Multi-attribute Decision Making Applied to Sets of Non-dominated Solutions of Financial Portfolios

Gustavo H. M. Mendonça, Fernando G. D. C. Ferreira, Rodrigo T. N. Cardoso, Flávio V. C. Martins
Centro Federal de Educação Tecnológica de Minas Gerais,
7675 Amazonas Av., Nova Gameleira, Belo Horizonte - Brazil 30510–000
Email: gurumendonca@gmail.com, fernandogdcf@gmail.com, rodrigoc@des.cefetmg.br, flaviocruzeiro@decom.cefetmg.br

***Abstract*—This paper presents three different methods of decision making used to select, according to the decision maker preferences, a solution from the set of non-dominated portfolios generated by an integer multiobjective optimization model with cardinality constraint performed using the NSGA-II. The decision making consist of two methods that determine weights analytically from the preferences of the decision makers - weight rank and DM Queries - and a method that determines the best selection using a neural network in unsupervised learning. Simulations were done using assets from the Brazilian stock exchange for the period between 2011 and 2015. At each beginning of the month, the previous portfolio is sold, the optimization is performed, and the decision-making method selects the new portfolio to be purchased. Results considered two metrics: monthly maximum Drawdown and cumulative return during the entire study period, and show that the optimization model is robust, always presenting cumulative returns above safe investments for the analyzed period, and decision methods can deliver higher returns for investors who prioritize return and reduce maximum drawdown for those who prioritize risk.**

I. Introduction

The use of computational artificial intelligence techniques has been increasingly used in supporting investors in the selection of promising financial assets, especially when it comes to portfolio management, where several assets should be analyzed based on a large mass of data.

Markowitz, in [13], advocates diversification of investment, demonstrating that such a process reduces the variance of investment. Despite his great model's success, criticisms were made to this measure because it takes into account the above and below average dispersions, with only below-average dispersions being undesired. Conditional value-at-risk (CVaR) became to be considered as a measure of the risk of a portfolio, thinking risk as the worst loss for a given set of scenarios, given a degree of confidence.

Considering a bi-objective model, which aims to minimize risk and maximize return, all these changes aiming to approximate the model to reality such as the inclusion of transaction costs and, especially, the inclusion of the cardinality constraints, greatly increase its complexity [4]. So that the use of computational techniques as evolutionary algorithms is advisable to ensure good solutions in a viable time. Several papers show good performance of NSGA-II for this problem.

Frequently in our daily lives, we are faced with situations where we need to make a decision about a choice, selecting an alternative and discarding the other ones. When we have only one goal, this choice is trivial, and we simply have to choose the alternative that best fits our purpose. For example, if our goal is to buy a laptop spending as less money as possible, we just need to choose the cheapest laptop available, and that is it, our goal is achieved. However, in most cases, we have more than one attribute to consider when we need to make a decision, and it is impossible to get the better of both.

Still thinking about the example of the computer, if we want to buy a laptop spending as less money as possible but getting the best configuration of the computer, we are in a situation where we can not get the best of both attributes. The computer with better configuration, with a better motherboard, with a good RAM and the best video card, certainly will not be the cheapest one. Moreover, apparently, the laptop that costs less is not the one with the best configuration. In these cases, it is only possible to improve one attribute to the detriment of others, for example, paying more for a laptop with better components. We will then have several alternatives, none of which is better than another in both attributes, and we will have to choose one according to our possibilities and preferences. This is a typical case of multiattribute decision-making (MADM).

A problem arises after the multiobjective optimization of real-world problems: the difficulty of selecting a suitable solution among the several solutions present in the Pareto Frontier. When we have only one goal, this choice is trivial, and we just have to choose the alternative that best fits our purpose. However, in a multiobjective problem, we have more than one attribute to consider when we need to make a decision, and it is impossible to get the better of both since the objectives are conflicting. We will then have several alternatives, none of which is better than another in both attributes, and we will have to choose one according to our possibilities and preferences, making it MADM problem.

Several works that perform multiobjective optimization of portfolios use unreliable criteria that do not take into account the preferences of the investor, such as the work of [7], which uses the return-to-risk index for portfolio selection on the Pareto Frontier. This paper proposes the modeling of the decision criterion based on the preferences of the investor and another contribution of the work is the proposal and comparison of three different decision methods: rank weights, decision maker queries and artificial neural network, taking into account the financial gain that each one provides in an out-of-sample analysis.

Results show that the decision-making methods applied to the non-dominated front obtained by the optimization work very well, generating satisfactory results and allowing to obtain the preferences of the decision-maker in a simple way.

The article is organized as follow. Section II presents the different methods of decision making used for selecting a solution after performing the optimization: rank weights, DM queries, and artificial neural networks. Section III presents the mathematical model for the portfolio optimization and the proposed algorithm to optimize this model. Section IV shows and discusses the results of the trading simulation for each of the different methods of decision making from the bi-objective optimization solutions and comparing them. Finally, Section V summarizes and analyzes the implications of the results.

II. Multiattribute Decision-Making

In this paper, the multiattribute decision-making selects the best solution on Pareto Frontier, based on the investor preferences, after the portfolio optimization has been completed. Monthly, the previous portfolio is sold and this new portfolio is bought and held during one more month using one of these methods of decision making.

A. Multi-Attribute Utility Theory

MultiAttribute Utility Theory (MAUT) assumes that the preferences of the Decision Maker (DM) can be represented by an additive MultiAttribute value function defined in an indirectly way through holistic pair-wise judgments. To do so, the DM has to compare alternatives and decide which one is better preferred, represented by aSb if alternative ”a” is at least as good as alternative ”b” [20].

It is important to emphasize that in this type of situation we deal with the uncertainty of the decision maker who has not always clearly defined his preference, which can directly impact the results.

Decision alternatives can be interpreted as a set of attributes in MAUT, and in order to evaluate an alternative, all the attributes are evaluated. Every single attribute offers a utility value between 0 and 1 through a single-attribute utility (SAU) function. The value provided by all SAU functions get integrated into a MultiAttribute utility (MAU) function, resulting in one utility value for every alternative [18].

The most common MAU function is the additive function. Let $v_i(X_k)$ be the value of alternative $X_k \in$ X, on attribute i $\in$ I, being $X = \{X_1, X_2, ..., X_n\}$ a finite set of alternatives evaluated by a set of attributes $I = \{1, 2, ..., j\}$. Also, w_i is the weight of the *i*th attribute, representing the relative importance of that attribute to the DM. The global value of alternative X_k can then be given by the following function of multiattribute values:

$$V(X_k) = \sum_{i=1}^{j} w_i v_i(X_k), \tag{1}$$

where V is the overall multiattribute value, $0 \leq V \leq 1$; X_k, k = 1, 2,. . . , n is a vector of attribute levels; $v_i(Xk)$ is a single attribute value function, $0 \leq v_i(X_k) \leq 1$; $w_i \geq 0$, the weight of attribute *i*, $\sum_{i=1}^{j} W_i = 1$ [1].

As shown in Equation (1), the main objective lies in get the weights for the attributes of alternatives, in which will be implicit the DM’s preferences.

B. Modeling DM’s Preferences

1) Rank Weights: It is therefore essential to extract from DM his preferences about which attribute is best preferred but, to do so, we are dealing with imprecision. This happens because DM has not always well defined which attribute is more important and, even when he does, it is challenging to define how better one attribute is in relation another (for example, one attribute is two times better than another, or 50% worse). According to [24], in uncertain circumstances, if the set of feasible probability distributions is not empty and contains more than one element, the dominance relationships must be verified. This would allow eliminating the dominated alternatives, making the task of determining which alternative is the best a little more natural.

In 1985, [23] presented a method for ranking multiattribute alternatives through a weighted-additive evaluation function, using partial information about the weighting coefficients and develop an algorithm that partially classifies the whole set of alternatives based on the ranking information.

Several approximate weighting schemes were presented to preserve the rank order of attributes, considering that attribute weights are arranged from the most important to the least important, like the schemes proposed by [21]:

- Rank Sum Weights

$$w_i = \frac{N - R_i + 1}{\sum_{j=1}^{N} N - R_j + 1}, \tag{2}$$

  In this scheme, N attributes are ranked and each attribute is weighted $(N - R_i + 1)$ where R_i is the rank position of the attribute. Each weight is then normalized by $\sum_{j=1}^{N} N - R_j + 1$.

- Rank Reciprocal Weights

$$w_i = \frac{\frac{1}{R_i}}{\sum_{j=1}^{N} \frac{1}{R_j}}, \tag{3}$$

  where w_i is the normalized weight for attribute i, R_i is the rank for the ith attribute, and N is the number of attributes.

- Rank Exponent Weights

$$w_i = \frac{(N - R_i + 1)^z}{\sum_{j=1}^{N} (N - R_j + 1)^z}, \tag{4}$$

  The respondent judges the weight of the most important attribute on a 0-1 scale. This weight is entered into the Equation (4), which may then be solved for z via an iterative process. N is the number of dimensions (number of attributes) in the ranking, and R_i is the rank of the ith dimension. Once z is known, weights for the rest of the dimensions are determined. It is interesting to observe that when z = 0 defines the equal weights case, and when z = 1 we get the rank sum weights. As z increases, the set of normalized weights gets steeper.

Following this research line, in 1996 [2] proposed the Rank order centroid weights scheme:

- Rank Order Centroid Weights

$$w_i = \frac{1}{N}\sum_{j=i}^{N}\frac{1}{j}, i = 1, 2, ..., N. \quad (5)$$

This scheme tries to identify a single set of weights that is representative of all possible weights combinations that are admissible and consistent with the established linear inequality constraints on the weights.

In literature, there are still several other schemes to define the attribute weights, and there is a great variety of decision-making methods, revealing the importance of multicriteria analysis in the current scenarios.

2) DM Queries: Another method used in this paper consists of asking queries to the DM considering pairs of alternatives, in order to direct the DMs preferences, without the need of knowing the weights of the attributes. For example, presented to DM two alternatives, with weights 0.2 and 0.8 (for attribute 1 and attribute 2 respectively), and another with weights 0.6 and 0.4. Also, it is determined an α variable in order to determine the next weights of attributes. The alternatives are presented without showing this weights to DM, instead, it is presented the consequences of that alternative. Considering a set of non dominated alternatives for the laptop buying problem, two alternatives would be selected, the first one values the price more, and the second has higher preference for performance. In this case, could be presented to DM one result of a benchmark execution and the corresponding percentage of the price of the laptop to the current minimum wage, for both alternatives. The DM would choose one of the alternatives without the knowing that one represents 0.2 and 0.8 weights against the other 0.6 and 0.4. After this first query, one direction will be made, valuing more the price or the laptop performance. Lets say, the first alternative was chosen (0.2 and 0.8 weights). Then, another query is made considering this first choice. The α variable would be added and subtracted from the weights, resulting in two new alternatives. Let α be 0.1 in this example. The new alternatives would be (0.2 and 0.8) $\pm$ α, making the alternatives 0.3 - 0.7 and 0.1 - 0.9. Again, the DM would not be aware of this weights, but would choose one of then considering the information presented. Now, DM chooses between the new alternatives or keep with the last one (0.2 and 0.8). If a new one is selected, then this process repeats with it. If DM still prefers the last one, then α is decreased (this decrease can be 0.01, 0.02, etc, is a methods parameter) and two new alternatives are presented. This queries continues to be asked until $\alpha = 0$, defining the attributes weights.

3) Artificial Neural Network: Artificial neural networks (ANN) are computational models inspired by the human brain. It is a powerful tool capable of learning from examples and its own decisions (such as the human brain) and has a great capacity for generalization. It can be described as a machine designed to model the way the brain performs a particular task or function of interest, having a natural propensity to store experimental knowledge and make it available for use. Among its characteristics that resemble the brain, are that its knowledge is acquired from its environment through a learning process, and the connecting forces between its basic structure (neurons), known as synaptic weights, are used to store the acquired knowledge [9]. In the feedforward neural network (used in this paper), the first layer has a connection from the network input. Each subsequent layer has a connection from the previous layer and the final layer produces the networkâs output. Figure 1 illustrates a ANN topology. A feedforward network with one hidden layer and enough neurons in the hidden layers is capable of fit any finite input-output mapping problem. Because it has this capacity, ANNs have been used for various purposes, such as dealing with liquidity risk assessment in banking [22], prognostics of aluminum electrolytic capacitors [12], dam break flow solution [19] and many other fields.

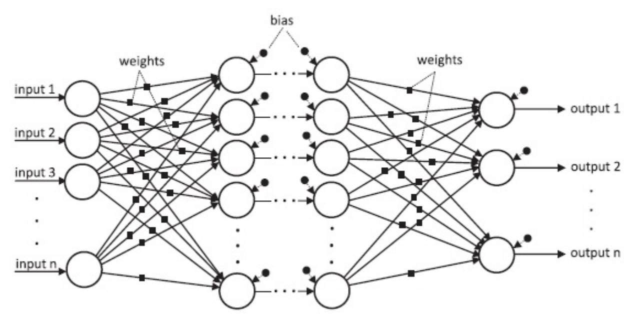

Figure 1. Topology of a ANN [10].

In this paper, the NN-DM Methodology was chosen for the Artificial Neural Network (ANN) in order to approximate the utility function, extracting the DM preferences. This method requires the following steps [15]:

1 Establish the Domain: Select the domain for the utility function approximation and construct a grid of simulated alternatives. These alternatives can be fictitious and will be used to train the ANN.
2 Build the Ranking: Build a partial ranking for the alternatives by assigning a scalar value to each one.
3 Approximate the Utility Function with ANN: Construct an artificial neural network in order to interpolate the results and represents the DM's preferences.

The domain of the approximation is defined as the smallest hyper-box with edges parallel to the coordinate axes that contain the set of available alternatives. Then, it is built a grid of alternatives in this domain. Considering alternatives on a curve, for example, a Pareto-Optimal front, refinement generates the information to find a suitable model for DM preferences in the whole domain, as illustrated in Figure 2.

To build the partial ranking is selected a method that uses features of divide and conquers. It initially sets the ranking

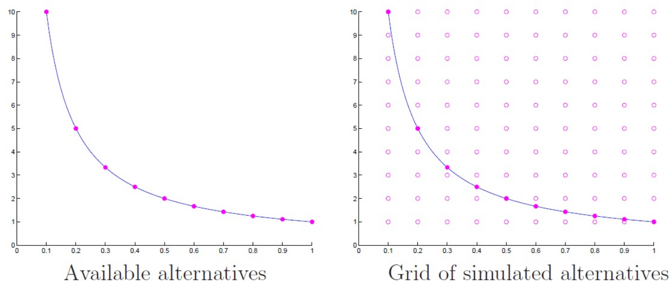

Figure 2. Refinement of a Pareto-optimal front [15].

of all alternatives as 0. Next, one alternative is selected as a pivot, and all other alternatives that are considered better than the pivot are separated on a new partition, and the rankings of these alternatives are set to 1. In this new partition, a new pivot is selected, and the procedure is repeated, separating the alternatives considered better than this new pivot in a new partition, increasing its rankings value. This procedure is repeated until the new partition has only one alternative, which is the best one. It is important to emphasize that these definitions, of which alternative is better than others, is done through queries to the DM.

This ranking level got by the last step is used as output (target), and the grid of fictitious alternatives is used as input for training the ANN which will approximate the DM preferences. Given a set of new alternatives in the same domain, the approximated function is used to choose the preferred alternative, without the need to consult decision-maker again, and the best alternative will be the one with greater value [16].

Weights and bias values in training were updates according to Levenberg-Marquardt (LM) optimization. The LM algorithm is a higher-order adaptive algorithm, member of a class of learning algorithms called pseudo second-order methods, and it is used to minimize the Mean Square Error of a neural network. In determining the best direction to move the weights in order to bring down the error, second-order methods use the Hessian or the matrix of second derivatives of the performance surface to determine the weight update, whereas, pseudo-second order methods estimate the Hessian. In order to discard second order derivatives of the error, LM method makes use of the Gauss-Newton approximation that accepts the Jacobian Matrix [11].

III. Optimization

Portfolio optimization is performed to define a set of non-dominated investments about the objectives of return maximization and risk minimization. This optimization is performed every month considering a historical data series of one year.

This process considers a historical trend data of the Brazilian stock market whereby 53 assets, which were participating in the Bovespa index [3] from January 2010 to December 2015 were chosen. Quotations relating to these six years were used to perform the optimization totaling 1,484 daily closing prices. 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. It is represents by $r(t) = \ln(closing(t)) - \ln(closing(t-1))$ so that it softens the difference between eventual exorbitant prices in the financial market.

A. Model Statement

Using the CVaR (Conditional Value-at-Risk) measure proposed by [17], an integer optimization model with the objectives of maximizing the portfolio's expected return and minimizing its risk was proposed, based on the one presented in [7]. The difference of the proposed model consists in using proportional and fixed modeling for transaction costs, bringing it closer to the reality of stock market tradings.

This way, the following model is proposed:

$$\min_{x_1,\ldots,x_N} \quad \zeta + (1-\alpha)^{-1} \sum_{j=1}^{J} \pi_j [f(x, y_j) - \zeta]^+ \tag{6}$$

$$\max_{x_1,\ldots,x_N} \quad \sum_{i=1}^{N} w_i \mu_i \tag{7}$$

$$s.t.: \begin{cases} w_i = \dfrac{m_i c_i x_i}{\sum_{i=1}^{N} m_i c_i x_i}, i = 1, ..., N & \text{(8a)} \\ \sum_{i=1}^{N} z_i = k & \text{(8b)} \\ \sum_{i=1}^{N} m_i c_i x_i (1+\gamma) \leq C - f & \text{(8c)} \\ z_i = \begin{cases} 0, & \text{if } x_i = 0 \\ 1, & \text{otherwise} \end{cases}, \forall i,\ i = 1, ..., n & \text{(8d)} \\ x_i \in \mathbb{N},\ \forall i,\ i = 1, ..., n & \text{(8e)} \end{cases}$$

with $[f(x, y_j) - \zeta]^+ = max(0, f(x, y_j) - \zeta)$.

The variables of the problem are: x_i is the *decision variable*, that represents the number of lots trades corresponding to the asset i; w_i is the proportion of the investment made in asset i; and z_i is a binary variable that indicates the presence of the asset i in portfolio. The parameters are: ζ is the reference VaR; α is the established significance level; π_j is the probability of a certain scenario; $f(x, y_j)$ is the loss function; σ_{ij} is the covariance between assets i and j; μ_i is the historical average return on asset i; m_i is the minimum number of shares that must be considered for each asset i; c_i is the cost of the minimum lot trade for the asset i; C is the available capital to be applied in the portfolio; γ and f are the proportional and fixed transaction cost, taking into account the brokerage, emoluments and the amount charged for the possession of shares.

The objective functions are described by the expressions 6 and 7, related to minimizing the CVaR *risk* and maximizing the *return* of the portfolio, respectively. CVaR function was used as defined in [17] while return refers to the expected portfolio return proposed in [13].

Expressions 8b and 8d describe the *cardinality constraint* of the portfolio, whereupon the sum of values of each z_i must be the stipulated by k. Expression 8c describes the *budget constraint*, that ensures that the cost of portfolio with the transaction costs is smaller than the amount available for investment. The *investment weights* w_i for each asset i, which are described by expression 8a, are calculated by the proportion of investment of such asset about the total cost of the portfolio.

B. Proposed Algorithm

An evolutionary algorithm for integer decision variables based on NSGA-II [6] are used here to optimize the model of investment portfolios. The algorithm performs selection, mutation, crossover and cutting operators after generating the initial population using repair operators in order to guarantee solutions feasibility.

Each individual, considered as a possible solution in evolutionary algorithms, represents a portfolio and it is encoded in a structure that contains two sets: *assets set* with a fixed 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. The algorithm performs mutation, crossover and selection operations until it reaches a large number of generations, as specified, or until the hypervolume indicator [8] reaches a very small number.

The proposed operators are:

- The **selection** selects one individual per iteration using the binary tournament method in which two individuals are chosen at random, and the best of them (the individual belonging to the best frontier or those with the highest crowding distance, when both belonging to the same frontier) is selected to be part of the new population. This procedure is repeated until N individuals are selected, where N is the number of individuals in the population.
- In **crossover**, two individuals of the new population are chosen randomly and 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.
- Finally, the **cutting** 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.

IV. Computational Experiment and Numerical Results

Stock market trading simulations were conducted in the period between January 2011 and December 2015. At each beginning of the month, the portfolio optimization is re-executed, and another portfolio is selected according to a method of decision making. In each month the monthly Drawdown was recorded, and this procedure was repeated until the end of 2015 so that a total of 60 months is used for each method considered. The cumulative return for the whole period was also analyzed for each method.

Three different methods of decision making were used:

(i) Rank weights;
(ii) DM Queries;
(iii) Artificial Neural Network.

For the optimization model, the values established for the parameters are: number of assets $N = 53$; significance level $\alpha = 5\%$; minimum number of shares $m_i, \forall i, i = 1, ..., N = 100$; available capital $C = \$\,100,000.00$; cardinality $K = 9$ assets; fixed and proportinal transaction costs f and γ was defined as $\$\,29$ and 0.45% of the portfolio value, respectively, including the sum of the brokerage, emoluments, amount charged for the possession of shares and a monthly variable tax.

The parameters of NSGA-II are: 500 individuals; at most, 500 generations to stop; the probability of crossover pc and mutation rate mr are adaptively determined during the executions according to the methodology presented in [5].

A. Rank Weights

The first method tested in this work was the Rank Order Centroid Weights. From the Pareto-optimal front, the values of the attributes of all the alternatives were normalized, and the method was applied. One of this paper's authors did the DM function to get the results in all tested methods. This method was executed twice, one simulating a DM with return preference and another giving more value to the risk. Using Equation 5, the weights were 0.75 and 0.25 for return and risk respectively in the first execution and 0.25 and 0.75 in the second execution. Figure 3 shows the result of the cumulative returns of the simulation, including the period values for the Ibovespa and Selic to compare. Selic is the basic interest rate of the Brazilian economy, used in the interbank market to finance operations with daily duration, backed by federal government bonds. Figure 4 shows the drawdown for the executions.

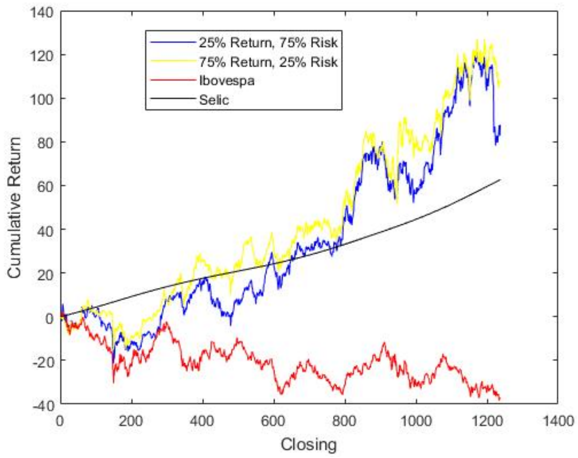

Figure 3. Cumulative returns using Rank Order Centroid Weights

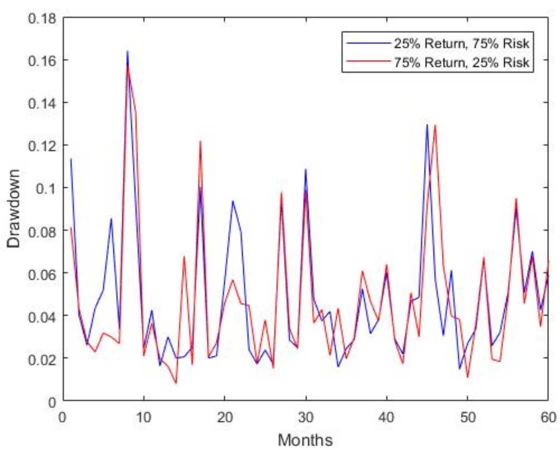

Figure 4. Drawdown using Rank Order Centroid Weights

B. DM Queries

The DM Queries method was also tested in this paper. Pairs of alternatives to DM were presented, starting with alternatives that represented 0.3 and 0.7 weights versus 0.5 and 0.5, with $\alpha = 0.2$ and decreasing 0.01 in each iteration. However, the DM was not aware of these values of weights, knowing only information regarding the return of that alternative in a historical period of one year and its greater loss in that period. The DM then chose one of them, directing their preference to one of the attributes. Next, the DM was presented with a new pair of alternatives and this scheme was repeated until the DM was satisfied. Following this method, four investor profiles were simulated, which in the end represented values of weights 0.25 - 0.75, 0.4 - 0.6, 0.6 - 0.4 and 0.75 - 0.25. Figures5 and Figure6 present the results obtained with this method.

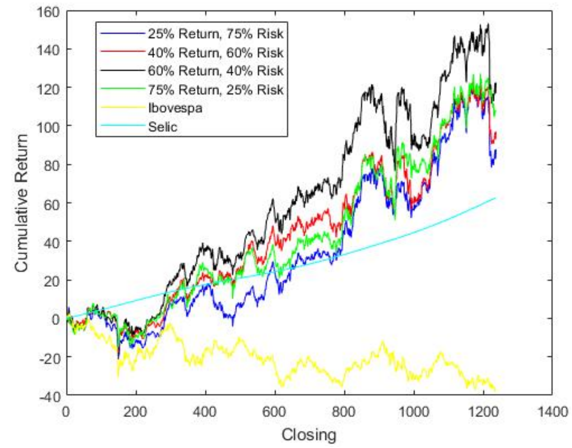

Figure 5. Cumulative returns using DM Queries

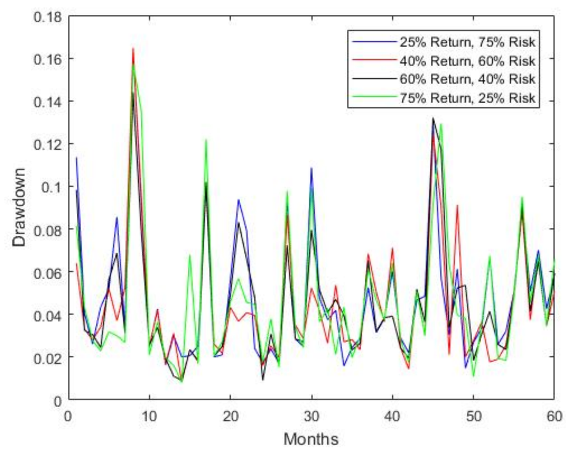

Figure 6. Drawdown using DM Queries

C. Artificial Neural Network

The ANN was built using the software MATLAB©. It was used a hidden layer with size 40 and the LM training method.

Levenberg-Marquardt optimization was chosen as the training method because it is often the fastest backpropagation algorithm in the toolbox, and it is highly recommended as a first-choice supervised algorithm, according to the official documentation of the software [14].

To verify the efficiency of the built ANN in approaching functions, a known function called "Sombreiro" (hat) was used, presented in Equation 9. The "Sombreiro" function was chosen because it is a bidimensional unimodal Gaussian and the DM's preference function should be unimodal as well. Also, the values for the preference regarding each alternative should be positive and worst alternatives should be zero, or close to it, and "Sombreiro" function behaves this way. The ANN was trained to approximate the original function. The graph of the original function was generated, the ANN was trained and used to approximate the original function. The graph of the original function and the approximate function are shown in Figure 7.

$$f(x, y) = \frac{sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}} \quad (9)$$

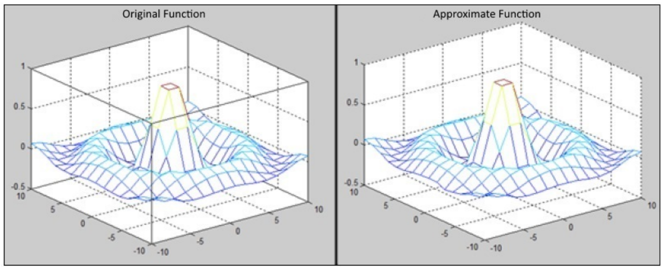

Figure 7. Original Function x ANN Approximate Function

After verifying the efficiency of the ANN, it was used to approximate the function that represented the weights 0.25 and 0.75. The comparative results are shown in figures 8 and 9. The third method is not included in this graph because the result for the Rank Order Centroid is the same as the DM Queries when the same decision maker applies the methods. They are two methods that can get the same result, being a mathematical method, through Equation 5, and another one extracting the information of the DM, but getting the same objective and consequently to the same result, represented by the weights 0.25-0.75. ANN, however, makes an approximation of the original function that models the decision maker's preference and, therefore, there may be small differences, justifying its presence in the graph.

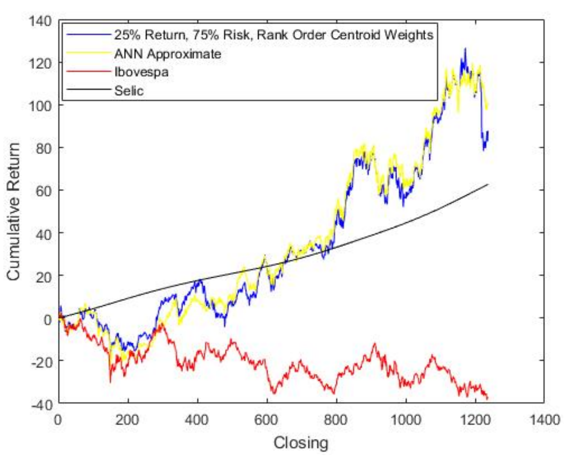

Figure 8. Cumulative Returns Comparison

V. Conclusions

With this work we were able to obtain expressive results with the optimization of investments, considering the preferences and particularities of the investor. In all the results and methods tested, the profit (cumulative return) obtained in the simulations after the optimization was satisfactory and higher than the Ibovespa, considered a safe investment in Brazil. It was possible to successfully implement the decision-making methods, selecting among the alternatives of the Pareto-frontier the one that best fits the profile of DM, based on the theory of utility.

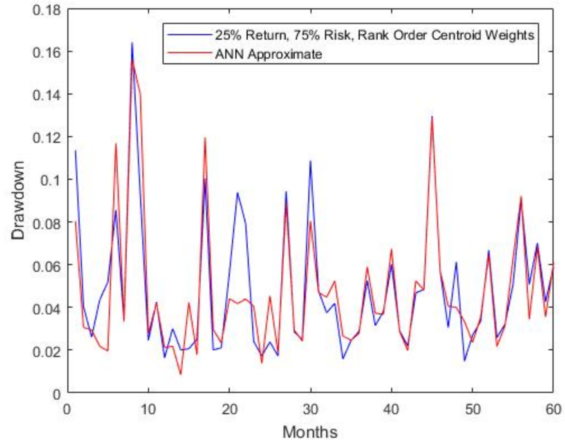

Figure 9. Drawdown Comparison

The result obtained using the Centroid Rank Order does not show a large difference between the accumulated returns for the two profiles presented, although the investor who values the return more than the risk, as expected, had a higher profit over of the simulation. However, since all the decisions were made considering optimized and non-dominated alternatives, both managed to obtain a good result.

When using the DM Queries method, the result was closer to the expected result, with the more conservative investor having a smaller cumulative return, but suffered a smaller decrease (drawdown) over most of the period.

It was also shown the efficiency of the use of ANN to approximate the function that represents the preferences of the DM and uses it in aid of decision making, obtaining with ANN results very close to the result obtained with the original function.

Finally, this article opens up a range of opportunities for further study and direct application of the methods presented. For future work, it is recommended to consider the preferences of the DM during the optimization process, coupling the decision-making methods in the financial optimization algorithm.

Acknowledgements

Authors would like to thank Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG.

References

[1] Byeong Seok Ahn. 2011. Compatible weighting method with rank order centroid: Maximum entropy ordered weighted averaging approach. *European Journal of Operational Research* 212, 3 (2011), 552 – 559. DOI: http://dx.doi.org/10.1016/j.ejor.2011.02.017

[2] F. Hutton Barron and Bruce E. Barrett. 1996. Decision Quality Using Ranked Attribute Weights. *Management Science* 42, 11 (1996), 1515–1523. DOI: http://dx.doi.org/10.1287/mnsc.42.11.1515

[3] BM&FBOVESPA. 2018. Índice Bovespa. http://www.bmfbovespa.com.br. (2018). Acessado: 2018-02-10.

[4] Runze Cheng and Jianjun Gao. 2015. On cardinality constrained mean-CVaR portfolio optimization, Vol. 27. Chinese Control and Decision Conference, IEEE, 1074–1079.

[5] Lucas de Carvalho Ribeiro, Alípio Monteiro Barbosa, and João Matheus de Oliveira Arantes. 2010. Algoritmo Genético Multiobjetivo. *Brazilian Conference on Dynamics, Control and their Applications* 9 (2010), 6.

[6] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. 2000. *A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II*. Springer Berlin Heidelberg, Berlin, Heidelberg, 849–858. DOI: http://dx.doi.org/10.1007/3-540-45356-3_83

[7] Fernando G. D. C. Ferreira, Bruno C. Barroso, Gustavo P. Hanaoka, Felipe D. Paiva, and Rodrigo T. N. Cardoso. 2017. Composition of investment portfolios through a combinatorial multiobjective optimization model using CVaR. *IEEE Congress on Evolutionary Computation* (2017), 8.

[8] J. L. Guerrero, L. MartÃ, A. Berlanga, J. GarcÃ-a, and J. M. Molina. 2010. Introducing a robust and efficient stopping criterion for MOEAs. In *IEEE Congress on Evolutionary Computation*. 1–8. DOI: http://dx.doi.org/10.1109/CEC.2010.5586265

[9] Simon Haykin and Paulo Martins Engel. 2001. *Redes Neurais - PrincÃpios e PrÃ¡tica* (2nd. ed.). Bookman, Porto Alegre, RS.

[10] Md Sazzad Hossain, Zhi Chao Ong, Zubaidah Ismail, and Shin Yee Khoo. 2017. A comparative study of vibrational response based impact force localization and quantification using radial basis function network and multilayer perceptron. *Expert Systems with Applications* 85 (2017), 87 – 98. DOI: http://dx.doi.org/10.1016/j.eswa.2017.05.027

[11] Nawaz Khan, Dhara Gaurav, and Thomas Kandl. 2013. Performance Evaluation of Levenberg-Marquardt Technique in Error Reduction for Diabetes Condition Classification. *Procedia Computer Science* 18 (2013), 2629 – 2637. DOI: http://dx.doi.org/10.1016/j.procs.2013.05.455 2013 International Conference on Computational Science.

[12] Neeraj Khera and Shakeb A. Khan. 2018. Prognostics of aluminum electrolytic capacitors using artificial neural network approach. *Microelectronics Reliability* 81 (2018), 328 – 336. DOI: http://dx.doi.org/10.1016/j.microrel.2017.11.002

[13] Harry Markowitz. 1952. Portfolio Selection. 7, 1 (1952), 77–91.

[14] MathWorks. 2017. TRAINLM - Levenberg-Marquardt backpropagation. (2017). Retrieved February 11, 2018 from https://www.mathworks.com/help/nnet/ref/trainlm.html

[15] Luciana Rocha Pedro. 2013. *The NN-DM Method â An Artificial Neural Network Model for Decision-Makerâs Preferences*. Ph.D. Dissertation. Faculty of Engineering at Universidade Federal de Minas Gerais.

[16] Luciana R. Pedro and Ricardo H. C. Takahashi. 2009. MODELLING THE DECISION-MAKER UTILITY FUNCTION THROUGH ARTIFICIAL NEURAL NETWORKS. In *Anais do 9 Congresso Brasileiro de Redes Neurais*. SBRN, Ouro Preto, MG, 1–5.

[17] R Tyrrell Rockafellar and Stanislav Uryasev. 2000. Optimization of conditional value-at-risk. *Journal of risk* 2 (2000), 21–42.

[18] Michael Scholz, Markus Franz, and Oliver Hinz. 2017. Effects of decision space information on MAUT-based systems that support purchase decision processes. *Decision Support Systems* 97 (2017), 43 – 57. DOI: http://dx.doi.org/10.1016/j.dss.2017.03.004

[19] Omid Seyedashraf, Abbas Rezaei, and Ali Akbar Akhtari. 2017. Dam break flow solution using artificial neural network. *Ocean Engineering* 142 (2017), 125 – 132. DOI: http://dx.doi.org/10.1016/j.oceaneng.2017.07.002

[20] Remy Spliet and Tommi Tervonen. 2014. Preference inference with general additive value models and holistic pair-wise statements. *European Journal of Operational Research* 232, 3 (2014), 607 – 612. DOI: http://dx.doi.org/10.1016/j.ejor.2013.07.036

[21] William G. Stillwell, David A. Seaver, and Ward Edwards. 1981. A comparison of weight approximation techniques in multiattribute utility decision making. *Organizational Behavior and Human Performance* 28, 1 (1981), 62 – 77. DOI: http://dx.doi.org/10.1016/0030-5073(81)90015-5

[22] Madjid Tavana, Amir-Reza Abtahi, Debora Di Caprio, and Maryam Poortarigh. 2018. An Artificial Neural Network and Bayesian Network model for liquidity risk assessment in banking. *Neurocomputing* 275 (2018), 2525 – 2554. DOI: http://dx.doi.org/10.1016/j.neucom.2017.11.034

[23] Craig W. Kirkwood and Rakesh K. Sarin. 1985. Ranking with Partial Information: A Method and an Application. 33 (02 1985), 38–48.

[24] Martin Weber. 1987. Decision making with incomplete information. *European Journal of Operational Research* 28, 1 (1987), 44 – 57. DOI: http://dx.doi.org/10.1016/0377-2217(87)90168-8