Learning Pairwise Comparisons with Machine Learning for Large-Scale Multi-Criteria Decision Making Problems

Marcos Antonio Alves
Graduate Program in Electrical Engineering
Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brazil
marcosalves@ufmg.br

Ivan Reinaldo Meneghini
Federal Institute of Education Science and Technology of Minas Gerais
Ibirité, Brazil
ivan.reinaldo@ifmg.edu.br

Frederico Gadelha Guimarães
Machine Intelligence and Data Science Laboratory (MINDS)
Department of Electrical Engineering
Universidade Federal de Minas Gerais
31270-901, Belo Horizonte, Brazil
fredericoguimaraes@ufmg.br

***Abstract*—Decision making is a complex task and requires a lot of cognitive effort from the decision maker. Multi-criteria methods, especially those based on pairwise comparisons, such as the Analytic Hierarchic Process (AHP), are not viable for large-scale decision-making problems. For this reason, the aim of this paper is to learn the preferences of the decision-maker using machine learning techniques in order to reduce the number of queries that are necessary in decision problems. We used a recently published parameterized generator of scalable and customizable benchmark problems for many-objective problems as a large-scale data generator. The proposed methodology is an iterative method in which a small subset of solutions are presented to the decision-maker to obtain pairwise judgments. This information is fed to an algorithm that learns the preferences for the remaining pairs in the decision matrix. The Gradient Boosting Regressor was applied in a problem with 5 criteria and 210 solutions. Subsets of 5, 7 and 10 solutions were used in each iteration. The metrics MSE, RMSE, MAPE and R2 were calculated. After the 8th iteration the ranking similarity stabilized, as measured by the tau distance. As the main advantage of the proposed approach is that it was necessary only 8 iterations presenting 5 solutions per time to learn the preferences and get an accurate final ranking.**

***Index Terms*—Preference learning, Multicriteria decision making, Many-objectives optimization, Machine Learning, Scalability**

I. Introduction

We are all decision-makers. In our daily lives, frequently, we deal with situations where we need to decide on a choice, selecting a single alternative and discarding the other ones. This choice is trivial when we have only one goal because we just need to order the values of that criterion and select the most advantageous alternative. However, in most cases, we are faced with dozens of alternatives and many criteria, making the ranking more complicated.

M.A. Alves declares that this work has been supported by the Brazilian agency CAPES. Also, this work has been supported by the Brazilian agencies (i) National Council for Scientific and Technological Development (CNPq); (ii) Coordination for the Improvement of Higher Education (CAPES) and (iii) Foundation for Research of the State of Minas Gerais (FAPEMIG, in Portuguese).

MINDS Laboratory – https://minds.eng.ufmg.br/

In the optimization field, the process of optimizing two or more conflicting objectives usually leads to a set of solutions, known as Pareto-optimal solutions, which cannot be ordered by a simple comparison of their nominal values. These solutions are incomparable, or non-dominated solutions, and often delivered by stochastic heuristic algorithms, particularly the Multi-Objective Evolutionary Algorithms (MOEAs). These solutions can come from benchmark functions (also called test functions or test suites), such as the Generalized Position-Distance (GPD) [1], an intelligent tool for generating scalable and customizable problems in Many-Objective Optimization (MaOP).

However, in real applications, the decision-maker (DM) is not interested in the overall Pareto front since the final decision that is going to be implemented is a single solution. A large number of variables, functions and parameters related to modeling information and in the application of decision-making methods requires a great effort in collecting, analyzing, evaluating and selecting the best alternatives for solving decision problems. Due to this, multi-criteria decision-making methods (MCDM) have been developed to help the decision-makers choose the most preferred alternative, that is, those that had the best scores in each objective. These solutions are under conflicting criteria of cost (lower value is better) or benefit (greater is better). Each method, as those revised in [2]–[4], use different strategies to offer, at the end of the process, an ordinal ranking of the alternatives that reflects the decision-maker preferences.

Derived from MCDM, multi-attribute decision making (MADM) uses several criteria to rank the available solutions and it is divided into three groups: (i) Multi-attribute utility theory (MAUT) whose best known method is the Analytic Hierarchy Process (AHP) [5], (ii) outranking and (iii) interactive methods [2]. From MAUT, there are a variety of utility functions (called MAUF) that can model the decision-maker preferences. Their form is expected to be quite complex [6]. For a better contextualization, please see subsection II-A.

In the AHP, for example, it is assumed that the preferences

of the DM can be represented by a value function defined indirectly through pairwise judgments among the available solutions. That is, the alternatives are presented two by two and the DM uses an scale, as the 9-point Saaty's scale, to elicit his/her preferences between them. At this point it is clear that as the complexity of the problem increases (in number of criteria and alternatives), the number of queries made to the DM also increases – and this translates into a greater cognitive effort. Due to several factors, such as exhaustion, inattention, ambition or even lack of knowledge of the problem during evaluations needed, another hindrance may arise: the comparisons can be inconsistent. This requires the DM to reevaluate a whole (sub)set of alternatives until the problem becomes consistent.

An important research line in this field is related with the approximation of the utility function that represents the preferences of the DM through computational intelligence. It involves the application of Machine Learning (ML) techniques for preference learning, such as Artificial Neural Networks (ANN) architectures [7], [8], ANNs with interactive evolutionary algorithms [9], [10] and those based on Support Vector Machines, named "SVM-rank" [11].

Even with the great advances, criticisms have been made about the difficulty in obtaining consistent pairwise matrices in the sense of faithfully reflecting the preferences of the DM, see for instance [12], [13]. Also, there is no consensus as to which is the best ML/AI technique. Efforts are in reducing the number of comparisons and making it reproducible to other domains, as in [7], [14], [15]. The main advantage is that it can model the preferences for any MAUF regardless of the form of decomposition it has, such as linear, as in the case of AHP, distance-based, fuzzy logic-based [14], for a single DM or a group of them (this field is known as Group MCDM, or GMCDM) [16], based on partial ordering or even outranking. The closest work with our proposal is [7] that aimed to help the DM to construct the preferences through an ANN architecture. However, this work does not introduce the idea of reducing inconsistency in evaluations and still suffers from scalability.

Thus, the purpose of this paper is to model the decision-maker preferences reducing the number of queries that are necessary in multi-criteria problems, minimizing the cognitive effort and making the problem scalable. To do so, a small number of solutions is presented per iteration and, once the MAUF is learned, the preferences of the remaining solutions are predicted with the trained model. As a test case, a set of solutions from a many-objective problem formulated with the GPD [1] is used. The expected result can be either one where the ranking does not vary as new solutions are inserted or one where the ranking generated is similar to that of the AHP.

This paper is organized as follows: Section II describes in II-A the literature overview about the MCDM methods and in II-B the revised papers that focus on learning the DM preferences. Section III presents the methodology, which is followed by a step-by-step method description. Section IV shows the results and Section V the conclusion and future works.

II. Modelling Preferences

A. Brief Overview about MCDM

Multi-criteria methods can be separated into three groups: 1) aggregation methods, whose main representatives are the multi-attribute utility theory, such as AHP [5], Analytic Network Process (ANP) [17] and Best-Worst Method (BWM) [18]; 2) outranking methods, such as the Preference Ranking Organisation Method of Enrichment Evaluations (PROMETHEE) [19] and Elimination Et Choix Traduisant la Realité (ELECTRE) [20] and their derivatives, and 3) interactive methods, such as multi-objective linear programming (MOLP) [21]. For the reader interested in an extensive review and classification of the multi-criteria decision-making methods, see [2] (in Portuguese).

Consider C_j as the j-th criterion and a_i the i-th alternative solution. In a deterministic decision problem, the evaluation of the alternative a_i under the criterion C_j is given by $C_j(a_i)$. This decision matrix, or $[D]_{ij} = C_j(a_i)$, illustrated in (1), is built at the beginning of the decision process and it represents the nominal values of each alternative in each criterion achieved in the objective space during the optimization.

$$D = \begin{array}{c} \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{array} \begin{array}{c} \begin{array}{cccccc} C_1 & C_2 & \dots & C_j & \dots & C_m \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1j} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nj} & \dots & x_{nm} \end{bmatrix} \end{array} \quad (1)$$

where a_i, $i = 1, \dots, n$ are the feasible alternative solutions, C_j, $j = 1, \dots, m$ are the criteria, x_{ij} is the performance of the alternative a_i under criterion C_j in the objective space.

For some methods, the utility function is built by the multiplication of the nominal values of the alternatives by a vector of weights $W = (w_1, \dots, w_m)$ that represent the importance of each criterion, being $w_j \geqslant 0$ and $\sum_{i=1}^{m} w_j = 1$. The greater the weight, the greater the importance of that criterion. Given the decision matrix $[D]_{ij}$, some methods do not take into consideration the vector of weights, which is built internally. In its way, each method classifies and ranks the available alternatives.

However, for the aggregation methods, the literature has reported several inconsistencies in the formulation of these matrices. Inconsistency, as explained by [21], arises when some opinions of a comparisons matrix contradict others and, surely, the larger the matrix size the greater the chances of inconsistency in the evaluations. These authors described two distinct situations that the inconsistency may appear: inter-criteria and intra-criterion. The former involves the elicitation procedure for determining the weights of the criteria and in the latter, a value function is determined for each criterion.

The most exhausting way to solve the decision problem is to make all the queries to the decision-maker, who needs to explicit his/her preferences among the alternatives and

criteria with respect to the goal (or their preferences, if we consider GMCDM). If there is any inconsistency, the DM needs to redo the process. Although this kind of application is very common in the literature, including for methods that go through the calculation of pairwise comparisons, we have clearly a scalability problem for large problems, i.e. when the number of alternatives or criteria is huge.

In real-world practical applications, the time that analysts spend with DMs is increasingly scarce, and convincing an executive to spend hours, or even days, making judgments for alternatives and criteria is often not feasible. Several studies have reported concerns related to the applicability of this type of procedure in situations where the number of criteria and alternatives is quite large, e.g. [21] to cite a few.

B. Learning Decision-Maker Preferences

Learning or eliciting preferences, as stated by [22], are a central concept for decision making and it is related to "acquiring preference information in either a direct or indirect way, from preference statements, critiques to examples, observations of user's clicking behaviour, etc". In decision theory, particular emphasis has been given to assessing the decision maker's utility function very precisely. The preferences are conveyed through preference statements or queries made to the decision-maker.

As pointed out by [6], the simplest utility function is described in Equation (2)

$$\sum_{j=1}^{J} w_j x_{ij} \tag{2}$$

for i of N alternatives, J is the number of criteria, w_j is the relative weight of criterion j and x_{ij} is the value of the alternative i on criterion j. This function is an increasing function and linear (convex). However, there are a variety of forms of utility functions and they are expected to be quite complex. The idea behind the automatic approaches that we are going to discuss next is to find a model that makes a good approximation of these utility functions, regardless of how it is modelled (that is, the way the decision-maker thinks).

Years ago, Chen and Lin [7] pointed out evidence that ANNs could be a promising tool in solving MCDM problems in terms of approximating the multi-attribute utility function (MAUF) and representing the preference of the decision-maker. An interactive decision neural network (DNN) was proposed and used for reducing the decision-maker cognitive burden by capturing and representing the decision-maker's preference. Similarly, Matsuda [23] proposed a neural network model based on the ANP. The architecture was used for solving complete and incomplete matrices by converting the MCDM method in an optimization problem. In [8], an "Improved Decision Neural Network" (IDNN) focused on multicriteria group decision making was investigated aiming to reduce the number of iterations to map the utility function from the multicriteria methods as well as the exploitation of indirect methods for learning preference relations.

A Multilayer Perceptron (MLP) architecture was proposed in [9] to capture information of the decision-maker. The function was built based on a partial sorting process, denoted ranking. With this ranking, the architecture was employed to model the partial sorting of the alternatives. Later, in [10], the authors presented the "NN-DM", an MLP-based approach to approximate arbitrary preference functions, including those in which there are nonlinear dependencies among different decision criteria. The Kendall-tau distance [24] was used to measure the similarity between the answers obtained with the ANN and the analytical model. In both pieces of research, the approach was able to deliver new answers to alternatives that have not been presented yet, even in new and similar domains. In [14], an ANN was used to model the decision-maker preferences in a supplier selection problem. The goal was to learn the preferences for each criterion given by fuzzy assessments and to produce a score (model output) for each supplier. The architecture provided a satisfactory mean square error (MSE) metric and can be used for future applications.

Through the aforementioned works, it can be seen that the efforts regarding the decision-maker's preference modelling are centred on simplifying the complex nature of the AHP, such as the AHP-Express [25], observations of past DM behaviours [11], combined MCDA model and ML to achieve better prediction performance while capturing the relationships between individual attributes and the prediction [26] or even in an interactive optimization process such as in [27] that showed preferred region in the Pareto front to the DM and built the preference relationships.

This present study is significantly different from those listed, all of which make use of pairwise preferences but each addressing a different goal. None of these studies focused on learning utility functions in the view of making the problem more scalable and reducing the cognitive effort of the DM. The baseline is the analytical AHP model that requires $n(n-1)/2$ evaluations to compare the alternatives in respect to the criteria. Instead of making all the pairwise comparisons, the idea is to ask the DM the minimum number of times necessary to obtain a sufficient number of samples that the ML method can learn the function that represents his/her preferences.

III. METHODOLOGY

In this paper, it is considered that a typical MCDM approach arranges a finite number of alternatives in a preferable sort way. These preferences can be represented by using a two-dimensional matrix as in (1). Also, it is considered a *post hoc* decision making problems. It refers to the choice of the alternative after the optimization process, where the decision-maker has information about both the decision space and objective space.

The alternatives used in this work come from the GPD [1]. A many-objective problem was formulated to generate a more complex problem. Using a MOEA algorithm, a set of feasible solutions were generated. Then, a ML method is employed to learn the MAUF while the DM explicit his/her preferences among the alternatives. Subsets of Q alternatives are presented

per time. In the 1st iteration, the first Q alternatives presented to the DM are used as input (training step) for the ML method and the remaining ones for the test. Some error metrics are calculated as well as the similarity of the rankings which indicates convergence. In the next iteration, more Q alternatives are aggregated, the DM makes the pairwise comparisons and the training input increases.

Figure 1 illustrates the main steps of this work. For sake of simplicity, a step-by-step is explained below.

- *Step 1 : Obtain a set of feasible alternatives*
  This step is about the optimization process. Here, GPD was used following the steps presented by the authors in [1]. The parameters were: $N = 13$ variables in decision space. $M = 5$ objectives, all of which are minimization, 210 non-dominated solutions were selected, using the MOEA algorithm. A convex Pareto front is generated setting up the p-norm equal to 2.
- *Step 2: Apply a MOEA*
  It was used the NSGA-III from PlatEMO [28]. The stop criterion was the maximum number of function evaluations ($5e6$). Note that the alternatives could be (i) analytically obtained through the GPD and (ii) from other optimization problems.
- *Step 3: Get N alternatives from PF*
  N solutions were selected after the stop criterion. The solutions in the variable space are distributed in the $[0-1]$ range.
- *Step 4: Select Q randomly*
  In the 1st iteration, Q alternatives are randomly selected from N. They represent the alternatives that are going to be presented to the DM to build the pairwise comparisons by means of queries. Different values for Q were evaluated, being $Q = [5, 7, 10]$.
- *Step 5: Generate PC matrices for Q*
  The usual process for constructing the pairwise comparison matrices is to present the alternatives, two by two, asking questions such as: "*How much do you prefer the alternative A_i over A_j?*". For this part, it was used the nominal values of the alternatives in the objective space labelling the answers according to the known 9-points Saaty's scale. This scale can be found in [2], [5]. For short: for each objective (i) calculate the minimum and maximum values achieved for the alternatives, (ii) calculate 9 intervals among these values, representing the ranges in the Saaty's scale, (iii) calculate the difference among all the alternatives – building a matrix with dimensions Q x Q, (iv) replace the differences by the corresponding values of the Saaty scale (step ii).
- *Step 6: Apply a ML method*
  The ML algorithm is applied to predict the targets for all objectives given the input, that is the concatenation of the alternatives in the variable space. To illustrate, consider $Q = 2$ and the chosen alternatives were A_1 and A_2, it leads to 4 entries: $[A_1 + A_2]$, $[A_2 + A_1]$, $[A_1 + A_1]$ and $[A_2 + A_2]$ totaling Q^2 entries.
  It was used the Multi-Output Regressor class and Gradient Boosting Regressor as the estimator. The Randomized Search CV class was used to search the best hyperparameters for this model among those described in Table I. In the end, the tuned model is used to predict the outputs, i.e. the preference value for each objective. As error metrics, it was measured the known MAPE (Mean absolute percentage error), MSE (Mean squared error), RMSE (Root MSE) and R2 (coefficient of determination - regression score function) available in the Scikit-learn package [29].

TABLE I: List of hyperparameters

Parameters	**Values tested**
Learning rate	[0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 0.9]
Loss	['ls', 'lad', 'huber']
# of estimators	[10, 50, 100, 300, 500, 700, 1000]
Criterion	[friedman_mse, mse]
Min samples split	[2, 4, 7, 10]
Max depth	[3, 5, 10, 15, 20, 30]
Min samples leaf	[1, 2, 3, 5, 8, 10]
Min impurity decrease	[0, 0.2, 0.4, 0.6, 0.8]
Max leaf nodes	[5, 10, 20, 30, 50, 100, 300]

  The parameters used for the Randomized Search CV were: (i) model: Gradient Boosting Regressor, (ii) hyperparameters: see Table I, (iii) random_state = 42, (iv) n_iter = 10, (v) refit = $True$, (vi) cv = 5, (vii) verbose = $True$, (viii) pre_dispatch = $2 * n\_jobs$, (ix) error_score = $raise$, (x) return_train_score = $True$.
- *Step 7: Make predictions*
  The best model is used to predict the $N * N - Q * Q$ samples. The idea behind this step is that a model that has used few alternatives (and consequently a few evaluations and comparisons) is capable of performing a good prediction. This means that the model was able to learn the MAUF that represents the decision maker's preferences.
- *Step 8: Apply MCDM and generate ranking*
  The predicted preferences among the alternatives in each objective are equivalent to that of the analytical AHP method. So, the AHP steps are followed to generate the ranking. Basically, it consists of the creation of the normalized matrix dividing all values by the total of the column, add the values per row and sort the alternatives. In this step, we do not take into account the importance of each criterion. Since all the objectives are minimization, this strategy is equivalent to using equal weights for all of them, that is $w_j = 1/M$.
- *Step 9: Calculate tau between rankings*
  In this step, the Kendall tau distance [24] was used to measure the (dis)similarity between the rankings. Tau distance is a known metric that counts the number of pairwise disagreements between two ranking lists. In this work, equal rankings mean that the algorithm has converged and the predictions the ML regressor is making match the DM's preferences in all the objectives.
- *Step 10: Stop criteria*

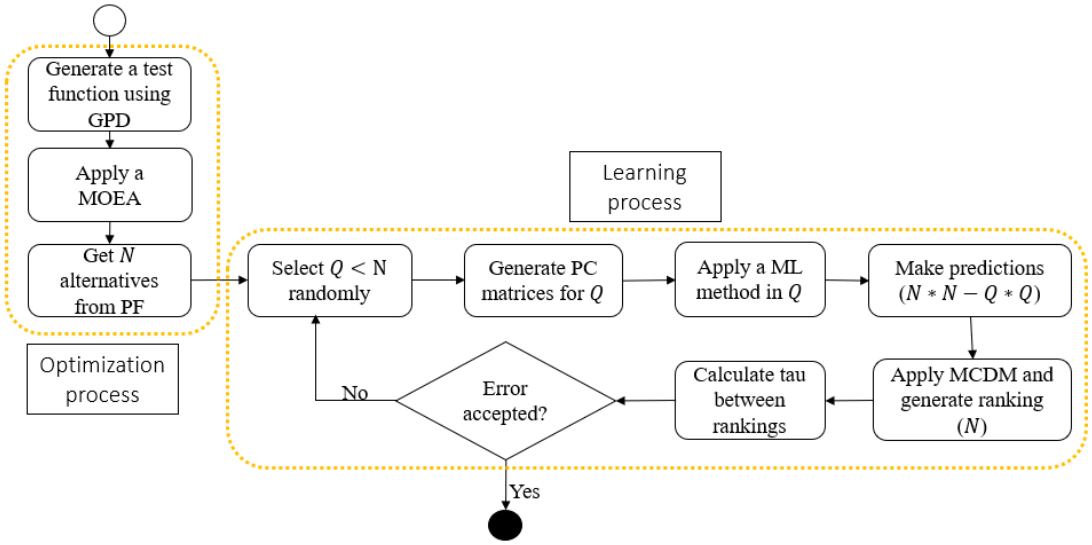

Fig. 1: Detailed overview of the proposed methodology for learning decision-maker preferences considering (i) the optimization process and (ii) machine learning procedures

If the difference between the rankings is acceptable, then the process is finished. Otherwise, Q new alternatives are presented. 5% of difference was setted. In means that the DM accept no more than this difference between the rankings.

The optimization part was implemented in Matlab by using the PlatEMO. For vizualization, CAPviz tool [30] was used. CAPviz integrates **C**hord diagram and **A**ngular mapping and **P**arallel coordinates into a single two-dimensional circular chart. In the Chord diagram, the range of each coordinate is normalized and arranged in a circle and the coordinate values are connected utilizing the Bezier curves. Angular mapping allows the observation of the distribution of points in space, its angular proximity to each axis and its proximity to the origin of the Cartesian coordinate system. In the Parallel coordinates, a point in n-dimensional space is represented as a polyline with vertices on the parallel axes. The position of the vertex on the i-th axis corresponds to the i-th coordinate of the point. An R script of the CAPviz is available at [31].

IV. Results and Discussion

The set of 210 solutions obtained with the GPD is illustrated through the CAPviz [30] in Figure 2. Once a p-norm equal to 2 was used, it generates a convex Pareto front. Based on a visual analysis on the Angular Mapping (norm in vertical per angle in horizontal), it is possible to see that the alternative solutions have a good dispersion of the points in the approximated Pareto front along the 5 objectives. Observing the Parallel Coordinates graph it can be seen that the points are detached from the others. It is also reflected in Chord Diagram.

With those alternatives, subsets of sizes 5, 7 and 10 were evaluated, symbolizing the number of solutions presented to the DM per time (or iteration). The key idea, as aforementioned explained, was to use just some solutions to learn the

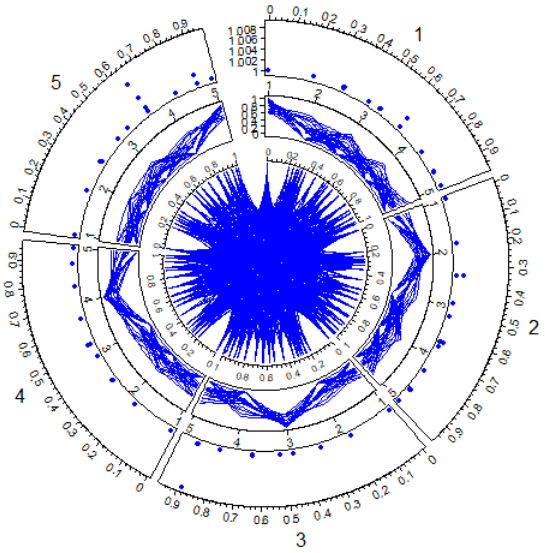

Fig. 2: Distribution of the 210 solutions over the 5 objectives from the optimization problem using CAPviz tool

DM preferences and to predict the remaining ones. Considering the PF with 210 solutions, when 5 of them were used for training (it means that they were evaluated by the DM), the rest, or 205, used for test, i.e., predicted by the ML regressor. In a new iteration, more 5 were presented to the DM that made the pairwise comparisons and these preferences were aggregated with the previous subset. Updated, evaluations of 10 solutions were used for training and 200 for test and so on.

Figure 3 illustrates the results for MSE, RMSE, R2 and MAPE while subsets of 5 solutions were presented to the DM (represented by the iterations). Figure 4, in its turn, represents

the tau distance both with the AHP if the classical method had been used (Current vs AHP) and with the previous iteration (Current vs Previous). Note that after the 8th iteration there was a little variation in the metrics ($< 5\%$), indicating the convergence. It other words, at this time, it is possible to say that only 8 queries were made to the decision maker presenting 5 evaluations at a time. To know, if the AHP was used in its classical form, 210 evaluations were needed in a single round.

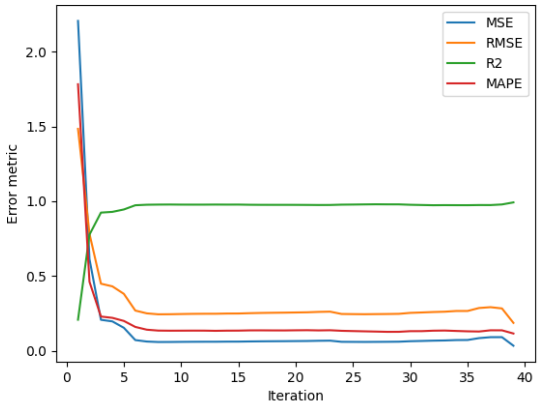

Fig. 3: Error metrics with 5 solutions per time during the iterations

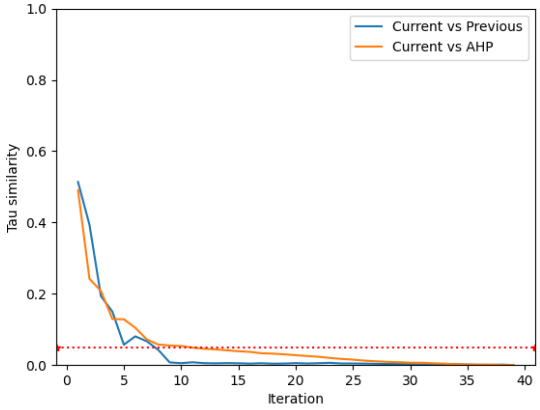

Fig. 4: Tau distance with 5 solutions per time during the iterations

Different values for Q were tested. By presenting $Q = 7$, the method converged after 7 iterations. The difference between the rankings was below 5% until 10 iterations. For $Q = 10$, the convergence was after the 5th iteration, meaning that 5 queries were made by presenting 10 solutions per time.

The ML regressor model used in the Figure 3, trained after 8 iterations and 5 solutions per time, was employed to predict the remaining ones. Once the preference matrix was complete (real + predicted), the AHP was applied and the ranking calculated. The top 10 alternatives are shown in the Figure 5 and listed in the Table II.

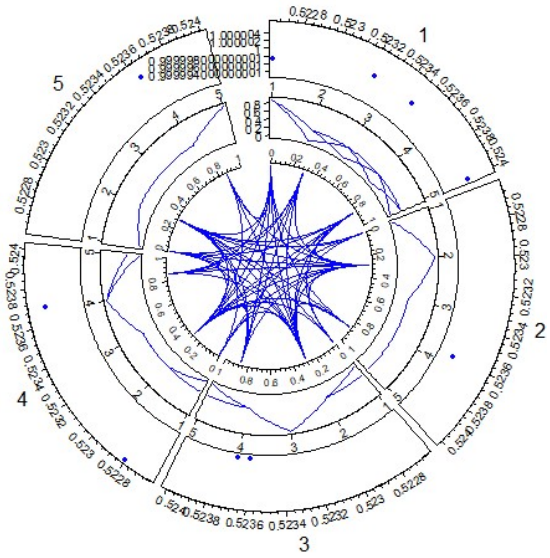

Fig. 5: Top 10 solutions obtained with the preferences predicted through the proposed approach after make 8 queries to the DM presenting 5 solutions per time

TABLE II: Ranking of the top 10 alternatives and its values in the objective space

Rank.	Ai	Obj1	Obj2	Obj3	Obj4	Obj5
1	76	0.8658	0.2893	0.0000	0.2887	0.2887
2	0	0.8662	0.2881	0.2888	0.2886	0.0000
3	164	0.8665	0.2882	0.2877	0.0000	0.2887
4	2	0.0000	0.2887	0.8660	0.2887	0.2887
5	135	0.8661	0.0000	0.2885	0.2887	0.2887
6	194	0.0000	0.2887	0.2887	0.2887	0.8660
7	19	0.0000	0.8660	0.2887	0.2887	0.2887
8	22	0.0000	0.2881	0.2881	0.8664	0.2886
9	6	0.2887	0.2887	0.8660	0.2887	0.0000
10	140	0.2888	0.2888	0.2886	0.8660	0.0000

To compare the rankings between the proposed approach with those that would be ranked by the classical AHP, Table III shows the top 10 alternatives. Note that the alternatives are not exactly in the same position, but there are known in both rankings – in the sense that the same alternatives, out of 210, are well ranked by both approaches. As the main advantage of the proposed approach, it was needed only $8 * 5^2$ comparisons to get this ranking, instead of 210^2, which makes the problem much more scalable and realistic to be implemented in real-world problems. In addition, the approach can be extended in new problems with similar domains without the need of training, which saves time and makes the decision-making process more agile.

V. Conclusion

Even with the advance of the methods for solving optimization problems, choose only one alternative from the entire

TABLE III: Comparison of the top 10 alternatives using the predicted preferences through the proposed approach and only with the AHP

Ranking	**1o**	**2o**	**3o**	**4o**	**5o**	**6o**	**7o**	**8o**	**9o**	**10o**
Predicted	76	0	164	2	135	194	19	22	6	140
AHP	2	76	0	61	164	140	44	83	135	6

Pareto set to be implemented is still a difficult task for the decision-maker. Some multicriteria methods, such as the AHP, requires too many efforts in the evaluation of the solutions that makes it impracticable for large problems. This paper aimed to help in building the preference matrices among the alternatives making the problem more scalable and saving the decision maker's time.

Queries were made to the DM with small subsets of alternatives per time. According to these initial evaluations, a machine learning regressor model learnt the utility function and was able to predict the preferences for the remaining alternatives. After 8 queries with 5 alternatives per time, the ranking with the predicted values was about 95% similar with that using the classical AHP. The main difference is that in the proposed approach it was necessary $8 * 5^2$ evaluations, instead of 210^2. We believe that five or seven solutions is a more plausible number to be estimated than dozen/hundred of them. In fact, in practical problems, it is unlikely that such large sets of solutions will be presented to the decision maker for pairwise preference construction. Therefore, the idea of dealing with smaller sets is more plausible.

For future studies, new possibilities can be explored. To cite a few: (i) new geometries to reflects other decision-makers, (ii) to extend to GMCDM, (iii) spatial location of the solutions and the DM behaviour, and (iv) although boosting is one of the strongest ML-techniques, other algorithms and approaches could be tested to see if there is an improvement in terms of performance.

For reproducibility, the codes are available at https://github.com/mvoicer/cbic-2021-learning-preferences

REFERENCES

[1] I. R. Meneghini, M. A. Alves, A. Gaspar-Cunha, and F. G. Guimarães, "Scalable and customizable benchmark problems for many-objective optimization," *Applied Soft Computing*, vol. 90, p. 106139, 2020.

[2] M. A. Alves, "Proposta de agregação robusta de múltiplos métodos com incertezas em problemas de tomada de decisão multicritério," Master thesis, Universidade Federal de Minas Gerais, 5 2018. [Online]. Available: 10.5281/zenodo.3762766

[3] M. R. Asadabadi, E. Chang, and M. Saberi, "Are mcdm methods useful? a critical review of analytic hierarchy process (ahp) and analytic network process (anp)," *Cogent Engineering*, vol. 6, no. 1, p. 1623153, 2019.

[4] M. Singh and M. Pant, "A review of selected weighing methods in mcdm with a case study," *International Journal of System Assurance Engineering and Management*, pp. 1–19, 2020.

[5] T. L. Saaty, "What is the analytic hierarchy process?" in *Mathematical models for decision support*. Springer, 1988, pp. 109–121.

[6] D. L. Olson, "Multiattribute utility theory," in *Decision Aids for Selection Problems*. Springer, 1996, pp. 19–33.

[7] J. Chen and S. Lin, "An interactive neural network-based approach for solving multiple criteria decision-making problems," *Decision Support Systems*, vol. 36, no. 2, pp. 137–146, 2003.

[8] R. K. Singh, A. Choudhury, M. Tiwari, and R. Shankar, "Improved decision neural network (idnn) based consensus method to solve a multi-objective group decision making problem," *Advanced Engineering Informatics*, vol. 21, no. 3, pp. 335–348, 2007.

[9] L. R. Pedro and R. H. Takahashi, "Modelling the decision-maker utility function through artificial neural networks," in *Anais do IX Congresso Brasileiro de Redes Neurais/Inteligência Computacional (IX CBRN)*, vol. 1, 2009, pp. 550–563.

[10] ——, "Modeling decision-maker preferences through utility function level sets," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2011, pp. 550–563.

[11] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 133–142.

[12] W. W. Koczkodaj, J.-P. Magnot, J. Mazurek, J. F. Peters, H. Rakhshani, M. Soltys, D. Strzałka, J. Szybowski, and A. Tozzi, "On normalization of inconsistency indicators in pairwise comparisons," *International Journal of Approximate Reasoning*, vol. 86, pp. 73–79, 2017.

[13] C. Lin, G. Kou, and D. Ergu, "An improved statistical approach for consistency test in ahp," *Annals of Operations Research*, vol. 211, no. 1, pp. 289–299, 2013.

[14] D. Golmohammadi, "Neural network application for fuzzy multi-criteria decision making problems," *International Journal of Production Economics*, vol. 131, no. 2, pp. 490–504, 2011.

[15] L. R. Pedro and R. H. Takahashi, "Inspm: An interactive evolutionary multi-objective algorithm with preference model," *Information Sciences*, vol. 268, pp. 202–219, 2014.

[16] A. B. Leoneti, "Utility function for modeling group multicriteria decision making problems as games," *Operations Research Perspectives*, vol. 3, pp. 21–26, 2016.

[17] T. L. Satty, "Decision making with dependence and feedback: The analytic network process," *RWS Publication*, 1996.

[18] J. Rezaei, "Best-worst multi-criteria decision-making method," *Omega*, vol. 53, pp. 49–57, 2015.

[19] J. P. Brans and B. Mareschal, "The promethee methods for mcdm; the promcalc, gaia and bankadviser software," in *Readings in multiple criteria decision aid*. Springer, 1990, pp. 216–252.

[20] B. Roy, "Classement et choix en présence de points de vue multiples," *Revue française d'informatique et de recherche opérationnelle*, vol. 2, no. 8, pp. 57–75, 1968.

[21] G. R. Vasconcelos and C. M. d. M. Mota, "Exploring multicriteria elicitation model based on pairwise comparisons: Building an interactive preference adjustment algorithm," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[22] G. Pigozzi, A. Tsoukias, and P. Viappiani, "Preferences in artificial intelligence," *Annals of Mathematics and Artificial Intelligence*, vol. 77, no. 3, pp. 361–401, 2016.

[23] S. Matsuda, "A neural network model for the decision-making process based on anp," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006, pp. 1421–1426.

[24] M. G. Kendall, "A New Measure of Rank Correlation," *Biometrika*, vol. 30, pp. 81–93, 1938.

[25] J. E. Leal, "Ahp-express: A simplified version of the analytical hierarchy process method," *MethodsX*, vol. 7, p. 100748, 2020.

[26] M. Guo, Q. Zhang, X. Liao, F. Y. Chen, and D. D. Zeng, "A hybrid machine learning framework for analyzing human decision-making through learning preferences," *Omega*, p. 102263, 2020.

[27] L. R. Pedro and R. H. Takahashi, "Decision-maker preference modeling in interactive multiobjective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2013, pp. 811–824.

[28] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[30] I. R. Meneghini, R. H. Koochaksaraei, F. G. Guimarães, and A. Gaspar-Cunha, "Information to the eye of the beholder: data visualization for many-objective optimization," in *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2018, pp. 1–8.

[31] ——. Cap_vis.r. [Online]. Available: https://minds.eng.ufmg.br/download_resources