

Computational simulation results show the importance of the rebalancing process in the model and the efficiency of parallel evolutionary algorithms, which can provide diversified portfolios at a short runtime. In-sample comparisons show that parallel versions can obtain similar fronts to the sequential version, in a small computational time, while out-of-sample simulations of stock market trading suggest that the rebalancing process provides greater long-term profit.

The article is organized as follow. Section II shows the proposed mathematical model for the portfolio optimization. Section III presents the proposed parallel algorithms to solve the problem and explains the metrics chosen to compare the algorithms. Section IV shows and discusses the results of the comparison between sequential and parallel versions of the algorithms and the behavior of the best algorithm in a real-case stock market simulation. Finally, Section V summarizes and analyses the implications of the results.

II. Model Statement

Based on the multiobjective combinatorial model presented in [18], whose objectives are to maximize the expected

978-1-5090-6017-7/18/$31.00 ©2018 IEEE

return and minimize Conditional Value at Risk [11], with cardinality constraint, a new model was proposed that adds the rebalancing process, wich consists basically in considering a past investment to calculate a transaction cost based on the quantity and value of the operations to obtain a new optimized investment from the old investment. Therefore it is assumed that the cost difference when taking into account this transaction is relevant to the optimization process.

The constrained biobjective model with rebalancing is then formulated as follows:

$$\min_{x_1,\ldots,x_N} \quad \zeta + (1-\alpha)^{-1} \sum_{j=1}^{J} \pi_j [f(x, y_j) - \zeta]^+ \tag{1}$$

$$\max_{x_1,\ldots,x_N} \quad \sum_{i=1}^{N} w_i \mu_i \tag{2}$$

$$s.t.: \begin{cases} \sum_{i=1}^{N} z_i = k & \text{(3a)} \\ d_i = x_i - x_i^{(0)}, \forall i,\, i = 1, ..., N & \text{(3b)} \\ \sum_{i=1}^{n} m_i c_i [(d_i) + \beta |d_i|] \leq C - Fv & \text{(3c)} \\ x_i, m_i \geq 1, \forall i \mid z_i = 1, i = 1, ..., N & \text{(3d)} \\ x_i \in \mathbb{N}, \forall i,\, i = 1, ..., N & \text{(3e)} \\ z_i \in \{0, 1\}, \forall i,\, i = 1, ..., N & \text{(3f)} \\ v \in \{0, 1\} & \text{(3g)} \\ w_i = \dfrac{m_i c_i x_i}{\sum_{i=1}^{N} m_i c_i x_i}, i = 1, ..., N & \text{(3h)} \end{cases}$$

The variables of the problem are: x_i, the decision variable that represents the number of lots corresponding to the asset i that compose the portfolio; the weight w_i for each asset i is calculated based on the investment ratio of such asset in relation to the total cost of the portfolio; z_i is a binary variable that indicates the presence of the asset i in the portfolio, whose value is 1 if the asset is present and 0 otherwise; and v that is also a binary variable that holds 0 if $\sum_{i=1}^{N}(x_i - x_i^{(0)}) = 0$ and 1 otherwise.

The parameters are: N, the number of assets considered; ζ, the VaR measure of the portfolio; $alpha$, a given level of significance; π_j, the probability of a certain scenario j occurs; $f(x, y_j)$, the loss function for the scenario j; μ_i, the historical average of returns of the asset i; m_i, the minimum number of shares to be invested for each asset i; c_i, the cost of each share of the asset i; C, the maximum available capital to invest in the portfolio; $x_i^{(0)}$, the amount of asset lots i that made up a previous portfolio; β, called proportional transaction cost, that is a percentage that determines a proportion of the amount involved in the operations to acquire the optimized portfolio; F, a value, in monetary unit, charged by each financial transaction. β and F represent transaction costs, including brokerage, custody and emoluments charges.

The objective functions are described by the expression 1, representing the minimization of risk using CVaR, and by the expression 2, describing the maximization of the portfolio expected return, calculated as the average of the assets returns weighted by their investment proportions [8]. The expression 3a describes the portfolio cardinality constraint in which the sum of the values of each z_i must be the value stipulated by k. The minimum value constraint of treatable shares in the expression 3d ensures that the number of lots invested in the asset i is greater than or equal to 1 and it is an integer value. Finally, the expression 3c represents the proposed cost constraint including rebalancing, that is, it is added a transaction cost to the cost of the portfolio taking into account a previous portfolio, since it defines the amount of operations needed to get the next optimized portfolio.

The return of each day is calculated as the logarithmic difference between the closing price of the current day and the closing price of the previous day: $r(t) = \ln(closing(t)) - \ln(closing(t-1))$. Therefore, the return presents a rate softening the difference between possible exorbitant prices in the financial market.

It can be noticed that the model without rebalancing was presented in [18], which is a particular case of this proposed model where $x_i^{(0)} = 0, \forall i, i = 1, ..., N$.

III. Proposed algorithms

Parallel multiobjective evolutionary algorithms are used in this article in order to optimize the proposed model of financial portfolios selection. They are parallel versions of PDEA [20], NSGA-II [5] and SPEA2 [19].

Each individual, considered as a possible solution in evolutionary algorithms, represents a portfolio and is encoded in a structure that contains two integer variable sets: assets set with a fix cardinality and set of lot trades for each asset with the same size of the assets whereby each one have its associated lot trades number in this set. Binary variables are associated with the assets and each variable has value 1 if its corresponding asset composes the portfolios and has value 0 otherwise.

The **generation of the initial population** is made by filling in the assets set randomly. Each asset receives a random amount of lot trades, so that the total cost of the portfolio does not exceed the amount available for investment. Repair operators are used if they become infeasible. All of the algorithms perform mutation, crossover and selection operations until reaches a large number of generations, as specified, or until the hypervolume indicator [16] reaches a very small number.

For **NSGA-II** the operators are:

- **Crossover** choses randomly two individuals of the new population called *parent1* and *parent2* in each iteration. Then, these two individuals generate two new ones, called *child1* and *child2* as follows: a cutoff is chosen randomly and *child1* is formed by the assets at the left of this point present in *parent1* with their lot trades and the assets present at the right of that point in *parent2* with their respective lot trades. Similarly, the *child2* is formed by the complementary combination of *parent1* and *parent2*. Thus, two new children replace their parents in the new population at each iteration and the procedure continues until N children are generated. Random assets replace repeated assets and lot trades are reduced randomly when the solution is infeasible.

- In **mutation**, a small percentage of the assets of the new population individuals are chosen randomly and these are replaced by other assets also randomly selected. When this occurs, the portfolio containing the asset that was modified changes the values of lot trades of each of its assets randomly.
- The **selection** operator selects the N best individuals from the union of the current population with the new population. These selected individuals will be part of the current population on the next generation.

For **PDEA** the operators are:

- The **mutation** is performed according to the *DE/rand/1/bin* strategy, that causes the mutated individual generated from the difference between two random distinct individuals plus a third random individual. Since the difference between integer values (representing assets) can generate an infeasible solution, the module-n function is applied to this set of assets. If any number is repeated, the repair is made raffling an integer that is not in the set. Random quantities of lot trades for each asset in the portfolio are generated after this process, so that the final cost of the portfolio does not exceed the amount available for investment.
- The **crossover** operator in wich an individual is generated with assets belonging to an individual of the current population or the mutated individual. After this process, as in mutation, random amounts of lot trades are assigned to each asset in the portfolio so that the portfolio cost does not exceed the amount available for investment.
- The **selection** is performed as in NSGA-II algorithm.

Mutation and crossover operators of SPEA2 are the same as those described for NSGA-II, but the classical selection operator ([19]) was used. The PDEA parameters scaling factor of the differential mutation and probability of crossover as the NSGA-II and SPEA2 parameters probability of crossover and mutation rate are adaptively determined during the executions. Repair operators are developed here to guarantee feasibility.

All three algorithms (PDEA, NSGA-II and SPEA2) were paralleled in two different ways, using an island model and on GPU:

- In an **island model**, the original algorithm is broken down into several other evolutionary algorithms with smaller numbers of individuals (called islands), that evolve independently, sharing their individuals belonging to the non-dominated frontier in each fixed number of generations. The number of individuals on each island is equal to the number of individuals used in the original algorithm divided by the total number of islands. In this work, each island is executed by a thread.
- **Parallelization on GPU** of the sorting function, which has a high computational cost. This parallelization is performed with the CUDA platform and using the thrust library.

Three performance measures were used in order to evaluate the results of the algorithms: Spacing (S) metric [15], which measures how the solutions are diverse or how evenly distributed they are across the solution space, Hypervolume (HV) [14], which measures the volume of the space dominated by the solution set of Pareto front, and the execution time of each algorithm.

IV. Numerical Results

Tests were performed on an Intel Xeon E5-2630 v3, 2.4 GHz processor. The computer has 8GB of RAM and an AST2500 Advanced PCIe Graphics GPU. It used Linux Mint 18.1 and C language.

This case study considers a historical trend data of the Brazilian stock market, with 53 assets, those participating in the Bovespa index ([2]) in December 2015. Quotations relating to 6 years (January 2010 to December 2015) were used to perform the optimization, totalling 1,484 daily closing prices. First of all, the optimization was performed for the first year (2010) and every proposed parallel versions were compared to each other and to the original sequential algorithm. Then, the best version of each algorithm was compared with the best version of the other algorithms in order to verify which algorithm provides the best efficient front of solutions (portfolios). Finally, an out-of-sample simulation of financial transactions, from 2011 to 2015, compares the profits provided by the best parallel algorithm with its sequential version, in addition to comparing the proposed model with and without rebalancing. Therefore, a total of 60 sets of portfolios were generated for each algorithm.

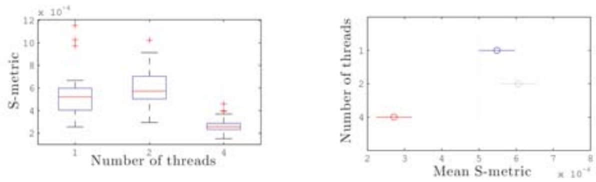
The model parameters chosen were: number of assets $N = 53$; significance level $\alpha = 5\%$; minimum number of shares $m_i, \forall i, i = 1, ..., N = 100$; available capital $C = \$\,100,000.00$; proportional transaction cost $\beta = 0.45\%$; fixed transaction cost $F = \$\,29.00$; cardinality $K = 9$ assets. Transaction cost parameters were determined based on information from Banco do Brasil, a Brazilian broker, and the cardinality was chosen among those who get the best performance in [18].

The algorithms PDEA, NSGA-II and SPEA2 are run for 30 times, using 500 individuals and the maximum generations equal to 500. Three types of parallelization were considered: parallelization using the island model, parallelization on GPU and parallelization using both. Tests were performed using 1, 2 and 4 islands or threads, with the algorithms sharing individuals to each 100 number of generations.

A. In-sample analysis

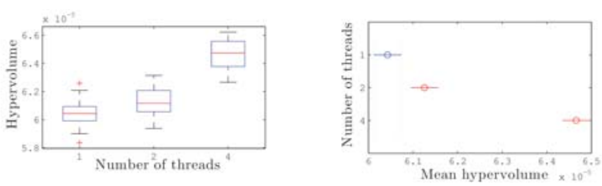
The aim of the in-sample analysis was to evaluate the performance of the parallel and sequential multiobjective optimization algorithms and to compare them, trying to find out if the parallelization affects the performance of the solutions generated by the algorithms. The algorithms are compared according to the metrics: C-metric, S-metric and execution time. The results shown correspond to the values of the metrics obtained for the combined nondominated front of 30 executions of each algorithm and type of parallelism. Hypothesis tests were performed to verify if parallel and sequential algorithms show the same nondominated front, chosen under the assumptions of normality, independence and homoscedasticity of the samples. Tukey's tests used a significance level of 5%.

For PDEA, boxplot in figure 1(a) shows that the island model with 4 threads presented better scattering of the solutions, with a lower S-metric value, whereas the Tukey's test for this metric in figure 1(b) does not detect statistical difference between 1 and 2 threads algorithm. The hypervolume boxplot shown in figure 2(a) also indicates the superiority of the 4-threaded algorithm for this metric, and the Tukey test in figure 2(b) reveals a superiority of the 2-thread algorithm over the sequential (named as 1-thread) algorithm.



(a) Boxplot (b) Tukey's test

Figure 1. Boxplot and Tukey's test of S-metric for sequential and parallel PDEA



(a) Boxplot (b) Tukey's test

Figure 2. Boxplot and Tukey's test of hypervolume for sequential and parallel PDEA

As the statistical tests suggest, it is also possible to observe by the non-dominated fronts of figure 3 that PDEA solutions improve with increasing number of islands in this model of parallelism. Supposedly this occurs due to the low adequacy of the PDEA operators to the problem addressed, since these operators are made for continuous problems and the model proposed in this paper is an integer variables model. Thus, a greater effect of randomness with increasing number of islands improves both the uniform scattering of the solutions and their proximity to the optimal front.

For SPEA2, the boxplot in figure 4(a) shows a little S-metric difference for the island model algorithms with 2 and 4 threads and the sequential algorithm, whereas the Tukey's test in figure 4(b) does not detect statistical difference for

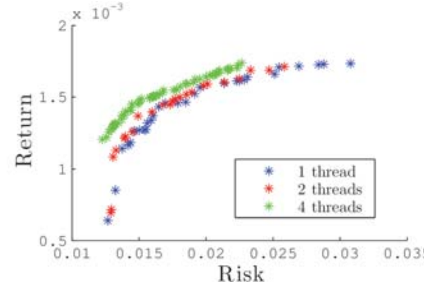
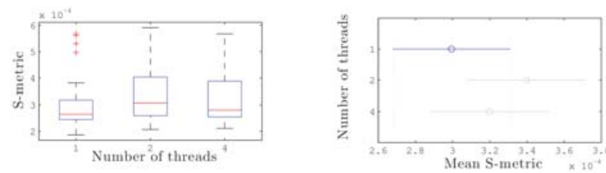


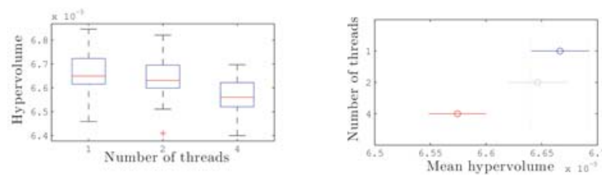
Figure 3. Efficient front of parallel and sequential PDEA

any algorithm in relation to this metric. On the other hand, the hypervolume boxplot shown in figure 5(a) reveals smaller values for the 4-thread algorithm and then Tukey's test shown in figure 5(b) confirms the inferiority of the 4-thread algorithm.



(a) Boxplot (b) Tukey's test

Figure 4. Boxplot and Tukey's test of S-metric for sequential and parallel SPEA2



(a) Boxplot (b) Tukey's test

Figure 5. Boxplot and Tukey's test of hypervolume for sequential and parallel SPEA2

SPEA2, unlike PDEA, uses appropriate operators for integer decision variables. In this way it was expected that the quality of the solutions would decrease with the increase in the number of islands, since the small amount of individuals in each island may not be sufficient for the generation of good individuals. Even so, this difference of the 4-threaded algorithm for the others was small, since this algorithm presents a front similar to the others, as it can be observed in figure 6.

For NSGA-II, the S-metric boxplot shown in figure 7(a) already reveals a great difference of this metric for the algorithm of 4 threads in relation to the others and the Tukey's test in figure 7(b) detects the inferiority of this algorithm. As for the hypervolume, the boxplot in figure 8(a) shows a little difference between the algorithms while the Tukey's test in figure 8(b) fails to detect statistical difference for the algorithms.

This worse spread presented by the algorithm of 4 threads was expected since the cut based on the distance of agglomeration is more effective, providing the maintenance of more

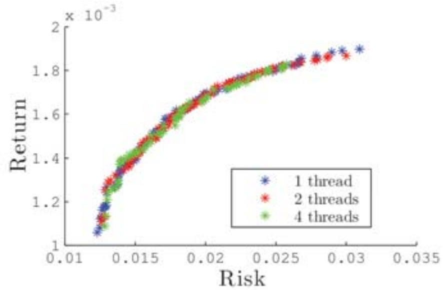
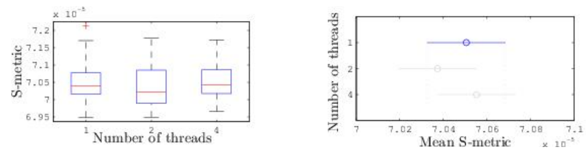
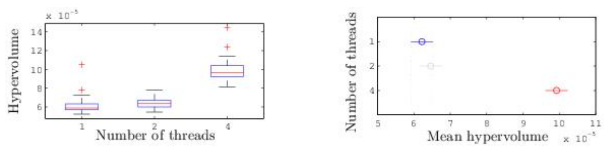


Figure 6. Efficient front of parallel and sequential SPEA2



(a) Boxplot (b) Tukey's test

Figure 7. Boxplot and Tukey's test of S-metric for sequential and parallel NSGA-II



(a) Boxplot (b) Tukey's test

Figure 8. Boxplot and Tukey's test of hypervolume for sequential and parallel NSGA-II

diversified solutions in populations with greater number of individuals. But, as with SPEA2, it can be seen in figure 5 that this difference is still relatively small and the fronts generated by the algorithms are visually similar.

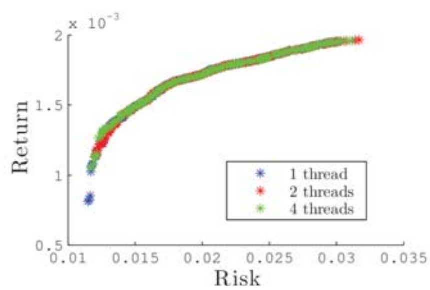


Figure 9. Efficient front of parallel and sequential NSGA-II

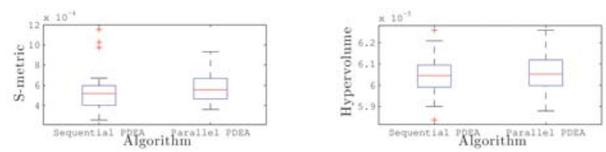
The execution time decreased with increasing number of islands as expected and a higher gain can be observed for SPEA2, which requires higher computational cost. Table I shows the minimum, average and maximum execution times observed for each algorithm.

Parallelization on GPU was also performed for the three algorithms. Tukey tests did not detect statistical difference of S-metric and hypervolume for the parallel and sequential algorithms, as expected, since just the ordering methods were

Table I
EXECUTION TIME FOR ISLAND MODEL ALGORITHMS IN SECONDS

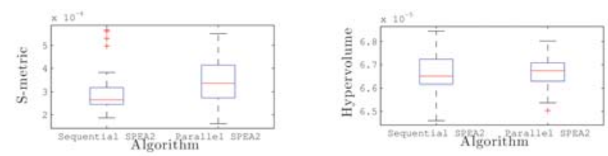
		Minimum	**Mean**	**Maximum**
	1 thread	26.730000	26.822667	26.920000
PDEA	**2 threads**	12.264487	12.308142	12.536879
	4 threads	8.458658	8.493440	8.671959
	1 thread	127.619995	166.675666	176.789993
SPEA2	**2 threads**	51.059544	54.483468	62.562866
	4 threads	15.737564	16.140833	18.212560
	1 thread	28.860001	28.947000	29.040001
NSGA-II	**2 threads**	12.747342	13.190849	18.819960
	4 threads	10.006701	10.059078	10.085508

parallelized. For the PDEA, the S-metric and hypervolume boxplots are represented in figures 10(a) and 10(b), respectively; the tests resulted in p-values of 0.46 and 0.58. For SPEA2, the S-metric and hypervolume boxplots are represented in figures 11(a) and 11(b), respectively; and the tests resulted in p-values of 0.86 and 0.16. For NSGA-II, the S-metric and hypervolume boxplots are represented in figures 12(a) and 12(b), respectively; and the tests resulted in p-values of 0.17 and 0.33.



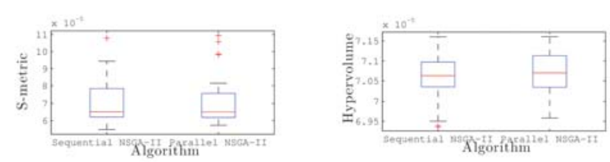
(a) S-metric (b) Hypervolume

Figure 10. Boxplot of S-metric and hypervolume for sequential and parallel PDEA



(a) S-metric (b) Hypervolume

Figure 11. Boxplot of S-metric and hypervolume for sequential and parallel SPEA2



(a) S-metric (b) Hypervolume

Figure 12. Boxplot of S-metric and hypervolume for sequential and parallel NSGA-II

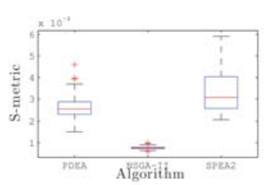
Table II shows the execution time for the parallel versions on GPU of each algorithm, comparing them with their sequential versions. As desired, the parallel versions have less execution time, despite the parallelization with island model with 2 or 4 threads present time even lower. Again, SPEA2, which has the highest computational cost, was the algorithm with the highest gain after parallelization, which can be explained by the fact that SPEA2 executes sorting methods several times during its cutting operation.

Table II
EXECUTION TIME FOR GPU-PARALLEL ALGORITHMS IN SECONDS

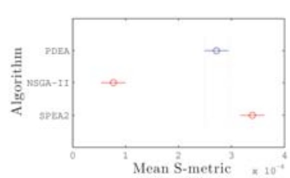
	Minimum	Mean	Maximum
parallel PDEA	21.375999	21.507733	21.936001
parallel SPEA2	63.179800	63.559401	65.116002
parallel NSGA-II	23.072001	23.155733	23.232000

Based on the S-metric, hypervolume and execution time, a parallel version was chosen for each algorithm. For the PDEA was chosen the algorithm with 4 islands, which presented better values for all the metrics. For the SPEA2 and NSGA-II algorithms, the 2-island versions were chosen because they presented S-metric and hypervolume similar to the respective sequential version with shorter execution time.

Comparing these three algorithms, it is revealed that the 2-islands NSGA-II overcomes the other algorithms in relation to the diversity of the solutions scattered in the front efficient and in the proximity of its front with the optimal front. SPEA2 takes the second place, surpassing the PDEA in S-metric and hypervolume. The boxplot in figure 13(a) reveals the large S-metric difference for the algorithms, and the Tukey's test in figure 13(b) identifies statistical difference from the NSGA-II to the others and from the SPEA2 to the PDEA. The same occurs for hypervolume, whose boxplot is shown in figure 14(a) and Tukey's test is shown in figure 14(b). Finally, figure 15 shows the combined front of the best parallel version of each algorithm. By looking at the figure, the results shown by the hypothesis tests performed can be easily confirmed and the dominance of the solutions in the NSGA-II over the other solutions can be noticed. Although the solutions of SPEA2 and the PDEA are overlapped, it can be seen that the solutions of SPEA2 dominate PDEA solutions in most of the time.

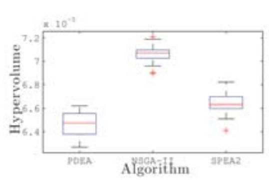


(a) Boxplot

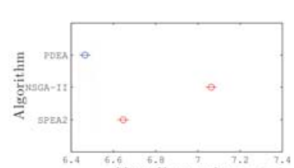


(b) Tukey's test

Figure 13. Boxplot and Tukey's test of S-metric for selected algorithms



(a) Boxplot



(b) Tukey's test

Figure 14. Boxplot and Tukey's test of hypervolume for selected algorithms

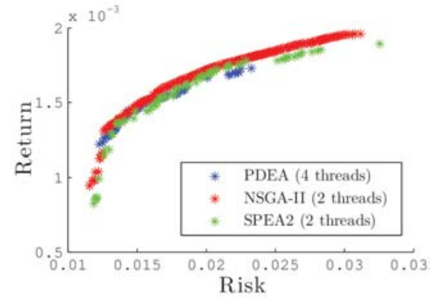


Figure 15. Efficient front of selected algorithms

B. Out-of-sample analysis

The out-of-sample analysis has the aim to evaluate the performance of some selected portfolios, obtained by the optimization of the proposed model, with and without rebalancing, using four algorithms:

- Algorithm 1, corresponding to the sequential NSGA-II with rebalancing;
- Algorithm 2, the 2-islands parallel NSGA-II with rebalancing;
- Algorithm 3, the sequential NSGA-II without rebalancing;
- Algorithm 4, corresponding to buying and holding the Ibovespa shares, with rebalancing.

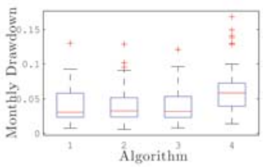
This section just considers NSGA-II, given its superiority in the simulation results of the in-sample analysis, as previously shown.

This out-of-sample test compared the portfolios over 60 months, from January 2011 to December 2015. For algorithms 1, 2 and 3, the point of the combined nondominated front with the best value of Sharpe ratio, the return over risk, was chosen. Ibovespa (Ibov) is considered in algorithm 4 because it is the most important index of average performance index of prices of shares traded in São Paulo Stock Exchange, Brazil. Besides, Selic, a brazilian interest rate considered to be low risk, is also used here as a benchmark. Portfolios generated by sequentials ans parallel NSGA-II were rebalanced on the first day of each month, with a new optimization process and Ibov were rebalanced every four months.

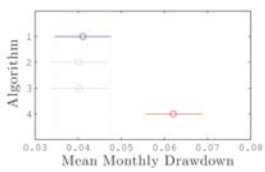
The portfolios were compared according to three metrics: return, risk and drawdown. Each metric was calculated monthly for each portfolio, totalling 60 samples. In this way, the return measure corresponds to the variation between the opening price of the first day of the month and the closing price of the last day of the month, considering the transaction costs included in the model. The risk considered was the CVaR of each portfolio in each month, and drawdown corresponds to the largest percentage drop in value of the yield curve.

After obtaining the samples of those metrics, the Analysis of Variance was performed, chosen according to assumptions of normality, independence and homoscedasticity of the samples, with a significance level of 5%. Later, the multiple comparison test was performed to find out which portfolio was statistically superior, always using a significance level of 5%. Finally, the algorithms and the benchmarks were compared in relation to the cumulative return provided by its generated portfolios over the 5 years.

Results of monthly Drawdown in boxplot of figure 16(a) shows approximate values for all algorithms, with a slight difference for algorithm 4, which reveals higher values of Drawdown. Then, after an Analysis of Variance, Tukey's test in figure 16(b) reveals the superiority of sequential and parallel NSGA-II (algorithms 1 to 3) on the Ibovespa benchmark (algorithm 4).



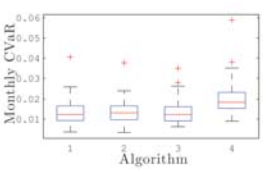
(a) Boxplot



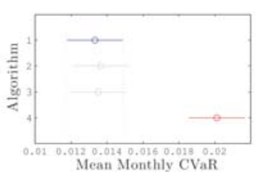
(b) Tukey's test

Figure 16. Boxplot and Tukey's test of monthly Drawdown

For monthly CVaR risk, the results were similar to those of Drawdown, with boxplot in figure 17(a) showing a little difference between all the algorithms and Tukey's test, showed in figure 17(b), revealing inferiority of Ibovespa in relation to the other algorithms.



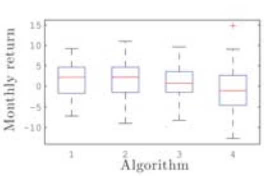
(a) Boxplot



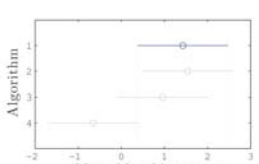
(b) Tukey's test

Figure 17. Boxplot and Tukey's test of monthly CVaR

The boxplot of the monthly return in figure 18(a)) shows a very similar values between algorithms 1 and 2, a slightly lower median for algorithm 3 and even lower values for the algorithm 4. Tukey's test in figure 18(b) reveals statistical difference just between algorithms 2 and 4, although the confidence intervals of algorithms 1 and 4 are almost separated.



(a) Boxplot



(b) Tukey's test

Figure 18. Boxplot and Tukey's test of monthly return

Thus, it can be stated that algorithms 1 and 2 show very similar and apparently better performance than those of algorithm 3, although this is not detected in the tests performed. On the other hand, algorithm 4 was overcame by the other algorithms in both risk metrics and statistically surpassed by the parallel NSGA-II even in relation to the monthly return.

Cumulative returns in figure 19 show that algorithms 1 and 2 presented very close values, as expected by the risk and monthly return tests, and that both presented accumulated returns visibly better than the algorithm 3 from 2014 onwards, reinforcing the idea that algorithms 1 and 2, with rebalancing, provide better result than algorithm 3, without rebalancing, in the long term. In addition to the expected result of the superiority of the NSGA-II algorithms (1 to 3) on the Ibovespa benchmark, it can be observed that these optimization algorithms also outperformed the Selic rate, highlighting the importance of using efficient portfolio optimization. Finally, the low Ibovespa performance in this period stands out, with returns well below of the Selic.

Figure 19. Cumulative return for each algorithm

V. Conclusion

This paper presents parallel versions of three portfolio optimization algorithms that allow users to obtain diversified and efficient portfolios in a short time. Results show that there are statistical evidences that the proposed parallel algorithms can provide solutions of the same quality as those generated by sequential algorithms, sometimes surpassing them, with a smaller runtime.

In-sample analysis shows that for the PDEA, island model parallel algorithms provide a better efficient front while SPEA2 and NSGA-II presented a 2-islands parallel algorithm that had no statistical difference detected in relation to the sequential version, with the 4-island version was shown to be inferior. For parallelization on GPU, it was shown that the quality of the generated solutions was not in relation to the efficient front provided by the sequential algorithms. This result was already expected, since this kind of parallelization affects only the sorting methods. As expected, all parallel versions had shorter execution time than the sequential version, and the island model presented a runtime greater than the parallelization on GPU and the more islands it is used, the smaller is the run time but shorter is the speedup. Comparing the different algorithms with respect to the efficient front, it was shown that NSGA-II presented the best results, followed by SPEA2, which surpassed the PDEA, even in its version of 4 islands.

Out-of-sample analysis showed the superiority of evolutionary algorithms for optimizing portfolios in relation to Selic, a low risk asset, even in a period of poor market performance, identified by the large loss provided by the Ibovespa. This fact highlights the importance of a good optimization in the financial investment process. It is also worth mentioning a higher accumulated return for the rebalancing algorithms, especially from 2014 onwards, which may indicate that the rebalancing process, which considers transaction costs more realistically, provides better gains for the long-term investments.

For future works, the number of islands and the number of individuals per island should be varied, trying to minimize the execution time without affecting the quality of the solutions. In addition, short-term investments (intra day) should be carried out in order to verify the effectiveness of the model for this type of investment as well.

Acknowledgments

Authors would like to thank to CAPES, CNPq, FAPEMIG and CEFET-MG.

References

[1] K.P. Anagnostopoulos, G. Mamanis, The mean-variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms, Expert Systems with Applications, v. 38, n. 11, pp. 14208-14217, 2011.

[2] Bovespa Index, http://www.bmfbovespa.com.br/pt_br/produtos/indices/indices-amplos/indice-ibovespa-composicao-da-carteira.htm (visited on 12-20-2016).

[3] T.-J. Chang, et al. Heuristics for cardinality constrained portfolio optimisation, Computers & Operations Research, v. 27, n. 13, pp. 1271-1302, 2000.

[4] R. Cheng, J. Gao, On cardinality constrained mean-CVaR portfolio optimization, In: The 27th Chinese Control and Decision Conference (2015 CCDC), IEEE, pp. 1074-1079, 2015.

[5] K. Deb, et al., A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE transactions on Evolutionary Computation, v. 6, n. 2, pp. 182-197, 2002.

[6] K. Deb, et al., Bi-objective portfolio optimization using a customized hybrid NSGA-II procedure. In: International Conference on Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, pp. 358-373, 2011.

[7] M. A. Ibrahim, M. El-Beltagy, M. Korshid, Evolutionary Multiobjective Optimization for Portfolios in Emerging Markets: Contrasting Higher Moments and Median Models, In: European Conference on the Applications of Evolutionary Computation, Springer International Publishing, pp. 73-87, 2016.

[8] H. Markowitz, Portfolio selection, The journal of finance, v. 7, n. 1, pp. 77-91, 1952.

[9] J. Pekár, Z. Čičková, I. Brezina, Portfolio performance measurement using differential evolution, Central European Journal of Operations Research, v. 24, n. 2, pp. 421-433, 2016.

[10] R.T. Rockafellar, S. Uryasev, Optimization of conditional value-at-risk, Journal of risk, v. 2, pp. 21-42, 2000.

[11] R.T. Rockafellar, S. Uryasev, "Conditional value-at-risk for general loss distributions." Journal of banking & finance, v. 26, n. 7, pp. 1443-1471, 2002.

[12] D. Roman, K. Darby-Dowman, G. Mitra, Mean-risk models using two risk measures: a multi-objective approach, Quantitative Finance, v. 7, n. 4, pp. 443-458, 2007.

[13] B. Sawik, Downside risk approach for multi-objective portfolio optimization, In: "Operations Research Proceedings", Springer Berlin Heidelberg, pp. 191-196, 2012.

[14] E. Zitzler, et al., Performance Assessment of Multiobjective Optimizers: An Analysis and Review, IEEE Transactions on Evolutionary Computation, 2003.

[15] J.R. Schott, Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH, 1995.

[16] J. L. Guerrero, et al., Introducing a Robust and Efficient Stopping Criterion for MOEAs, IEEE Congress on Evolutionary Computation (CEC 2010). Barcelona, Spain, July 2010.

[17] H. Soleimani, H.R. Golmakani, M.H. Salimi, Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm, Expert Systems with Applications, v. 36, n. 3, pp. 5058-5063, 2009.

[18] B. C. Barroso, F. G. D. C. Ferreira, G. P. Hanaoka, F. D. Paiva, R. T. N. Cardoso, Composition of investment portfolios through a combinatorial multiobjective optimization model using CVaR, IEEE Congress on Evolutionary Computation, pp. 1795-1802, 2017.

[19] E. Zitzler, M. Laumanns, L. Thiele, et. al., SPEA2: Improving the strength Pareto evolutionary algorithm, Tik-report, 2001.

[20] N. K. Madavan, Multiobjective optimization using a Pareto differential evolution approach, IEEE Congress on Evolutionary Computation, 2002.

[21] M. Dhurandhar, Multi-Objective Multi-Stage Portfolio Optimization with Parallel GA, Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 662-670, 2003.

[22] J. E. Beasley, Portfolio Optimisation: Models and Solution Approaches, Tutorials in Operations Research, 2013.

[23] P. Maguire, D. O'Sullivan, P. Moser and G. Dunne, Risk-adjusted portfolio optimisation using a parallel multi-objective evolutionary algorithm, 2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr), New York, NY, pp. 1-8, 2012.

[24] D. J. Bodas-Sagi, et. al., A parallel evolutionary algorithm for technical market indicators optimization, Journal of Natural Computing, pp. 195-207, 2013.

[25] W. Sun, et al., Optimal rebalancing for institutional portfolios: minimizing costs using dynamic programming, Journal of Portfolio Management, pp. 33-43, 2003