CRANFIELD UNIVERSITY

JUAN ANTONIO JUST AMARGÓS

SENSITIVITY OF MULTI-CRITERIA DECISION MAKING IN SUSTAINABLE MANUFACTURING SYSTEMS

SCHOOL OF AEROSPACE TRANSPORT AND MANUFACTURING
Management & Information Systems

MSc
Academic Year: 2018 - 2019

Supervisor: Dr Emanuele Pagone
September 2019

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE TRANSPORT AND MANUFACTURING
Management & Information Systems

MSc

Academic Year 2018 – 2019

JUAN ANTONIO JUST AMARGÓS

Sensitivity of Multi-Criteria Decision Making in Sustainable Manufacturing Systems

Supervisor: Emanuele Pagone
September 2019

This thesis is submitted in partial fulfilment of the requirements for the degree of Master of Science
(NB. This section can be removed if the award of the degree is based solely on examination of the thesis)

ABSTRACT

Multi-Criteria Decision Making (MCDM) is a powerful tool that can support decision makers in choosing among alternatives combining their characteristics in order to consider conflicting features.

The aim of this study is to find an innovative approach to make a sensitivity analysis to MCDM methods, in order to demonstrate the robustness of the decision process, specifically applied to the MCDM method Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). This analysis is suited to be applied in the area of sustainable manufacturing to weight the importance of traditional performance indicators (e.g. time, cost, quality) with a wide spectrum of sustainability areas (i.e. economic, environmental, social…).

Results of the Sensitivity Analysis will determine the amount of variations of rank reversals when the different criterion weights are changed and will measure the change needed to create these variations, resulting in a new and accurate value of each criterion's sensitivity.

Keywords:

MULTI-CRITERIA DECISION MAKING, SENSITIVITY ANALYSIS, WEIGHT, SENSITIVITY COEFFICIENT, TOPSIS

ACKNOWLEDGEMENTS

I would like to express my total gratitude to my supervisor ÇDR Emanuele Pagone for providing me the opportunity to work on this project, guiding and helping me during it. It was a great learning experience on a really interesting topic I barely knew before it began.

I would like to thank too my closest friends, not only the ones I met in Cranfield University, but also the ones back in my city that supported me along my year hear, with special mention to the ones that helped me to achieve me important data for the project's completion. Finally, my deference to my parents and family without whom I wouldn't have had the chance to live such an enrichment experience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

SC	Sensitivity Coefficient
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
MCMD	Multi-Criteria Decision Making
MADM	Multiple Attribute Decision Making
MODM	Multiple Objective Decision Making
AHP	Analytic Hierarchy Process
VIKOR	Multicriteria Optimization and Compromise Solution
WPM	Weighted Product Model
WSM	Weighted Sum Model
rAHP	revised Analytic Hierarchy Process
COPRAS	Complex proportional Assessment
HPDC	High Pressure Die Casting
PROMETHEE	Preference Ranking Organization Method for Enrichment of Evaluations
OWA	Ordered Weighted Averaging
GPU	Graphic Processing Unit
ELECTREE	Elimination and Choice Expressing Reality
GIS	Geographical Information Software
OPSC	Operating Point Sensitivity Coefficient
TSC	Total Sensitivity Coefficient
MSI	Measurement Scale Independency
CFI	Criteria Formulation Independence
HDT	Heat Deflection Temperature

1 Introduction

The following Individual Research Project focuses on the objectives of coming up with a new definition of the Sensitivity Coefficient (SC) and a complete methodology for a specific weight related Sensitivity Analysis, oriented to the Sustainable Manufacturing sector and to the Technique for Order of Preference by Similarity to the Ideal Solution (TOPSIS).
The new definition of SC and the methodology used have been selected after numerous researches over what has been done recently in the Sensitivity Analysis world, investigating the lacks in the definition and developing new parameters to create a more complete description of the coefficient.

This approach to the Sustainable Manufacturing market is accomplished thanks to the company chosen for the Case Study developed in section 4. Not only the Environmental Sustainability of the production process of the company is analysed, but it also fulfils a series of conditions to be considered into this Sustainable Manufacturing market. Numerous examples defend this classification such as three of the key targets they have proposed for 2020 to improve the company environmental performance:

- Total Energy Usage (amount of energy used against the Cost of Goods Sold): The target for 2020 is a 40% reduction normalized to cost of goods sold from 2008 baseline year.
- Water Usage (amount of water used against the Cost of Goods Sold): the target for 2020 is a 40% reduction normalized to cost of goods sold from 2008 baseline year.
- 20% of Waste Redirected from Landfill.

The Sensitivity Analysis is applied to the Case Study, calculating all the results and parameters to define the robustness of the decision-making process.

2 Literature Review

2.1 Introduction

The purpose of the literature review focuses on the idea of defining a new approach to Sensitivity Analysis application and demonstrate the robustness in a Multi-Criteria Decision Making problem using a method known as TOPSIS. The literature review questions are:

- What is Multi-Criteria Decision Making?
- What is TOPSIS?
- How is a Sensitivity Analysis performed?

2.2 Multi-Criteria Decision Making

MCDM tools are techniques used in operations research that evaluate the different criteria involved in decision making, criteria which are usually in conflict among them. These methods can be used not only in daily life situations but also in any other aspects of life and business that involve a decision with multiple options (in our case we will focus on a Sustainable Manufacturing situation). Frequently, in the decision making process, certain criteria are inversely proportional in terms of suitability for the decision maker, such as price and quality for example. The Multi-Criteria Decision Analyses tries to solve this habitual conflict between the different criteria and the conditions of each alternative related to the criteria.

Structuring complex problems and appropriately weighting the different criteria leads to make better decisions with the support of an appropriate basis of knowledge. Since the start of the MCDM discipline in the 1960s there have been numerous advances, many developed through specialised decision-making software.

The problems of MCDM can be classified into two categories: Multiple Attribute Decision Making (MADM) and multiple objective decision making (MODM):

- MODM methods have decision variable values that are determined in a continuous or integer domain, with either an infinite or a large number of choices, the best of which should satisfy the decision maker's constraints and preference priorities.

- MADM methods are generally discrete, with a limited number of predetermined alternatives. MADM is an approach employed to solve problems involving selection from among a finite number of alternatives. An MADM method specifies how to process attribute information in order to arrive at a choice. MADM methods require both inter- and intra-attribute comparisons and involve appropriate explicit trade-offs.

"Each decision situation requires at least 4 main parts: alternatives (A_i with i from 1 to N), criterions (C_j with j from 1 to M), weights (w_j with j from 1 to M) and measure of performance of the alternative in terms of the criterions (m_{ij} with I from 1 to N and j from 1 to M)." [1]

Along the history of MCDM, numerous researchers have used many different Multi-Criteria Decision Making methods, such as Weighted Product Method (WPM), Analytic Hierarchy Process (AHP) Method, Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) Method, Compromise Ranking Method (VIKOR), etc…

Depending on the situation studied and the belief of the researchers, the different techniques have been applied in many case studies and not only their validity, but their robustness has been demonstrated. Some authors have suggested that it is important to use alternative MCDM methods in order to achieve reliable and viable ranking results while others just deeply research the methods and choose the one that seems to be the most trustworthy.

For example, Vida Maliene et al. uses five commonly used MCDM models: Weighted Sum Model (WSM), Weighted Product Model (WPM), revised Analytic Hierarchy Process (rAHP), Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), and Complex proportional Assessment (COPRAS). The environment they work in is related to the affordability of sustainable housing and the acquired data comes from conducting interviews and literature review. A total of eighteen decision criteria and eleven alternatives were used in the assessment of sustainable housing affordability, as an empirical case study [2].

Regarding the paper to be presented, this literature review will focus on the TOPSIS method.

2.3 TOPSIS Method

The TOPSIS method was created in 1981 by Hwang and Yoon, and it is based on the idea that there is an ideal best solution and an ideal negative solution. The best alternative among the possible ones should have the shortest Euclidean distance to the best ideal solution and the largest distance to the negative ideal solution, according to the method. The ideal solution is a hypothetical solution for which all the attribute values are equal to the maximum attribute values in the database comprising the satisfying solutions, while the negative ideal solution is the one for which all attribute values represent the minimum values of the different attributes among the options.

Different authors have used TOPSIS as their key method to solve their MCDM problems in their papers. E. Pagone et al. used the TOPSIS method to select the most suitable material for the manufacture of an automotive component using the High Pressure Die Casting (HPDC) process.
The performance of three different alternatives (Aluminium-A380, Magnesium-AZ91D and Zinc-ZA8) was assessed based on four different criterions. The four different performance measurements of the criterions were evaluated, and each one of the four classes of criteria examined was assessed using a number of metrics normalised by mass to generalise the yielded results. After applying all the algorithm calculations and making the comparison with the ideal solutions, the decision about which was the most suitable material was taken [3].

2.4 Sensitivity Analysis

We can use Sensitivity as an indicator in multiple fields with different meanings or definitions. However, different studies agree when they mathematically define it as an index that shows the dependence of the output in terms of the input. For instance, in their research paper, S. Ray et al., [4] state their sensitivity index using Jorgensen's (1994) definition as:

$$S = \frac{\frac{dx}{x}}{\frac{dp}{p}} \tag{2-1}$$

Where S stands for sensitivity, x is the state variable, P parameter, dx and dp are change of values of state variables, parameters, respectively.

In the case we will focus on, the studied and considered Sensitivity Analysis can be defined as the study of how uncertain the output of a mathematical model can be as a consequence of variations allocated to the different inputs.

This process of recalculating the output under assumptions that change the initial input values to determine the impact in a decision under sensitivity analysis helps to:

- Understand better the relationship between the input variable and the output variables.
- Test the robustness of the system.
- Check the model to find for hidden errors in it.

However, the way of applying this sensitivity analysis varies among the different researchers who want to check the stability and robustness of the decision they made. Several ways of challenging the robustness of a decision have been created during the recent decades and each of them focuses on a particular characteristic of the decision process. After an intense research, the following classification of sensitivities has been established:

- Weights Sensitivity
- Scale Sensitivity
- Criteria Formulation Sensitivity

2.4.1 Weights Sensitivity

2.4.1.1 Most Critical Criterion

The authors previously mentioned ([2]) consider that the Sensitivity Analysis should be carried out through two approaches:

- First, by quantifying the level of crosstalk between criteria and ranking by making small (5%) or large (50%) changes in the criterions weights to get specific values of relative sensitivity coefficients.
- Second, by determining the most critical criterion (criterion for which the smallest relative change, in percentage, in its weight value must occur to alter the existing ranking of the alternatives) of the criterions so that it could furtherly be treated carefully.

The final definition of the sensitivity term is described with the following expression:

$$SC_j = \frac{1}{D_j}, \ j=1,2\ldots n \qquad \textbf{(2-2)}$$

Where SC_j is the measurement of sensitivity per criteria and D_j was defined as the smallest relative change (defined in the weight value). that must occur to change the rank reversal of the most critical criterion.

This mathematical definition does not seem to be accurate enough as it considers a criterion with a high number of rank reversal changes as sensitive as one with just one change as long as the distance to the closest change is equal for the both of them. This definition will be modified in the development of this study in order to really assess the stability and robustness of a MCDM mathematical model.

This idea of determining the most critical criterion continues appearing in research papers such as [5], where this criterion is defined in four ways:

- The criterion that makes any rank reversal with the smallest absolute weight change.
- The criterion that changes the first ranked alternative with the smallest absolute weight change.

- The criterion that makes any rank reversal with the smallest relative weight change.
- The criterion that changes the first ranked alternative with the smallest relative weight change.

However, it lacks a mathematical definition for the sensitivity coefficient derived from the identification of this criterion.

2.4.1.2 Stability Interval

A different analysis is applied in [6], the Sensitivity Analysis is performed over PROMETHEE and TOPSIS weights by using stability intervals. The calculations that the PROMETHEE algorithm applies to reach the rank reversal are also applied to TOPSIS method and the different stability regions are obtained. In this particular case, PROMETHEE and TOPSIS results are in the same trend and they have similar stability analysis results. In the meantime, PROMETHEE and TOPSIS results vary little because the notations of the methods and the additional parameters that the decision makers decide subjectively.

These stability intervals are widely used among researchers and, expressed in different or similar ways, they represent the same. For example, in [7] the following framework is proposed to calculate the stability interval's thresholds:

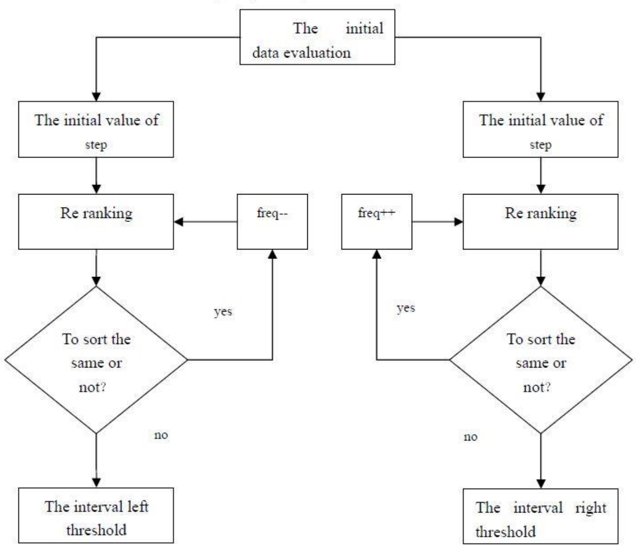


Figure 2-1 – Framework methodology for calculating the interval's threshold.

As we can see, the approach is alike, starting with some initial data, iterating with a determined frequency and increasing it until the ranking is modified, then the thresholds of the intervals are set.
Moreover, in A. Rosa et al. this stability interval is applied to Multi objective linear programming problems, so this interval becomes a zone in the 2 dimensional cartesian plane defined as a hypercube centred on the estimated weight with a radius called the allowable tolerance which fulfils the condition of maintaining the current rank [8].
Furthermore, in N. Zhao et al. the Ordered Weighted Averaging (OWA) method is used to define the sensitivity of the case study. With a similar approach to the calculation of the stability region, the orness measure is calculated and the tolerance of the region (ηi) is determined [9].
Finally, M. Jirí proposes an approach full of similarities with the methodology defended in this study. His idea of sensitivity analysis is set forth in the following five steps [10]:

1. Perform standard TOPSIS while saving the matrixes R^+, R^- (the two vectors of the distances of unweighted criteria), W (vector of weights) and the resulting order of the alternatives for further use.
2. Change one weight and adjust the other weights, construct W' out of the changed values.
3. Calculate the new distances from the saved vectors.
4. Calculate the new relative closeness using the standard TOPSIS procedure.
5. Evaluate the new order of alternatives and check if it equals the original, saved order.

This paper, despite proposing a good theoretical working framework where the stability region is in fact being calculated, lacks a mathematical calculation of the sensitivity index itself (as most of the papers) and doesn't give an example of an application of the framework, there is no real sensitivity analysis suggested. However, the basis of the methodology relates to the one propounded in the following study and can be considered as a good starting point in order to perform a complete sensitivity analysis.

2.4.1.3 Sampling Scenarios

Simpler approaches such as sampling different scenarios with weight variations and checking the ranking variations have been widely performed too. This modality is followed in the study of Y. Gülsah et al., in order to compare different preference scenarios, an equal weights case and four additional cases are analysed, each of the additional sensitive cases expresses a different point of view (technocratic, mercantilist, eco-social and administrative), with a different weight distribution reinforcing the specific weight of the criterion most valuable according to the decision maker's point of view. [11]

H. Dinçer et al. apply the same methodology for nine different cases with variance among the criteria weights in order to prove the stability of the decision made. [12]

N. Favretto et al. assess six different situations in which certain criterion weights are doubled or halved in order to visualise the sensitivity of the analysed criterion. [13]

Moreover, D. Martín et al. uses five alternative criteria weighting schemes and one alternative scoring scheme to observe the alteration in the output values and check their behaviour. [14]

In addition, S. Ray et al. suggest an approach of variations of 10% in the criteria weights in order to determine the robustness of the model. [4]

C. Erlacher et al. try to go a bit further and change the common manual sampling by using a Graphic Processing Unit (GPU)-based approach to sample random scenarios with computational aid evaluating weight modifications to assess the sensitivity of the case study. [15]

Finally, the research paper, N. Kokaraki et al. try to give a new point of view about the Weights Sensitivity Analysis. In this case, the sensitivity analysis is carried out by analysing three causes of uncertainty: criteria weightings, stakeholders' design preferences and stakeholders' relative importance over the decision process. In order to test these ideas, four different scenarios were chosen and simulated: [16]

1. The first scenario is the base case study with the initial weight distribution (Stakeholder1= 11%, Stakeholder2= 57%, Stakeholder3=27%, Stakeholder4= 5%.)
2. In scenario 2, the selected performance criteria have equal importance for all the stakeholders. Equally-weighted criteria is a common situation, against which the sensitivity of the results is tested. In this case study the selected performance criteria are six and thus each one has a weight factor of 100%/6=16.667%.
3. Scenario 3 incorporates a change in the stakeholders' relative importance (10% 50%, 35% and 5% respectively).
4. In scenario 4, the client considers the options that have an specific characteristic to be a little more desirable, concluding to alternatives No.25 to No.32 as more preferred.

Despite the two last scenarios seem to be innovative approaches, they end up being a different way of modifying the specific weights of the criteria, so the three uncertainty causes appear to be alike. If you modify the preferences of the stakeholders (decision makers), you modify the weight distribution assigned to the criteria; and if you modify the importance of these stakeholders, you also modify the importance of the one that supports more or less a criterion, which ends up in a modification of the specific weights of the criteria.

Furthermore, as there is no mathematical way of defining and comparing the sensitivity indexes of the situation. The authors make a comparison of the different scenarios two by two with the different MCDM techniques used (AHP, TOPSIS, ELECTRE III – Minimum Rank, ELECTRE III – Maximum Rank and PROMETHEE II) so they are able to graphically see which is the method that has the lowest occurrence rate of rank reversal despite no figure of sensitivity is stated.

These ways of analysing sensitivity by spotting specific situations can result interesting as it is a quick way of evaluating sensitivity depending on the decision maker’s character. However, the analysis itself it is poor as it only considers some specific situations and does not give a value for the sensitivity itself (a lack seen in a big quantity of articles related to MCDM sensitivity that tries to be solved by this study).

2.4.1.4 Graphical Sensitivity

A new way of representing the weights sensitivity is shown in [17]. The main difference between this paper and the ones above mentioned is that the desired result of the sensitivity analysis in its case study is not a figure of the sensitivity index, but a graphical solution. The application of the MCDM technique together with the usage of GIS (Geographical Information Software) result in heat maps of the analysed provinces with the distribution of the criteria performance in the entire region. This leads to a sensitivity analysis represented by different graphs showing the weight distribution per criterion in the X-axis and the hectares (ha) classified as highly desirable in the studied province in the Y-axis. There is one graph for each province and each graphic explains graphically the behaviour of each criterion depending on the weight variation; in another words, the slope of the graph represents the sensitivity of each criterion depending on the weight variation.

Exactly the same method but with different structure is used to as they create one graph for each alternative representing the variation of the output (the cost) in relation with the input (the cubic meters of land removed) [18]. This technique can represent the sensitivity of each alternative depending on the percentage of completion of the duty.

2.4.1.5 Remaining Sensitivities

E. Hernández et al. propose a different mathematical definition for sensitivity. They assume that all of the performance values and weights are considered as inputs and their uncertainties are modelled as random variables properly characterized through known probability distribution functions [19]. This means that the ranking is now a random variable, and its elements have files that represent the distributions of the possible ranks. With this new scenario, some constrains (weight change scenarios) are set and a random number of samples is taken as result of the scenarios. Recording the value of y^k (being y^k the status of the kth sample), being 1 if the constrain is satisfied or 0 if it is not, the sensitivity is defined as:

$$Sensitivity = \frac{TP}{TP + FN} \tag{2-3}$$

Where:

- TP is the number of examples where the constrain is satisfied.
- FP TP is the number of examples where the constrain is not satisfied.

The same exact definition is used in N. Chai et al.'s article where they use not only the sensitivity but also the specificity to draw the Receiver Operating Characteristic (ROC) curve of the one parameter against the other one.[20]
A diverse expression for sensitivity is defined in H. Chen's et al. article, introducing two new indicators to determine the sensitivity's characteristics of a decision-making problem [21]:

- The Operating Point Sensitivity Coefficient (OPSC): defined as the shortest distance from the current contribution value to the edges of its tolerance. Depends on the operating point and the direction of the weights change (if it is increasing or decreasing). Mathematically is defined as:

$$OPSC = Min\{|\delta_{l-}^{O}|, |\delta_{l+}^{O}|\} \tag{2-4}$$

- The Total sensitivity Coefficient (TSC): Specifies that the shorter the tolerances of a decision element's contributions are, the more sensitive the final decision is to variations of that decision element. Mathematically is defined as:

$$TSC = \left|\delta_{l+}^{O} - \delta_{l-}^{O}\right| \tag{2-5}$$

Where:

- The allowable range of perturbation is $\left[\delta_{l-}^{O}, \delta_{l+}^{O}\right]$ to preserve the initial ranking.

A different definition is set out in J. Song and E. Chung's article, where the researchers focus their efforts in developing Guillen et al. (1998) definition for the robustness of the decision [22]. This definition gives a value for the robustness between two of the selected alternatives being only able to assess them by pairs. Mathematically can be expressed as:

$$\gamma(a_1, a_2) = \frac{W_1 \times (X_{1,1} - X_{1,2}) + \cdots + W_m \times (X_{1,m} - X_{1,m})}{W_1 \times |(X_{1,1} - X_{1,2})| + \cdots + W_m \times |(X_{1,m} - X_{1,m})|} \quad \textbf{(2-6)}$$

Where:

- $\gamma(a_1, a_2)$ is the robustness between alternatives 1 and 2. The result can range from -1 to 1.
- W_m is the criteria weight applied to criterion m
- $X_{1,m}$ is the performance value of criterion m of Alternative 1.

Once the robustness values are calculated, we can calculate the new criteria weights (W^*) that reverses the ranking between the paired alternatives with the following equations:

- If $X_{1,1} > X_{1,2}$, then $W_1^* = W_1 - W_1 \times \gamma(a_1, a_2)$ **(2-7)**
- If $X_{1,1} \leq X_{1,2}$, then $W_1^* = W_1 + W_1 \times \gamma(a_1, a_2)$ **(2-8)**

The limitations of this method are:

- It is only applicable to the Weighted Summation Method (WSM)
- Critical Criteria weights cannot be identified, as an equal proportion adjusts all weights.
- The sum of the optimized criteria weights no longer equals the sum of the initial criteria weights.

2.4.2 Scale Sensitivity

This weight sensitivity approach does not seem to be enough for A. Randjelovic. In this paper, the researcher assesses the consistency evaluation of an MCDM method on an example of the logistical centre location selection, defending that the Sensitivity Analysis, in order to be complete and trustful, should consist of:

- Solution stability assessment in the case that weight of criteria is changed. This approach is similar to the one proposed in the previous reference and focuses mainly in the variation of original weights distribution and its effects. [23]

- Result consistency analysis according to changes in the measurement scale used to depict qualitative criteria. Using French's (1988) "Normative theory of decision making in risk and uncertainty conditions" definition of the Measurement Scale Independency (MSI): "The value of the outcome of an action A_i during the realization of circumstance θ_j is labelled as v_{ij} and expressed in cardinal units of usefulness of the decision maker (it is measured on an interval scale). We can measure cardinal values on different measurement scales, whereby outcome values are measured on two scales, v_{ij} and v_{ij}^+, mutually connected by positive affine transformation [23]:

$$V_{ij}^{+} = a * V_{ij} + b \quad \textbf{(2-9)}$$

Where a and b are constants under condition of a > 0." Regarding this condition, when evaluating a criterion according to a specific scale, the measurement of performance of these criteria must be tested in different scales and fulfil the MSI condition and the results of the initial weighted criteria must not change for the decision maker.

2.4.3 Criteria Formulation Sensitivity

Finally, in the paper above mentioned, an innovative approach related to the sensitivity of the way the different criteria are formulated takes place. The researchers state that the result of the consistency analysis considering criteria formulation in a case when the same criterion can be shown in two normatively equivalent ways can differ. Using "Kahneman & Tversky (1981) definition of Criteria Formulation Independence (CFI), which is formulated based on the descriptive invariability condition described in the behavioural theory of decision making as the condition of choice rationality of an individual decision maker." [23] defending that, despite the multiple existing ways of formulating a criterion (with a positive or negative framework), this should not affect the decision maker evaluation. According to these observations, Pamučar & Ćirović (2015) defend that "Positive and negative framework are connected with a function:

$$X_j^- = \frac{C}{X_j^+} \quad \textbf{(2-10)}$$

Where C is a constant and X_j refers to the positive or negative framework depending on the sign they have." Regarding this condition, when evaluating a criterion according to a specific framework or formulation, the measurement of performance of these criteria must be tested within the different frameworks, they must fulfil the CFI condition and the results of the initial weighted criteria must not change for the decision maker.

This analysis was applied to the TOPSIS, COPRAS, VIKOR, and ELECTRE methods and showed that these methods are sensitive to changes in weight but they maintain the priorities of alternatives.

These two innovative approaches referred to the independency of scale and criteria formulation represent interesting ways of checking the validity of the initial conditions of the MCDM analysis. However, they do not really seem to fulfil the definition suggested for sensitivity analysis as they do not challenge the robustness of the method (TOPSIS in this case), only testing really the correct formulation of the case study and the initial weight distribution selected by the decision maker. They can be considered as good complements for a Sensitivity Analysis in order to pre-check the situation where the analysis will be applied but will not probably be applied in this study.

2.5 Summary

This literature review has served to describe the Multi-Criteria Decision Making methods, their categories, their common uses, and mentioned some of the different types using references found during the research.

Among this broad list of techniques, TOPSIS has been chosen for the case study concerning this project. The TOPSIS method has been analysed, reinforcing this choice with the different references to papers where it has also been used. In the following parts of this work further information about TOPSIS, the algorithm, and the steps will be explained.

Finally, a complete view of the Sensitivity Analysis and the Sensitivity index itself is shared. Several papers of authors of the recent decades have been analysed and challenged if their description of sensitivity lacked a mathematical description, an analysis or any characteristic.

These three mains (MCDM techniques, TOPSIS and Sensitivity Analysis) topics already covered enable the reader to have the appropriate knowledge to understand the next steps of this research project and be aware of the techniques used.

3 Methodology

3.1 Introduction

Several ways of challenging the robustness in a Multi-Criteria Decision Making problem have been developed during the recent years, these Sensitivity Analysis have also come up with different possibilities of defining mathematically the term of Sensitivity.

In this research project, a new methodology for this Sensitivity Analysis, that differs from those previously formulated, is suggested and defended.

3.2 Case Definition

In order to proceed with this analysis, firstly we must define the case. This involves deciding on the problem we are facing with the alternatives among which we can choose, the different criterions and sub criterions for the decision makers, and the measures of performance of the alternatives in relation to the criterions.

3.3 TOPSIS Method

Once these data are stablished it is time to apply the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method, the steps for the calculation are the following:

Step 1: The first step is to determine the situation to be studied, the objective, and defining the four above mentioned main parts of the decision situation (alternatives, criterions, weights, and measurement performances).

Step 2: This step represents a matrix based on all the information available on criterion (Decision Table). In this matrix, each row is allocated to one alternative, and each column to one attribute.

Therefore, an element m_{ij} of the decision table gives the value of the j-th criterion in original real values (non-normalized form and units) for the i-th alternative.

In the case of a subjective criterion, a ranked value judgement on a scale is adopted, and then the adopted value is calculated in the same manner as the objective ones.

Step 3: Calculating the normalized decision matrix, R_{ij}. We can calculate this matrix with the following expression:

$$R_{ij} = \frac{m_{ij}}{\sqrt{\sum_{j=1}^{M} m_{ij}^2}} \tag{3-1}$$

Step 4: Decide the weights of the different criterions (w_j (for j from 1 to M)) fulfilling the rule that $\sum W_j = 1$.

Step 5: Calculate the weighted normalized matrix V_{ij}, multiplying of each element of the column of the matrix R_{ij} by its associated weight w_j. The elements of the matrix V_{ij} can be expressed by means of the following formula:

$$V_{ij} = w_j \times R_{ij} \tag{3-2}$$

Step 6: Obtain the ideal and negative ideal solutions in this step. These can be expressed as:

$$V^{+} = \left\{ \frac{\sum_i^{max} V_{ij}}{j \epsilon J}, \frac{\frac{\sum_i^{min} V_{ij}}{j \epsilon J'}}{i} = 1,2, \ldots N \right\} \tag{3-3}$$

$$= \{V_1^{+}, V_2^{+}, V_3^{+}, \ldots V_m^{+}\}$$

$$V^{-} = \left\{ \frac{\sum_i^{min} V_{ij}}{j \epsilon J}, \frac{\sum_i^{max} V_{ij}}{j \epsilon J'} / i = 1,2, \ldots N \right\} \tag{3-4}$$

$$= \{V_1^{-}, V_2^{-}, V_3^{-}, \ldots V_m^{-}\}$$

Where J = (j = 1, 2, …, M) /j is associated with beneficial attributes, and J' = (j = 1, 2, …, M) /j is associated with non-beneficial attributes. V_j^+ indicates the ideal value of the considered criterion among the values that criterion for different alternatives. In the case of beneficial criterion, V_j^+ indicates the higher value of the attribute. In the case of non-beneficial criterion, V_j^+ indicates the lower value of the attribute.

V_j^- indicates the negative ideal value of the considered criterion among the values of that criterion for different alternatives. In the case of beneficial attributes, V_j^- indicates the lower value of the criterion. In the case of non-beneficial attributes, V_j^- indicates the higher value of the attribute.

Step 7: Calculate the separation measures. The separation of each alternative from the ideal one is given by the Euclidean distance in the following equations.

$$S_i^+ = \left\{\sum_{J=1}^{m} \sqrt{\left(V_{ij} - V_j^+\right)^2}\right\}, i = 1,2, \dots, N \tag{3-5}$$

$$S_i^- = \left\{\sum_{J=1}^{m} \sqrt{\left(V_{ij} - V_j^-\right)^2}\right\}, i = 1,2, \dots, N \tag{3-6}$$

Step 8: The relative closeness of an alternative to the ideal solution (P_i) can be expressed as:

$$P_i = \frac{S_i^-}{S_i^+ + S_i^-} \tag{3-7}$$

Step 9: The different alternatives are ranked in descending order, according to the value of P_i indicating the most preferred (highest P_i) and least preferred feasible solution (lowest P_i). P_i may also be called the overall performance score of an alternative.

3.4 Sensitivity Analysis

Once the ranking for the initial weights that the decision maker set is achieved, the Sensitivity Analysis takes place. In first place, using an iteration algorithm written with R © software, we reproduce the above mentioned TOPSIS procedure changing the weights of each different criterion from 1% to 100% with steps of 1%, and distributing the difference in percentages between the new weight and the original one equally among the remaining criterions. This procedure will give us enough information to create graphs such as the following one:

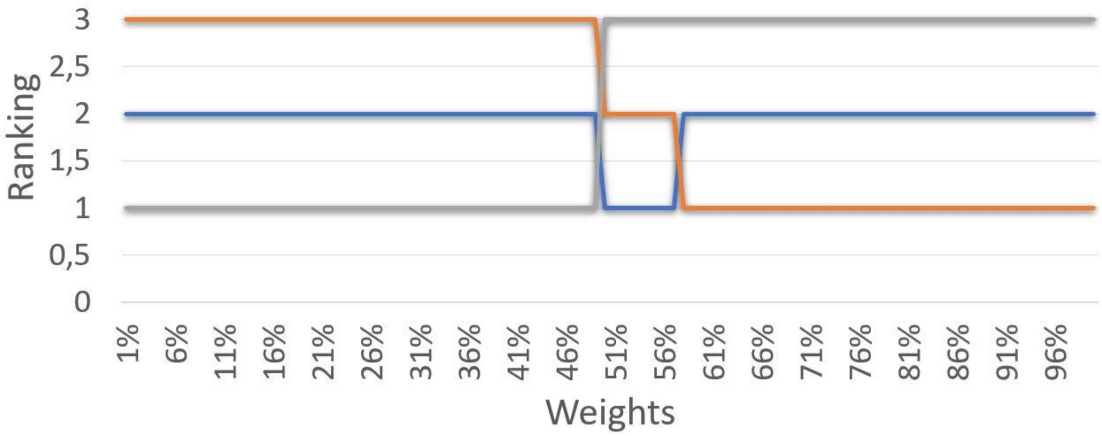


Figure 3-1 – Ranking/Weights graph example

Once the stability and the changing intervals are defined (like in the picture above), we should quantify the distance in terms of percentage from the original weight decided by the decision maker to the changes of rank reversal.

With all these data, the sensitivity of each criterion can be calculated according to the following formula:

$$S_j = \frac{1}{\Delta D_j} + \frac{\boldsymbol{n_j}}{\boldsymbol{\Delta D_j}} \qquad \textbf{(3-8)}$$

Where:

- ΔD_j is the minimum weight variation to the first ranking change from the original weights decided by the decision maker in percentage units.

- $\boldsymbol{n_j}$ is the number of rank reversals during the 100 iterations.
- $\overline{\Delta D_j}$ is the average of the weight variations from the original weight to the changes of rank reversal expressed in percentage units.

According to this definition, we manage to reflect the inversely proportional impact in the sensitivity term of the ΔD term, the closest the first rank reversal change is, the highest the sensitivity is. The effect of the number of changes during the iteration is expressed with the term $\boldsymbol{n}$, these changes represent instability and increase the value of the sensitivity as the number of changes grows. Finally, the term $\overline{\Delta D}$ allows to distinguish the difference between criterions where changes of rank reversal appear closer to the initial weight of the decision maker (what clearly means instability as a smaller modification in decision weighting is needed) and the criterions where the changes appear almost in the final zones of the graph.

4 Case Study

4.1 Introduction

The proposed Case Study focuses on the selection process of the most adequate material for the barrel of a pre-fillable syringe, a product designed to enable the performance for large volume drug-delivery.

Several specifications describing the performance of the different resins used to produce this part of the product have been studied, and the most significant have been finally taken into account and measured. Their positive or negative impact has been assessed considering the effect of an increase in the quantity of the metric for each criterion. Furthermore, the different metrics have been classified according to four criteria:

1. Cost
2. Quality
3. Environmental Sustainability
4. Time

4.2 Criteria

4.2.1 Cost

Some of the parameters needed to calculate the Cost section are expressed in table 1.

Cost Parameters	Resin 1	Resin 2	Resin 3	Impact
Density (kg/m^3)	1200*	1023*	900*	Negative
Extrusion Energy (MJ/kg)	6.43*	10.9*	6.5*	Negative
Moulding Energy (MJ/kg)	20.6*	28.4*	22.6*	Negative

Recycling Energy (MJ/kg)	38.7*	63.8*	27.2*	Negative
Raw Material Cost (£/kg)	7.325	11.459	2.851	Negative

Table 4-1 –Initial Parameters for Cost calculations

* This data were acquired from CES EduPack 2018 software. [24]

The different parameter's importance is explained following:

- Density: Once the dimensions of the barrel are measured and the volume of raw material required to produce it is calculated, we have to know the density in order to calculate the mass of raw material needed, and, therefore, its price per unit. This price will comprise not only the raw material itself but also the price of the energy needed to produce it.
  After measuring the size and dimensions of a unit of the product, the volume of the finished good is of 38,02 cm^3. The production processes of extrusion and moulding have a material utilisation fraction of [0.9-0.99], in our case study we will consider the value of 0.9. This value has been found in CES EduPack 2018 software [24]. The software's sources only refer to the processes, not to the materials, so the utilisation fraction has been considered equal for the three resins. This gives us a quantity of 42.24 cm^3-of-required raw material in order to achieve the desired volume in the finished good.
- Extrusion and Moulding Energy: the process of transforming the raw material involves these two actions. Using this data, we can calculate the amount of energy needed and, considering that this process uses only electricity, calculate the cost of this process per unit.
- Recycling Energy: we are focusing on the entire life cycle of the product, so the recycling part is also important in our study. Using the same path than in the property afore mentioned, the cost of energy for the recycling process can be estimated (due to lack of information about this process, it has been assumed that the totality of it can be executed with electric

energy, using the same energy cost than n the extrusion and moulding parameter to calculate the final value).

- The Raw Material cost has been provided by the company of the case study in £/Kg, this unit will be converted into an extensive magnitude (£) using the density and the volume of each unit.

Moreover, the intensive magnitudes of energy will be transformed from MJ/Kg to kWh/Kg applying the following unit's conversion:

$$1\frac{MJ}{kg} = 0.277778\frac{kWh}{kg} \quad \textbf{(4-1)}$$

Finally, in order to calculate the cost of energy consumption for the process the price of the kWh of electric energy in 2018 in UK has been used [25]:

$$1\ kWh = 12.376\ pennies \quad \textbf{(4-2)}$$

As a result, the following table that summarises the different Economic parameters related to the Cost Criterion:

Cost Parameters	**Resin 1**	**Resin 2**	**Resin 3**	**Impact**
Manufacturing Process Cost (£)	0.047	0.058	0.038	Negative
Raw Material Cost (£)	0.371	0.495	0.108	Negative
Recycling Cost (£)	0.067	0.095	0.036	Negative

Table 4-2 – Definitive Cost Parameters

The metrics initially considered finally provided us with three main indicators:

- Manufacturing process cost: Involves the cost related to the energy consumed in the extrusion and moulding processes.

- Raw material cost: Involves the cost of the raw material itself plus the cost of the energy needed to produce it.
- Recycling process cost: Involves the cost of the energy needed to recycle the material once the life cycle of the product is over.

4.2.2 Quality

The different properties that have been evaluated for the Quality section are expressed in table 3.

Quality Parameters	**Resin1**	**Resin 2**	**Resin 3**	**Impact**
Density (kg/m^3)	1200*	1023*	900*	Negative
Heat Deflection Tª (ºC)	[132-142]*	[131-134]*	80*	Positive
Transparency	Transparent	Opaque	Translucent	Positive

Table 4-3 – Initial Quality Parameters table with qualitative data

* This data were acquired from CES EduPack 2018 software [24].

We can classify the characteristics above mentioned in three different criteria:

- Density: Its importance lies in the weight of the material, the denser it is, the heaviest and uncomfortable it will be for the final user.
- Heat Deflection Temperature: is the temperature at which a polymer or plastic sample deforms under a specified load. This is taken into account during storage in bad conditions, the highest the property is, the less risks of deformation the product will have. As two of the resins present intervals as a performance measure for this sub-criterion, the middle value of the ranges has been considered for the case study.

Transparency: The syringe will contain drug and an appropriate visual contact with it is necessary, as the user may need to know how full the device is or if it is clean or not before applying it. The information obtained this parameter is qualitative, but TOPSIS needs quantitative values in order to apply the algorithm, so these values should be transformed. Following the steps of A. Randjelovic, Fuzzificated Likert scale was used in order to evaluate the qualitative criteria (Camparo, 2013). The qualitative values were transformed into triangular fuzzy distributions that after were defuzzificated with the following formula [23]:

$$A = \left[\left(a^{(r)} - a^{(l)}\right) + \left(a^{(m)} - a^{(l)}\right)\right] \times 3^{-1} + a^{(L)} \qquad \textbf{(4-3)}$$

Where:

- $a^{(r)} and\ a^{(l)}$ respectively represent the right and left confidence interval of the triangular distributions.
- A represents the value of the defuzzificated parameter, the final transformation of the qualitative initial value.
- $a^{(m)}$ represents the peak value of the triangular distribution.

Quality Parameters	Resin 1	Resin 2	Resin 3	Impact
Density (kg/m^3)	1200	1023	900	Negative
Heat Deflection Tª (ºC)	137	132,5	80	Positive
Transparency	4.11	0.88666667	2.49666667	Positive

Table 4-4 – Definitive Quality Parameters table with defuzzificated quantitative parameters

4.2.3 Sustainability

In terms of sustainability, the aim is to focus on the environmental sustainability parameters that are involved in the total life cycle of the product, including: production, extrusion, moulding and recycling processes. Table 5 represents the initial data measured and the initial metrics taken into account to evaluate this criterion:

Sustainability Parameters	**Resin 1**	**Resin 2**	**Resin 3**	**Impact**
Production CO_2 Footprint (Kg/Kg)	4.99*	9.93*	1.89*	Negative
Production Water Usage (l/Kg)	182*	451*	41.2*	Negative
Extrusion CO_2 Footprint (Kg/Kg)	0.43*	0.874*	0.488*	Negative
Extrusion Water Usage (l/Kg)	7.24*	9.79*	7.28*	Negative
Moulding CO_2 Footprint (Kg/Kg)	1.55*	2.27*	1.69*	Negative
Moulding Water Usage (l/Kg)	18.9*	23.3*	20*	Negative
Recycling CO_2 Footprint (Kg/Kg)	2.56*	3.38*	1.17*	Negative
Recycling Water Usage	0.742*	0*	5.81*	Negative

(l/Kg)

Table 4-5 – Initial Environmental Sustainability Parameters table

* This data were acquired from CES EduPack 2018 software [24]

In order to facilitate comprehension, the data from the different steps of the process has been merged depending on its nature, and transformed into extensive values. The water usage per production process was calculated by multiplying the data from Table 5 times the mass of each unit of product. For the calculation of the CO_2 footprint, the value was calculated using the MJ/Kg values appearing in Table 1, multiplying them by the mass of each unit of product depending on its material, transforming the energies into kWh and using the Carbon Intensity of Electricity value for UK 2018 [26]:

$$1kWh = 173\ gCO_2 \qquad \textbf{(4-4)}$$

The final results for the parameters taken into account for the sustainability criterion is shown in table 6:

Sustainability Parameters	**Resin 1**	**Resin 2**	**Resin 3**	**Impact**
Raw Material Production CO_2 Footprint (Kg)	0.243	0.351	0.121	Negative
Raw Material Production Water Usage (l)	8.303	17.541	1.410	Negative
Manufacturing CO_2 Footprint (Kg)	0.059	0.073	0.048	Negative

Manufacturing Water Usage (l)	1.193	1.287	0.933	Negative
Recycling CO_2 Footprint (Kg)	0.085	0.119	0.045	Negative
Recycling Water Usage (l)	0.034	0.051	0.199	Negative

Table 4-6 – Definitive Environmental Sustainability Parameters table

The six main sub-criteria consider the CO_2 footprint that each process (production, manufacturing and recycling) with each material creates measured in kg of CO_2 per unit of product produced, and the water usage described in litres per unit of finished goods produced. These measures clearly have a negative impact, as the higher they are, the more they contaminate or the more they consume, which means a negative effect towards the environmental sustainability pursued in this study.

The energy consumption of each process can't be taken into account individually as a parameter in this criterion as it has already been taken into account in the Cost criterion and TOPSIS forces its criteria to be independent among them and do not repeat data, otherwise the results will lead to errors.

4.2.4 Time

Despite this criterion is commonly one of the most complex ones when referring to production processes. the capacity of mass production of the producing plant and the various moulding machines in the studied location allows us to consider as equal the production time of the different resins, considering also their similarity in terms of the production related characteristics. Therefore, the only metric to be taken into consideration is the lead time of the supplier to deliver the material. These measures can be found below in table 7:

Time criterion	Resin 1	Resin 2	Resin 3	Impact
Lead Time (days)	63	60	90	Negative

Table 4-7 – Time Parameters table

The impact of this criterion is obviously negative as the higher the delivery time is, the longer it will take to buy new raw material needed for the production process.

4.3 Weight Selection Process

The weights along the different criteria and sub-criteria have been defined thanks to a survey answered by 40 students from the Cranfield School of Aerospace. Transport and Manufacturing belonging to different MSc courses with knowledge related to the discussed fields.

The results obtained will define the initial weight distribution for our case study, presented in the following tables:

Table 8 shows the initial weight distribution among the four different criteria, these values will be used as the original point of our sensitivity analysis.

Criterion	Cost	Quality	Environmental Sustainability	Time
Weights Distribution	26%	31%	22%	21%

Table 4-8 – Criteria Weight Distribution

Some of the criteria have sub-criteria, that is why the different specific weights of these sub-criteria should be determined too in order to introduce it in the TOPSIS algorithm.

Table 9 shows the initial weight distributions of the three sub-criteria of the Cost criterion.

Cost Sub-criteria	Manufacturing Energy Cost	Raw Material Cost	Recycling Energy Cost
Weights Distribution	39%	33%	28%

Table 4-9 – Cost's Sub-criteria Weight distribution

Following the path above, Table 10 expresses the division of relative importance among the different sub-criteria of the Quality criterion.

Quality Sub-criteria	Density	Heat Deflection Temperature	Transparency
Weights Distribution	36%	35%	29%

Table 4-10 – Quality's Sub-criteria Weight distribution

Finally, Table 11 expresses the weight distribution of the six sub-criteria for the Environmental Sustainability criterion. Time criterion does not have a weight distribution, as it is the only criterion without sub-criteria.

Environmental Sustainability Sub-criteria	CO_2-RM[1]	Water- RM[2]	CO_2-M[3]	Water-M[4]	CO_2-R[5]	Water-R[6]
Weights Distribution	15%	21%	14%	16%	20%	14%

Table 4-11 – Environmental Sustainability's Sub-criteria Weight distribution

1. CO_2-RM[1] stands for the CO_2 footprint created in the Ram Material production process.
2. Water- RM[2] stands for the Water Usage in the Ram Material production process.

3. CO_2-M[3] stands for the CO_2 footprint created in the manufacturing process (extrusion and moulding).
4. Water-M[4] stands for the Water Usage in the manufacturing process (extrusion and moulding).
5. CO_2-R[5] stands for the CO_2 footprint created in the recycling process.
6. Water-R[6] stands for the Water Usage in the recycling process.

As a combination of the weights of the criteria and sub-criteria, the final Weight Distribution Vector that will appear in the TOPSIS algorithm is shown in Table 12:

Cost	Density (kg/m^3)	Heat Deflection Tª (ºC)	Transparency	CO_2 Footprint (Kg)	Water Usage (l)	Time (days)
26%	11%	11%	9%	11%	11%	21%

Table 4-12 – Final Weight Distribution Vector

5 Results & Discussion

5.1 Introduction

In the following section, the methodology of section 3 will be applied to the case study of section 4 in closeness to get the results for our proposed Case Study.

The four criteria and their multiple sub-criteria for which the methodology will be applied are shown in Figure 3:

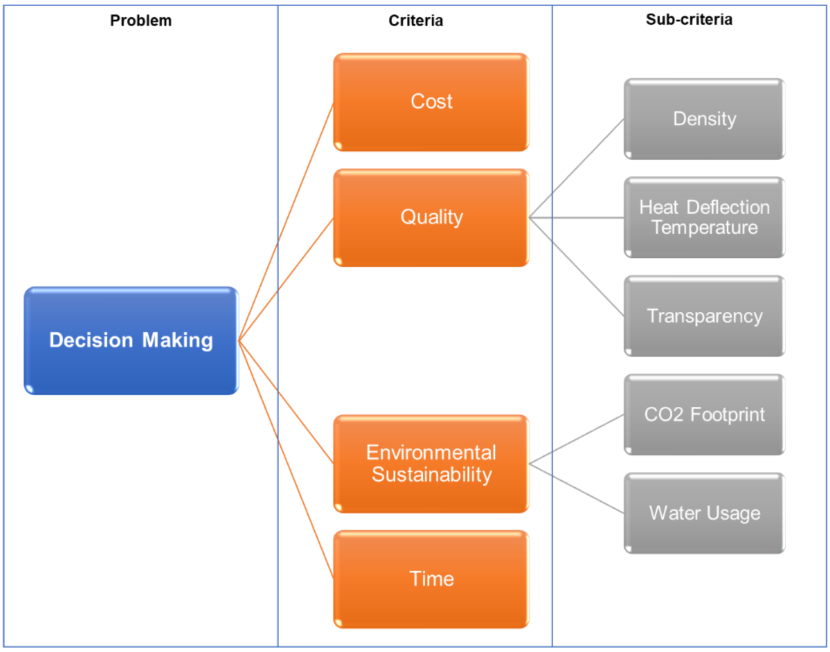


Figure 5-1 – Breakdown of the Case Study Decision Making Problem

5.2 Original weight distribution

Once the weights vector and the measure of the performance matrix were determined, adding them to the optimisation vector (the one that determines if a positive variation in the measurement performance of an indicator is positive or negative for the desired outcome), the TOPSIS algorithm is applied with the initial weights is applied. In Table 13 we can see the answers provided:

Resins	Relative closeness to ideal solution	Ranking
1	0.3626375	2
2	0.2637381	3

3	0.7121498	1

Table 5-1 – Ranking and Relative Closeness to Ideal Solution Values with the initial specific weights as input

It can be stated that, based on the results according to the initial conditions, Resin 3 is considered the best alternative with a big difference compared to the next competitor, Resin 1, that is closely followed by Resin 2.

After collecting the results, the next step will be executing the Sensitivity Analysis proposed in the Methodology section. An iteration of the specific weights of each criterion from 0% to 100% with 1% steps will be performed. The results obtained are commented in the following sections.

5.3 Cost criterion

After applying the algorithm to iterate the weight values of the cost criterion, the following graphs where obtained:

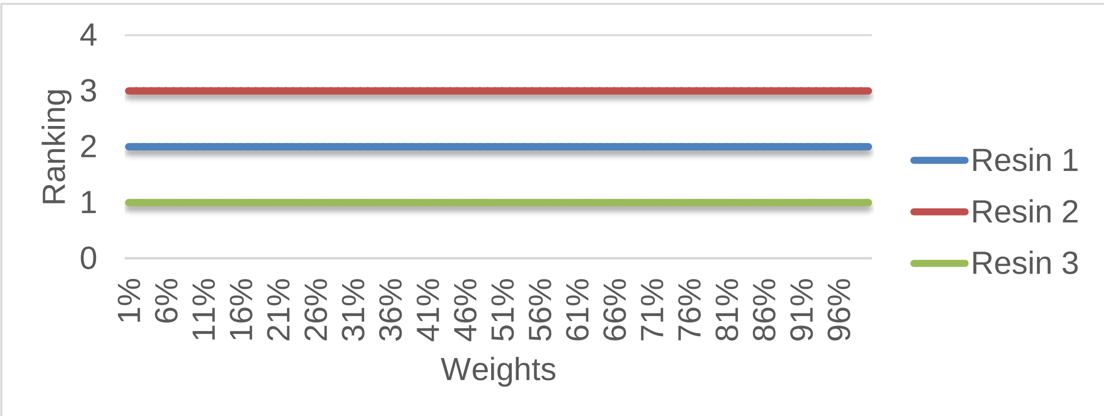


Figure 5-2 – Ranking behaviour during the iteration process of the Cost criterion

Figure 4 represents the ranking of the different alternatives along the different weight associated to the Cost criterion. As we can clearly see, Resin 3 is ranked as the first option, independently of the weight associated to the criterion. The same happens with Resin 1, ranked as the second best; and with Resin 2 ranked as the third option. Moreover, according to what Figure 3 is showing, the relative closeness to the ideal solution of Resin 3 grows until reaching 1 when the weight of the Cost criterion reaches 100%, while the value for the other two resins diminishes until Resin 2 reaches 0.

This means that, not only Resin 3 is considered the best option along the entire process, demonstrating the robustness of this criterion, but also that the decision gets more robust as its weight increases.

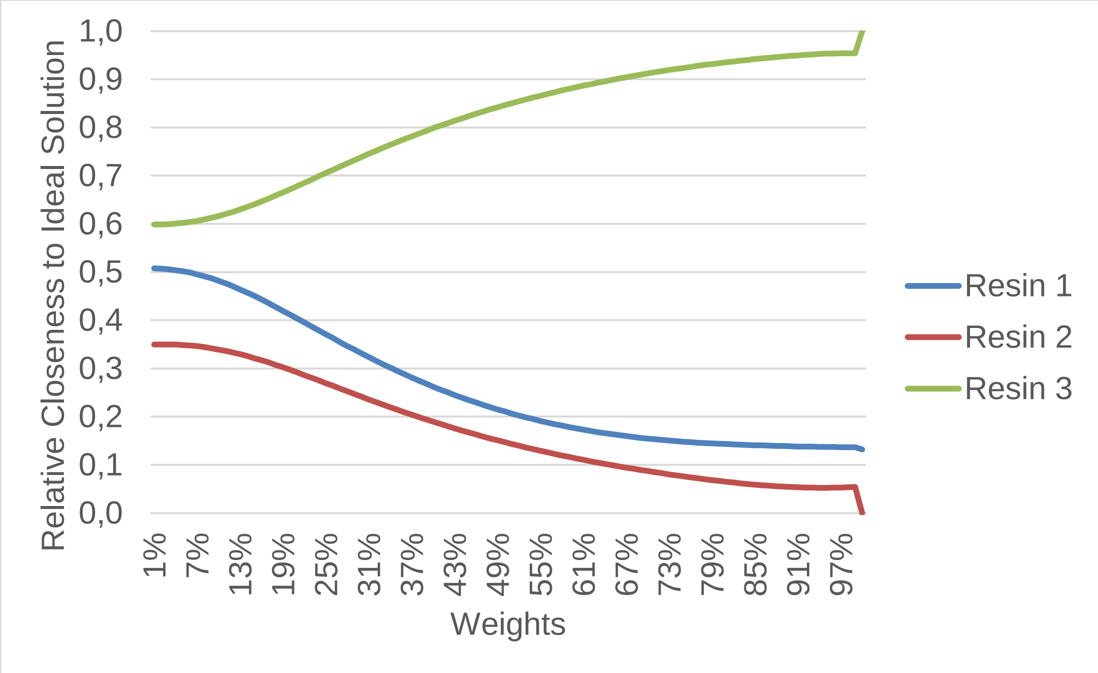


Figure 5-3 – Relative Closeness to the Ideal Solution Value behaviour during the iteration process of the Cost criterion

According to the information provided by the graphs, is easy to guess that the sensitivity of the Cost criterion is SC=0 as there are no rank reversals and there is no distance to the closest change in the ranking (the ranking never changes).

5.4 Quality criterion

As has been explained in the Case study section of the report, the Quality criterion is formed by three sub-criterion whose measurement performances are expressed in diverse units. In this section we will cover the robustness of each of them individually.

5.4.1 Density

The following graphs express the behaviour of the Density sub-criterion:

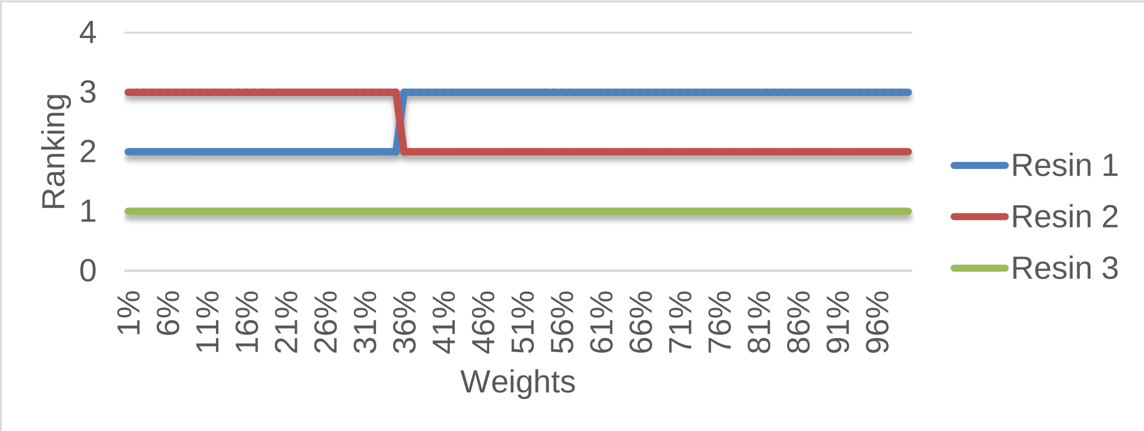


Figure 5-4 - Ranking behaviour during the iteration process of the Density sub-criterion

As is shown in Figure 6, Resin 3 maintains itself as the best alternative along the entire process, while Resin 2 and Resin 1 switch their rank when the specific weight of the criterion reaches 36%. At that moment Resin 1 goes from second position to third, and Resin 2 does the opposite movement.

Moreover, Figure 7 also demonstrates the different tendencies of resins 2 and 1 in the rank reversal in iteration. In addition, its closeness to the ideal solution grows almost during all the process, reinforcing the leadership of Resin 3 referring to the density criterion.

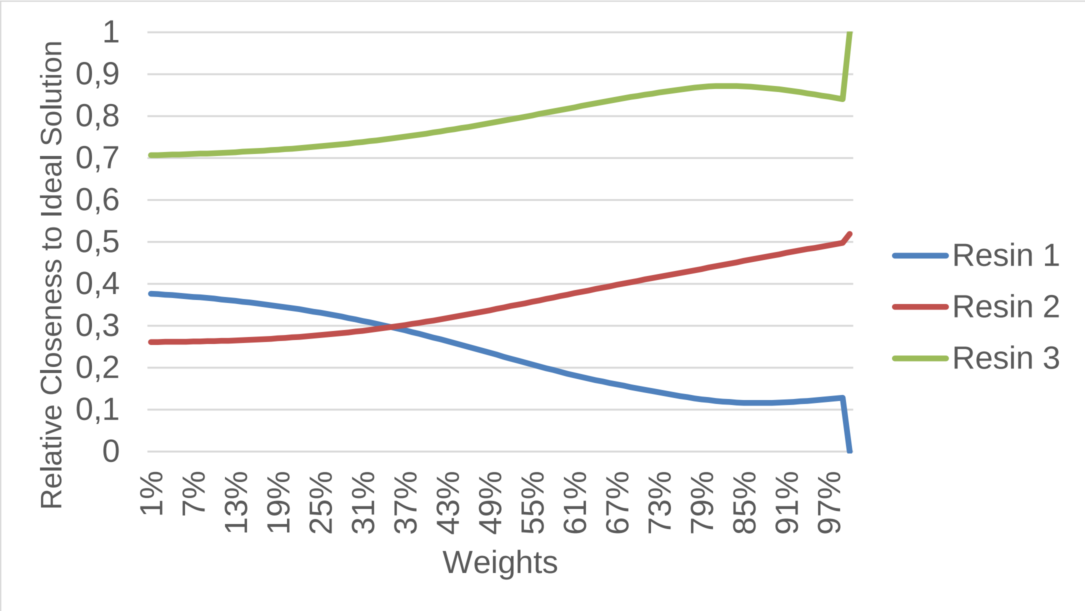


Figure 5-5 - Relative Closeness to the Ideal Solution Value behaviour during the iteration process of the Density sub-criterion

As opposed to the previous criterion, Density does have a change of rank reversal, so its measurement of Sensitivity Coefficient cannot be zero. Its Sensitivity measurement can be expressed as SC=0.08.

5.4.2 Heat Deflection Temperature (HDT)

The second sub-criterion inside the cost criterion is the Heat Deflection Temperature property of the resins.

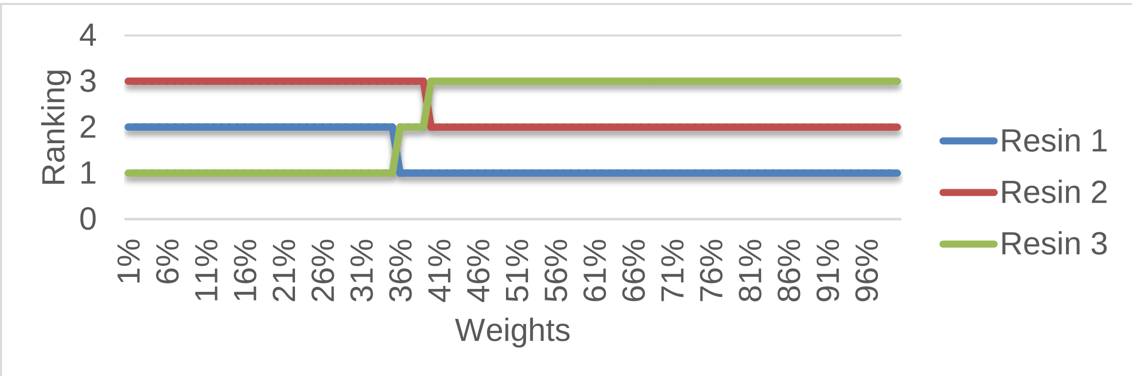


Figure 5-6 - Ranking behaviour during the iteration process of the Heat Deflection Temperature sub-criterion

It can clearly be observed that the Heat Deflection Temperature sub-criterion is the most critical out of the three previously studied. It has two rank reversals, with three alternative position changes, in iteration 36 and iteration 40, and each of the alternatives appears in at least 2 different positions of the rank, while Resin 3 is ranked in every position along the iteration process. The two alternatives that can be considered as the best options are Resin 3 (with specific HDT weights from 1% to 36%) and Resin 1 (with specific HDT weights from 36% to 100%).

While the 2nd position of the ranking is held by all the 3 alternatives: Resin 1 from iteration 1 until iteration 36, Resin 3 from iteration 36 until the 40th, and last but not least, the Resin 2 maintains itself ranked there until the end of the process.

Finally, the worst option starts being Resin 2, but as times passes and the 40th iteration arrives; Resin 3 takes the lead and stays as the worst option.

The graph shown in Figure 9 reinforces these behaviour descriptions. Despite Resin 3 starts with the highest relative closeness to ideal solution value, its tendency is clearly downward and in its way to the value 0 it crosses the other two alternatives, which present a growing tendency. It is remarkable to mention also that, according to the graph, Resin 2’s tendency seems to have a higher slope than Resin 1 and, if the experiment would have carried on with iterations over 100% in specific weight terms, Resin 2 would have probably became the best ranked solution.

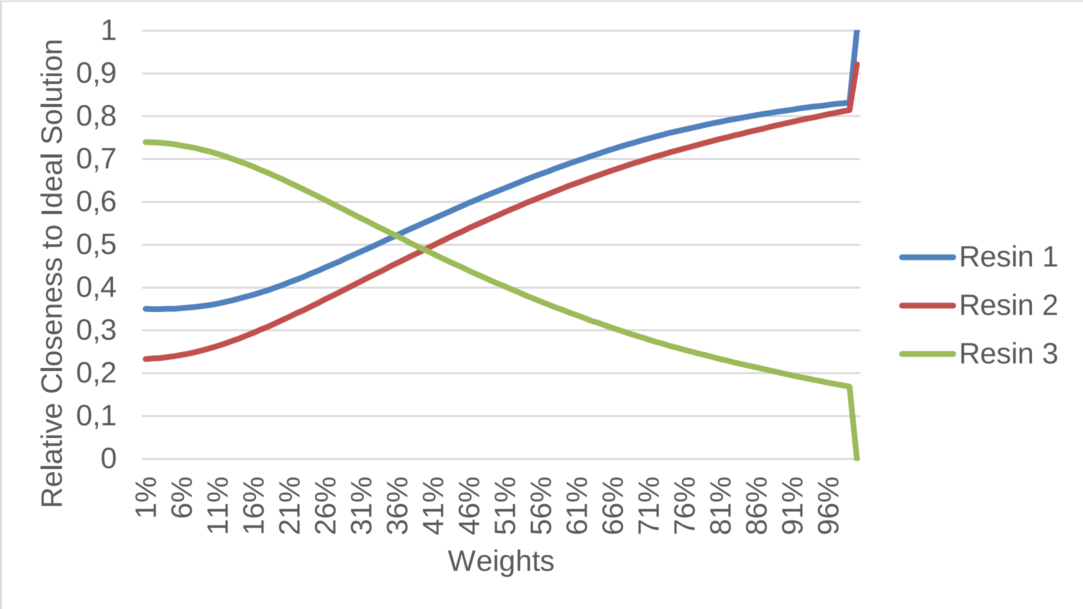


Figure 5-7 – Relative Closeness to the Ideal Solution Value behaviour during the iteration process of the Heat Deflection Temperature sub-criterion

To conclude, the Sensitivity Coefficient of this sub-criterion was calculated, attending to the amount of changes, and the specific weight difference of this changes from the initial value, giving a score of SC=0.11407.

5.4.3 Transparency

The last sub-criterion that forms the Quality main criterion has also been analysed. The following graphs define its behaviour.

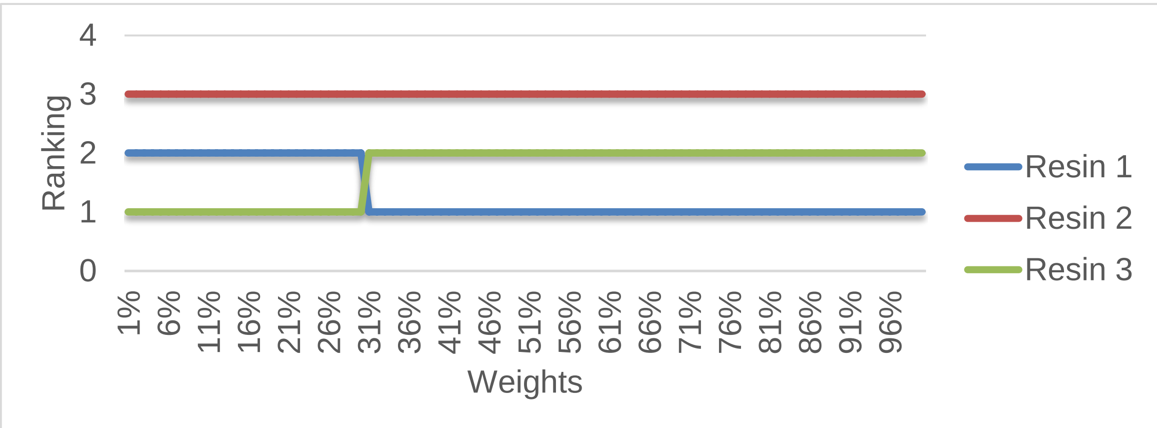


Figure 5-8 - Ranking behaviour during the iteration process of the Transparency sub-criterion

This sub-criterion's behaviour is similar to the one for density. The main difference between both of them is that the ranking position held always by the same alternative is the worst one, not the first one (as happened in the Density criterion analysis). Undoubtedly, Resin 2 ranks as the worst alternative along the totality of the process while Resins 1 and 3 share the first and second position. Resin 3 starts at the top of the ranking but falls into the second position after iteration number 31, where Resin 1 takes the lead and keeps it until the end of the process.

As Figure 11 shows below, Resin 1 has a positive tendency while Resin 3 has a negative slope, this leads to the change of ranking in the 31st iteration. Resin 2 starts as the less close to the Ideal Solution and its decreasing tendency prevents it from changing the position it is ranked along the process. However, at the starting point it is very close to Resin 1, and with a measurement performance slightly better, the initial rank for the first iteration could have been different.

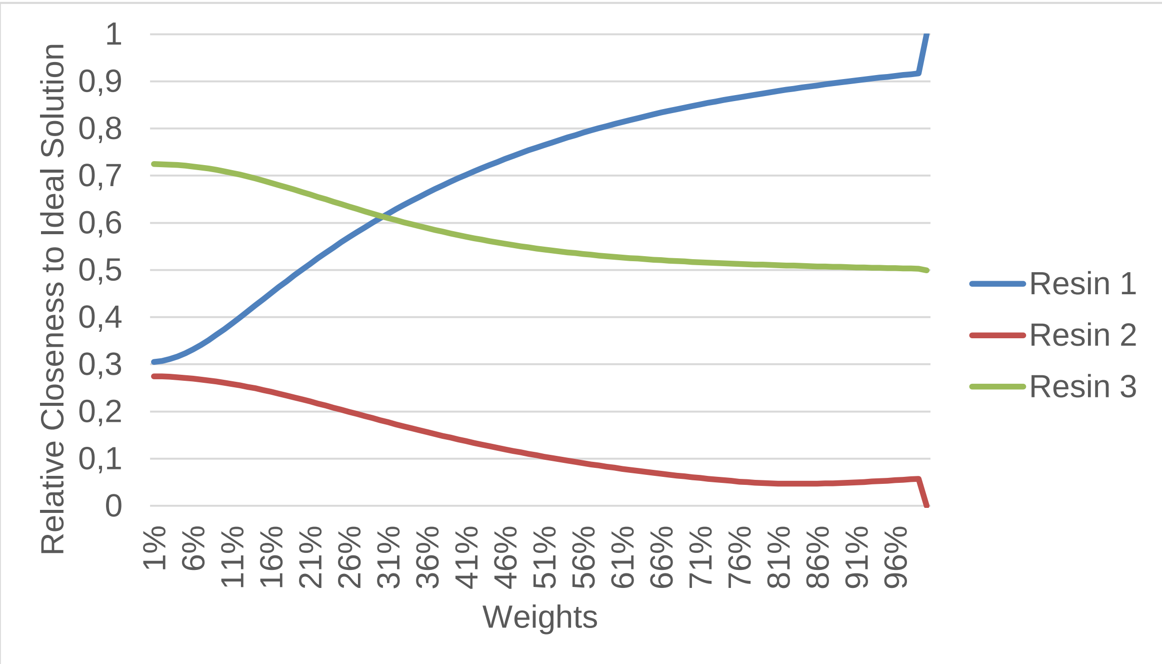


Figure 5-9 - Relative Closeness to the Ideal Solution Value behaviour during the iteration process of the Transparency sub-criterion

Finally, the Sensitivity Coefficient of the Transparency sub-criterion was calculated, giving a value of SC=0.09091. This value is higher than the Density's SC, despite having the same amount of rank reversals, because of the ranking change appearing before the Density's one. However, the value is lower than the HDT's one as there is only 1 change of rank.

5.4.4 Environmental Sustainability

As explained in the Case study section, the Environmental Sustainability criterion is also formed by more than one sub-criterion and their measurement performance are expressed in different units. In this section, we will cover the robustness of the two of them individually.

5.4.5 CO_2 Footprint

The ranking along the process and the Relative Closeness to the Ideal Solution of the CO2 Footprint sub-criterion are plotted in the following graphs.

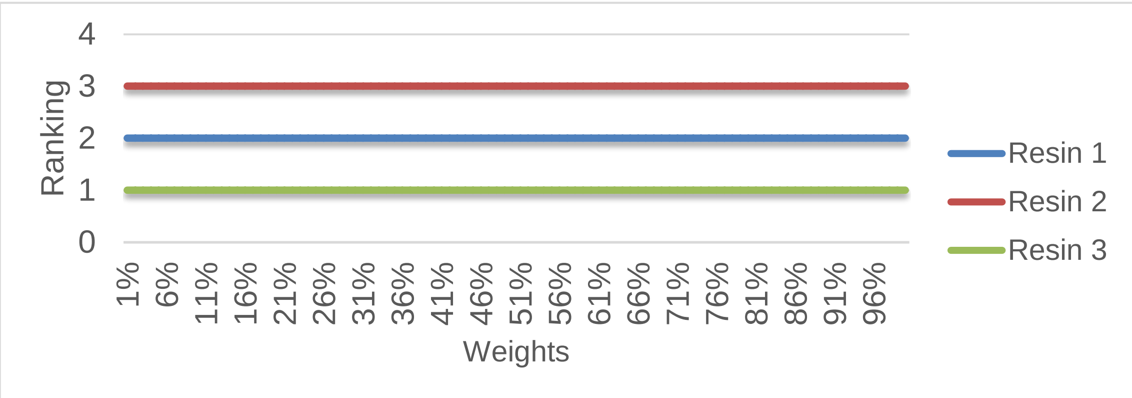


Figure 5-10 - Ranking behaviour during the iteration process of the CO2 Footprint sub-criterion

This specific sub-criterion shows no rank reversal during the process, the three different resins are ranked in the same order constantly. Resin 3 is classified as the best option, followed by Resin 1 and Resin 2 completes the classification.

Moreover, Figure 11 shows the negative slope of 2 while Resin 3 has a clearly positive gradient, which marks the difference between the alternatives, increasing as the iterations advance. Finally, Resin 1 shows an almost neutral slope (slightly negative), so the initial difference with the other two resins increases too as they have clearly negative or positive gradients.

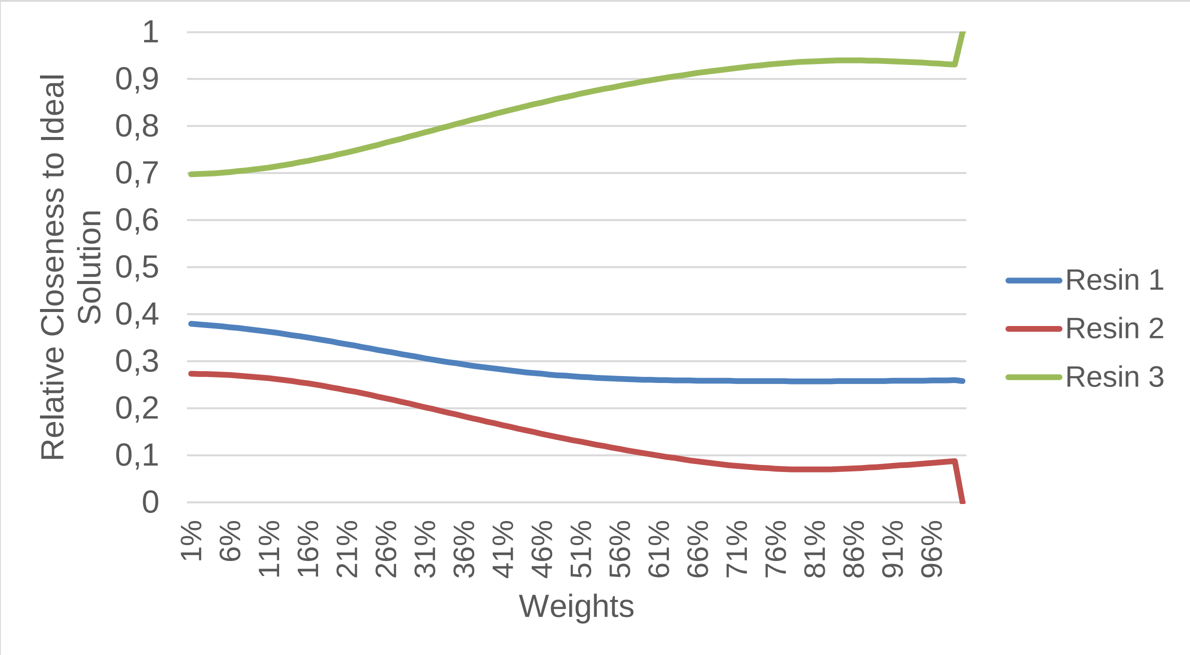


Figure 5-11 - Relative Closeness to the Ideal Solution Value behaviour during the iteration process of the CO2 Footprint sub-criterion

It is plain to see that the figure for the Sensitivity Coefficient of this sub-criterion will be- SC=0, as there is no rank reversal along the process.

5.4.6 Water Usage

If we analyse the results obtained from the second sub-criterion of the Environmental Sustainability criterion, similar conclusions will be found.

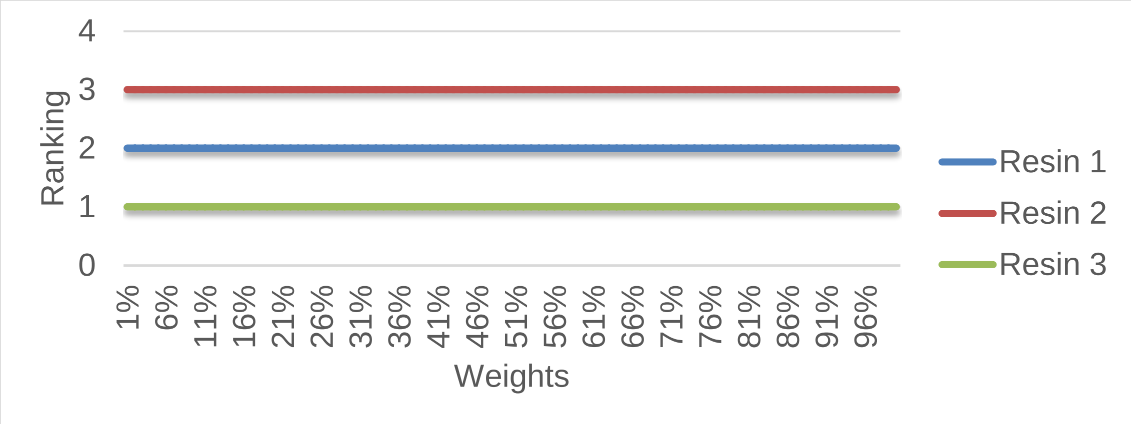


Figure 5-12 - Ranking behaviour during the iteration process of the Water Usage sub-criterion

The results shown in Figure 14 are exactly the same as the ones previously seen in Figure 10. Resin 3 leads the ranking along the entire process, Resin 1 ranks second and Resin 2 ranks third. However, the results do slightly differ from the previous sub-criterion if we take into account Figure 13 and 15. The main difference lies in the tendency of the second ranking criterion. In this case, Resins 2 and 1 have clear negative slopes and Resin 3 shows a mainly positive one.

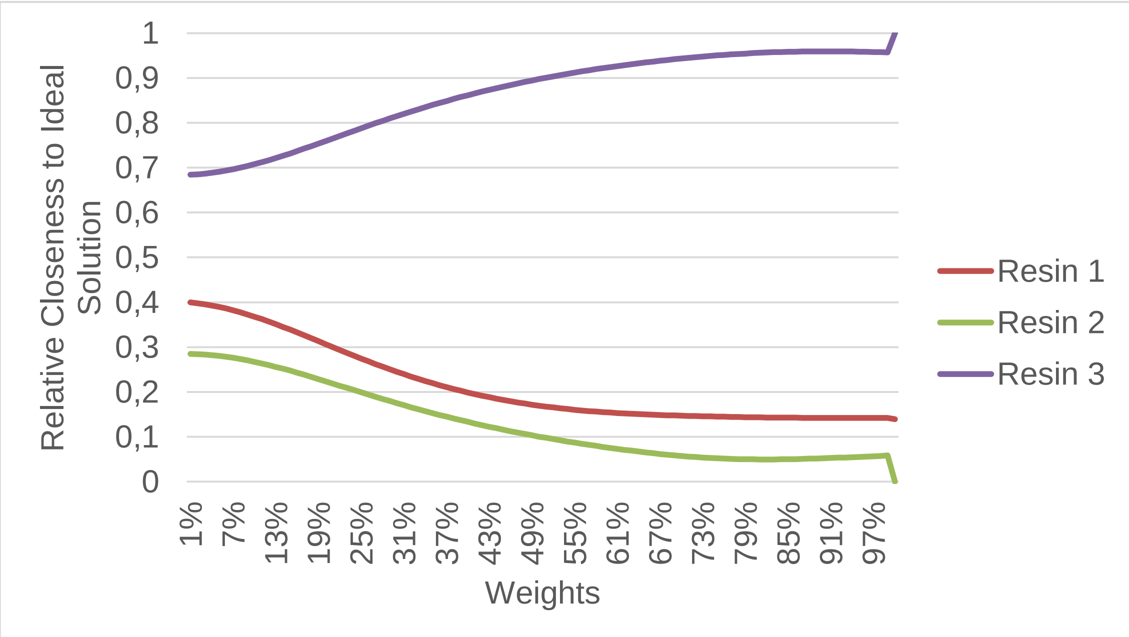


Figure 5-13 - Relative Closeness to the Ideal Solution Value behaviour during the iteration process of the Water Usage sub-criterion

It is obvious to state that the Sensitivity Coefficient for this sub-criterion will be also SC=0.

5.5 Time

The last criterion to analyse is the lead-time criterion. Exactly like happened with the Cost criterion, this criterion lacks sub-criterions, so the analysis is simpler than the previous ones. The following graphs show its behaviour.

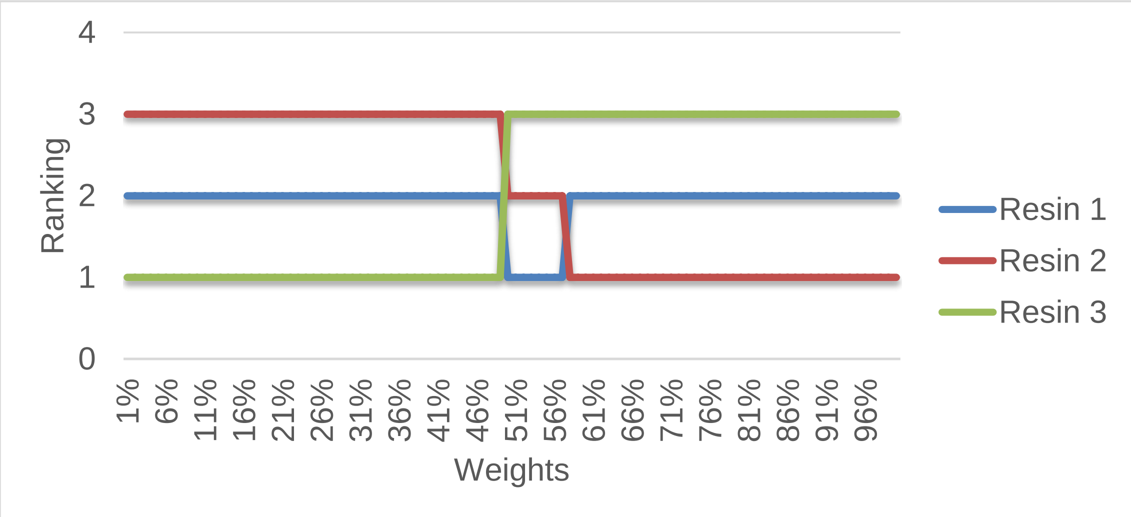


Figure 5-14 - Ranking behaviour during the iteration process of the Time criterion

As Figure 16 shows, the Time criterion presents some variances along the iteration process. Resin 3 is ranked as the best option from the first iteration until the fiftieth, where it changes its rank directly to the 3rd position. Resin 1 shows an uncommon behaviour, at first it ranked 2nd from the beginning of the process until the 50th iteration, where it ranked the first position, but just 8 iterations later, goes back to the second position and remains secondly ranked until the end of the process. Finally, Resin 2 starts as the worst ranked option; it changes into the 2nd best ranked option in the 50th iteration and in the 58th iteration swaps its position with Resin 1 becoming the new ranking leader.

This behaviour description of the alternatives is supported by Figure 17. Although the zone where the 3 criteria get closer is too narrow to appreciate it clearly, we can see how, right after the iteration where they cross, Resin 3 goes from the highest Relative Closeness to Ideal Solution value to the lowest, and

Resin 1 presents a higher value than Resin 2 only during some iterations from the crossing moment. Following, despite both have positive slopes; Resin 2 maintains the 1st position of the ranking while Resin 1 follows it closely. However, Resin 3 showed a negative tendency from the beginning of the iteration and ends the experiment with a Relative Closeness to Ideal Solution value of zero.

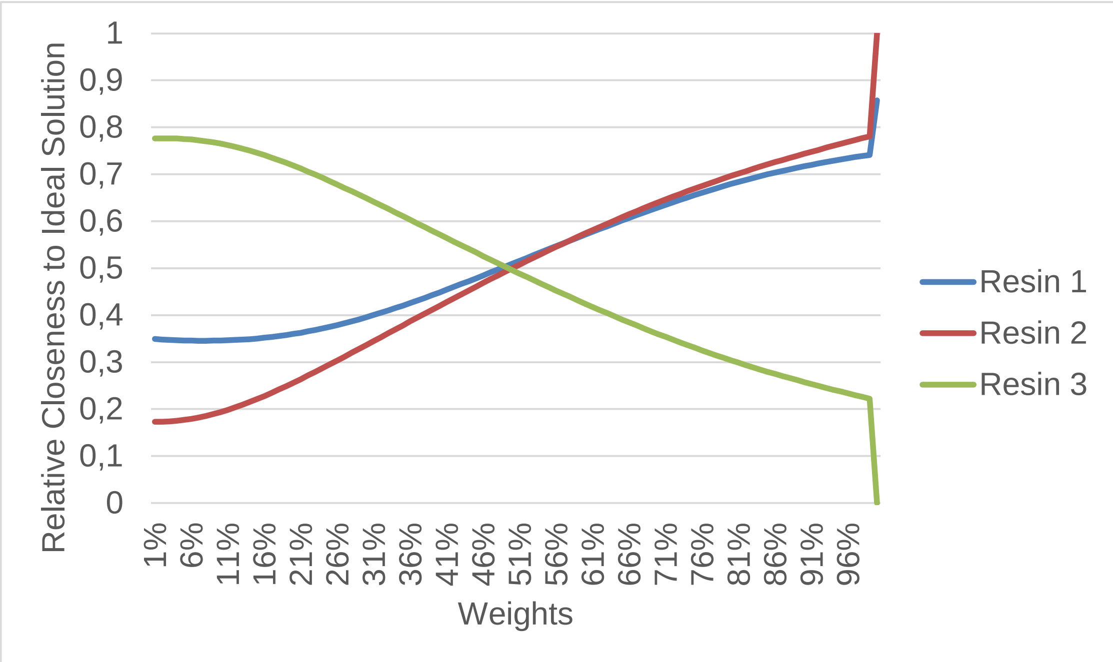


Figure 5-15 - Relative Closeness to the Ideal Solution Value behaviour during the iteration process of the Time criterion

To end this section, the value of the Sensitivity Coefficient of this criterion must be calculated. It is important to notice that, in spite of the fact that several changes take place along the process, these changes appear in points that are distant from the original specific weight value, which decreases the Sensitivity Coefficient significantly. The figure the Sensitivity formula gives us is SC=0.09508.

5.6 Comparison with VIKOR

With the objective of determining the sensitivity analysis validity, the same methodology has been applied but using VIKOR as MCDM technique instead of TOPSIS to see how the results diverge depending on the tools used.

5.6.1 Original Weight Distribution

Attending to the same original weight's distribution utilised for the TOPSIS algorithm, the following table has been completed:

Resins	Ranking
1	2
2	3
3	1

Table 5-2 - Ranking with the initial specific weights as input using VIKOR

As It can be seen, based on the results according to the initial conditions, the order of the alternatives is exactly the same than the obtained using the TOPSIS algorithm.

The following step will be applying the Methodology defined in Section 3 but using the VIKOR (Appendix A3) code instead of the TOPSIS code (Appendix A2).

5.6.2 Cost Criterion

The ranking of the cost criterion slightly diverges from the one previously obtained with the TOPSIS algorithm, as it can be seen in Figure 16:

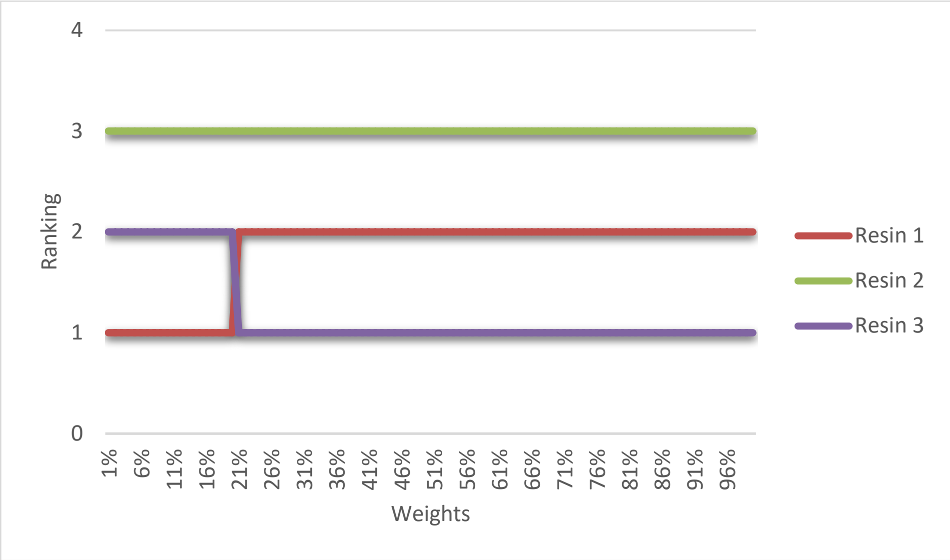


Figure 5-16 - Ranking behaviour during the iteration process of the Cost criterion

The iteration process starts with Resin 1 as the best ranked but when the specific weight of the Cost criterion reaches 21%, Resin 1 swaps its position in the ranking with Resin 3 and both maintain this new order during the entire process. Meanwhile, Resin 2 is ranked as the worst alternative during the totality of the process. Despite there is only one change in the Cost Sensitivity Analysis, we can appreciate a high figure for the Sensitivity Coefficient SC=0.4. This appears to be because the rank reversal happens in iteration 25^{th} which is really close to the original specific weight value of the Cost criterion (26%).

5.6.3 Density Sub-Criterion

The first sub-criterion of the Quality criterion represents a less stable behaviour when analysing it with VIKOR than with TOPSIS as the following figure shows:

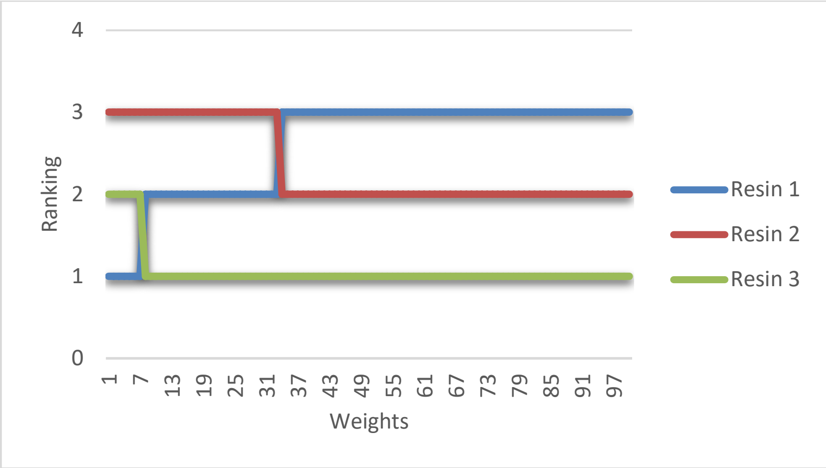


Figure 5-17 - Ranking behaviour during the iteration process of the Density sub-criterion

As it can be seen, the iteration starts with Resin 1 as the best ranking, but this only happens until the 8th iteration where Resin 3 takes the lead. Further in the process, in iteration 34, it swaps its position with Resin 2 and keeps ranked as the worst option until the end of the process; it has a clear negative tendency. On the other side, both Resin 2 and 3 are able to maintain their ranking after the swapping with Resin 1 and end the process in 2nd and 1st position, respectively.

This sub-criterion also shows a high value of Sensitivity Coefficient, SC=0.48718 due to the position of the first change (8th iteration), which appears to be very near to the initial specific weight value (11%).

5.6.4 Heat Deflection Temperature Sub-Criterion

Just like with the TOPSIS algorithm, this sub-criterion appears to be one of the most sensitive with VIKOR too. As the following figure shows, 2 ranking changes appear during the iteration process.

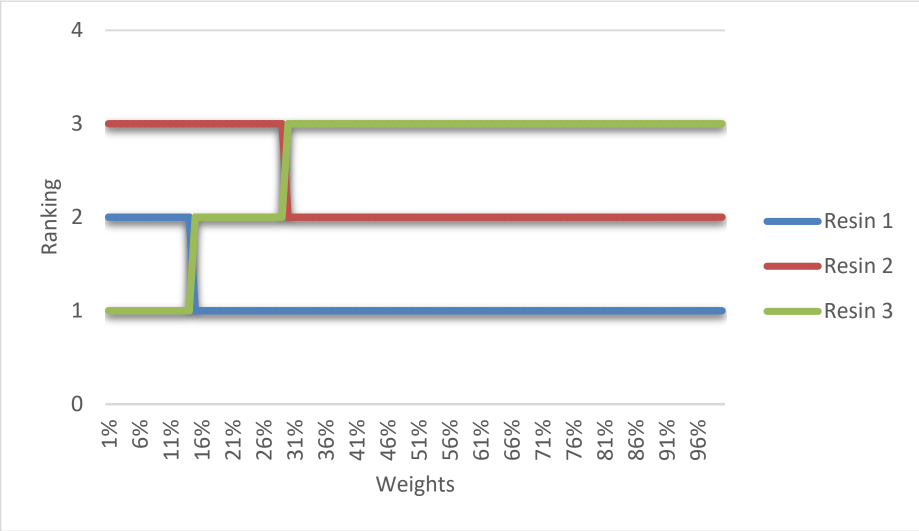


Figure 5-18 - Ranking behaviour during the iteration process of the HDT sub-criterion

These changes appear to happen in the 15th and 30th iteration, the first one is also near to the initial weight value of the sub-criterion (11%), that is the reason why the Sensitivity Coefficient gives a high figure as result, SC=0.42391.

5.6.5 Transparency Sub-Criterion

When it comes to talk about transparency, its behaviour with VIKOR seems to be almost the same than with TOPSIS, as the following figure shows:

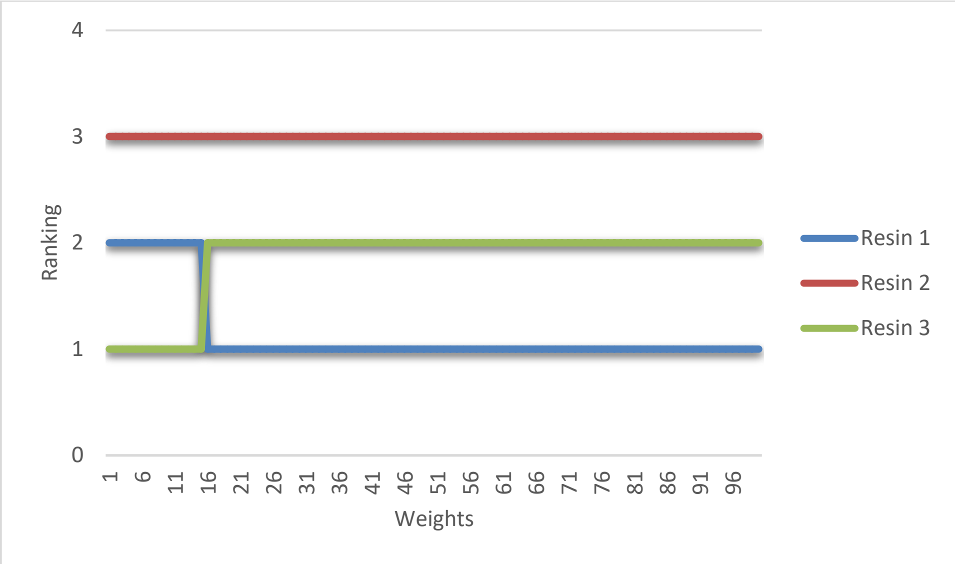


Figure 5-19 - Ranking behaviour during the iteration process of the Transparency sub-criterion

As the figure shows, Resin 2 is ranked as the 3rd alternative during the totality of the process. However, the ranking of Resin 1 and 3 varies, Resin 1 starts in the 2nd position but after iteration 16, swaps its position with Resin 3 and both maintain the 1st and 2nd position in the ranking respectively.

Despite this criterion is quite stable, it does have a significant value of Sensitivity Coefficient (SC=0.28571) because the rank reversal appears in the 16th iteration, just 7% away from the initial specific weight value.

5.6.6 CO_2 Footprint Sub-Criterion

The analysis of the CO2 footprint shows a sub-criterion that tends to be very stable as the iterations advance. Resin 1 starts as first ranked and Resin 3 as second, but in the 5th iteration they swap their position and keep them until the end of the experiment. Resin 2 is considered the worst alternative during the entire process.

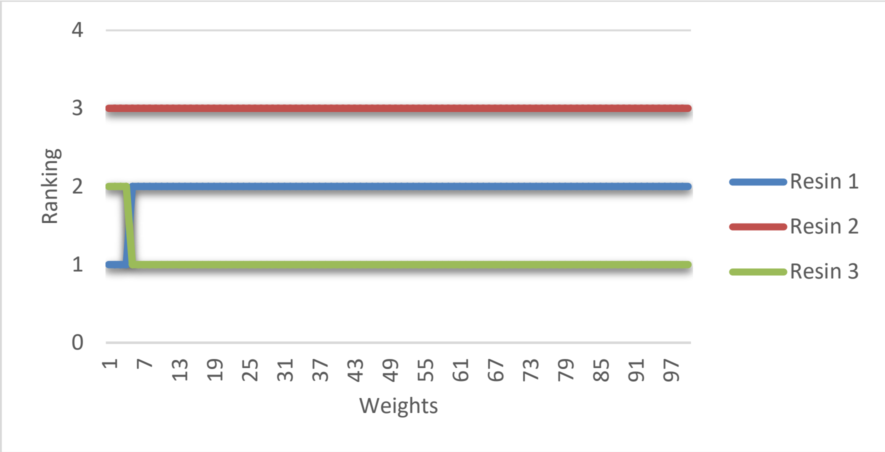


Figure 5-20 - Ranking behaviour during the iteration process of the CO2 Footprint sub-criterion

Despite this apparent stability, in the first steps of the experiment the rank reversal (fifth iteration), which tends to be very close to its initial weight value (11%), provokes the high figure of Sensitivity Coefficient the algorithm gives SC=0.33.

5.6.7 Water Usage Sub-Criterion

Its behaviour is similar to the previous described sub-criterion, as the following figure represents:

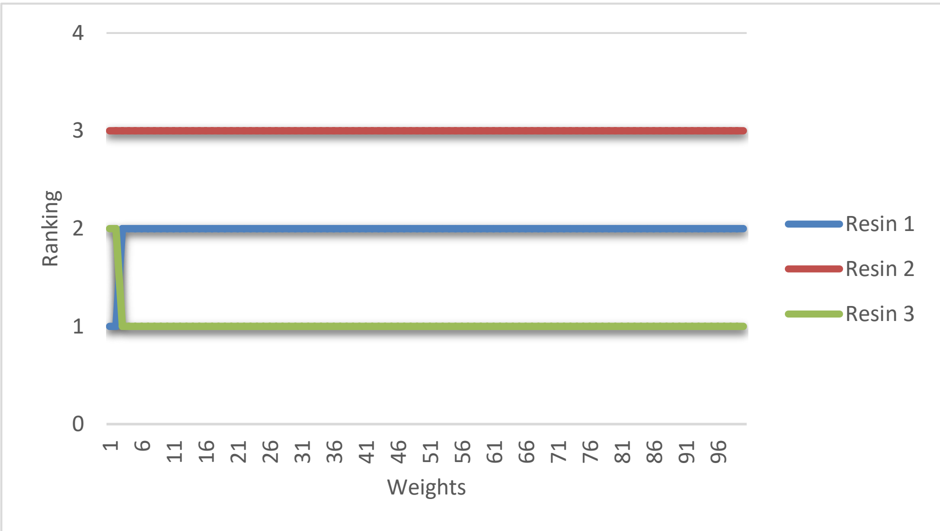


Figure 5-21 – Ranking behaviour during the iteration process of the Water Usage sub-criterion

In it, Resin 2 is ranked during the entire process as the worst one and Resin 1 and 3 swap their ranking positions in the third iteration, leaving Resin 3 as the best ranked until the end of the process and Resin 1 as the 2nd ranked.

The value for the Sensitivity Coefficient of this sub-criterion (SC=0.25) is still a significant figure, despite is slightly smaller than the previous one due to the appearance of the rank reversal in a slightly further iteration to the initial weight than in the previous sub-criterion (3rd iteration).

5.6.8 Time Criterion

Finally, the Time criterion appears to be one of the most sensitive, just like in the TOPSIS SA, due to the 3 rank reversals that the following graph shows:

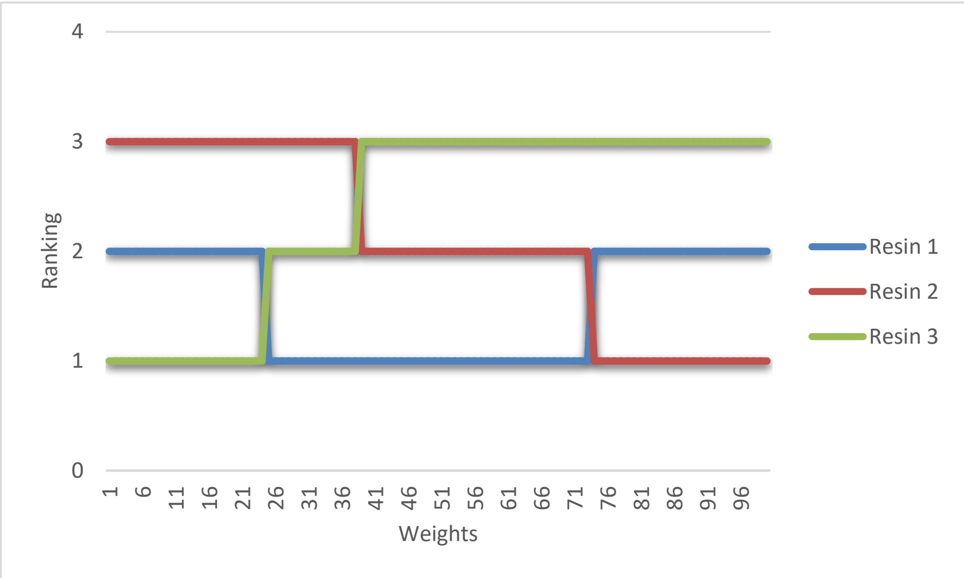


Figure 5-22 - Ranking behaviour during the iteration process of the Time criterion

As it can be clearly seen, Resin 2 starts the iteration process in the last position of the ranking, but advances to the second position in iteration number 39 to finally end as the first ranked after the 74th iteration. On the other hand, Resin 3 makes the opposite path: it starts ranked as the 1st option, but switches to the 2nd position in the rankling after 25 iterations and ends last in the ranking from the 39th iteration until the end of the process. Finally, Resin 1 starts as the second alternative, but during 49 iterations is ranked as the best one, until the 74th iteration, where it swaps with Resin 2, going back to the 2nd position of the ranking. Despite the high amount of rank reversals, the Time criterion is not the one with the highest figure of Sensitivity Coefficient (SC=0.37) due to the remoteness of the rank reversal compared to the initial specific weight of the criterion.

5.7 Summary

Table 15 shows the different calculated values and interest points of the behaviour of all the criteria and sub-criteria after the performance of the totality of the analysis applying TOPSIS method.

Criterion	Sub-Criterion	Number of Rank Reversals	Iteration of the Rank changes	Sensitivity Coefficient	Sensitivity Coefficient (SC)
Cost	Cost	0	0	0	0
Quality	Density	1	36	0.08	0.09499
	Heat Deflection Tª	2	36 40	0.11407	
	Transparency	1	31	0.09091	
Environmental Sustainability	CO_2 Footprint	0	0	0	0
	Water Usage	0	0	0	
Time	Time	2	50 58	0.09508	0. 09508

Table 5-3 – Sensitivity Coefficient, number and position of rank changes per criterion and sub-criterion using TOPSIS technique

While Table 16 shows the same values but for the VIKOR method:

Criterion	Sub-	Number of	Iteration	Sensitivity	Average

	Criterion	Rank Reversals	of the Rank changes	Coefficient (SC)	Sensitivity Coefficient
Cost	Cost	1	21	0.4	0.4
Quality	Density	2	8 34	0.48718	0.39893
	Heat Deflection T[a]	2	15 30	0.42391	
	Transparency	1	16	0.29571	
Environmental Sustainability	CO_2 Footprint	1	5	0.333	0.29165
	Water Usage	1	3	0.25	
Time	Time	3	25 39 74	0.37	0. 37

Table 5-4 - Sensitivity Coefficient, number and position of rank changes per criterion and sub-criterion using VIKOR technique

Comparing the 2 methods, we can state that TOPSIS provides more robust results than VIKOR, as after all the analysis performed. Two of the criteria have a Sensitivity Coefficient SC=0, and despite two other Criteria show some changes in the ranking along the process, the value of their SC is not very high and the first rank reversal appears when Transparency sub-criterion is worth a 31%, when its initial value (mentioned in Table 13, section 4) is 9% (22% of specific weight difference). The other changes appear in higher iterations, so they compromise even less the stability of the decision-making process. However, VIKOR not only provokes more rank reversals (one more per each criterion or sub-criterion but for HDT which maintains the 2 changes), but these changes tend to appear generally in earlier moments of the iteration process, this tendency is the one that results in such a difference between the Sensitivity Coefficients calculated with the two different methods. However, the Sensitivity Analysis technique positively analyses the robustness of the different decisions and gives representative values of the Sensitivity Coefficient, so it is easy to determine just by reading the coefficient, which is more sensitive or robust, characteristic which is graphically demonstrated with the figures.

5.8 Future Work

After the results of the two MCDM techniques were analysed and evaluated as positive, two targets remain to be met. Firstly, despite the methodology has been proved to be successful with TOPSIS and VIKOR, it stills need to be assessed with many other MCDM techniques that involve weight selection process (such as ELECTREE, PROMETHEE, WSM, WPM…) and their results must be analysed. Finally, a new definition of a Global Sensitivity Coefficient able to merge the various criteria Sensitivity Coefficient needs to be invented and evaluate its validity.

6 Conclusion

Multi-Criteria Decision Making techniques are powerful tools decision makers use when facing difficult decisions with various alternatives and criteria to evaluate. Sensitivity Analysis is an accurate way of assessing the reliability of these decisions in order to reinforce the conviction of having taken the right decision.

Due to the lack of research in these practises, a deep research about the current and past Sensitivity Analysis approaches was done in the Literature Review. Once a knowledge basis was established, an innovative methodology for performing this technique, together with a new mathematical definition of the Sensitivity Coefficient were created and exposed. Moreover, a case study was analysed, and the methodology was applied to it, giving as a result a detailed analysis of the robustness of the initial decision and an accurate measurement of the sensitivity of each criterion.

As a conclusion, the aim of the Individual Research Project has been fulfilled and the objectives have been met, considering the new Sensitivity definition a success based on the results obtained and all the references found in the Literature Review.

REFERENCES

[1] Rao, R. V. (no date a) *Decision Making in the Manufacturing Environment*.

[2] Maliene, V., Dixon-gough, R. and Malys, N. (2018) 'Dispersion of relative importance values contributes to the ranking uncertainty: Sensitivity analysis of Multiple Criteria', *Applied Soft Computing Journal*. Elsevier B.V. doi: 10.1016/j.asoc.2018.03.003.

[3] E. Pagone, K. S. & M. J. (2014) 'Multi-criteria decision-making for the life cycle of sustainable high pressure die casting products', (Mcdm).

[4] Ray, S., Mukherjee, J. and Mandal, S. (2015) *Modelling nitrogen and carbon cycles in Hooghly estuary along with adjacent mangrove ecosystem*. 1st edn, *Advanced Modelling Techniques for Studying Global Changes in Environmental Sciences*. 1st edn. Elsevier B.V. doi: 10.1016/B978-0-444-63536-5.00013-2.

[5] Triantaphyllou, E. and SsInchez, A. (1997) 'A Sensitivity Analysis Approach for Some Deterministic Multi-Criteria Decision-Making Methods *', (I), pp. 151–194.

[6] Genc, T. (2014) 'Sensitivity analysis on PROMETHEE and TOPSIS weights', 13(4).

[7] Simulation, S. (2014) 'Sensitivity Analysis to Attribute Values Based on Grey Correlation in Multiple Attribute Decision Making', 851, pp. 927–931. doi: 10.4028/www.scientific.net/AMR.850-851.927.

[8] Rosa, A., Borges, P. and Antunes, C. H. (2002) 'A visual interactive tolerance approach to sensitivity analysis in MOLP', 142, pp. 357–381.

[9] Zhao, N., Wei, C. and Xu, Z. (2013) 'Sensitivity Analysis of Multiple Criteria Decision Making Method Based on the OWA Operator', 28, pp. 1124–1139. doi: 10.1002/int.

[10] Jirí, M. (2019) 'The Robustness of TOPSIS Results Using Sensitivity Analysis Based on Weight Tuning', pp. 83–86. doi: 10.1007/978-981-10-9038-7.

[11] Gülsah Yilan, M.A. Neset Kadirgan, G. A. Ç. (2020) ‘Analysis of electricity generation options for sustainable energy decision making : The case of Turkey’, 146. doi: 10.1016/j.renene.2019.06.164.

[12] Dinçer, H., Yüksel, S. and Martínez, L. (2019) ‘Interval Type 2-based Hybrid Fuzzy Evaluation of Financial Services in E7 Economies with DEMATEL-ANP and MOORA Methods’, *Applied Soft Computing Journal*. Elsevier B.V. doi: 10.1016/j.asoc.2019.03.018.

[13] Favretto, N., Stringer, L. C., Dougill, A. J., Dallimer, M., Perkins, J. S., Reed, M. S., Atlhopheng, J. R. and Mulale, K. (2016) ‘Multi-Criteria Decision Analysis to identify dryland ecosystem service trade-offs under different rangeland land uses’, 17, pp. 142–151. doi: 10.1016/j.ecoser.2015.12.005.

[14] Martin, D. M. and Mazzotta, M. (2018) ‘Non-monetary valuation using Multi-Criteria Decision Analysis : Sensitivity of additive aggregation methods to scaling and compensation assumptions’, *Ecosystem Services*, 29, pp. 13–22. doi: 10.1016/j.ecoser.2017.10.022.

[15] Erlacher, C., Şalap, S., Jankowski, P. and Anders, K. (2016) ‘A GPU-based Solution for Accelerating Spatially-Explicit Uncertainty- and Sensitivity Analysis in Multi-Criteria Decision Making’, (2014), pp. 305–312.

[16] Kokaraki, N., Hopfe, C. J., Robinson, E. and Nikolaidou, E. (2019) ‘Testing the reliability of deterministic multi-criteria decision-making methods using building performance simulation’, *Renewable and Sustainable Energy Reviews*. Elsevier Ltd, 112(October 2018), pp. 991–1007. doi: 10.1016/j.rser.2019.06.018.

[17] Nadizadeh, S., Karimi, M., Nematollahi, O., Karimi, H. and Jelokhani-niaraki, M. (2019) ‘A risk-based multi-criteria spatial decision analysis for solar power plant site selection in different climates : A case study in Iran’, *Renewable Energy*. Elsevier Ltd, 143, pp. 958–973. doi: 10.1016/j.renene.2019.05.063.

[18] Yi, C., Asce, S. M., Lu, M. and Asce, M. (2019) ‘Mixed-Integer Linear

Programming – Based Sensitivity Analysis in Optimization of Temporary Haul Road Layout Design for Earthmoving Operations', 33(3), pp. 1–14. doi: 10.1061/(ASCE)CP.1943-5487.0000838.

[19] S, C. M. R. and Hernandez, E. (2015) 'Robustness and sensitivity analysis in multiple criteria decision problems using rule learner techniques', *Reliability Engineering and System Safety*. Elsevier, 134, pp. 297–304. doi: 10.1016/j.ress.2014.04.022.

[20] Chai, N., Wu, B., Yang, W. and Shi, B. (2019) 'A Multicriteria Approach for Modeling Small Enterprise Credit Rating : Evidence from China', pp. 2523–2543.

[21] Chen, H. and Kocaoglu, D. F. (2008) 'A sensitivity analysis algorithm for hierarchical decision models', 185, pp. 266–288. doi: 10.1016/j.ejor.2006.12.029.

[22] Song, J. Y. and Chung, E. (2016) 'Robustness , Uncertainty and Sensitivity Analyses of the TOPSIS Method for Quantitative Climate Change Vulnerability : a Case Study of Flood Damage', *Water Resources Management*. Water Resources Management, pp. 4751–4771. doi: 10.1007/s11269-016-1451-2.

[23] Randjelovic, A. (2017) 'Multi-criteria decision making : An example of sensitivity analysis MULTI-CRITERIA DECISION MAKING : AN EXAMPLE OF SENSITIVITY ANALYSIS', (January). doi: 10.5937/sjm12-9464.

[24] 'CES EduPack 2018 - [Material Universe]' (no date x).

[25] *Electric Energy Cost UK* (no date y). Available at: https://www.ukpower.co.uk/home_energy/tariffs-per-unit-kwh.

[26] *Canbon Intensity of Electricity* (no date z). Available at: https://www.icax.co.uk/Grid_Carbon_Factors.html.

[27] Martin, B. A. C. (2016) 'Package " MCDM "', pp. 1–9. Available at: https://cran.r-project.org/web/packages/MCDM/index.html%0A.

APPENDICES

Appendix A - R Simulation Code

A.1 Introduction

In order to calculate the results shown in section 5, the software called R has been used. Using the TOPSIS code already created in the [27], an iteration algorithm that reproduced the explained technique in the Methodology section has been developed so that not only the ranking order along the iteration but also the Relative Closeness to the ideal solution values were saved in vectors, and afterwards plotted using Excel to create the graphs shown in the Results & Discussion section.

A.2 Code using TOPSIS algorithm

```
library(MCDM)

MP <-
matrix(c(0.1438,0.19145,0.0545,1200,1023,900,137,132.5,80,4.11,0.88667,2.4
9667,0.06177,0.08684,0.03375,1.93925,3.8966,0.47323,63,60,90),nrow=3,ncol
=7)

Ncrit<-7

newweight<-rep(0,7)

w1<-0.26

w2<-0.11

w3<-0.11

w4<-0.09

w5<-0.11

w6<-0.11

w7<-0.21
```

```
MinDistC1<-0

MinDistC2<-0

MinDistC3<-0

MinDistC4<-0

MinDistC5<-0

MinDistC6<-0

MinDistC7<-0

AvgDistC1<-0

AvgDistC2<-0

AvgDistC3<-0

AvgDistC4<-0

AvgDistC5<-0

AvgDistC6<-0

AvgDistC7<-0

AcumC1<-0

AcumC2<-0

AcumC3<-0

AcumC4<-0

AcumC5<-0

AcumC6<-0

AcumC7<-0

sensitivityC1<-0

sensitivityC2<-0
```

```
sensitivityC3<-0

sensitivityC4<-0

sensitivityC5<-0

sensitivityC6<-0

sensitivityC7<-0

C1order1<-rep(0,100)

C1order2<-rep(0,100)

C1order3<-rep(0,100)

C2order1<-rep(0,100)

C2order2<-rep(0,100)

C2order3<-rep(0,100)

C3order1<-rep(0,100)

C3order2<-rep(0,100)

C3order3<-rep(0,100)

C4order1<-rep(0,100)

C4order2<-rep(0,100)

C4order3<-rep(0,100)

C5order1<-rep(0,100)

C5order2<-rep(0,100)

C5order3<-rep(0,100)

C6order1<-rep(0,100)

C6order2<-rep(0,100)

C6order3<-rep(0,100)
```

```
C7order1<-rep(0,100)

C7order2<-rep(0,100)

C7order3<-rep(0,100)


C1closeness1<-rep(0,100)

C1closeness2<-rep(0,100)

C1closeness3<-rep(0,100)

C2closeness1<-rep(0,100)

C2closeness2<-rep(0,100)

C2closeness3<-rep(0,100)

C3closeness1<-rep(0,100)

C3closeness2<-rep(0,100)

C3closeness3<-rep(0,100)

C4closeness1<-rep(0,100)

C4closeness2<-rep(0,100)

C4closeness3<-rep(0,100)

C5closeness1<-rep(0,100)

C5closeness2<-rep(0,100)

C5closeness3<-rep(0,100)

C6closeness1<-rep(0,100)

C6closeness2<-rep(0,100)

C6closeness3<-rep(0,100)

C7closeness1<-rep(0,100)
```

```
C7closeness2<-rep(0,100)

C7closeness3<-rep(0,100)



optimise<-c('min','min','max','max','min','min','min')

changesC1<-0

changesC2<-0

changesC3<-0

changesC4<-0

changesC5<-0

changesC6<-0

changesC7<-0

changemomentC1<-rep(1000,100)

changemomentC2<-rep(1000,100)

changemomentC3<-rep(1000,100)

changemomentC4<-rep(1000,100)

changemomentC5<-rep(1000,100)

changemomentC6<-rep(1000,100)

changemomentC7<-rep(1000,100)

counterC1<-1

counterC2<-1

counterC3<-1

counterC4<-1
```

```
counterC5<-1

counterC6<-1

counterC7<-1

plotvectorC1<-c()

plotvectorC2<-c()

plotvectorC3<-c()

plotvectorC4<-c()

plotvectorC5<-c()

plotvectorC6<-c()

plotvectorC7<-c()

for (i in 1:100){

weight<-c(round(i/100,digits=20),round((w2+((w1-i/100)/(Ncrit-1))),digits=20),round((w3+((w1-i/100)/(Ncrit-1))),digits=20),round((w4+((w1-i/100)/(Ncrit-1))),digits=20),round((w5+((w1-i/100)/(Ncrit-1))),digits=20),round((w6+((w1-i/100)/(Ncrit-1))),digits=20),round((w7+((w1-i/100)/(Ncrit-1))),digits=20))

#print(weight)


normalise<-sum(c(round(i/100,digits=20),round((w2+((w1-i/100)/(Ncrit-1))),digits=20),round((w3+((w1-i/100)/(Ncrit-1))),digits=20),round((w4+((w1-i/100)/(Ncrit-1))),digits=20),round((w5+((w1-i/100)/(Ncrit-1))),digits=20),round((w6+((w1-i/100)/(Ncrit-1))),digits=20),round((w7+((w1-i/100)/(Ncrit-1))),digits=20)))

#print (normalise)
```

```
weight<-c(round(i/100,digits=20)/normalise,round((w2+((w1-i/100)/(Ncrit-1))),digits=20)/normalise,round((w3+((w1-i/100)/(Ncrit-1))),digits=20)/normalise,round((w4+((w1-i/100)/(Ncrit-1))),digits=20)/normalise,round((w5+((w1-i/100)/(Ncrit-1))),digits=20)/normalise,round((w6+((w1-i/100)/(Ncrit-1))),digits=20)/normalise,round((w7+((w1-i/100)/(Ncrit-1))),digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)


if (i==100){

weight<-c(1,0,0,0,0,0,0)

}


TOPSISLinear(MP,weight,optimise)

C1order1[i]<-TOPSISLinear(MP,weight,optimise)[1,3]

C1order2[i]<-TOPSISLinear(MP,weight,optimise)[2,3]

C1order3[i]<-TOPSISLinear(MP,weight,optimise)[3,3]


C1closeness1[i]<-TOPSISLinear(MP,weight,optimise)[1,2]

C1closeness2[i]<-TOPSISLinear(MP,weight,optimise)[2,2]

C1closeness3[i]<-TOPSISLinear(MP,weight,optimise)[3,2]


if (i>1){
```

```
if (C1order1[i]!=C1order1[i-1]||C1order2[i]!=C1order2[i-1]||C1order3[i]!=C1order3[i-1]){

changesC1=changesC1+1

changemomentC1[i]=i

}}}


for (i in 1:100){

weight<-c(round((w1+((w2-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w3+((w2-i/100)/(Ncrit-1))),digits=20),round((w4+((w2-i/100)/(Ncrit-1))),digits=20),round((w5+((w2-i/100)/(Ncrit-1))),digits=20),round((w6+((w2-i/100)/(Ncrit-1))),digits=20),round((w7+((w2-i/100)/(Ncrit-1))),digits=20))

#print(weight)


normalise<-sum(c(round((w1+((w2-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w3+((w2-i/100)/(Ncrit-1))),digits=20),round((w4+((w2-i/100)/(Ncrit-1))),digits=20),round((w5+((w2-i/100)/(Ncrit-1))),digits=20),round((w6+((w2-i/100)/(Ncrit-1))),digits=20),round((w7+((w2-i/100)/(Ncrit-1))),digits=20)))

#print (normalise)


weight<-c(round((w1+((w2-i/100)/(Ncrit-1))),digits=20)/normalise,round(i/100,digits=20)/normalise,round((w3+((w2-i/100)/(Ncrit-1))),digits=20)/normalise,round((w4+((w2-i/100)/(Ncrit-1))),digits=20)/normalise,round((w5+((w2-i/100)/(Ncrit-1))),digits=20)/normalise,round((w6+((w2-i/100)/(Ncrit-1))),digits=20)/normalise,round((w7+((w2-i/100)/(Ncrit-1))),digits=20)/normalise)
```

```
#print (weight)

check<-sum(weight)

#print (check)



if (i==100){

weight<-c(0,1,0,0,0,0,0)

}


TOPSISLinear(MP,weight,optimise)

C2order1[i]<-TOPSISLinear(MP,weight,optimise)[1,3]

C2order2[i]<-TOPSISLinear(MP,weight,optimise)[2,3]

C2order3[i]<-TOPSISLinear(MP,weight,optimise)[3,3]


C2closeness1[i]<-TOPSISLinear(MP,weight,optimise)[1,2]

C2closeness2[i]<-TOPSISLinear(MP,weight,optimise)[2,2]

C2closeness3[i]<-TOPSISLinear(MP,weight,optimise)[3,2]



if (i>1){

if (C2order1[i]!=C2order1[i-1]||C2order2[i]!=C2order2[i-1]||C2order3[i]!=C2order3[i-1]){

changesC2=changesC2+1
```

```
changemomentC2[i]=i

}}}


for (i in 1:100){

weight<-c(round((w1+((w3-i/100)/(Ncrit-1))),digits=20),round((w2+((w3-
i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w4+((w3-i/100)/(Ncrit-
1))),digits=20),round((w5+((w3-i/100)/(Ncrit-1))),digits=20),round((w6+((w3-
i/100)/(Ncrit-1))),digits=20),round((w7+((w3-i/100)/(Ncrit-1))),digits=20))

#print(weight)


normalise<-sum(c(round((w1+((w3-i/100)/(Ncrit-1))),digits=20),round((w2+((w3-
i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w4+((w3-i/100)/(Ncrit-
1))),digits=20),round((w5+((w3-i/100)/(Ncrit-1))),digits=20),round((w6+((w3-
i/100)/(Ncrit-1))),digits=20),round((w7+((w3-i/100)/(Ncrit-1))),digits=20)))

#print (normalise)


weight<-c(round((w1+((w3-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w2+((w3-i/100)/(Ncrit-
1))),digits=20)/normalise,round(i/100,digits=20)/normalise,round((w4+((w3-
i/100)/(Ncrit-1))),digits=20)/normalise,round((w5+((w3-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w6+((w3-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w7+((w3-i/100)/(Ncrit-1))),digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)
```

```
if (i==100){

weight<-c(0,0,1,0,0,0,0)

}


TOPSISLinear(MP,weight,optimise)

C3order1[i]<-TOPSISLinear(MP,weight,optimise)[1,3]

C3order2[i]<-TOPSISLinear(MP,weight,optimise)[2,3]

C3order3[i]<-TOPSISLinear(MP,weight,optimise)[3,3]


C3closeness1[i]<-TOPSISLinear(MP,weight,optimise)[1,2]

C3closeness2[i]<-TOPSISLinear(MP,weight,optimise)[2,2]

C3closeness3[i]<-TOPSISLinear(MP,weight,optimise)[3,2]



if (i>1){

if (C3order1[i]!=C3order1[i-1]||C3order2[i]!=C3order2[i-1]||C3order3[i]!=C3order3[i-1]){

changesC3=changesC3+1

changemomentC3[i]=i

}}}
```

```
for (i in 1:100){

weight<-c(round((w1+((w4-i/100)/(Ncrit-1))),digits=20),round((w2+((w4-
i/100)/(Ncrit-1))),digits=20),round((w3+((w4-i/100)/(Ncrit-
1))),digits=20),round(i/100,digits=20),round((w5+((w4-i/100)/(Ncrit-
1))),digits=20),round((w6+((w4-i/100)/(Ncrit-1))),digits=20),round((w7+((w4-
i/100)/(Ncrit-1))),digits=20))

#print(weight)


normalise<-sum(c(round((w1+((w4-i/100)/(Ncrit-1))),digits=20),round((w2+((w4-
i/100)/(Ncrit-1))),digits=20),round((w3+((w4-i/100)/(Ncrit-
1))),digits=20),round(i/100,digits=20),round((w5+((w4-i/100)/(Ncrit-
1))),digits=20),round((w6+((w4-i/100)/(Ncrit-1))),digits=20),round((w7+((w4-
i/100)/(Ncrit-1))),digits=20)))

#print (normalise)


weight<-c(round((w1+((w4-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w2+((w4-i/100)/(Ncrit-
1)))/normalise,digits=20),round((w3+((w4-i/100)/(Ncrit-
1))),digits=20)/normalise,round(i/100,digits=20)/normalise,round((w5+((w4-
i/100)/(Ncrit-1))),digits=20)/normalise,round((w6+((w4-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w7+((w4-i/100)/(Ncrit-1))),digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)



if (i==100){
```

```
weight<-c(0,0,0,1,0,0,0)
}

TOPSISLinear(MP,weight,optimise)
C4order1[i]<-TOPSISLinear(MP,weight,optimise)[1,3]
C4order2[i]<-TOPSISLinear(MP,weight,optimise)[2,3]
C4order3[i]<-TOPSISLinear(MP,weight,optimise)[3,3]

C4closeness1[i]<-TOPSISLinear(MP,weight,optimise)[1,2]
C4closeness2[i]<-TOPSISLinear(MP,weight,optimise)[2,2]
C4closeness3[i]<-TOPSISLinear(MP,weight,optimise)[3,2]

if (i>1){
if (C4order1[i]!=C4order1[i-1]||C4order2[i]!=C4order2[i-1]||C4order3[i]!=C4order3[i-1]){
changesC4=changesC4+1
changemomentC4[i]=i
}}}

for (i in 1:100){
weight<-c(round((w1+((w5-i/100)/(Ncrit-1))),digits=20),round((w2+((w5-i/100)/(Ncrit-1))),digits=20),round((w3+((w5-i/100)/(Ncrit-1))),digits=20),round((w4+((w5-i/100)/(Ncrit-
```

```
1))),digits=20),round(i/100,digits=20),round((w6+((w5-i/100)/(Ncrit-
1))),digits=20),round((w7+((w5-i/100)/(Ncrit-1))),digits=20))

#print(weight)


normalise<-sum(c(round((w1+((w5-i/100)/(Ncrit-1))),digits=20),round((w2+((w5-
i/100)/(Ncrit-1))),digits=20),round((w3+((w5-i/100)/(Ncrit-
1))),digits=20),round((w4+((w5-i/100)/(Ncrit-
1))),digits=20),round(i/100,digits=20),round((w6+((w5-i/100)/(Ncrit-
1))),digits=20),round((w7+((w5-i/100)/(Ncrit-1))),digits=20)))

#print (normalise)


weight<-c(round((w1+((w5-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w2+((w5-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w3+((w5-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w4+((w5-i/100)/(Ncrit-
1))),digits=20)/normalise,round(i/100,digits=20)/normalise,round((w6+((w5-
i/100)/(Ncrit-1))),digits=20)/normalise,round((w7+((w5-i/100)/(Ncrit-
1))),digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)



if (i==100){

weight<-c(0,0,0,0,1,0,0)

}
```

```
TOPSISLinear(MP,weight,optimise)

C5order1[i]<-TOPSISLinear(MP,weight,optimise)[1,3]

C5order2[i]<-TOPSISLinear(MP,weight,optimise)[2,3]

C5order3[i]<-TOPSISLinear(MP,weight,optimise)[3,3]


C5closeness1[i]<-TOPSISLinear(MP,weight,optimise)[1,2]

C5closeness2[i]<-TOPSISLinear(MP,weight,optimise)[2,2]

C5closeness3[i]<-TOPSISLinear(MP,weight,optimise)[3,2]


if (i>1){

if (C5order1[i]!=C5order1[i-1]||C5order2[i]!=C5order2[i-1]||C5order3[i]!=C5order3[i-1]){

changesC5=changesC5+1

changemomentC5[i]=i

}}}


for (i in 1:100){

weight<-c(round((w1+((w6-i/100)/(Ncrit-1))),digits=20),round((w2+((w6-i/100)/(Ncrit-1))),digits=20),round((w3+((w6-i/100)/(Ncrit-1))),digits=20),round((w4+((w6-i/100)/(Ncrit-1))),digits=20),round((w5+((w6-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w7+((w6-i/100)/(Ncrit-1))),digits=20))
```

```
#print(weight)

normalise<-sum(c(round((w1+((w6-i/100)/(Ncrit-1))),digits=20),round((w2+((w6-i/100)/(Ncrit-1))),digits=20),round((w3+((w6-i/100)/(Ncrit-1))),digits=20),round((w4+((w6-i/100)/(Ncrit-1))),digits=20),round((w5+((w6-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w7+((w6-i/100)/(Ncrit-1))),digits=20)))

#print (normalise)

weight<-c(round((w1+((w6-i/100)/(Ncrit-1))),digits=20)/normalise,round((w2+((w6-i/100)/(Ncrit-1))),digits=20)/normalise,round((w3+((w6-i/100)/(Ncrit-1))),digits=20)/normalise,round((w4+((w6-i/100)/(Ncrit-1))),digits=20)/normalise,round((w5+((w6-i/100)/(Ncrit-1))),digits=20)/normalise,round(i/100,digits=20)/normalise,round((w7+((w6-i/100)/(Ncrit-1))),digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)

if (i==100){

weight<-c(0,0,0,0,0,1,0)

}

TOPSISLinear(MP,weight,optimise)

C6order1[i]<-TOPSISLinear(MP,weight,optimise)[1,3]
```

```
C6order2[i]<-TOPSISLinear(MP,weight,optimise)[2,3]

C6order3[i]<-TOPSISLinear(MP,weight,optimise)[3,3]


C6closeness1[i]<-TOPSISLinear(MP,weight,optimise)[1,2]

C6closeness2[i]<-TOPSISLinear(MP,weight,optimise)[2,2]

C6closeness3[i]<-TOPSISLinear(MP,weight,optimise)[3,2]


if (i>1){

if (C6order1[i]!=C6order1[i-1]||C6order2[i]!=C6order2[i-1]||C6order3[i]!=C6order3[i-1]){

changesC6=changesC6+1

changemomentC6[i]=i

}}}



for (i in 1:100){

weight<-c(round((w1+((w7-i/100)/(Ncrit-1))),digits=20),round((w2+((w7-i/100)/(Ncrit-1))),digits=20),round((w3+((w7-i/100)/(Ncrit-1))),digits=20),round((w4+((w7-i/100)/(Ncrit-1))),digits=20),round((w5+((w7-i/100)/(Ncrit-1))),digits=20),round((w6+((w7-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20))

#print(weight)
```

```
normalise<-sum(c(round((w1+((w7-i/100)/(Ncrit-1))),digits=20),round((w2+((w7-
i/100)/(Ncrit-1))),digits=20),round((w3+((w7-i/100)/(Ncrit-
1))),digits=20),round((w4+((w7-i/100)/(Ncrit-1))),digits=20),round((w5+((w7-
i/100)/(Ncrit-1))),digits=20),round((w6+((w7-i/100)/(Ncrit-
1))),digits=20),round(i/100,digits=20)))

#print (normalise)


weight<-c(round((w1+((w7-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w2+((w7-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w3+((w7-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w4+((w7-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w5+((w7-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w6+((w7-i/100)/(Ncrit-
1))),digits=20)/normalise,round(i/100,digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)


if (i==100){

weight<-c(0,0,0,0,0,0,1)

}


TOPSISLinear(MP,weight,optimise)

C7order1[i]<-TOPSISLinear(MP,weight,optimise)[1,3]

C7order2[i]<-TOPSISLinear(MP,weight,optimise)[2,3]

C7order3[i]<-TOPSISLinear(MP,weight,optimise)[3,3]
```

```
C7closeness1[i]<-TOPSISLinear(MP,weight,optimise)[1,2]

C7closeness2[i]<-TOPSISLinear(MP,weight,optimise)[2,2]

C7closeness3[i]<-TOPSISLinear(MP,weight,optimise)[3,2]



if (i>1){

if (C7order1[i]!=C7order1[i-1]||C7order2[i]!=C7order2[i-1]||C7order3[i]!=C7order3[i-1]){

changesC7=changesC7+1

changemomentC7[i]=i

}}}



MinDistC1= min(changemomentC1)

SensDistC1<-round(abs(MinDistC1-w1*100),digits=20)

if (MinDistC1==1000){

SensDistC1<-0

}

MinDistC2= min(changemomentC2)

SensDistC2<-round(abs(MinDistC2-w2*100),digits=20)

if (MinDistC2==1000){

SensDistC2<-0
```

```
}
MinDistC3= min(changemomentC3)
SensDistC3<-round(abs(MinDistC3-w3*100),digits=20)
if (MinDistC3==1000){
SensDistC3<-0
}
MinDistC4= min(changemomentC4)
SensDistC4<-round(abs(MinDistC4-w4*100),digits=20)
if (MinDistC4==1000){
SensDistC4<-0
}
MinDistC5= min(changemomentC5)
SensDistC5<-round(abs(MinDistC5-w5*100),digits=20)
if (MinDistC5==1000){
SensDistC5<-0
}
MinDistC6= min(changemomentC6)
SensDistC6<-round(abs(MinDistC6-w6*100),digits=20)
if (MinDistC6==1000){
SensDistC6<-0
}
MinDistC7= min(changemomentC7)
SensDistC7<-round(abs(MinDistC7-w7*100),digits=20)
```

```
if (MinDistC7==1000){
SensDistC7<-0
}

for (i in 1:100){
 if (changemomentC1[i]!=1000){
 AcumC1=AcumC1+round(abs(changemomentC1[i]-w1*100),digits=20)
 plotvectorC1[counterC1]<-changemomentC1[i]
 counterC1<-counterC1+1
}
if (changemomentC2[i]!=1000){
 AcumC2=AcumC2+round(abs(changemomentC2[i]-w2*100),digits=20)
plotvectorC2[counterC2]<-changemomentC2[i]
 counterC2<-counterC2+1
}
if (changemomentC3[i]!=1000){
 AcumC3=AcumC3+round(abs(changemomentC3[i]-w3*100),digits=20)
plotvectorC3[counterC3]<-changemomentC3[i]
 counterC3<-counterC3+1
}
if (changemomentC4[i]!=1000){
 AcumC4=AcumC4+round(abs(changemomentC4[i]-w4*100),digits=20)
plotvectorC4[counterC4]<-changemomentC4[i]
```

```
 counterC4<-counterC4+1
}
if (changemomentC5[i]!=1000){
 AcumC5=AcumC5+round(abs(changemomentC5[i]-w5*100),digits=20)
plotvectorC5[counterC5]<-changemomentC5[i]
 counterC5<-counterC5+1
}
if (changemomentC6[i]!=1000){
 AcumC6=AcumC6+round(abs(changemomentC6[i]-w6*100),digits=20)
plotvectorC6[counterC6]<-changemomentC6[i]
 counterC6<-counterC6+1
}
if (changemomentC7[i]!=1000){
 AcumC7=AcumC7+round(abs(changemomentC7[i]-w7*100),digits=20)
plotvectorC7[counterC7]<-changemomentC7[i]
 counterC7<-counterC7+1
}
}

AvgDistC1<-AcumC1/changesC1
sensitivityC1<-(1/SensDistC1)+(changesC1/AvgDistC1)
if (SensDistC1==0||AvgDistC1==0){
sensitivityC1<-0
```

```
}

AvgDistC2=AcumC2/changesC2
sensitivityC2=(1/SensDistC2)+(changesC2/AvgDistC2)
if (SensDistC2==0||AvgDistC2==0){
sensitivityC2<-0
}

AvgDistC3=AcumC3/changesC3
sensitivityC3=(1/SensDistC3)+(changesC3/AvgDistC3)
if (SensDistC3==0||AvgDistC3==0){
sensitivityC3<-0
}

AvgDistC4=AcumC4/changesC4
sensitivityC4=(1/SensDistC4)+(changesC4/AvgDistC4)
if (SensDistC4==0||AvgDistC4==0){
sensitivityC4<-0
}

AvgDistC5=AcumC5/changesC5
sensitivityC5=(1/SensDistC5)+(changesC5/AvgDistC5)
if (SensDistC5==0||AvgDistC5==0){
```

```
sensitivityC5<-0
}

AvgDistC6=AcumC6/changesC6
sensitivityC6=(1/SensDistC6)+(changesC6/AvgDistC6)
if (SensDistC6==0||AvgDistC6==0){
sensitivityC6<-0
}

AvgDistC7=AcumC7/changesC7
sensitivityC7=(1/SensDistC7)+(changesC7/AvgDistC7)
if (SensDistC7==0||AvgDistC7==0){
sensitivityC7<-0
}
```

A.3 Code using TOPSIS algorithm

```
library(MCDM)
MP <- matrix(c(0.1438,0.19145,0.0545,1200,1023,900,137,132.5,80,4.11,0.88667,2.49667,0.06177,0.08684,0.03375,1.93925,3.8966,0.47323,63,60,90),nrow=3,ncol=7)
Ncrit<-7
newweight<-rep(0,7)
v<-1
w1<-0.26
```

```
w2<-0.11

w3<-0.11

w4<-0.09

w5<-0.11

w6<-0.11

w7<-0.21

MinDistC1<-0

MinDistC2<-0

MinDistC3<-0

MinDistC4<-0

MinDistC5<-0

MinDistC6<-0

MinDistC7<-0

SensDistC1<-0

SensDistC2<-0

SensDistC3<-0

SensDistC4<-0

SensDistC5<-0

SensDistC6<-0

SensDistC7<-0

AvgDistC1<-0

AvgDistC2<-0

AvgDistC3<-0
```

```
AvgDistC4<-0

AvgDistC5<-0

AvgDistC6<-0

AvgDistC7<-0

AcumC1<-0

AcumC2<-0

AcumC3<-0

AcumC4<-0

AcumC5<-0

AcumC6<-0

AcumC7<-0

sensitivityC1<-0

sensitivityC2<-0

sensitivityC3<-0

sensitivityC4<-0

sensitivityC5<-0

sensitivityC6<-0

sensitivityC7<-0

C1order1<-rep(0,100)

C1order2<-rep(0,100)

C1order3<-rep(0,100)

C2order1<-rep(0,100)

C2order2<-rep(0,100)
```

```
C2order3<-rep(0,100)
C3order1<-rep(0,100)
C3order2<-rep(0,100)
C3order3<-rep(0,100)
C4order1<-rep(0,100)
C4order2<-rep(0,100)
C4order3<-rep(0,100)
C5order1<-rep(0,100)
C5order2<-rep(0,100)
C5order3<-rep(0,100)
C6order1<-rep(0,100)
C6order2<-rep(0,100)
C6order3<-rep(0,100)
C7order1<-rep(0,100)
C7order2<-rep(0,100)
C7order3<-rep(0,100)

C1closeness1<-rep(0,100)
C1closeness2<-rep(0,100)
C1closeness3<-rep(0,100)
C2closeness1<-rep(0,100)
C2closeness2<-rep(0,100)
C2closeness3<-rep(0,100)
```

```
C3closeness1<-rep(0,100)

C3closeness2<-rep(0,100)

C3closeness3<-rep(0,100)

C4closeness1<-rep(0,100)

C4closeness2<-rep(0,100)

C4closeness3<-rep(0,100)

C5closeness1<-rep(0,100)

C5closeness2<-rep(0,100)

C5closeness3<-rep(0,100)

C6closeness1<-rep(0,100)

C6closeness2<-rep(0,100)

C6closeness3<-rep(0,100)

C7closeness1<-rep(0,100)

C7closeness2<-rep(0,100)

C7closeness3<-rep(0,100)



optimise<-c('min','min','max','max','min','min','min')

changesC1<-0

changesC2<-0

changesC3<-0

changesC4<-0

changesC5<-0
```

```
changesC6<-0

changesC7<-0

changemomentC1<-rep(1000,100)

changemomentC2<-rep(1000,100)

changemomentC3<-rep(1000,100)

changemomentC4<-rep(1000,100)

changemomentC5<-rep(1000,100)

changemomentC6<-rep(1000,100)

changemomentC7<-rep(1000,100)

counterC1<-1

counterC2<-1

counterC3<-1

counterC4<-1

counterC5<-1

counterC6<-1

counterC7<-1

plotvectorC1<-c()

plotvectorC2<-c()

plotvectorC3<-c()

plotvectorC4<-c()

plotvectorC5<-c()

plotvectorC6<-c()

plotvectorC7<-c()
```

```
for (i in 1:100){

weight<-c(round(i/100,digits=20),round((w2+((w1-i/100)/(Ncrit-1))),digits=20),round((w3+((w1-i/100)/(Ncrit-1))),digits=20),round((w4+((w1-i/100)/(Ncrit-1))),digits=20),round((w5+((w1-i/100)/(Ncrit-1))),digits=20),round((w6+((w1-i/100)/(Ncrit-1))),digits=20),round((w7+((w1-i/100)/(Ncrit-1))),digits=20))

#print(weight)


normalise<-sum(c(round(i/100,digits=20),round((w2+((w1-i/100)/(Ncrit-1))),digits=20),round((w3+((w1-i/100)/(Ncrit-1))),digits=20),round((w4+((w1-i/100)/(Ncrit-1))),digits=20),round((w5+((w1-i/100)/(Ncrit-1))),digits=20),round((w6+((w1-i/100)/(Ncrit-1))),digits=20),round((w7+((w1-i/100)/(Ncrit-1))),digits=20)))

#print (normalise)


weight<-c(round(i/100,digits=20)/normalise,round((w2+((w1-i/100)/(Ncrit-1))),digits=20)/normalise,round((w3+((w1-i/100)/(Ncrit-1))),digits=20)/normalise,round((w4+((w1-i/100)/(Ncrit-1))),digits=20)/normalise,round((w5+((w1-i/100)/(Ncrit-1))),digits=20)/normalise,round((w6+((w1-i/100)/(Ncrit-1))),digits=20)/normalise,round((w7+((w1-i/100)/(Ncrit-1))),digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)


if (i==100){

weight<-c(1,0,0,0,0,0,0)
```

```
}

VIKOR(MP,weight,optimise,v)
C1order1[i]<-VIKOR(MP,weight,optimise,v)[1,5]
C1order2[i]<-VIKOR(MP,weight,optimise,v)[2,5]
C1order3[i]<-VIKOR(MP,weight,optimise,v)[3,5]

C1closeness1[i]<-VIKOR(MP,weight,optimise,v)[1,4]
C1closeness2[i]<-VIKOR(MP,weight,optimise,v)[2,4]
C1closeness3[i]<-VIKOR(MP,weight,optimise,v)[3,4]

if (i>1){
if (C1order1[i]!=C1order1[i-1]||C1order2[i]!=C1order2[i-1]||C1order3[i]!=C1order3[i-1]){
changesC1=changesC1+1
changemomentC1[i]=i
}}}

for (i in 1:100){
weight<-c(round((w1+((w2-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w3+((w2-i/100)/(Ncrit-1))),digits=20),round((w4+((w2-i/100)/(Ncrit-1))),digits=20),round((w5+((w2-i/100)/(Ncrit-1))),digits=20),round((w6+((w2-i/100)/(Ncrit-1))),digits=20),round((w7+((w2-i/100)/(Ncrit-1))),digits=20))
```

```
#print(weight)


normalise<-sum(c(round((w1+((w2-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w3+((w2-i/100)/(Ncrit-1))),digits=20),round((w4+((w2-i/100)/(Ncrit-1))),digits=20),round((w5+((w2-i/100)/(Ncrit-1))),digits=20),round((w6+((w2-i/100)/(Ncrit-1))),digits=20),round((w7+((w2-i/100)/(Ncrit-1))),digits=20)))

#print (normalise)


weight<-c(round((w1+((w2-i/100)/(Ncrit-1))),digits=20)/normalise,round(i/100,digits=20)/normalise,round((w3+((w2-i/100)/(Ncrit-1))),digits=20)/normalise,round((w4+((w2-i/100)/(Ncrit-1))),digits=20)/normalise,round((w5+((w2-i/100)/(Ncrit-1))),digits=20)/normalise,round((w6+((w2-i/100)/(Ncrit-1))),digits=20)/normalise,round((w7+((w2-i/100)/(Ncrit-1))),digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)



if (i==100){

weight<-c(0,1,0,0,0,0,0)

}


VIKOR(MP,weight,optimise,v)
```

```
C2order1[i]<-VIKOR(MP,weight,optimise,v)[1,5]

C2order2[i]<-VIKOR(MP,weight,optimise,v)[2,5]

C2order3[i]<-VIKOR(MP,weight,optimise,v)[3,5]


C2closeness1[i]<-VIKOR(MP,weight,optimise,v)[1,4]

C2closeness2[i]<-VIKOR(MP,weight,optimise,v)[2,4]

C2closeness3[i]<-VIKOR(MP,weight,optimise,v)[3,4]



if (i>1){

if (C2order1[i]!=C2order1[i-1]||C2order2[i]!=C2order2[i-1]||C2order3[i]!=C2order3[i-1]){

changesC2=changesC2+1

changemomentC2[i]=i

}}}


for (i in 1:100){

weight<-c(round((w1+((w3-i/100)/(Ncrit-1))),digits=20),round((w2+((w3-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w4+((w3-i/100)/(Ncrit-1))),digits=20),round((w5+((w3-i/100)/(Ncrit-1))),digits=20),round((w6+((w3-i/100)/(Ncrit-1))),digits=20),round((w7+((w3-i/100)/(Ncrit-1))),digits=20))

#print(weight)
```

```
normalise<-sum(c(round((w1+((w3-i/100)/(Ncrit-1))),digits=20),round((w2+((w3-
i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w4+((w3-i/100)/(Ncrit-
1))),digits=20),round((w5+((w3-i/100)/(Ncrit-1))),digits=20),round((w6+((w3-
i/100)/(Ncrit-1))),digits=20),round((w7+((w3-i/100)/(Ncrit-1))),digits=20)))

#print (normalise)


weight<-c(round((w1+((w3-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w2+((w3-i/100)/(Ncrit-
1))),digits=20)/normalise,round(i/100,digits=20)/normalise,round((w4+((w3-
i/100)/(Ncrit-1))),digits=20)/normalise,round((w5+((w3-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w6+((w3-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w7+((w3-i/100)/(Ncrit-1))),digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)


if (i==100){

weight<-c(0,0,1,0,0,0,0)

}


VIKOR(MP,weight,optimise,v)

C3order1[i]<-VIKOR(MP,weight,optimise,v)[1,5]

C3order2[i]<-VIKOR(MP,weight,optimise,v)[2,5]

C3order3[i]<-VIKOR(MP,weight,optimise,v)[3,5]
```

```
C3closeness1[i]<-VIKOR(MP,weight,optimise,v)[1,4]

C3closeness2[i]<-VIKOR(MP,weight,optimise,v)[2,4]

C3closeness3[i]<-VIKOR(MP,weight,optimise,v)[3,4]



if (i>1){

if (C3order1[i]!=C3order1[i-1]||C3order2[i]!=C3order2[i-1]||C3order3[i]!=C3order3[i-1]){

changesC3=changesC3+1

changemomentC3[i]=i

}}}



for (i in 1:100){

weight<-c(round((w1+((w4-i/100)/(Ncrit-1))),digits=20),round((w2+((w4-i/100)/(Ncrit-1))),digits=20),round((w3+((w4-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w5+((w4-i/100)/(Ncrit-1))),digits=20),round((w6+((w4-i/100)/(Ncrit-1))),digits=20),round((w7+((w4-i/100)/(Ncrit-1))),digits=20))

#print(weight)


normalise<-sum(c(round((w1+((w4-i/100)/(Ncrit-1))),digits=20),round((w2+((w4-i/100)/(Ncrit-1))),digits=20),round((w3+((w4-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w5+((w4-i/100)/(Ncrit-
```

```
1))),digits=20),round((w6+((w4-i/100)/(Ncrit-1))),digits=20),round((w7+((w4-
i/100)/(Ncrit-1))),digits=20)))

#print (normalise)


weight<-c(round((w1+((w4-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w2+((w4-i/100)/(Ncrit-
1)))/normalise,digits=20),round((w3+((w4-i/100)/(Ncrit-
1))),digits=20)/normalise,round(i/100,digits=20)/normalise,round((w5+((w4-
i/100)/(Ncrit-1))),digits=20)/normalise,round((w6+((w4-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w7+((w4-i/100)/(Ncrit-1))),digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)



if (i==100){

weight<-c(0,0,0,1,0,0,0)

}


VIKOR(MP,weight,optimise,v)

C4order1[i]<-VIKOR(MP,weight,optimise,v)[1,5]

C4order2[i]<-VIKOR(MP,weight,optimise,v)[2,5]

C4order3[i]<-VIKOR(MP,weight,optimise,v)[3,5]
```

```
C4closeness1[i]<-VIKOR(MP,weight,optimise,v)[1,4]

C4closeness2[i]<-VIKOR(MP,weight,optimise,v)[2,4]

C4closeness3[i]<-VIKOR(MP,weight,optimise,v)[3,4]


if (i>1){

if (C4order1[i]!=C4order1[i-1]||C4order2[i]!=C4order2[i-1]||C4order3[i]!=C4order3[i-1]){

changesC4=changesC4+1

changemomentC4[i]=i

}}}



for (i in 1:100){

weight<-c(round((w1+((w5-i/100)/(Ncrit-1))),digits=20),round((w2+((w5-i/100)/(Ncrit-1))),digits=20),round((w3+((w5-i/100)/(Ncrit-1))),digits=20),round((w4+((w5-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w6+((w5-i/100)/(Ncrit-1))),digits=20),round((w7+((w5-i/100)/(Ncrit-1))),digits=20))

#print(weight)


normalise<-sum(c(round((w1+((w5-i/100)/(Ncrit-1))),digits=20),round((w2+((w5-i/100)/(Ncrit-1))),digits=20),round((w3+((w5-i/100)/(Ncrit-1))),digits=20),round((w4+((w5-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w6+((w5-i/100)/(Ncrit-1))),digits=20),round((w7+((w5-i/100)/(Ncrit-1))),digits=20)))
```

```
#print (normalise)

weight<-c(round((w1+((w5-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w2+((w5-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w3+((w5-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w4+((w5-i/100)/(Ncrit-
1))),digits=20)/normalise,round(i/100,digits=20)/normalise,round((w6+((w5-
i/100)/(Ncrit-1))),digits=20)/normalise,round((w7+((w5-i/100)/(Ncrit-
1))),digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)

if (i==100){

weight<-c(0,0,0,0,1,0,0)

}

VIKOR(MP,weight,optimise,v)

C5order1[i]<-VIKOR(MP,weight,optimise,v)[1,5]

C5order2[i]<-VIKOR(MP,weight,optimise,v)[2,5]

C5order3[i]<-VIKOR(MP,weight,optimise,v)[3,5]

C5closeness1[i]<-VIKOR(MP,weight,optimise,v)[1,4]
```

```
C5closeness2[i]<-VIKOR(MP,weight,optimise,v)[2,4]

C5closeness3[i]<-VIKOR(MP,weight,optimise,v)[3,4]


if (i>1){

if (C5order1[i]!=C5order1[i-1]||C5order2[i]!=C5order2[i-1]||C5order3[i]!=C5order3[i-1]){

changesC5=changesC5+1

changemomentC5[i]=i

}}}


for (i in 1:100){

weight<-c(round((w1+((w6-i/100)/(Ncrit-1))),digits=20),round((w2+((w6-i/100)/(Ncrit-1))),digits=20),round((w3+((w6-i/100)/(Ncrit-1))),digits=20),round((w4+((w6-i/100)/(Ncrit-1))),digits=20),round((w5+((w6-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w7+((w6-i/100)/(Ncrit-1))),digits=20))

#print(weight)


normalise<-sum(c(round((w1+((w6-i/100)/(Ncrit-1))),digits=20),round((w2+((w6-i/100)/(Ncrit-1))),digits=20),round((w3+((w6-i/100)/(Ncrit-1))),digits=20),round((w4+((w6-i/100)/(Ncrit-1))),digits=20),round((w5+((w6-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20),round((w7+((w6-i/100)/(Ncrit-1))),digits=20)))

#print (normalise)
```

```
weight<-c(round((w1+((w6-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w2+((w6-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w3+((w6-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w4+((w6-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w5+((w6-i/100)/(Ncrit-
1))),digits=20)/normalise,round(i/100,digits=20)/normalise,round((w7+((w6-
i/100)/(Ncrit-1))),digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)


if (i==100){

weight<-c(0,0,0,0,0,1,0)

}


VIKOR(MP,weight,optimise,v)

C6order1[i]<-VIKOR(MP,weight,optimise,v)[1,5]

C6order2[i]<-VIKOR(MP,weight,optimise,v)[2,5]

C6order3[i]<-VIKOR(MP,weight,optimise,v)[3,5]


C6closeness1[i]<-VIKOR(MP,weight,optimise,v)[1,4]

C6closeness2[i]<-VIKOR(MP,weight,optimise,v)[2,4]

C6closeness3[i]<-VIKOR(MP,weight,optimise,v)[3,4]
```

```
if (i>1){

if (C6order1[i]!=C6order1[i-1]||C6order2[i]!=C6order2[i-1]||C6order3[i]!=C6order3[i-1]){

changesC6=changesC6+1

changemomentC6[i]=i

}}}




for (i in 1:100){

weight<-c(round((w1+((w7-i/100)/(Ncrit-1))),digits=20),round((w2+((w7-i/100)/(Ncrit-1))),digits=20),round((w3+((w7-i/100)/(Ncrit-1))),digits=20),round((w4+((w7-i/100)/(Ncrit-1))),digits=20),round((w5+((w7-i/100)/(Ncrit-1))),digits=20),round((w6+((w7-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20))

#print(weight)


normalise<-sum(c(round((w1+((w7-i/100)/(Ncrit-1))),digits=20),round((w2+((w7-i/100)/(Ncrit-1))),digits=20),round((w3+((w7-i/100)/(Ncrit-1))),digits=20),round((w4+((w7-i/100)/(Ncrit-1))),digits=20),round((w5+((w7-i/100)/(Ncrit-1))),digits=20),round((w6+((w7-i/100)/(Ncrit-1))),digits=20),round(i/100,digits=20)))

#print (normalise)


weight<-c(round((w1+((w7-i/100)/(Ncrit-1))),digits=20)/normalise,round((w2+((w7-i/100)/(Ncrit-
```

```
1))),digits=20)/normalise,round((w3+((w7-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w4+((w7-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w5+((w7-i/100)/(Ncrit-
1))),digits=20)/normalise,round((w6+((w7-i/100)/(Ncrit-
1))),digits=20)/normalise,round(i/100,digits=20)/normalise)

#print (weight)

check<-sum(weight)

#print (check)


if (i==100){

weight<-c(0,0,0,0,0,0,1)

}


VIKOR(MP,weight,optimise,v)

C7order1[i]<-VIKOR(MP,weight,optimise,v)[1,5]

C7order2[i]<-VIKOR(MP,weight,optimise,v)[2,5]

C7order3[i]<-VIKOR(MP,weight,optimise,v)[3,5]


C7closeness1[i]<-VIKOR(MP,weight,optimise,v)[1,4]

C7closeness2[i]<-VIKOR(MP,weight,optimise,v)[2,4]

C7closeness3[i]<-VIKOR(MP,weight,optimise,v)[3,4]




if (i>1){
```

```
if (C7order1[i]!=C7order1[i-1]||C7order2[i]!=C7order2[i-1]||C7order3[i]!=C7order3[i-1]){
changesC7=changesC7+1
changemomentC7[i]=i
}}}

MinDistC1= min(changemomentC1)
SensDistC1<-round(abs(MinDistC1-w1*100),digits=20)
if (MinDistC1==1000){
SensDistC1<-0
}
MinDistC2= min(changemomentC2)
SensDistC2<-round(abs(MinDistC2-w2*100),digits=20)
if (MinDistC2==1000){
SensDistC2<-0
}
MinDistC3= min(changemomentC3)
SensDistC3<-round(abs(MinDistC3-w3*100),digits=20)
if (MinDistC3==1000){
SensDistC3<-0
}
```

```
MinDistC4= min(changemomentC4)

SensDistC4<-round(abs(MinDistC4-w4*100),digits=20)

if (MinDistC4==1000){

SensDistC4<-0

}

MinDistC5= min(changemomentC5)

SensDistC5<-round(abs(MinDistC5-w5*100),digits=20)

if (MinDistC5==1000){

SensDistC5<-0

}

MinDistC6= min(changemomentC6)

SensDistC6<-round(abs(MinDistC6-w6*100),digits=20)

if (MinDistC6==1000){

SensDistC6<-0

}

MinDistC7= min(changemomentC7)

SensDistC7<-round(abs(MinDistC7-w7*100),digits=20)

if (MinDistC7==1000){

SensDistC7<-0

}


for (i in 1:100){

 if (changemomentC1[i]!=1000){
```

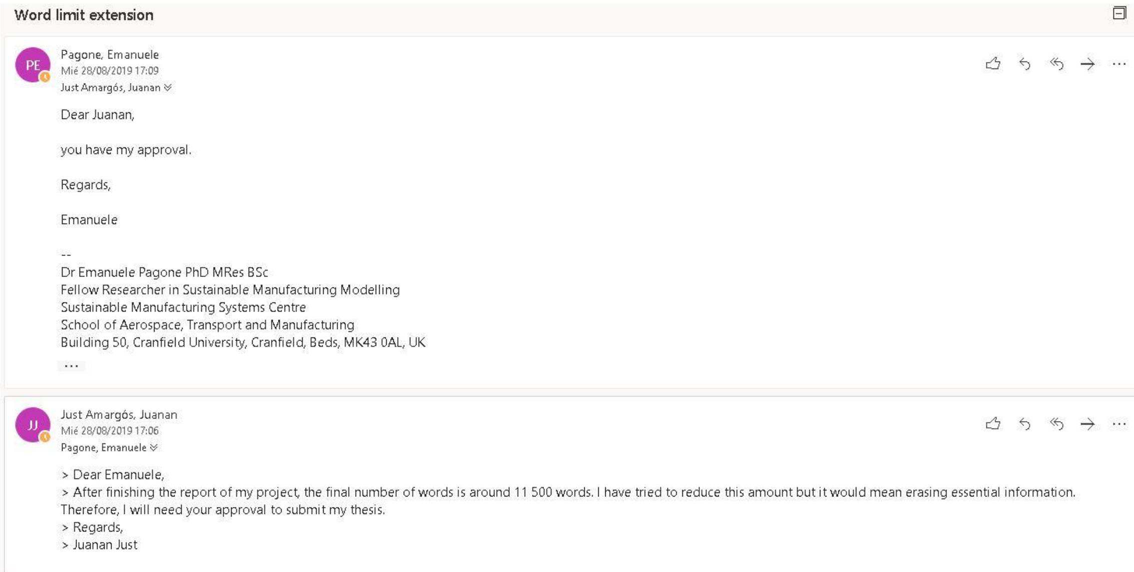
```
 AcumC1=AcumC1+round(abs(changemomentC1[i]-w1*100),digits=20)
 plotvectorC1[counterC1]<-changemomentC1[i]
 counterC1<-counterC1+1
}
if (changemomentC2[i]!=1000){
 AcumC2=AcumC2+round(abs(changemomentC2[i]-w2*100),digits=20)
plotvectorC2[counterC2]<-changemomentC2[i]
 counterC2<-counterC2+1
}
if (changemomentC3[i]!=1000){
 AcumC3=AcumC3+round(abs(changemomentC3[i]-w3*100),digits=20)
plotvectorC3[counterC3]<-changemomentC3[i]
 counterC3<-counterC3+1
}
if (changemomentC4[i]!=1000){
 AcumC4=AcumC4+round(abs(changemomentC4[i]-w4*100),digits=20)
plotvectorC4[counterC4]<-changemomentC4[i]
 counterC4<-counterC4+1
}
if (changemomentC5[i]!=1000){
 AcumC5=AcumC5+round(abs(changemomentC5[i]-w5*100),digits=20)
plotvectorC5[counterC5]<-changemomentC5[i]
 counterC5<-counterC5+1
```

```
}
if (changemomentC6[i]!=1000){
 AcumC6=AcumC6+round(abs(changemomentC6[i]-w6*100),digits=20)
plotvectorC6[counterC6]<-changemomentC6[i]
 counterC6<-counterC6+1
}
if (changemomentC7[i]!=1000){
 AcumC7=AcumC7+round(abs(changemomentC7[i]-w7*100),digits=20)
plotvectorC7[counterC7]<-changemomentC7[i]
 counterC7<-counterC7+1
}
}

AvgDistC1<-AcumC1/changesC1
sensitivityC1<-(1/SensDistC1)+(changesC1/AvgDistC1)
if (SensDistC1==0||AvgDistC1==0){
sensitivityC1<-0
}

AvgDistC2=AcumC2/changesC2
sensitivityC2=(1/SensDistC2)+(changesC2/AvgDistC2)
if (SensDistC2==0||AvgDistC2==0){
sensitivityC2<-0
```

```
}

AvgDistC3=AcumC3/changesC3
sensitivityC3=(1/SensDistC3)+(changesC3/AvgDistC3)
if (SensDistC3==0||AvgDistC3==0){
sensitivityC3<-0
}

AvgDistC4=AcumC4/changesC4
sensitivityC4=(1/SensDistC4)+(changesC4/AvgDistC4)
if (SensDistC4==0||AvgDistC4==0){
sensitivityC4<-0
}

AvgDistC5=AcumC5/changesC5
sensitivityC5=(1/SensDistC5)+(changesC5/AvgDistC5)
if (SensDistC5==0||AvgDistC5==0){
sensitivityC5<-0
}

AvgDistC6=AcumC6/changesC6
sensitivityC6=(1/SensDistC6)+(changesC6/AvgDistC6)
if (SensDistC6==0||AvgDistC6==0){
```

```
sensitivityC6<-0

}


AvgDistC7=AcumC7/changesC7

sensitivityC7=(1/SensDistC7)+(changesC7/AvgDistC7)

if (SensDistC7==0||AvgDistC7==0){

sensitivityC7<-0

}
```

A.4 Word Limit Extension



Word limit extension

Pagone, Emanuele
Mié 28/08/2019 17:09
Just Amargós, Juanan

Dear Juanan,

you have my approval.

Regards,

Emanuele

--
Dr Emanuele Pagone PhD MRes BSc
Fellow Researcher in Sustainable Manufacturing Modelling
Sustainable Manufacturing Systems Centre
School of Aerospace, Transport and Manufacturing
Building 50, Cranfield University, Cranfield, Beds, MK43 0AL, UK

Just Amargós, Juanan
Mié 28/08/2019 17:06
Pagone, Emanuele

> Dear Emanuele,
> After finishing the report of my project, the final number of words is around 11 500 words. I have tried to reduce this amount but it would mean erasing essential information. Therefore, I will need your approval to submit my thesis.
> Regards,
> Juanan Just