mathematics

MDPI

Article

Machine Learning-Driven Approach for Large Scale Decision Making with the Analytic Hierarchy Process

Marcos Antonio Alves [1,†,‡], Ivan Reinaldo Meneghini [2,†,‡], António Gaspar-Cunha [3,†,‡] and Frederico Gadelha Guimarães [4,*,†,‡]

[1] Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais, Av. Antonio Carlos 6627, Belo Horizonte 31270-901, MG, Brazil

[2] Federal Institute of Education Science and Technology of Minas Gerais (IFMG), Campus Ibirité, Ibirité 32407-190, MG, Brazil

[3] Institute of Polymers and Composites, University of Minho (Uminho), Campus Azurém, 4800-058 Guimarães, Portugal

[4] Department of Electrical Engineering, Universidade Federal de Minas Gerais, Belo Horizonte 31270-000, MG, Brazil

[*] Correspondence: fredericoguimaraes@ufmg.br

[†] Current address: Machine Intelligence and Data Science (MINDS) Laboratory, Federal University of Minas Gerais, Belo Horizonte, Brazil.

[‡] These authors contributed equally to this work.

Abstract: The Analytic Hierarchy Process (AHP) multicriteria method can be cognitively demanding for large-scale decision problems due to the requirement for the decision maker to make pairwise evaluations of all alternatives. To address this issue, this paper presents an interactive method that uses online learning to provide scalability for AHP. The proposed method involves a machine learning algorithm that learns the decision maker's preferences through evaluations of small subsets of solutions, and guides the search for the optimal solution. The methodology was tested on four optimization problems with different surfaces to validate the results. We conducted a one factor at a time experimentation of each hyperparameter implemented, such as the number of alternatives to query the decision maker, the learner method, and the strategies for solution selection and recommendation. The results demonstrate that the model is able to learn the utility function that characterizes the decision maker in approximately 15 iterations with only a few comparisons, resulting in significant time and cognitive effort savings. The initial subset of solutions can be chosen randomly or from a cluster. The subsequent ones are recommended during the iterative process, with the best selection strategy depending on the problem type. Recommendation based solely on the smallest Euclidean or Cosine distances reveals better results on linear problems. The proposed methodology can also easily incorporate new parameters and multicriteria methods based on pairwise comparisons.

Keywords: scalable decision making; pairwise matrices; Multi-Attribute decision methods; online machine learning; Analytic Hierarchy Process

MSC: 90B50

check for updates

Citation: Alves, M.A.; Meneghini, I.R.; Gaspar-Cunha, A.; Guimarães, F.G. Machine Learning-Driven Approach for Large Scale Decision Making with the Analytic Hierarchy Process. *Mathematics* **2023**, *11*, 627. https://doi.org/10.3390/math11030627

Academic Editor: Violeta Keršulienė, Zenonas Turskis

Received: 26 December 2022
Revised: 17 January 2023
Accepted: 17 January 2023
Published: 26 January 2023

1. Introduction

According to the Paradox of Choice [1], the greater the number of alternatives available, the harder the decision process is going to be. The Decision Maker (DM) needs to have different alternatives to choose from. However, if the alternatives are too many, the process can become time-consuming and tedious; also, the evaluations may be inconsistent over time [2,3]. Furthermore, in many projects, such as engineering problems, the DM (or user, manager, consumer, specialist, etc.) is required to participate in several stages of the process, requiring even more time to obtain the ranking of the different alternatives.

In decision theory, there are several Multicriteria Decision Making (MCDM) methods that help to rank the existing solutions for multifaceted problems. The ones based on the Multi-Attribute Utility Theory (MAUT) subclass assume the existence of a degree of utility that reflects the preferences of the DM. It is often used to identify trade-offs and to obtain a utility value for items or alternatives over more than one criterion in a consistent way.

In the MAUT subclass, the Analytic Hierarchy Process (AHP) [4] is the best-known and most widely used method [5–10]. It is based on pairwise comparison matrices, a divide-and-conquer technique that analyzes two alternatives at a time in order to determine their relative utilities or preferences. However, the high number of queries (NQ) that the DM is required to make at once to construct the comparison matrices can make the method complex and unfeasible for large-scale problems [2,11–14]. The main criticisms of the use of this method are related to the high cognitive effort [15], inconsistency [2,3,11,16,17], and the time required by the specialists [2,15].

The AHP described later in Section 2.1 has been applied in different areas such as the polymer extrusion process [18], sustainable supplier selection [9], engineering [5,19], operational risk in power substations [20], online shopping [10], plant location [21], undergraduate elective course planning [22], the management of architectural heritage in smart cities [23], supplier selection in the automotive industry [24], universities rankings [13], and others [2,5,25,26]. However, this method suffers from a notable drawback: it requires $n \times (n-1)/2$ comparisons of n alternatives for each criterion to solve the decision problem [11]. The time investment of experts, the required cognitive effort, and the possibility of ambiguity in the judgments pose challenges to using this method in large-scale problems [2,10–12,15,25].

Learning the DM's preferences has been an alternative to making the MCDM methods, particularly those based on pairwise comparisons that are practical to use. Comparisons take a long time to perform, and currently, there are no guarantees on scalability in larger problems [2,15]. In companies, data scientists and statisticians often need to find simpler ways to present the best alternatives to managers. This is achieved by excluding dominated solutions, selecting the most diverse ones on the Pareto front, and those closest to the utopian or another reference point to facilitate the sorting of the solutions from the least to the most desirable. Nevertheless, the problem remains difficult, especially for Multi-Objective (MOO) (two and three objectives) and Many-Objective Optimization problems (MaOP) (four or more objectives) [27] in which the alternatives are all in the Pareto front and the preferences are not known.

Accordingly with Tuljak-Suban and Bajec [12], "decision makers need a relatively simple, reliable method which is not terribly time consuming". For this, an approach that couples the AHP multicriteria method and online machine learning to facilitate the task of obtaining preferences is proposed in this paper. The scalability is improved by presenting fewer alternatives to the DM. In the classical approach, all of the solutions are presented and evaluated at once. It is not feasible for large-scale problems, since those decisions must be taken rapidly and always with timely information.

To surpass these challenges, we employ a machine-learning-driven approach that acts as an interactive process. Between the two phases, small subsets of solutions are chosen using different strategies to query the DM and to capture their preferences. An existing regression method is applied to learn these relationships while trying to predict the remaining ones. In other words, the model learns the Multi-Attribute Utility Function (MAUF) that represents the DM, but it only considers some comparisons per iteration. This reduces the number of comparisons compared to the original method. To measure the agreements and disagreements between the rankings, the Kendall tau (KDT) distance [28] is used, as in [15,29]. Similar rankings mean that the model was able to approximate the DM's behavior.

In the related literature, the learning phase is viewed as an offline process [30], the preferences relations are treated as binary [29,31], and there are no strategies in the selection of new alternatives [15]. Although it reduces the problem dimensions, it also restricts the preferences between one alternative to another, rather than a range of preferences, and it may still require many evaluations compared to the original method.

The main contributions of this paper are summarized as follows.

1. The proposal of a new version of the AHP method to make it scalable;
2. Recommendation strategies and different solution selections are applied in an interactive process to learn more about the DM's preferences;
3. A reduction in the number of solutions that need to be evaluated;
4. A reduction in the time and cognitive effort to evaluate the solutions until solving the decision problem;
5. Re-use of the trained model without new queries to the DM is possible in problems with the same domain.

The remainder of this work is as follows: Section 2 presents the bibliography review, introduces the AHP method and related works; Section 3 details the proposed approach; Section 4 presents the results and a discussion of the main findings; and Section 5 concludes the paper and points out new directions.

2. Background

2.1. Introduction to the Analytic Hierarchy Process

MCDM is a two-part method that includes MOO and MaOP problems in the first part, and Multiple Criteria Decision Analysis in the second part [32]. Optimization plays an important role in the design cycle, and solving large-scale problems poses challenges among practitioners [27,33]. Modeling the problems under multiple objectives and disciplines is known as MCDM [32]. Fundamentally, the goal of these methods is to solve a decision problem and to help the DMs choose a solution that best portrays their preferences among all of the objectives.

There are several multicriteria decision making methods available in the literature, which are classified into Multi-Attribute and Multi-Objective Decision Methods, MADM and MODM, respectively. The methods from MADM, the focus of this study, can be separated into (i) aggregation methods, whose main representatives are the MAUT, such as Analytic Hierarchy Process (AHP) [4] and Analytic Network Process (ANP) [34], (ii) outranking methods, such as the Preference Ranking Organization Method of Enrichment Evaluations (PROMETHEE) [35] and Élimination et Choix Traduisant la Realité (ELECTRE) [36] and their derivatives, and (iii) interactive methods, such as Multi-Objective Linear Programming (MOLP) [8,37]. For a more complete overview of optimization and MCDM, see [9,12,19–21,26,27,38,39].

In general, these methods are structured in a two-dimensional matrix, $D_{n\mathbf{x}m}$, as in (1), where C_j is the j-th criterion, a_i the i-th alternative solution, and $x_{ij} = C_j(a_i)$ the evaluation of a_i under C_j.

$$D = \begin{array}{c} \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \dots & C_m \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \end{array} \tag{1}$$

Some methods normalize/standardize these values using the Min-Max normalization ($\mathbf{x}_i^* = (\mathbf{x}_i - \mathbf{x}_{min})/(\mathbf{x}_{max} - \mathbf{x}_{min})$) or the z-score ($z = (\mathbf{x}_i - \mu)/\sigma$), where μ is the mean and σ is the standard deviation. Then, these scaled values are multiplied by a vector of weights $W = w_1, \dots, w_m$ that should represent the importance of each criterion, where $w_j \geqslant 0$ and $\sum\limits_{j \in m} w_j = 1$. Others, such as the AHP, are able to elicit its weights directly, also evaluating both the quantitative and qualitative criteria [25]. This integrated approach is often used in the literature, where analysts use the weights extracted from the AHP as an input to another method that relies on the decision maker to inform the importance of each criterion; see for instance [5,8,9,12,22,39].

Pairwise comparisons are a vital part of the prioritization procedure in AHP. When conducting an assessment, the decision problem is built in a hierarchical structure to obtain the D matrix. Then, the nine-point Likert scale known as Saaty's scale presented in Table 1 is used. For each criterion/objective, the DMs explicit their numeric and gradual preferences for each pair sampling.

Table 1. Fundamental scale of factors in pairwise comparisons.

Importance	Description	Reciprocal
1	Equal importances (or equivalents) of i and j	1 (1.000)
2	Equal to moderate importance	1/2 (0.500)
3	Moderate importance, or slightly more important	1/3 (0.333)
4	Moderate plus importance	1/4 (0.250)
5	Strong importance of i over j	1/5 (0.200)
6	Strong plus importance	1/6 (0.167)
7	Very strong or demonstrated importance	1/7 (0.143)
8	Very, very strong importance	1/8 (0.125)
9	Absolute importance	1/9 (0.111)

Source: Adapted from Saaty [40].

The steps of the AHP, adapted from its creator Prof. Saaty [4], are explained below:

1. Structuring the hierarchy:
The problem is decomposed into three parts: goal, subcriteria, and alternatives, in a hierarchical form. The approach of the AHP involves the structure of any complex problem into different hierarchy levels intending to accomplish the stated objective of the problem.
2. Perform pairwise comparisons:
Construct a matrix of pairwise comparisons for the set $\mathcal{X}$ of alternatives where the entries indicate how much the DM prefers a solution or criterion to another, using the description or the importance value from Table 1. At this point, it is important to highlight that the NQ is $n \times (n-1)/2$ evaluations for each matrix $n \times n$. Depending on the number of alternatives n and/or criteria j, it could be a lengthy process.
3. Calculate the consistency of the DM's assessments:
The matrix of evaluations is considered to be consistent if all of its elements are transitive and reciprocate, such as $x_{ij} = x_{ik} \times x_{jk}$ and $x_{ij} = 1/x_{jk}$ where i, j, and k are any elements of the matrix $\mathcal{Q} \in \mathcal{X}$, and if $i = j$ (main diagonal), then $x_{ij} = 1$. To obtain the relative priority of each criterion, a normalized matrix is defined according to Equation (2).

$$w_{ij} = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}} \tag{2}$$

where $\sum_{i=1}^{n} x_{ij}$ is the sum of the elements in column j.
Then, the relative weight of each row is computed by dividing the sum of the values of each row by the number of elements n, see Equation (3).

$$w_i = \frac{\sum_{j=1}^{n} w_{ij}}{n} \tag{3}$$

Calculate the eigenvector for each comparison matrix. The maximum eigenvalue (λ_{max}) is a measure of consistency within the pairwise comparison matrix [17]. To obtain the λ_{max}, also called the Eigenvalue problem [24], simply calculate the arithmetic mean of the elements of the vector. The largest eigenvalue is greater than or equal to

n ($\lambda_{max} \geq n$). The closer λ_{max} is to n, the more consistent is $\mathcal{Q}$. The Consistency Index (*CI*) is given by Equation (4).

$$CI = \frac{\lambda_{max} - n}{n - 1} \tag{4}$$

4. Calculate the Consistency Ratio (*CR*):
The CR is calculated according to Equation (5).

$$CR = \frac{CI}{RI} \tag{5}$$

where *RI* is a consistency index, proposed in the seminal work of Prof. Saaty [4], and is largely discussed in the literature [41]. Accordingly, with Saaty [4], if $CR \leq 0.10$ the level of inconsistency in the judgments made by the DM is acceptable. Otherwise, the process needs to be redone, partially or fully.

5. Synthesizing the results:
The pairwise matrices are synthesized to calculate the overall priorities for the alternative solutions. Sort the priorities and select the alternative with the highest priority.

The most exhausting manner to solve the decision problem is to take all the solutions to the DM, at once, and ask him/her to evaluate them. This leads to a scalability problem when the number of alternatives and/or objectives is large [10,14,25]. This approach also carries a greater risk of inconsistency during the elicitation process, which can increase the number of evaluations needed and further complicate the process [2,10,11,15,37]. As a result, the size of the pairwise comparison matrix remains a limitation.

The complexity of dynamic systems has prompted efforts to improve the scalability of the decision making process in the literature. There are many possible approaches, such as data reduction [32], the selection of regions of interest (ROI) in the Pareto front [29,42], offline learning [30], and the selection of the prominent alternative [43]. However, they do not use the benefit of online learning to capture the DM's preferences by presenting only a few solutions at a time in an iterative process.

2.2. Decision Maker Preference Learning

The term preference may assume different contexts, but here, it is interpreted as subjective comparative evaluations that translate the DM's specific desires into a declarative way [44], subjectively, since it is typically attributed to a human. It is comparative because the evaluations are expressed

as an item relative to another item. Additionally, it is evaluations because it concerns matters of value, typically concerning practical reasoning.

Learning, or eliciting preferences, is a central concept in decision making, and they can be obtained explicitly, such as ratings, statements, or queries, or implicitly, such as through user observation or inference. In the end, it generates utility functions from single to complex [15]. Fürnkranz and Hüllermeier [45] described different types of preferences that follow some properties presented in Salvatore [46]. Preference learning, in turn, has emerged as a new sub-field of machine learning (ML) dealing with the learning of (predictive) models from observed, revealed, or automatically extracted information [45]. It has been successfully used in Decision Theory and Multicriteria Problems [44,45]. It is often applied to form a total order relation on a collection of alternatives [44], also called ranking problems.

In online learning, the number of alternatives presented to the DM for review must, necessarily, be very limited and the data becomes available in sequential order. It is different, for instance, from active learning (or query learning [47]), which is a weak supervised learning technique where both labeled and unlabeled data may be used, and offline learning (or batch learning), where data are collected and the model is trained once [47].

In the learning phases, the DM is demanded to maximize a utility function $\mathcal{U}$, indicating the priority between two alternatives $\{i, j\} \in \mathcal{X}$ with respect to each criterion in the sequence of t iterations [48]. The learner, then, makes a prediction p_t, then the correct answer y_t, taken from a target domain $\mathcal{Y}$, is revealed and the learner suffers a loss $l(p_t, y_t)$. For binary (yes/no) answers and predictions, namely $\mathcal{Y} \in \{0, 1\}$, is called online classification [48]. In regression problems, the focus of this research, $\mathcal{X} \in \mathbb{R}^d$, corresponds to a set of features that represents the solutions in the variable space, and $\mathcal{Y} \in \mathbb{R}$. After the learner prediction, a loss function is computed to measure the difference between p and y. The most common loss functions are: Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), Root MSE (RMSE), and Coefficient of Determination (R2) [49].

Those are the explicit manners of gathering the DM's preferences. However, these evaluations can also be obtained implicitly or even inferred. To improve the learning process and to reduce the number of queries, it may be beneficial to personalize the presentation of new solutions based on the decision maker's past evaluations and preferences.

Chen and Lin [30] stated that the key process for solving an MCDM problem is to capture the preference structure of the DM. Machine learning-driven approaches have made great progress because they enable the learning of the utility function, regardless of its structure or property. These authors developed an interactive Decision Neural Network (DNN) architecture to model the DM's preferences. The scalability is intrinsic in their approach, since some solutions are chosen to test whether the DNN model is satisfactory. If the trained model is not consistent with the results given by the DM, they suggest adding new solutions and retraining the algorithm. Chen and Lin [50] proposed a DNN with a "twin-topology" addressed to MOO problems in the search for the most desirable solution. These approaches are quite similar to that proposed by Alves et al. [15] which suggested an online learning methodology based on the Extreme Gradient Boosting (XGBoost) algorithm. The Kendall tau (KDT) [28] distance was used to evaluate model convergence through iterations.

Pedro and Takahashi [14] proposed a Multilayer Perceptron (MLP) architecture to capture information from the DM and to model a utility function based on a partial sorting process. Later, these authors proposed in [51] a method called Neural Network Decision Maker (NNDM) that uses the MLP and queries to approximate the utility function, extracting the DM's preferences. Mendonça et al. [31] extended the NNDM to another version, called NNDM-2, for portfolio optimization. In this research, the authors proposed a multiobjective financial portfolio optimization model analyzed with the decision methods NNDM and NNDM-2, considering the investor risk profiles of conservative, moderate, and aggressive. Although these proposals are contributions to the computational finance area, they do not focus on learning pairwise matrices and reducing the DM's cognitive efforts.

Even with the reduction in pairwise comparison matrices, many-objective optimization problems (MaOPs) can demand high computational effort and require good visualization techniques to make the decision process simple and efficient. In this direction, Pedro and Takahashi [29] worked on a selection of alternatives in the ROI. The DM interacts with the method by evaluating the alternatives, and a Radial Basis Function network is trained to construct the preference function. By limiting comparisons to a specific region of the Pareto front, this strategy helps to avoid redundant and unnecessary queries.

These works are quite interesting and provide relevant approaches to reducing the number of assessments, but most of them suffer from scalability, bias, and ambiguity in the comparisons. These factors may lead to inconsistency or even the lack of smart strategies for presenting new solutions to the DM over the iterations.

3. Proposed Approach

This section introduces a novel technique that improves the AHP multicriteria decision making method with machine learning to gain scalability, as illustrated in Figure 1.

The scalability is improved by reducing the number of alternatives presented to the DM. It follows the main arguments of the Paradox of Choice [1], leading to two benefits: first, it shows that the decision making process is easier with fewer options and, second, it facilitates the task of eliciting preferences.

In this way, instead of presenting all the solutions at once and asking a DM's preferences, it works iteratively with only a few alternatives at a time to be compared. A machine learning regression algorithm is invoked to learn these preferences and, after some iterations, the model is expected to be able to infer the behavior of the DM and to guide the search for an optimal solution to the decision problem.

The approach follows the steps detailed below. The numbers in the list correspond to the numbers on each box shown in Figure 1.

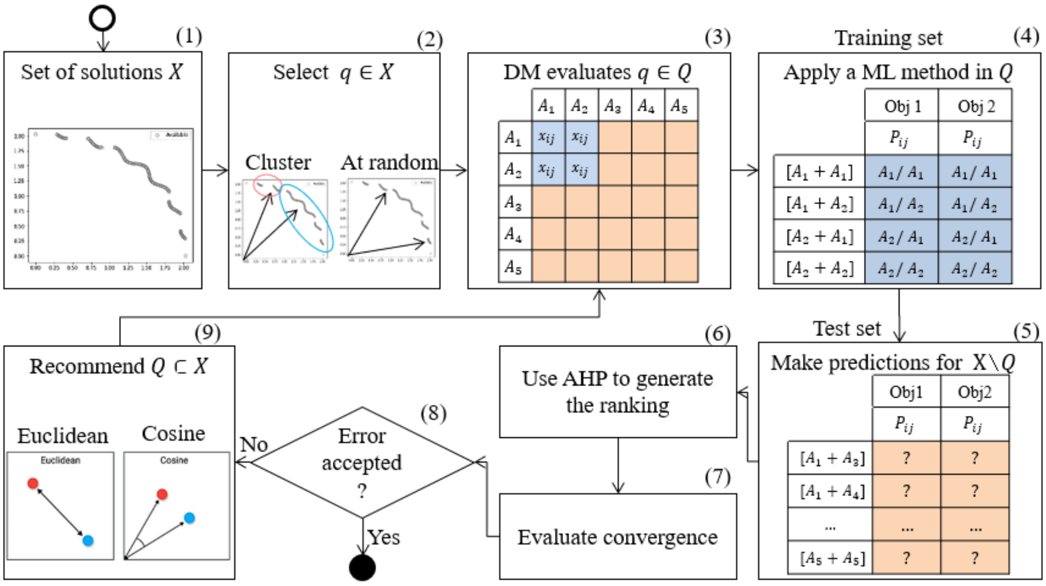
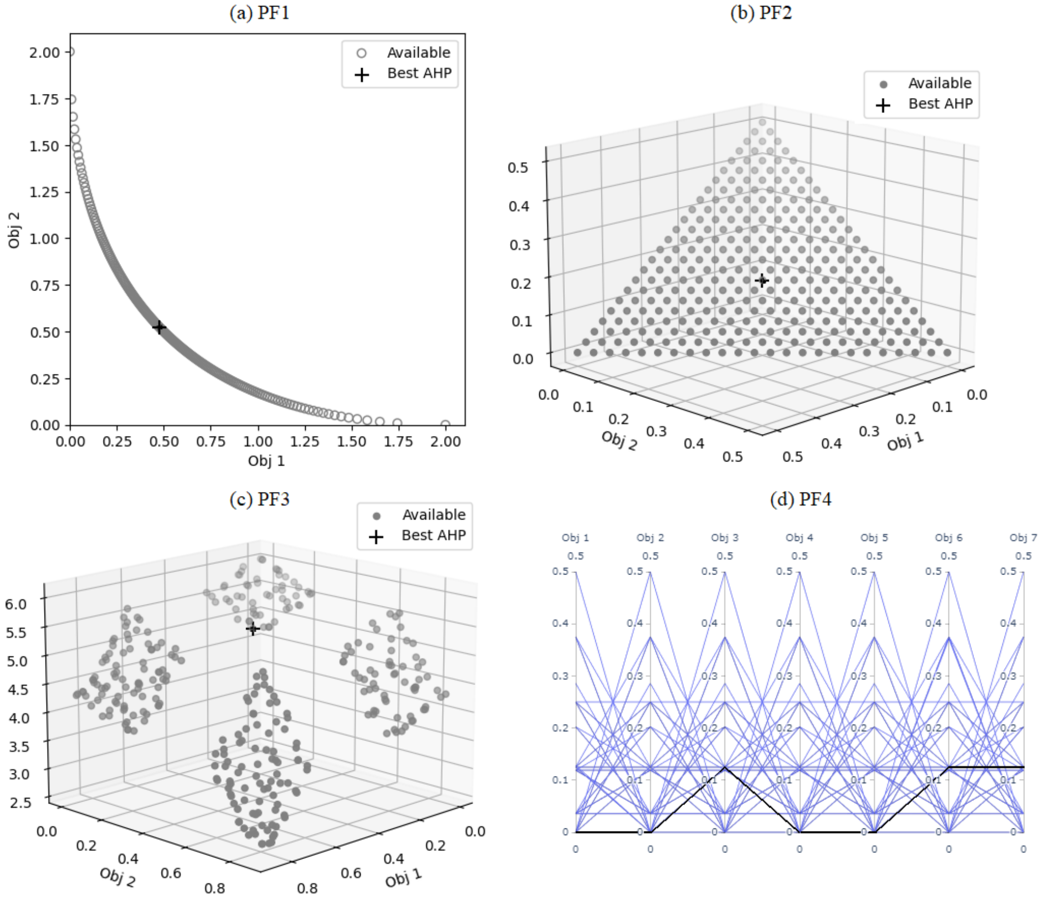


Figure 1. Flowchart illustrating the main steps of the proposed methodology to make the decision process with AHP scalable.

1. Set of available solutions $\mathcal{X}$.
   In this paper, we use the Generalized Position-Distance (GPD) [27] tool to simulate different decision problems, as illustrated in Figure 2, to validate the applicability of the proposed methodology. We vary the number of objectives (two, three, and seven) and the surface of the regions in the Pareto front (convex, linear, and discontinuous). It allows us to simulate problems such as those that managers in organizations deal with on a daily basis. From a set of available alternatives, the managers have to evaluate these alternatives to choose the one that best meets their desires. The best solution using the AHP method is highlighted.
2. Select q alternatives from $\mathcal{X}$.
   When the DMs start evaluating solutions, very little is known about their preferences. Therefore, we implement two ways of choosing the initial solutions: random and clustering. In the former, q is randomly chosen, such that $q \in \mathcal{X}$. The latter builds q clusters and chooses one alternative from each cluster. It may guarantee diversity and preserve diversity by presenting alternatives in different regions of the Pareto surface.
3. Query the DM and build the training set.
   The training set is composed of all combinations of $\mathcal{Q} \subset \mathcal{X}$ to gather the utilities. For each pair, ask the DM: "*How much do you prefer the alternative i over j for objective k?*". Following the illustration (3) in Figure 1, suppose that the first two alternatives are $q = [A_1, A_2]$ in the blue part. Using Table 1, the DM states that A_1 has very strong importance over A_2 in objective 1, which reveals that $u(A_1)/u(A_2) = \langle 7 \rangle$, then $u(A_2)/u(A_1) = \langle 1/7 \rangle$. In addition, it is said that A_1 has moderate importance over A_2 in objective 2, which leads to $u(A_1)/u(A_2) = \langle 3 \rangle$ and $u(A_2)/u(A_1) = \langle 1/3 \rangle$. Equal comparisons are discarded because they have equal importance. Each sample is the concatenation of the variables in the decision space (feature vector), and the

target/label are the collected preferences. Thus, $[A_1 + A_2] = \langle 7, 3\rangle$, and $[A_2 + A_1] = \langle 0.1428, 0.3333\rangle$.

4. Train the machine learning regressor.
In the training phase, a multi-output regressor method was employed. The model targets the expected utility value, i.e., importance on the Saaty scale, for each pair of alternatives for the set $\mathcal{Q}$ in each objective, as the DM does. In this stage, different methods were implemented, based on the Scikit-learn package [49]: (i) Gradient Boosting for Regression (GBR), (ii) Multitask Lasso, (iii) Multitask ElasticNet, both trained with the L1/L2 mixed-norm as a regularizer, (iv) Ridge Regression, and (v) Random Forest (RF) Regressor. It is worth mentioning that the Randomized SearchCV method was used to find the best hyperparameters for each method in the first round of executions.


(a) PF1

(b) PF2

(c) PF3

(d) PF4

Figure 2. Pareto fronts (PF) representing the decision problems with 2 objectives in (**a**)—GPD02, 3 objectives in (**b**)—DTLZ1 and (**c**)—DTLZ7, and 7 objectives in (**d**)—DTLZ1.

5. Make the predictions on the test set.
The test set consists of all pairs of the remaining alternatives, $\mathcal{X} \setminus \mathcal{Q}$. The trained model is used to predict the label for this entire set, simulating the DM's behavior. Note that the alternatives to be evaluated are those that were not selected in step (3). However, as an iterative process, more alternatives are added to $\mathcal{Q}$, and once the training set increases, the test set decreases.
6. Apply the AHP and generate the ranking.
In this step, the classical AHP [4] is applied to generate the total ordering of the alternatives in $\mathcal{X}$. To do this, the method merges both of the preferences given according to step 1, and those predicted by the machine learning model in step 5.
7. Compute the convergence measure.
The KDT [28] defined in Equation (6) was used as a convergence measure. This

metric measures the dissimilarity between the two rankings lying in the interval [0, 1]. The lower the dissimilarity between the ranking generated from the model's predictions at iteration t and the ranking generated at iteration $t-1$, the greater the model's ability to approximate the DM preferences.

$$K(\tau_1, \tau_2) = |(i,j) : i < j, (\tau_1(i) < \tau_1(j) \wedge \tau_2(i) > \tau_2(j)) \vee (\tau_1(i) > \tau_1(j) \wedge \tau_2(i) < \tau_2(j))| \quad (6)$$

where $\tau_1(i)$ and $\tau_2(i)$ are the rankings for the elements i in the indexes. $K(\tau_1, \tau_2)$ is 0 if the lists (in our case, the rankings) are identical, and 1 otherwise.
Additionally, the MAPE, MSE, RMSE, and R2 regressor metrics [49] are computed in order to assess overfitting and the preferences predicted by the model.

8. Stopping criterion.
In this work, a KDT that is less than or equal to 5% between the iteration t and $t-1$ was defined as the stopping criterion. To obtain the model that best fits the utility function $\mathcal{U}$ that represents the DM, the algorithm is retrained and tunned whenever it does not reach the desired the minimum similarity between the rankings. This is performed so that the model can reduce the number of iterations until the stopping criterion.
9. Recommending new solutions to be compared by the DM.
Depending on the utility function that represents the DM, it is interesting to explore new regions of the decision boundary, to select and to recommend new solutions that he/she is going to likely be interested in, and also in a way in which it is going to positively impact the performance of the model. To do this, a parameter θ was implemented that represents the percentage of random solutions picked from the second iteration. $\theta = 0.0$ indicates that the recommended solutions are going to be based on either the Cosine distance or the Euclidean distance [52], and $\theta = 1.0$ means that all solutions are going to be randomly recommended.

Finally, Table 2 details the parameters and hyperparameters of the proposed approach, defining the expected values that are currently implemented. New ones can be easily added.

Table 2. Description of the parameters and hyperparameters of the proposed approach, and their expected values.

Hyperparameters	Description	Values
MCDM	AHP method based on pairwise comparison matrices.	'AHP'
weights	The weights of the criteria, when applicable.	$[0-1]$ array
cb	If the criterion is benefit (maximize) or cost (minimize).	'cost' or 'benefit' array
q	Number of solutions presented to the DM.	$\mathbb{N}^*$
initial recommend.	Strategy to select q solutions in the first iteration.	'aleatory' or 'cluster'
similarity distance	Measure used to recommend new solutions.	'Euclidean' or 'Cosine'
θ	% of random solutions to be recommended.	$0-1$ float
ml	Multi-output ML regressor method.	'gbr', 'lasso', 'elasticnet', 'rf', 'ridge'

The baseline was the classical AHP [4]. The NQ required until the ranking reaches the stop criterion is computed based on both the proposed approach and the original method.

4. Results and Discussion

This section initially focuses on discussing the results regarding the optimization of the hyperparameters described in Table 2. The ranking obtained by the proposed scalable method is presented and is discussed further at the end. It is demonstrated that the proposal is effective for different problems and that provides satisfactory results when compared to the classical method.

The results are presented in subsections for sake of organization. Section 4.1 describes the case studies, Section 4.2 shows the learners' performance on each problem, Section 4.3 details the effects of choosing initial solutions via cluster or at random, Section 4.4 analyzes the results of the model convergence when the recommended solutions are based on

Euclidean or Cosine distance, Section 4.5 explains the best choice of the q, Section 4.6 shows how θ guides the local search to minimize the NQ until the stop condition, Section 4.7 provides a ranking obtained with the proposed approach, and Section 4.8 gives comparisons with other works, and future directions.

4.1. Case Studies

The problems PF1 to PF4 illustrated in Figure 2 are used to validate the applicability of the proposed approach. They represent different types of problems that managers may face on a daily basis. Then, the hyperparameters are explored one at a time, to improve the learner's performance and, consequently, to reduce the NQ. The analysis of the results is carried out over four runs of each experiment.

4.2. Machine Learning Methods Performance

The ML methods GBR, Lasso, ElasticNet, Ridge, and RF were applied to learn preferences for the problems PF1 to PF4. Although the ML algorithms vary, the other hyperparameters are kept fixed, that is, $q = 5$, $\theta = 0.2$. The initial recommendation was selected based on clustering, and the remaining solutions in the smallest Cosine distance.

Both the GBR and RF-based models reached the stop criterion first at the 10th iteration for the decision problem PF1. From PF2 to PF4 was RF at iterations 11, 15, and 10, respectively. Table 3 shows the order of models in each problem. This order is based on the number of iterations that each model spent to reach the stopping condition.

Table 3. Selection of the best ML method for each decision problem. The order of the models and the iteration where the best one (in bold) achieved the stop criterion.

Problem	Order	Iteration
PF1	**GBR** $\succeq$ **RF** $\succ$ Ridge $\succ$ Lasso $\succ$ ElasticNet	10
PF2	**RF** $\succ$ GBR $\succ$ Lasso $\succ$ Ridge $\succ$ ElasticNet	11
PF3	**RF** $\succ$ GBR $\succ$ Ridge $\succ$ ElasticNet $\succ$ Lasso	15
PF4	**RF** $\succ$ GBR $\succ$ Ridge $\succ$ ElasticNet $\succ$ Lasso	10

4.3. Initial Selection of the Solutions

The first alternatives chosen for the DMs to express their preferences can be selected in two ways: randomly or via cluster. Both have pros and cons. The former allows a greater possibility of choosing one or more solutions that are already in the region where the global optimum is located. On the other hand, it does not guarantee the coverage of the entire Pareto front region as the latter. In the MOO and MaOP problems, diversity on the PF is crucial [27].

Figure 3 shows the learning rate for problem PF3, which behaved differently than the others. The ranking stability throughout the iterations was greater when using cluster to select the first q solutions. In the figure, the lines with dots and the shaded area represent the mean and the 95% confidence interval, respectively. A smaller shaded area indicates that the trained model consistently predicted the preferences more accurately in that iteration. Additionally, the rankings generated between iterations was closer, as seen at t_5.

The random choice of initial solutions resulted in a larger standard deviation over the iterations, since it does not guarantee the coverage of the entire Pareto surface as the cluster with $q = 5$ possibly does. This problem represents the DTLZ7 function whose surface is also highly nonlinear and exhibits many discontinuities. These characteristics make it difficult to predict the learner behavior and they pose more challenges to the model.

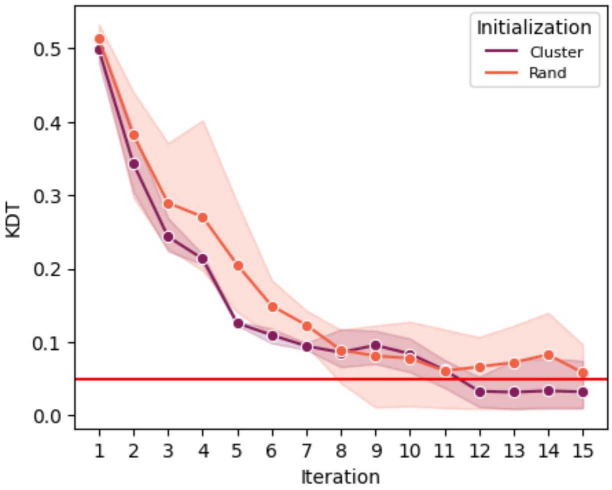


Figure 3. Learning curve of problem PF3 varying the selection of the alternatives between cluster or at random.

4.4. Similarity among the Recommended Solutions

This part refers to the selection of alternatives during the interactions with the DM. These alternatives are picked iteratively from the set $\mathcal{X} \setminus \mathcal{Q}$, based on proximity metrics. This analysis considers the best models mentioned above, initialization with cluster, $q = 5$, and $\theta = 0.2$. Euclidean and Cosine are two values used as reference distances between solutions.

The main difference also occurred in problem PF3, as illustrated in Figure 4. It is possible to observe that the model that selected the alternatives based on the shortest Cosine distance outperformed the Euclidean one. The first reached the stopping condition in iteration t_{10}, and the second in t_{12}.

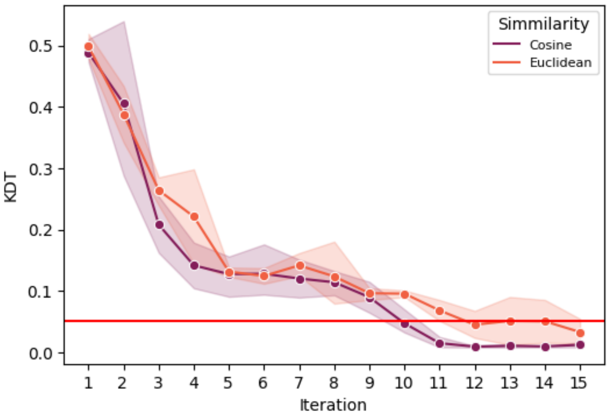


Figure 4. Learner behavior in PF3 by using either Euclidean or Cosine distance to select new solutions to query the DM.

The Euclidean distance is a measure of the distance between two points in a Euclidean space, while the Cosine targets the similarity between two vectors. A possible explanation for this difference is that the Euclidean distance is known to perform poorly in high dimensions [53], or even in discontinuous surfaces. This may cause the later convergence and larger standard deviation in predicting preferences at the final iterations. More studies can be carried out to specifically investigate this difference.

In this work, these recommendation strategies are utilized to minimize bias during the process of selecting the alternatives that will be presented to the decision maker. They identify the most relevant solutions for the DM, leading to more informed choices.

4.5. Number of Alternatives Presented to the DM

For this analysis, we considered $q = [3, 4, 5, 7, 10]$, with a focus on the minimum NQ. The most appropriate hyperparameters from the previous analyses were taken into consideration. Figure 5 exhibits the results for the different values of q. By utilizing KDT as a reference, the models reached the stopping criterion at iterations 5, 7, 10, and 11 for PF1, PF2, PF3, and PF4, respectively.

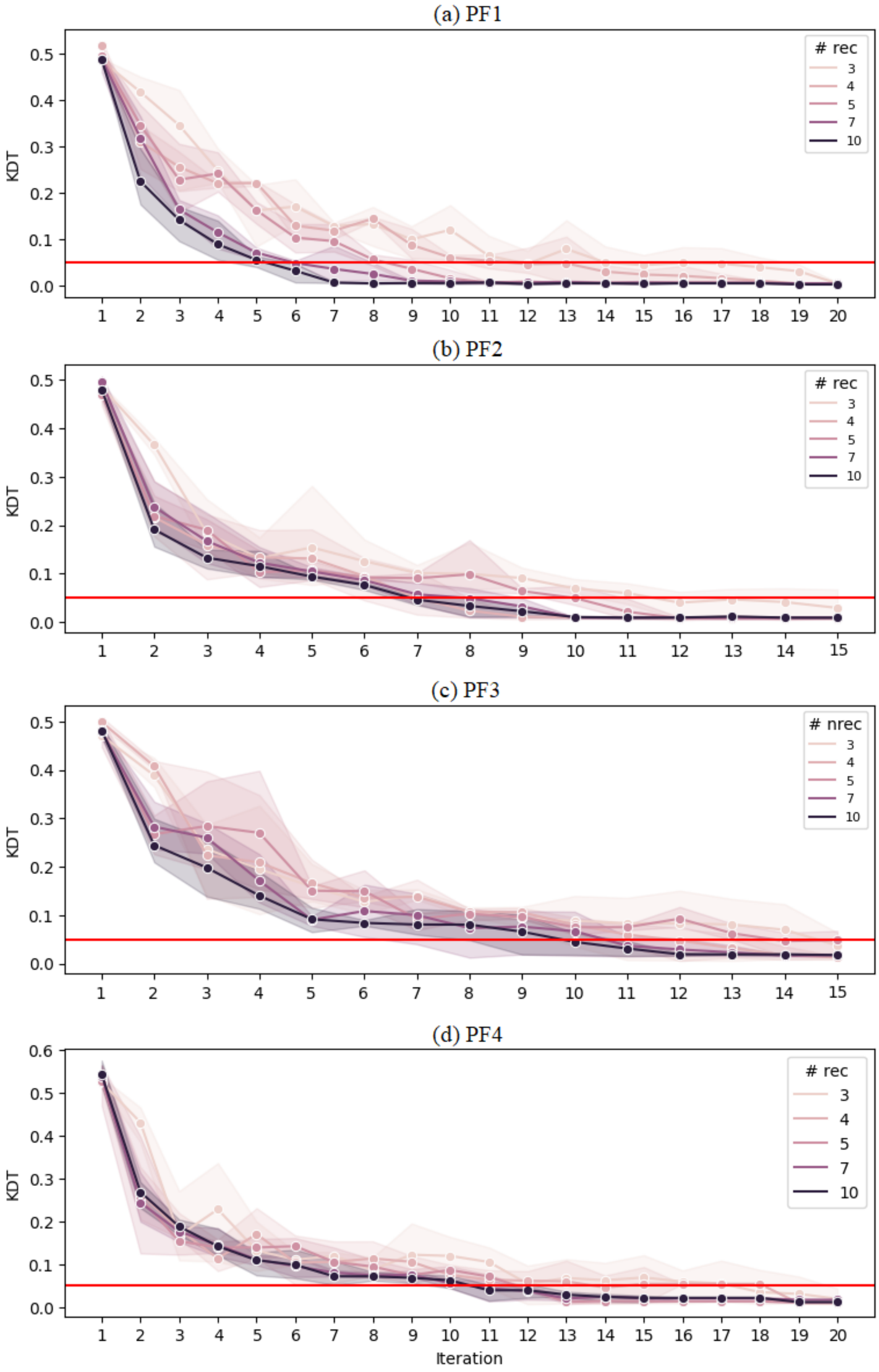


Figure 5. Learning rate for $q = [3, 4, 5, 7, 10]$ alternatives selected at a time for PF1 in (**a**), PF2 in (**b**), PF3 in (**c**), and PF4 in (**d**).

Although more solutions can bring more information to the learner, a slight improvement in the model's performance is observed by increasing q. On the other side, more comparisons are necessary. Fewer interactions translate into less effort and time-consumption from the DM and, consequently, a faster selection of the preferred solution. It is also worth mentioning that there are greater chances of obtaining consistent assessments and achieving the appropriate CR (5).

The NQ until the learner reached the stopping condition is described in Table 4.

Table 4. NQ needed for each decision problem, varying the parameter q until achieving the stop condition.

Problem	**NQ**				
	$q = 3$	$q = 4$	$q = 5$	$q = 7$	$q = 10$
PF1	45	72	90	126	270
PF2	36	60	80	168	315
PF3	45	84	120	231	450
PF4	57	108	130	252	495

Based on these problems, it can be assumed that the most appropriate number of alternatives to present to the DM is three at a time, which requires three queries per objective. It should be clear that, in practical terms, the managers should allocate the availability of time and effort to the analyst to perform such assessments. In this proposed approach, q can be easily modified according to the business needs.

Based on the Paradox of Choice main argument, the lower value is better. In other words, a smaller q implies fewer comparisons, even if this necessitates more interactions. After that, the model learns the utility function and can predict the preferences for the remaining alternatives and build the final ranking.

Two limitations of AHP are addressed in this point: complexity and limited scope. Complexity is reduced since the number of required comparisons is smaller. The scope of AHP applications is improved because scalability allows for complex problems to be broken down and solved iteratively.

4.6. Strategies for Local Search

The hyperparameter θ guides the local search towards more promising solutions picked in each iteration t. The search can be from totally random ($\theta = 1.0$) to only distance-based ($\theta = 0.0$)—see Section 4.4. Considering $q = 3$ from the previous analysis, we vary θ to test the search for $[0 - q]$ solutions.

The models converged similarly in problems PF1 and PF3. However, an interesting finding is observed in problems PF2 and PF4—see Figure 6. The search for purely distance-based solutions was more efficient in terms of reaching conditions with fewer iterations and, thus, less NQ.

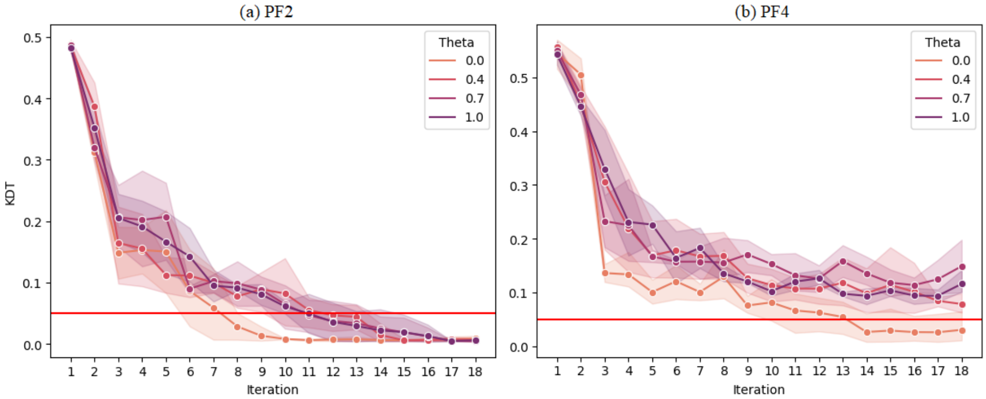


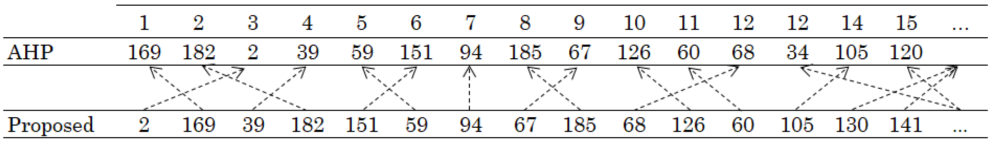
Figure 6. Effects of varying θ in problems PF2 in (**a**) and PF4 in (**b**), whose Pareto surfaces are linear.

For problems whose decision boundaries are nonlinear and/or discontinuous, such as PF1 and PF3, a randomness factor can help the learner to find new search spaces and to escape from local minima. However, in linear ones, this problem is minimized, since convergence is the major challenge.

4.7. Ranking with the Scalable Approach

Since this paper deals with solving decision making problems with many criteria/objectives, the final ranking is expected at the end of the process. To illustrate an example, the ranking obtained for the PF2 problem with $\theta = 0.0$ presented earlier in Section 4.6 and Figure 6a is going to be used.

Notice that the RF-based model required eight iterations to learn the preferences. After that, it predicts the preferences among all the remaining pairs. Although only up to 18 iterations are shown, the model can be applied to all $\mathcal{X}/q$ iterations without requiring new queries or retraining. The ranking obtained with the predicted preferences at iteration t_{18} is presented in Figure 7. The model created with $\theta = 0.0$ requires six assessments per iteration. At t_{18}, the preferences predicted by the model generate a ranking that is quite similar to that provided by the analytical AHP using the entire set $\mathcal{X}$. The arrows indicate the swaps between the indices that are necessary to obtain the exact ranking.



	1	2	3	4	5	6	7	8	9	10	11	12	12	14	15	...
AHP	169	182	2	39	59	151	94	185	67	126	60	68	34	105	120	
Proposed	2	169	39	182	151	59	94	67	185	68	126	60	105	130	141	...

Figure 7. Ranking of the top 15 alternatives using the classical AHP and the proposed scalable approach.

Figure 8 illustrates the solutions in the objective space. Using the AHP method, the best alternative is 169. It was ranked 2 using the proposed approach. It is possible to notice that the best solutions predicted by the proposed method are in the same region where the best AHP solution is.

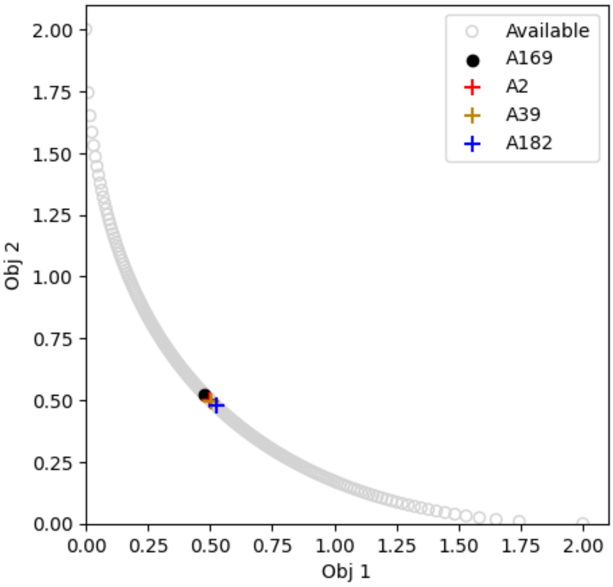


Figure 8. Best solutions for the PF1 based on the AHP and the proposed methodology.

4.8. Other Analysis and Directions

As argued before, preferences can be obtained either explicitly—as in the case of querying the DMs, or implicitly—when it is collected and/or inferred. Step 3 of the AHP, see Section 2.1, explains the consistency of the preferences. That transitivity is now used to obtain implicit relations. In each objective, the difference between A_i, A_j, A_j, A_q (recommended), and inferred to A_i regarding A_q is computed. Thus, the preference is simply the difference between A_i and A_q. For this, $q - 1$ are recommended, and one is used to extract the preferences. Based on some rounds of experiments, it was noticed that this strategy did not help the model to reduce the number of iterations to reach the stopping criterion. The principal justification is that the number of data entered implicitly at each iteration grows linearly with the number of training samples. It means that in the first iterations, as the method converged quickly, there was not yet much data imputed in a way that accelerated the model's convergence. The authors believe this strategy may work for more complex problems and can be investigated in future analyses.

A comparison with other works in the literature was conducted. We analyzed the proposed method by Alves et al. [15] and the classical AHP in [4]. In the former, the proposal is more simple. The initial selection is based solely at random, as well as the recommendation, which is equivalent to ours when $\theta = 1.0$. Based on the results discussed in Section 4.3, the cluster strategy shows less variance in the problems such as PF3 and PF4. In addition, in Section 4.6, distance-based recommendation works better for problems with linear boundaries. Thus, the results of this current research are expected to outperform those presented in [15] in these cases. Compared to the latter, our proposal requires only $\mathcal{Q}$ alternatives to be evaluated in subgroups of size q for a few iterations. AHP, on the other hand, requires the evaluation of $\mathcal{X}$ alternatives in a single round.

This approach is an alternative for solving large-scale multicriteria decision making problems, where the number of alternatives and/or criteria is very large, making it hard for the decision maker to compare solutions. The methodology is customizable and can be incremented with new features and problems. Investigations aimed at reducing dimensionality and then reducing DM effort (cognitive or in NQ), correlation effect between criteria, consistency indicators, and others can be incorporated to acting as new hyperparameters and functionalities, for instance [3,10,33,54].

5. Conclusions

Managers and organizations are looking for tools that will speed up the decision process. However, methods such as the AHP, based on pairwise comparisons, make this difficult to achieve. An approach to making the classical Analytic Hierarchy Process method scalable is described in this paper. Instead of presenting all of the solutions to the DMs at once, it is achieved through successive iterations with help of a machine learning method to learn the preferences and to predict the remaining ones.

The scalability is improved through the optimization of the hyperparameters, and it acts directly on both the reduction in queries and the probability of inconsistency in the evaluations. In addition, it indirectly affects the wasted time and the possibility of having an automated model to use in new problems of the same domain.

The methodology has the advantage of different parameters that can help to further explore the decision problem and to accelerate convergence. A higher number of interactions with the DM allows for better convergence to the desired location on the Pareto front. However, as the number of queries depends on the number of interactions, analysts and DMs need to agree on a stopping condition that meets the organization's needs. In this article, the KDT was used as a merit function, and new ones can be implemented. Once the model has been trained, it can be reused without requiring new queries to the DM, except when the domain changes.

The parameters were analyzed one at a time on four problems with different shapes, including convex, linear, and discontinuous. Other findings include the number of solutions that require fewer queries ($q = 3$) and that searching for solutions based on the shortest distance tends to accelerate the models' learning when the problem has a linear surface. When θ is set to 0.0, the search is distance-based and tends to be more efficient in terms of reaching conditions with fewer iterations. In practical applications, the DM's effort is reduced from $n \times (n-1)/2$ to $q \times (q-1)/2$ assessments, with $q \ll n$, in approximately 15 iterations.

Among the criticisms made of the AHP method, this article directly addresses the complexity, limited scope, and bias; and it addresses the subjectivity indirectly, and it does not address the lack of transparency. Scalability acts on complexity and scope limitation. Between two question phases, the model can learn the utility function with fewer alternatives. Bias is reduced using the recommendation strategies, and it is also controlled by the parameter θ, focusing on the most relevant solutions. Subjectivity is improved by reducing the number of comparisons at a time, which decreases the chances of inconsistent evaluations. However, the uncertainty or vagueness during evaluations is not directly discussed, and it may be a target of future studies, such as the application of fuzzy logic or scenarios investigation. In addition, the AHP method may be difficult to understand or to explain to stakeholders, causing a lack of transparency. Future improvements may involve, for instance, explainability methods.

For future research, we also suggest the implementation of a grid-search to automate the process of tuning the parameters, and we extend the methodology in more problems, either with artificial or real data. Furthermore, evaluating solutions in MaOPs is a very hard task, from objective function calculation, and visualization, to choosing the best solution. The proposed approach can support this class of problems, and it was modeled in a manner such that other features can be easily inserted, such as new ML algorithms, recommendation and solution selection strategies, and even other multicriteria methods based on pairwise comparisons.

Author Contributions: Conceptualization, F.G.G. and M.A.A.; Methodology, M.A.A. and F.G.G.; Validation, M.A.A., I.R.M., F.G.G. and A.G.-C.; Formal analysis, I.R.M.; Investigation, M.A.A.; Data curation, I.R.M.; Writing—original draft preparation, M.A.A.; Writing—review and editing, I.R.M., F.G.G. and A.G.-C.; Visualization, M.A.A.; Software, M.A.A.; Supervision, F.G.G.; Project administration, F.G.G.; Funding acquisition, F.G.G. and A.G.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Funds through the FCT—Portuguese Foundation for Science and Technology, References UIDB/05256/2020 and UIDP/05256/2020.

Data Availability Statement: The data supporting the findings of this study are available at: https://github.com/mvoicer/doutorado.

Acknowledgments: M.A. Alves declares that this work has been supported by the Brazilian agency CAPES. This work has been supported by the Coordination for the Improvement of Higher Education Personnel (CAPES, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), the Foundation for Research of the State of Minas Gerais (FAPEMIG, Fundação de Amparo à Pesquisa do Estado de Minas Gerais), and by the National Council for Scientific and Technological Development (CNPq), Brazil , Grants no. 306850/2016-8 and 312991/2020-7. The authors would like to thank the anonymous reviewers for their earnest efforts and the thorough review of our manuscript. The first author also thanks O. Orang, R.C.P. Silva, T.M. Rezende, G.Z. Castro, and P.C.L. Silva for suggestions and revisions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Schwartz, B. *The Paradox of Choice: Why More Is Less*; Harper Collins Publishers: New York, NY, USA, 2004; ISBN 978-0-06-146158-3.
2. Sakhardande, M.J.; Gaonkar, R.S.P. On solving large data matrix problems in Fuzzy AHP. *Expert Syst. Appl.* **2022**, *194*, 116488. [CrossRef]
3. Kuo, T. An Ordinal Consistency Indicator for Pairwise Comparison Matrix. *Symmetry* **2021**, *13*, 2183. [CrossRef]
4. Saaty, T.L. What is the analytic hierarchy process? In *Mathematical Models for Decision Support*; Springer: New York, NY, USA, 1988; pp. 109–121.
5. Ho, W. Integrated analytic hierarchy process and its applications—A literature review. *Eur. J. Oper. Res.* **2008**, *186*, 211–228. [CrossRef]
6. Basílio, M.P.; Pereira, V.; Costa, H.G.; Santos, M.; Ghosh, A. A Systematic Review of the Applications of Multi-Criteria Decision Aid Methods (1977–2022). *Electronics* **2022**, *11*, 1720. [CrossRef]

7. Madzík, P.; Falát, L. State-of-the-art on analytic hierarchy process in the last 40 years: Literature review based on Latent Dirichlet Allocation topic modelling. *PLoS ONE* **2022**, *17*, e0268777. [CrossRef]
8. Melnik-Leroy, G.A.; Dzemyda, G. How to influence the results of MCDM?—Evidence of the impact of cognitive biases. *Mathematics* **2021**, *9*, 121. [CrossRef]
9. Dang, T.T.; Nguyen, N.A.T.; Nguyen, V.T.T.; Dang, L.T.H. A Two-Stage Multi-Criteria Supplier Selection Model for Sustainable Automotive Supply Chain under Uncertainty. *Axioms* **2022**, *11*, 228. [CrossRef]
10. Huang, J.J. Analytic Hierarchy Process with the Correlation Effect via WordNet. *Mathematics* **2021**, *9*, 872. [CrossRef]
11. Munier, N.; Hontoria, E. *Uses and Limitations of the AHP Method*; Springer: New York, NY, USA, 2021; pp. 1–130. [CrossRef]
12. Tuljak-Suban, D.; Bajec, P. Integration of AHP and GTMA to make a reliable decision in complex decision-making problems: Application of the logistics provider selection problem as a case study. *Symmetry* **2020**, *12*, 766. [CrossRef]
13. Aliyev, R.; Temizkan, H.; Aliyev, R. Fuzzy analytic hierarchy process-based multi-criteria decision making for universities ranking. *Symmetry* **2020**, *12*, 1351. [CrossRef]
14. Pedro, L.R.; Takahashi, R.H. Modelling the Decision-Maker Utility Function through Artificial Neural Networks. In Proceedings of the Anais do IX Congresso Brasileiro de Redes Neurais/Inteligência Computacional (IX CBRN), Ouro Preto, Brazil, 25–28 October 2009; Volume 1, pp. 550–563.
15. Alves, M.A.; Meneghini, I.R.; Guimarães, F.G. Learning Pairwise Comparisons with Machine Learning for Large-Scale Multi-Criteria Decision Making Problems. In Proceedings of the Anais do 15 Congresso Brasileiro de Inteligência Computacional: Joinville, Brazil, 30 November 2021; Filho, C.J.A.B., Siqueira, H.V., Ferreira, D.D., Bertol, D.W., ao de Oliveira, R.C.L., Eds.; SBIC: Joinville, Brazil, 2021; pp. 1–7. [CrossRef]
16. Chu, P.; Liu, J.K.H. Note on consistency ratio. *Math. Comput. Model.* **2002**, *35*, 1077–1080. [CrossRef]
17. Saaty, T.L. A scaling method for priorities in hierarchical structures. *J. Math. Psychol.* **1977**, *15*, 234–281. [CrossRef]
18. Pedro, L.R.; Takahashi, R.H.C.; Gaspar-Cunha, A. A Model for a Human Decision-Maker in a Polymer Extrusion Process. In Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization, Guimarães, Portugal, 29 March–1 April 2015; Springer: New York, NY, USA, 2015, pp. 358–372. ._24. [CrossRef]
19. Zavadskas, E.K.; Turskis, Z.; Kildienė, S. State of art surveys of overviews on MCDM/MADM methods. *Technol. Econ. Dev. Econ.* **2014**, *20*, 165–179. [CrossRef]
20. Maia, W.; Ekel, P.; Vieira, D.A.G.; de Castro, E.A.; de Oliveira, M.A.D.; Reis, I.M.; Dos Santos, K.M.G. Evaluation of Operational Risk in Power Substations and Its Rational Reduction on the Basis of Multicriteria Allocating Resources. *IEEE Access* **2021**, *9*, 149383–149397. [CrossRef]
21. Zolfani, S.H.; Bazrafshan, R.; Ecer, F.; Karamaşa, Ç. The suitability-feasibility-acceptability strategy integrated with Bayesian BWM-MARCOS methods to determine the optimal lithium battery plant located in South America. *Mathematics* **2022**, *10*, 2401. [CrossRef]
22. Zolfani, S.H.; Nemati, A.; Reyes-Norambuena, P.J.; Monardes-Concha, C.A. A Novel MCDM Approach Based on OPA-WINGS for Policy Making in Undergraduate Elective Courses. *Mathematics* **2022**, *10*, 4211. [CrossRef]
23. Milošević, M.R.; Milošević, D.M.; Stanojević, A.D.; Stević, D.M.; Simjanović, D.J. Fuzzy and interval AHP approaches in sustainable management for the architectural heritage in smart cities. *Mathematics* **2021**, *9*, 304. [CrossRef]
24. Dweiri, F.; Kumar, S.; Khan, S.A.; Jain, V. Designing an integrated AHP based decision support system for supplier selection in automotive industry. *Expert Syst. Appl.* **2016**, *62*, 273–283. [CrossRef]
25. Russo, R.F.; Camanho, R. Criteria in AHP: A systematic review of literature. *Procedia Comput. Sci.* **2015**, *55*, 1123–1132. [CrossRef]
26. Mufazzal, S.; Khan, N.Z.; Muzakkir, S.; Siddiquee, A.N.; Khan, Z.A. A new fuzzy multi-criteria decision-making method based on proximity index value. *J. Ind. Prod. Eng.* **2022**, *39*, 42–58. [CrossRef]
27. Meneghini, I.R.; Alves, M.A.; Gaspar-Cunha, A.; Guimarães, F.G. Scalable and customizable benchmark problems for many-objective optimization. *Appl. Soft Comput.* **2020**, *90*, 106139. [CrossRef]
28. Kendall, M.G. A New Measure of Rank Correlation. *Biometrika* **1938**, *30*, 81–93. [CrossRef]
29. Pedro, L.R.; Takahashi, R.H. INSPM: An interactive evolutionary multi-objective algorithm with preference model. *Inf. Sci.* **2014**, *268*, 202–219. [CrossRef]
30. Chen, J.; Lin, S. An interactive neural network-based approach for solving multiple criteria decision-making problems. *Decis. Support Syst.* **2003**, *36*, 137–146. [CrossRef]
31. Mendonça, G.H.; Ferreira, F.G.; Cardoso, R.T.; Martins, F.V. Multi-attribute decision making applied to financial portfolio optimization problem. *Expert Syst. Appl.* **2020**, *158*, 113527. [CrossRef]
32. Mosavi, A. The Large Scale System of Multiple Criteria Decision Making; Pre-processing. *IFAC Proc. Vol.* **2010**, *43*, 354–359. [CrossRef]
33. Tanabe, R.; Ishibuchi, H. An easy-to-use real-world multi-objective optimization problem suite. *Appl. Soft Comput.* **2020**, *89*, 106078. [CrossRef]
34. Saaty, T.L. *Decision Making with Dependence and Feedback: The Analytic Network Process*; RWS Publication: Pittsburgh, PA, USA, 1996.
35. Brans, J.P.; Mareschal, B. The PROMETHEE methods for MCDM; the PROMCALC, GAIA and BANKADVISER software. In *Readings in Multiple Criteria Decision Aid*; Springer: New York, NY, USA, 1990; pp. 216–252.
36. Roy, B. Classement et choix en présence de points de vue multiples. *Rev. Française D'informatique Et De Rech. Opérationnelle* **1968**, *2*, 57–75. [CrossRef]

37. Vasconcelos, G.R.; Mota, C.M.d.M. Exploring Multicriteria Elicitation Model Based on Pairwise Comparisons: Building an Interactive Preference Adjustment Algorithm. *Math. Probl. Eng.* **2019**, *2019*, 2125740. [CrossRef]
38. Wang, C.N.; Yang, F.C.; Nguyen, V.T.T.; Vo, N.T. CFD analysis and optimum design for a centrifugal pump using an effectively artificial intelligent algorithm. *Micromachines* **2022**, *13*, 1208. [CrossRef]
39. Samanlioglu, F.; Ayağ, Z. Concept selection with hesitant fuzzy ANP-PROMETHEE II. *J. Ind. Prod. Eng.* **2021**, *38*, 547–560. [CrossRef]
40. Saaty, T.L. Analytic heirarchy process. In *Wiley statsRef: Statistics Reference Online*; Wiley: Hoboken, NJ, USA, 2014. [CrossRef]
41. Pant, S.; Kumar, A.; Ram, M.; Klochkov, Y.; Sharma, H.K. Consistency Indices in Analytic Hierarchy Process: A Review. *Mathematics* **2022**, *10*, 1206. [CrossRef]
42. Meneghini, I.R.; Guimarães, F.G.; Gaspar-Cunha, A.; Cohen, M.W. Incorporation of region of interest in a decomposition-based multi-objective evolutionary algorithm. In *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*; Springer: New York, NY, USA, 2021; pp. 35–50. [CrossRef]
43. Leal, J.E. AHP-express: A simplified version of the analytical hierarchy process method. *MethodsX* **2020**, *7*, 100748. [CrossRef] [PubMed]
44. Fürnkranz, J.; Hüllermeier, E. Preference Learning. In *Encyclopedia of Machine Learning*; Sammut, C., Webb, G.I., Eds.; Springer: Boston, MA, USA, 2010; pp. 789–795. [CrossRef]
45. Fürnkranz, J.; Hüllermeier, E. Preference learning and ranking by pairwise comparison. In *Preference Learning*; Springer: New York, NY, USA, 2010; pp. 65–82. [CrossRef]
46. Salvatore, D. *Microeconomics: Theory and Applications*; Oxford University Press: Oxford, UK, 2003.
47. Settles, B. *Active Learning Literature Survey*; Computer Sciences Technical Report 1648; University of Wisconsin–Madison: Madison, WI, USA, 2009.
48. Shalev-Shwartz, S. Online learning and online convex optimization. *Found. Trends® Mach. Learn.* **2012**, *4*, 107–194. [CrossRef]
49. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
50. Chen, J.; Lin, S. A neural network approach-decision neural network (DNN) for preference assessment. *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.* **2004**, *34*, 219–225. [CrossRef]
51. Pedro, L.R.; Takahashi, R.H. Modeling decision-maker preferences through utility function level sets. In Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization, Ouro Preto, Brazil, 5–8 April 2011; Springer: New York, NY, USA, 2011; pp. 550–563. [CrossRef]
52. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef]
53. Domingos, P. A few useful things to know about machine learning. *Commun. ACM* **2012**, *55*, 78–87. 2347736.2347755. [CrossRef]
54. Ferreira, J.C.; Fonseca, C.M.; Denysiuk, R.; Gaspar-Cunha, A. Methodology to select solutions for multiobjective optimization problems: Weighted stress function method. *J. Multi-Criteria Decis. Anal.* **2017**, *24*, 103–120. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.