

Université Libre de Bruxelles

CoDE - SMG

**An extension of PROMETHEE II to
temporal evaluations
CoDE-SMG – Technical Report Series**

I. BANAMAR, Y. DE SMET

CoDE-SMG – Technical Report Series

Technical Report No.

TR/SMG/2016-002

July 2016

CoDE-SMG – Technical Report Series
ISSN 2030-6296

Published by:

CoDE-SMG, CP 210/01
UNIVERSITÉ LIBRE DE BRUXELLES
Bvd du Triomphe
1050 Ixelles, Belgium

Technical report number TR/SMG/2016-002

The information provided is the sole responsibility of the authors and does not necessarily reflect the opinion of the members of CoDE-SMG. The authors take full responsibility for any copyright breaches that may result from publication of this paper in the CoDE-SMG – Technical Report Series. CoDE-SMG is not responsible for any use that might be made of data appearing in this publication.

An extension of PROMETHEE II to temporal evaluations

CoDE-SMG – Technical Report Series

I. BANAMAR

`issam.banamar@ulb.ac.be`

Y. DE SMET

`yves.de.smet@ulb.ac.be`

CoDE-SMG, Université Libre de Bruxelles, Brussels, Belgium

July 2016

An extension of PROMETHEE II to temporal evaluations

Issam Banamar, Yves De Smet

Service de Mathématiques de la gestion,
Ecole polytechnique de Bruxelles,
Université libre de Bruxelles, Belgium

issam.banamar@ulb.ac.be
yves.de.smet@ulb.ac.be

Abstract

Conventional PROMETHEE methods help decision makers to manage problems without considering temporal evaluations. The aim of this paper is to introduce a temporal extension of PROMETHEE II. A new approach to set the dynamic thresholds is introduced. The weighted arithmetic mean is used to aggregate net flow scores over time. In addition, an interactive query method is developed in order to determine decision maker's instances importance. Simulation results are used to show the effectiveness of this approach in terms of convergence and speed. Finally, an example is given to illustrate the model.

Keywords. Multicriteria decision aid, temporal evaluations, outranking methods, PROMETHEE, GAIA

1 Introduction

In this paper we focus ourselves on decision problems that involve multiple conflicting criteria. Different methods have been developed in this context. Among them we can cite UTA, Zionts-Wallenius ALGORITHM, STEM, MACBETH, AHP, MAUT, ELECTRE, PROMETHEE, etc. [18], [19], [20], [21], [2],[3].

Most of these approaches belong to three main categories:

- Multi-attribute methods: turning multicriteria problems into a single criterion optimization problem (ex. UTA, MACBETH [18], AHP [23]).
- Interactive methods: alternating calculation phases and dialogue with the decision maker (ex. Zionts-Wallenius ALGORITHM, STEM [19],[20]).

- Outranking methods: computing pairwise comparisons between alternatives which allows an enrichment of the traditional dominance relation (ex. ELECTRE, PROMETHEE [2],[22]).

In this contribution we focus ourselves on PROMETHEE II. This method has been widely used thanks to its simplicity and the existence of user-friendly softwares (D-Sight [24], Visual PROMETHEE [25]). Furthermore, without being exhaustive, Behzadian et al. [1] have listed more than 200 applications based on PROMETHEE published in more than 100 different journals covering various fields such as sport, health care, finance, environmental management, agriculture, etc.

The aim of this paper is to propose an extension of PROMETHEE II to temporal evaluations. In order to illustrate the importance of managing temporal multicriteria problems let us consider the three following examples:

Banks performance assessment: The principal regulator of a given country would like to assess the performances of commercial banks -that it controls- during the period between 2008 and 2014. This will be made with respect to the CAMEL Rating System [17] which is composed of five main criteria:

- Capital Adequacy: a measure of the bank's capital;
- Asset Quality: a measure of loans quality;
- Management Capacity: profit per employee, expenses per employee, gross earning assets, total assets, etc;
- Earning Ability: assessed through ratios like net investment margin, return on assets, net profit margin, etc;
- Liquidity: a measure of the ease with which assets can be converted into cash.

In this context the regulator will not only consider the current evaluation but also the global trend over the last six years.

Human Development Index: The assessment of the global ranking of 10 emerging economies with respect to the Human Development Index (HDI) [9] during the last three decades. The HDI consists of four main criteria:

- Long and healthy life: a measure of the life expectancy at birth;
- Expected years of schooling: measured for adults aged of 25 years and older;
- Access to knowledge: a measure of the enrolment by age at all level of education and the number of school age children in the population;
- Standard of living: measured by the Gross National Income per Capita.

This example will be further studied in section 6.

Football players assessment: The coach of a junior football club aims to have a global ranking of dozens of players who have been regularly controlled during one year of training. The assessment is made with respect to 3 criteria:

- Speed test: time needed for a 30m sprint;
- Anaerobic capacity: capacity to maintain an intense effort (through ergometer);
- Peak power: through Sargent Jump Test.

In this example, it will be inappropriate to adopt an approach based on one single

control because, in general, players performances are increasing during the year. In addition, during one given control, a player could be injured for instance leading to a local bad evaluation. This means that the ranking should consider the whole training period. One can imagine that the coach is interested in monitoring players evolution over one or more criteria during the year to better advise them afterwards. Hence, the method to be used should reflect the temporal aspect of the problem.

Several temporal MCDA methods have been proposed in the literature [14], [10]. However, in this context the time weighting vector determination remains, to our point of view, questionable. To give an example, Chunwei et al. [10] base the time weighting vector determination on an orness single value $\lambda \in [0, 1]$ given by the decision maker (DM). Then, in order to find weights that evolve smoothly, the entropy of these weights is maximised. Yet, we think that the nature of λ would remain abstract for many DMs. This paper addresses this subject with a new perspective.

The next sections are organised as following: first an overview of the PROMETHEE II method and GAIA plan is proposed. Then some assumptions about the new temporal model are stated. In the following section we explore the new model step by step with an emphasis on the dynamic thresholds setting issue. Then we introduce a new approach to determine the time weighting vector. Finally, an example is proposed to illustrate the application of the new model followed by conclusions.

2 PROMETHEE II and GAIA

PROMETHEE II is a multicriteria decision aid method that allows a complete ranking of the alternatives. PROMETHEE I is dedicated to partial rankings.

We consider a set of n alternatives (or actions) denoted by $A = \{a_1, a_2, \dots, a_n\}$ which are evaluated according to a set of k criteria denoted by $F = \{f_1, f_2, \dots, f_k\}$. Each criterion f_h is characterised by a weight $v_h \in V = \{v_1, v_2, \dots, v_k\}$.

We assume that: $\sum_{h=1}^k v_h = 1$ and $v_h > 0 \quad \forall h \in H = \{1, 2, \dots, k\}$.

The first step of PROMETHEE II is a pairwise comparison between each couple of alternatives $a_i, a_j \in A$. This is based on the computation of the difference of evaluations of a_i over a_j on each criterion $f_h \in F$:

$$d_h(a_i, a_j) = f_h(a_i) - f_h(a_j) \quad (1)$$

Then a preference degree $0 \leq P_h(a_i, a_j) \leq 1$ is computed based on the preference function G_h associated to criterion h such as:

$$P_h(a_i, a_j) = G_h(d_h(a_i, a_j)) \quad (2)$$

Six general preference functions are proposed in software implementing the methods of PROMETHEE. Figure 1 shows one example of these functions:

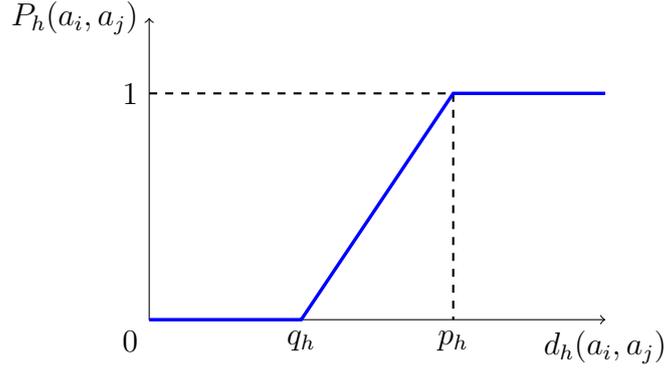


Figure 1: Example of a linear preference function

Practically, assuming that criterion f_h has to be maximized, we set a preference function and two parameters q_h and p_h which are, respectively, the indifference and preference thresholds (with $q_h \leq p_h$) such as:

- If $d_h(a_i, a_j) \leq q_h$ then $P_h(a_i, a_j) = 0$, a_j is preferred or indifferent to a_i with respect to f_h .
- If $q_h < d_h(a_i, a_j) < p_h$ then $0 < P_h(a_i, a_j) < 1$, the preference of a_i over a_j with respect to f_h is as strong as $d_h(a_i, a_j)$ is close to p_h .
- If $p_h \leq d_h(a_i, a_j)$ then $P_h(a_i, a_j) = 1$, a_i is strictly preferred to a_j with respect to f_h .

A global preference index of a_i over a_j with respect to all criteria is defined in the following way:

$$\pi(a_i, a_j) = \sum_{h=1}^k v_h \cdot P_h(a_i, a_j) \quad (3)$$

We have $\pi(a_i, a_j) \geq 0$ and $\pi(a_i, a_j) + \pi(a_j, a_i) \leq 1$

Finally, the alternative a_i is compared to the $n - 1$ other ones. The PROMETHEE net flow score is computed as a difference between the outgoing flow score $\phi^+(a_i)$ and the ingoing flow score $\phi^-(a_i)$ of a_i :

$$\phi(a_i) = \phi^+(a_i) - \phi^-(a_i) = \frac{1}{n-1} \sum_{a \in A} \pi(a_i, a) - \frac{1}{n-1} \sum_{a \in A} \pi(a, a_i) \quad (4)$$

$\phi^+(a_i)$ quantifies the mean preference degree of a_i over all the other alternatives. In the same way $\phi^-(a_i)$ quantifies the mean preference degree of all the alternatives over a_i .

The value $\phi(a_i) \in [-1, 1]$ is such that the higher this score, the better the alternative a_i is ranked. For further details, we refer the interested reader to [5],[4],[3].

In addition let us note the possibility to calculate unicriterion net flow scores as follows:

$$\phi_h(a_i) = \phi_h^+(a_i) - \phi_h^-(a_i) = \frac{1}{n-1} \sum_{a \in A} P_h(a_i, a) - \frac{1}{n-1} \sum_{a \in A} P_h(a, a_i) \quad (5)$$

Obviously, we have:
$$\phi(a_i) = \sum_{h=1}^{h=k} v_h \cdot \phi_h(a_i) \quad (6)$$

The PROMETHEE methodology includes an interactive visualization tool to represent alternatives and criteria on a 2 dimensional plan. The GAIA plan construction turns the k -dimensional unicriterion net flow scores space into a 2-dimensional plan that is called GAIA. For this purpose, a Principal Component Analysis (PCA) technique is applied on the $n \times k$ unicriterion net flow scores matrix. One computes the $k \times k$ variance-covariance matrix \mathcal{C}_u of the unicriterion net flow scores matrix. Then, the eigenvectors of \mathcal{C}_u are calculated. Finally, the two axes defining the GAIA plan are identified as the eigenvectors u and v with the highest associated eigenvalues denoted by λ_1 and λ_2 . For additional information, the interested reader is referred to [6],[7] and [8].

Finally, in order to evaluate the quality of the representation, the following ratio is computed :

$$\delta = \frac{\lambda_1 + \lambda_2}{\sum_{j=1}^k \lambda_j} \quad (7)$$

In practice, values of $\delta > 70\%$ are supposed to be associated to good projections. However, the DM should always keep in mind the approximative nature of this representation. The GAIA plan helps to better understand and structure the multicriteria problem. It allows to identify similar and contrasted alternatives profiles, redundant and conflicting criteria , best and worst performing alternatives with respect to each criterion,etc.

3 Model assumptions

The aim of the temporal model we propose is to give a global ranking of the set of alternatives A with respect to the criteria set F and after being evaluated during the period T . Forecasting is not the primary aim of this model. Nevertheless, since the model is based on instantaneous net flow scores this kind of applications could be investigated in future research.

All evaluations are considered to have been conducted in the past. Let $T = \{t_1, t_2, \dots, t_s\}$ denotes the set of evaluation instants. Each alternative $a_i \in A$ is assumed to be evaluated over each criterion f_h at each instant $t_\ell \in T$. Based on this assumption, one will compute s instantaneous net flow scores for every alternative. We also assume that there is no missing evaluation. In addition, we suppose that the evaluations are crisp and accurate. Consequently, this model is assumed to be synchronous.

The proposed approach supports both static and temporal preferences of the decision maker. Indeed, we suppose that the DM's preferences (indifference, preference thresholds and instants weights) may change over time. This will be discussed in subsections 4.2 and 4.3.

We also suppose that the weight v_h assigned to each criterion f_h is constant during the whole period T . However, as we will see in subsection 4.4, during the aggregation of instantaneous net flow scores the weight v_h will become implicitly dependent

on the instant t_ℓ .

The global ranking is obtained by aggregating the s net flow scores of every alternative (see subsection 4.4). For this purpose, we use the weighted arithmetic mean. We assume that each instant t_ℓ is characterised by a relative weight denoted $w_\ell \in W$ such as $W = \{w_1, w_2, \dots, w_s\}$. The set W represents the time weighting set that satisfies the starting constraints \mathcal{C}_0 such as: $\sum_{\ell=1}^s w_\ell = 1, w_\ell \geq 0 \quad \forall \ell \in L = \{1, 2, \dots, s\}$ and $w_1 \leq w_2 \leq \dots \leq w_\ell \leq \dots \leq w_s$ where T, W and L are considered fixed.

The weights w_1, w_2, \dots, w_s are increasing because we assume that DMs pay more attention to recent evaluations than to older ones (of course, this assumption can easily be relaxed). For instance, in an environmental problem the amount of CO_2 emissions is more important in 2015 than in 2005.

4 General overview of the model

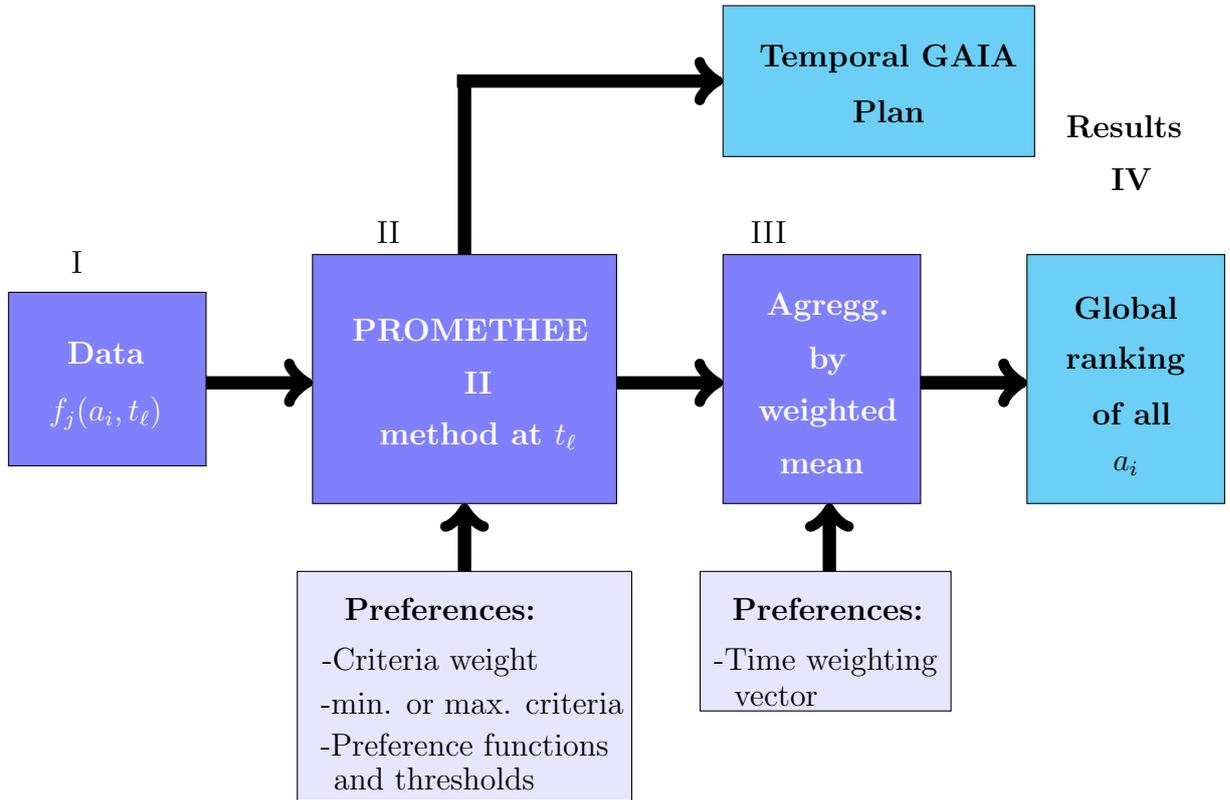


Figure 2: Overview of the temporal extension of PROMETHEE II

Figure 2 represents the main components of the model that we propose in this paper. In the next paragraphs we explore the components of the model one by one.

4.1 Evaluations data

As already outlined, we consider: the alternatives set A , the criteria set F and the instants set T as they have been defined in sections 2 and 3. These sets are assumed to be finite. Let us stress that, since we have n^2 pairwise comparisons to perform at each instant t_ℓ , the complexity of the procedure is $O(n^2)$. Thus, the size of A has an impact on the resolution duration. However, it is worth noting that in a recent paper [36] Calders and Van Assche have introduced an exact algorithm that finds PROMETHEE's net flow scores with a complexity of $O(q.n.\log(n))$, where q is the number of criteria and n the number of alternatives.

4.2 Preferences

Practically, we ask the DM to give his preferences in 5 steps:

- 1 Whether a given criterion has to be maximized or minimized;
- 2 The relative importance of each criterion;
- 3 The type of preference function associated to each criterion;
- 4 Temporal preference and indifference thresholds for every criterion;
- 5 The relative importance of each instant.

The three first steps are set as in the classical PROMETHEE II method. The last step will be detailed in subsection 4.6. For step 4, we consider that indifference and preference thresholds are likely to evolve during the assessment period T .

Indeed, if we refer to the football players example, one can imagine that in the first control, speed test results could be very disparate due to the fact that players were not prepared in the same way. On the contrary, in the last control, results are likely to be less disparate due to the common training. In this context, the coach should pay more attention to the divergence of results in the last control than in the first one. Accordingly, in order to have a realistic approach, the preference threshold of the speed test criterion should have a decreasing trend during T while the indifference threshold could remain static. Figure 3.a shows thresholds evolution corresponding to this example. For convenience we write p and q instead of p_h and q_h :

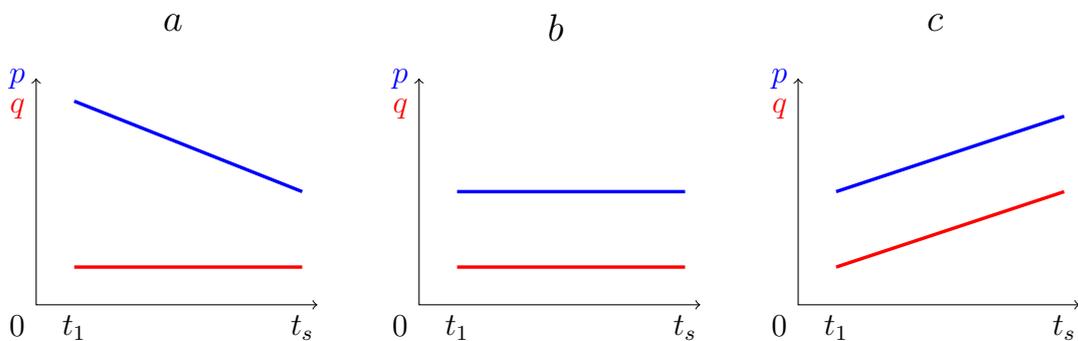


Figure 3: Examples of the possible evolution of indifference and preference thresholds during T

Figure 3.b shows the case of constant (static) indifference and preference thresholds p and q over the evaluation period.

Finally, figure 3.c gives an other potential trend where both q and p are increasing. To illustrate, let us imagine that in a given university, a team of researchers plans to evaluate, annually, the fitness of hundreds of children from their birth until they reach 20 years old. Among the criteria we consider the height of children. Obviously, the height values will be more and more disparate as children grow up. In this context, the researchers are likely to be more attentive to the divergence of results in the first years than in the last ones. Consequently, p has to be increasing. Moreover, if the researchers are only interested in the children with the largest heights, then indifference threshold q has to be increasing too.

The following subsection investigates the question of setting dynamic thresholds.

4.3 Setting dynamic thresholds

Asking the DM to give his indifference and/or preference instantaneous thresholds over the whole period T for a given criterion may be a tedious task. In this subsection we introduce an interpolation technique allowing to set these thresholds based on a few questions.

For the sake of simplicity, we choose to use the preference function of type III [7] and we maintain the monotonicity assumption of the dynamic thresholds (figure 3a, b, c). However, we also assume that the values of an increasing (or decreasing) thresholds set can eventually be constant during some instants. The next paragraphs treat the case of increasing preference thresholds. Of course, these results can be easily generalised, under assumptions just mentioned above, to the other types of thresholds seen in figure 3.

Let us assume $P_{real} = \{p_1, p_2, \dots, p_s\}$ denote the preference thresholds associated to a given criterion. These are associated to the instants t_1, t_2, \dots, t_s . Thus, based on a few known values, we shall find the best approximation to the set P_{real} . For this purpose, four different interpolation techniques are compared based on simulations:

- 1- Only two values are known, p_s and another $p_i \in P_{real}$. The other values are assumed to be interpolated by the straight line joining p_i to p_s (figure 4).
- 2- Three values are known: p_1 , p_i and p_s such that $1 < i < s$. The interpolation is done by two segments, the first one joins p_1 to p_i and the second joins p_i to p_s (figure 4).
- 3- $p_1 < p_i < p_s$ are known, the interpolation is done by the least squares straight line fit (figure 5).
- 4- $p_1 < p_i < p_s$ are known, the interpolation is done by the second order-polynomial line fit (figure 5).

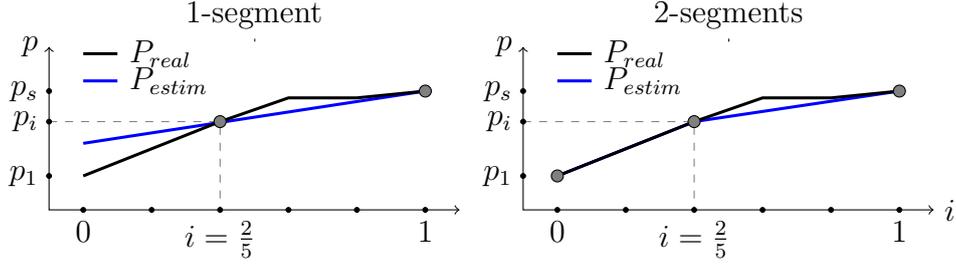


Figure 4: Example of 1-segment and 2-segments interpolation techniques

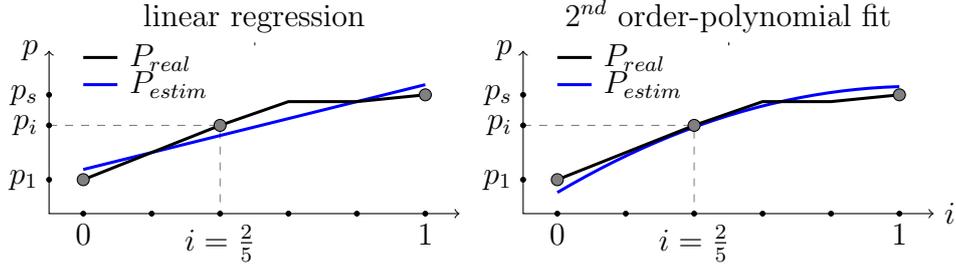


Figure 5: Example of interpolation by linear regression and 2nd order-polynomial fit

For convenience, i is assumed to be located at either $\frac{1}{5}$, $\frac{2}{5}$, $\frac{3}{5}$ or $\frac{4}{5}$ of the time range. These techniques are compared with each other at each location of i . On the other hand, since the preference threshold of the last instant t_s has the largest impact on the global ranking, p_s has to be known rather than approximated.

Two criteria are used to perform these comparisons:

- 1- The maximisation of the Kendall correlation tau between the two sets of unicriterion net flow scores: the first one being based on P_{real} and the second on the approximate values of P_{real} .
- 2- The minimisation of the sum of squared deviations \mathcal{S} between these 2 unicriterion scores.

Every simulation is repeated 100 times. Each time, a unicriterion temporal MCDA problem is created where the characteristic parameters are generated randomly by using the function *randint()* on *Matlab R2007b*. This function will create:

- Instantaneous evaluations over T for each $a_i \in A$ with an increasing standard deviation as shown in algorithm VI in appendix A.
- Instantaneous (dynamic) preference thresholds. Also, $0 \leq p_\ell \leq p_{max}$ such as $\ell = 1, 2, \dots, s$ and p_{max} is the maximum difference between two alternatives evaluations at the last instant t_s (algorithm VII in appendix A).
- Time weighting vector W whose generation will be detailed in section 5.

The simulations are made for three sizes of T (10, 20 and 30 instants) whereas alternatives set size A has been maintained at 30. The simulations based on the assumptions seen from the beginning of this subsection will be referred to as the

Dynamic Threshold Simulation (DTS).

Hereafter, figure 6 shows the results of the DTS. However, in the rest of this subsection, results related to Kendall correlation tau will be omitted since we have found that its values remained quasi constant (98-99%) during the different simulations.

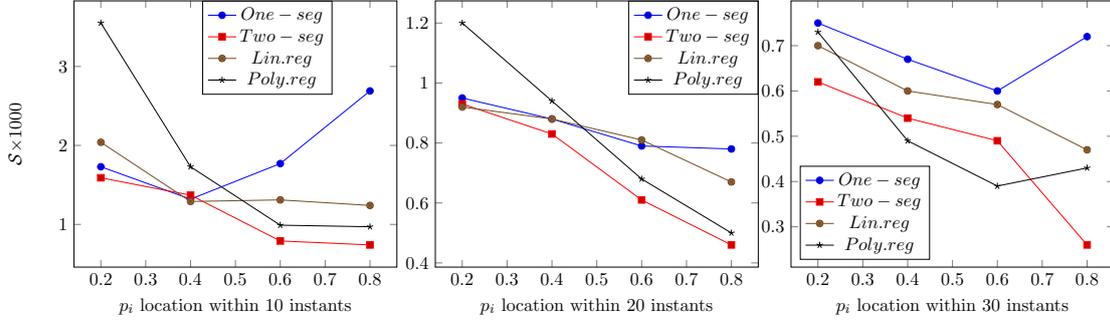


Figure 6: Average of fit quality for 4 interpolation techniques (uniform distribution)
 S is the sum of squared deviations between the 2 unicriterion sets.

According to figure 6, the 2-segments interpolation provides the best approximation to unicriterion net flow scores based on P_{real} . Furthermore, the approximation is better when the intermediate threshold p_i is located at $\frac{4}{5}$ of the time range (especially when the latter becomes larger). In order to confirm these results, we run the DTS by changing the random number generation process. This time a negative exponential distribution is used. In this case, λ varies randomly at each simulation such that $0, 1 \leq \lambda \leq 2$.

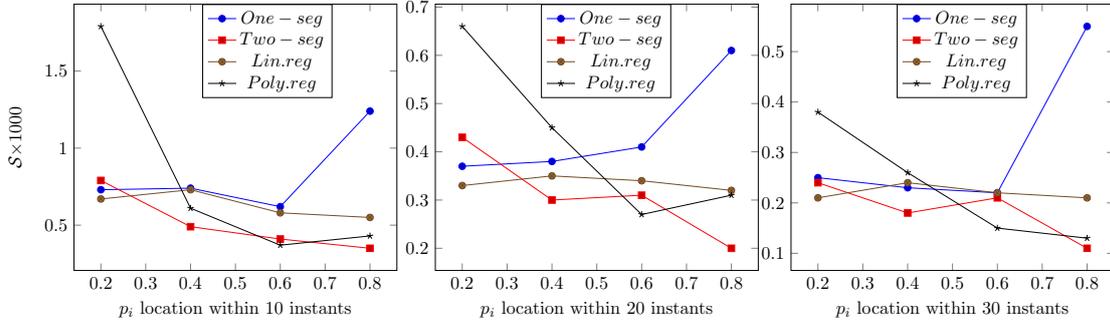


Figure 7: Average of fit quality for 4 interpolation techniques (negative exponential distribution)

As we can notice, figure 7 confirms the results seen in figure 6. To complete these results, we have applied the DTS in the case where the size of the alternatives set has been changed from 30 to 10, once again, the 2-segments interpolation holds its advantage. The related results will not be plotted.

In the continuity of this study, we investigate an eventual link between p_i locations and the time vector W . Thus, we will evaluate the best location of p_i according to different time weighting vectors W . We are going to run the DTS with two changes:

W will no longer be randomly generated and only the 2-segments technique will be considered. In fact, we apply these simulations to five different time weighting vectors W_1, W_2, \dots, W_5 whose slopes are, respectively, 0, 0.005, 0.01, 0.015, 0.02. The slope only relates to the non zero part of the components of each time weighting vector. Figure 8 gives more details about that:

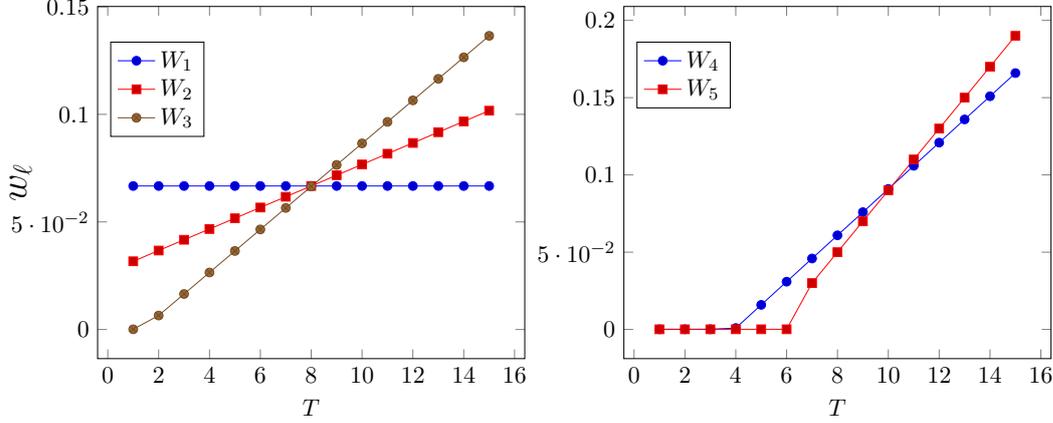


Figure 8: Five different time weighting vectors W_1, W_2, \dots, W_5 in the case of $s = 15$ instants (w_ℓ is the weight of instant ℓ)

Remember, for a given W the sum of its components is always equal to one. Thus, knowing the slope is sufficient to find all the corresponding components. Figure 9 stresses the behaviour of the 2-segments technique for each time weighting vector:

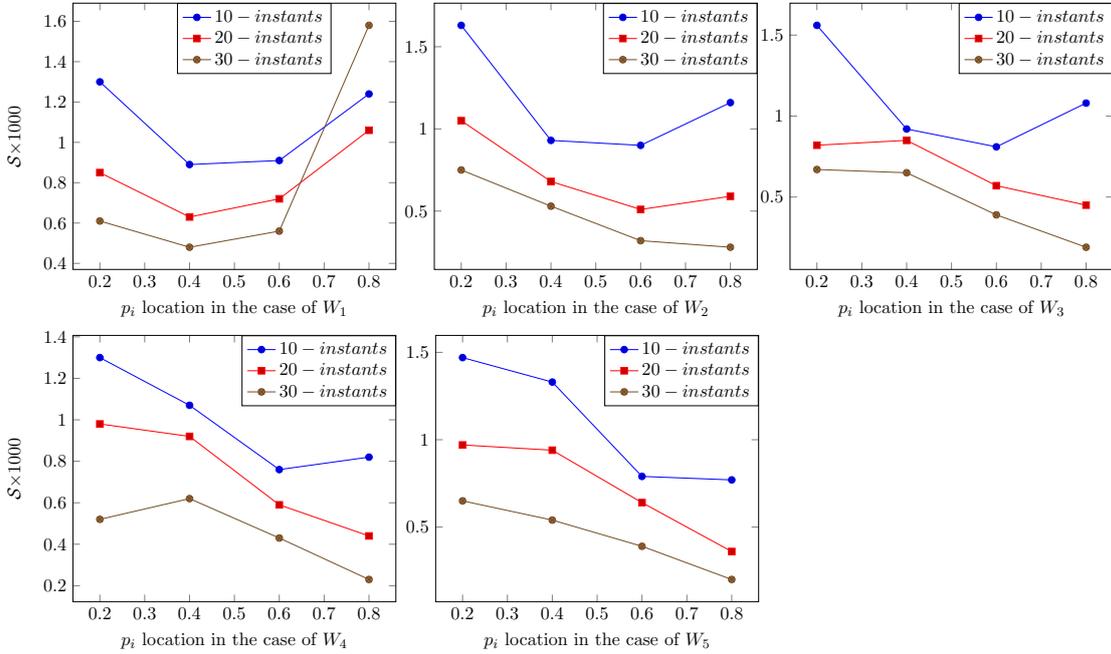


Figure 9: Average of fit quality of 2-segments technique for five different time weighting vectors (uniform distribution)

According to figure 9, for W_1 (equal time weights) the best location of p_i is $\frac{2}{5}$. For W_2 , $\frac{3}{5}$. For W_3, W_4 and W_5 the best location is $\frac{4}{5}$. We also note that $\frac{3}{5}$ would be bet-

ter while the size of instants becomes smaller (10 instants). Indeed, as the instants size gets larger the first instants gets less important within a time weighting vector W and vice-versa. That justifies the results differences between 10 instants and both 20 and 30 instants curves in figure 9. Results in the case of values generated with a negative exponential are shown in appendix A, they are concordant with those of figure 9.

As a conclusion to this subsection, we will select the 2-segments interpolation technique for the PROMETHEE temporal model. The intermediate threshold p_i will be set to around $\frac{4}{5}$. However, if later, at the time weighting W elicitation step we find out that the DM associates almost equal weights to W components, then we go back to the dynamic threshold preference elicitation step and set p_i to $\frac{2}{5}$. Of course, after that, there is no need to repeat the elicitation of W components, these are independent on dynamic threshold choices.

4.4 The temporal extension of PROMETHEE II

The PROMETHEE II method [6],[7] is applied at each instant t_ℓ . As a result we have:

$$f_h : \begin{array}{l} A \times T \rightarrow \mathfrak{R} \\ (a_i, t_\ell) \rightarrow f_h(a_i, t_\ell) \end{array}$$

Given a $f_h \in F$, the evaluation difference between $a_i, a_j \in A$ at instant $t_\ell \in T$ is defined as follows:

$$d_h(a_i, a_j, t_\ell) = f_h(a_i, t_\ell) - f_h(a_j, t_\ell) \quad (8)$$

Let us consider a preference function P_h associated to f_h denoted by:

$$P_h(a_i, a_j, t_\ell) = P_h[d_h(a_i, a_j, t_\ell)] \quad (9)$$

We have $\sum_{h=1}^k v_h = 1$ and $v_h > 0$. Then the instantaneous preference index is computed as follows:

$$\pi(a_i, a_j, t_\ell) = \sum_{h=1}^k P_h(a_i, a_j, t_\ell) \cdot v_h \quad (10)$$

We compute the instantaneous ingoing and outgoing flow scores associated to a_i :

$$\phi^+(a_i, t_\ell) = \frac{1}{n-1} \sum_{x \in A} \pi(a_i, x, t_\ell) \quad (11)$$

$$\phi^-(a_i, t_\ell) = \frac{1}{n-1} \sum_{x \in A} \pi(x, a_i, t_\ell) \quad (12)$$

Finally, the instantaneous net flow score of $a_i \in A$ for each $t_\ell \in T$ is given by:

$$\phi(a_i, t_\ell) = \phi^+(a_i, t_\ell) - \phi^-(a_i, t_\ell) \quad (13)$$

Each alternative from A is characterised by s instantaneous net flows scores at the output of the bloc II as illustrated in figure 10:

a_1	$\phi(a_1, t_1), \phi(a_1, t_2), \dots, \phi(a_1, t_s)$
a_2	$\phi(a_2, t_1), \phi(a_2, t_2), \dots, \phi(a_2, t_s)$
\vdots	$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$
a_n	$\phi(a_n, t_1), \phi(a_n, t_2), \dots, \phi(a_n, t_s)$

Figure 10: Succession of net flows during T for each alternative

Since we consider the temporal problem as a succession of s individual multicriteria problems, a weighted arithmetic mean is used in order to have the global ranking of all alternatives during T . One motivation for this choice is its additive property. Indeed, the weighted mean uses an additive aggregation as well as PROMETHEE II method. This allows to decompose the global score at the criterion level. Hence, the additive property makes the weighted mean operator consistent with PROMETHEE II.

The model aggregates instantaneous net flow scores as follows:

$$\phi_{A,T}(a_i) = \sum_{\ell=1}^s w_{\ell} \cdot \phi(a_i, t_{\ell}) \quad (14)$$

Where w_1, w_2, \dots, w_s denote the time weighting vector.

As already stated in section 3, v_h is supposed to remain static during T . However v_h is used to find $\phi(a_i, t_{\ell})$, the latter depends on t_{ℓ} . In addition, the global net flow score $\phi_{A,T}(a_i)$ requires both v_h and w_{ℓ} , such that :

$$\begin{aligned} \phi_{A,T}(a_i) &= \sum_{\ell=1}^s w_{\ell} \cdot \phi(a_i, t_{\ell}) \\ &= \sum_{\ell=1}^s w_{\ell} \cdot \frac{1}{n-1} (\sum_{x \in A} \pi(a_i, x, t_{\ell}) - \sum_{x \in A} \pi(x, a_i, t_{\ell})) \\ &= \frac{1}{n-1} \sum_{\ell=1}^s w_{\ell} \sum_{x \in A} (\sum_{h=1}^k P_h(a_i, x, t_{\ell}) \cdot v_h - \sum_{h=1}^k P_h(x, a_i, t_{\ell}) \cdot v_h) \\ &= \frac{1}{n-1} \sum_{\ell=1}^s w_{\ell} \sum_{x \in A} \cdot \sum_{h=1}^k v_h (P_h(a_i, x, t_{\ell}) - P_h(x, a_i, t_{\ell})) \end{aligned}$$

Finally, we have:

$$= \frac{1}{n-1} \sum_{\ell=1}^s \sum_{x \in A} \sum_{h=1}^k \cdot w_{\ell} \cdot v_h (P_h(a_i, x, t_{\ell}) - P_h(x, a_i, t_{\ell})) \quad (15)$$

Since w_{ℓ} depends on t_{ℓ} , the product $w_{\ell} \cdot v_h$ illustrates that the importance of criterion f_h is variable during T .

At this stage, one may wonder how the GAIA plan will look like if extended to temporal evaluations too. The following part is dedicated to this question.

4.5 The temporal GAIA plan

Once Bloc II has generated the successive net flow scores corresponding to each alternative, we are able to build a temporal GAIA plan in the same way as in the

classic version [7] of PROMETHEE. Since we have s instantaneous net flow scores, the GAIA plan construction is repeated s times. Then these plans are simply overlaid.

Figure 11 shows the temporal GAIA plan based on the footballer example presented in the introduction. The aim is to have a global ranking of players after a training period and with respect to three criteria A, B and C:

A- 30m sprint test: with average values between 4 and 5s [28].

B- Anaerobic Capacity : the body capacity to run without using O_2 (for a couple of minutes). It is important for high intensity and low duration activities. The corresponding average values are between 40 and 50 *Joules/kg* [29].

C- Peak Power test: the explosive power of lower limbs. The corresponding average values are between 14 and 17W/kg [30], [28], [31].

The corresponding evaluation table can be found in appendix B (table 5). Let us consider four players: P_1 , P_2 , P_3 and P_4 evaluated during a period of nine sessions of tests.

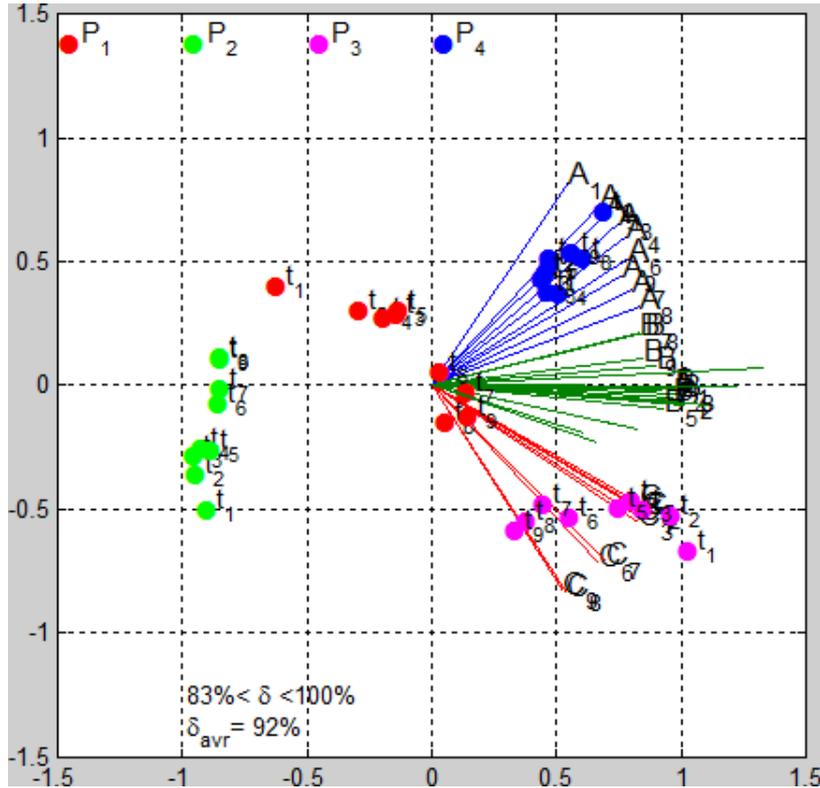


Figure 11: Temporal GAIA plan for the football players example

In figure 11, we observe that each criterion axis moves around a limited area during the nine instants (sessions). The instantaneous axes of B and the instantaneous decision axes are mixed. To facilitate the interpretation of this plan we only keep the criteria axes belonging to the last instant as shown in figure 12. Likewise, only

positions corresponding to P_1 and P_4 have been kept in order to stress out two different behaviours:

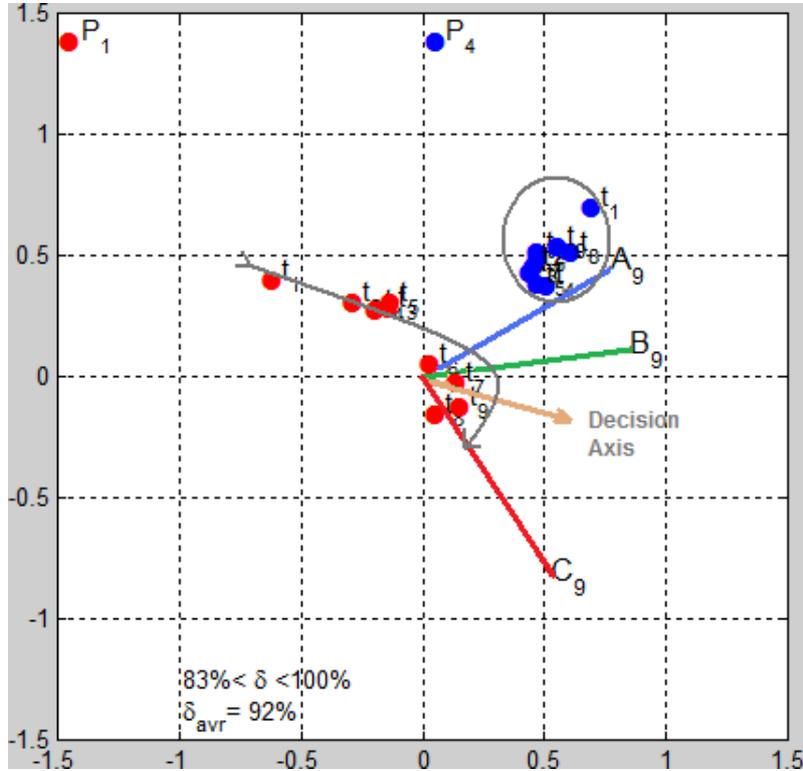


Figure 12: Temporal GAIA plan with last instant axes

In figure 12, we observe that:

- P_4 , on the right part of the figure, moves in a limited area. Thus he keeps stable scores with respect to the three criteria. He exhibits stable performances during the assessment period.
- P_1 , on the left part of the figure, improves his performances toward the decision axis, he gets also better scores with respect to criterion C .

Nonetheless, this example does not attempt to investigate all facets of the extension of the GAIA plan to the temporal evaluations. Indeed, many questions remain unanswered. For instance, under what conditions axes shift during T ? How pertinent is associating a relative importance to the GAIA plan of each instant t_ℓ ? How to interpret the evolution of δ_ℓ 's? ... These questions will be the topic of a future research.

4.6 Time weighting vector determination

In the context of additive temporal aggregation, several methods have been proposed in order to find the time weighting vector. For instance one can cite the maximum entropy method [11],[10], minimal variability method [12] or minimax disparity approach [13]. In all these methods the DM is asked to give an orness degree $\lambda \in [0, 1]$

which reflects his preferences over the period T . Instead of asking a technical value we would like to use here a preference elicitation method introduced in 2001 by Iyengar et al. [26]. A similar approach has been used by Eppe and De Smet [15] in order to find the criteria weights regarding the PROMETHEE II method.

The basic idea of this approach is transforming the weighting vector problem into a s -dimensional constrained polyhedron problem. The weighting vector W that we are looking for is assumed to be an unknown s -dimensional point $W = \{w_1, w_2, \dots, w_s\}$ in this polyhedron. So, the aim is to approach the coordinates (weights) of W . In practice, iteratively querying the DM through pairwise alternative comparisons leads to approach these weight values. So, at each query the DM gives his preference over two alternatives. The choice of this method is motivated by its ability to get closer to the DM's real preferences by an interactive way.

In the following, we illustrate the time weighting vector $W = \{w_1, w_2, w_3\}$ determination. We consider the determination of W in the context of a problem based on three dimensions as shown in the figure 13. We draw only w_1 and w_2 . w_3 is easily calculated since $\sum_{\ell=1}^3 w_\ell = 1$.

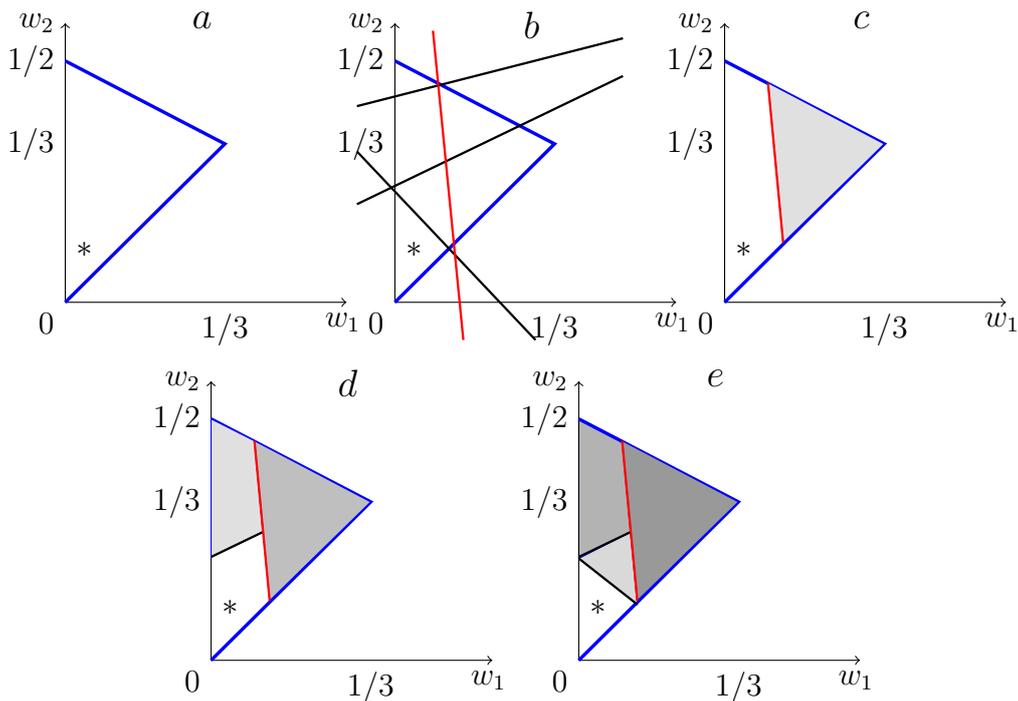


Figure 13: Reducing the area of possible time weighting vectors by querying

There are five triangles in figure 13, each of them represents one iteration. A triangle is the projection of the polyhedron on 2 dimensions (prior to querying the DM). The asterisk in each triangle represents the location of W to approach by the end of the querying process. At each step, the white area is the area of compatible weights regarding the previous queries. In the following figures 13.a, 13.b, 13.c, 13.d

and 13.e, the white area shrinks based on the DM's preference between two given alternatives. Each pair of alternatives is represented by a line in these figures.

Every query should have the optimum shrinking impact on the polyhedron. The DM is assumed to give his preference over two time series corresponding to two alternatives a_i and a_j . At this stage, we suppose that the DM's current answer is consistent with the previous ones. We assume also the fact that he is able to give an answer at each query.

In the next paragraphs, we describe the main steps of this approach which is based on the computation of the analytic center y^a ("a" for analytic center) of the current feasible region, the selection of the closest hyperplane to y^a , and the querying process. In order to better understand the flow of these steps, we are going, first, to see how to generate an optimum query.

First let us consider figure 13. Each line is the geometric equivalent of a query addressed to the DM. His answer (preference) determines the side of the line where lies W (white area). Now, consider figure 13.a. Prior to the DM's answer, we are not able to know the side of the triangle where W lies. The optimum way to deal with this uncertainty is to select the closest line to the center of the feasible region. This yields the minimisation of the feasible region whatever the answer will be. In the case of a s -dimensional polyhedron, we would select the closest hyperplane to this analytic center.

As stated by Bous et al. [16], a bounded polyhedron \mathcal{P} in \mathbb{R}^s defined by m linear inequalities can be defined such as:

$$\mathcal{P} = \{y \in \mathbb{R}^s : c - G.y \geq 0\} \quad (16)$$

Where $G \in \mathbb{R}^{m \times s}$, $m > s$ and $c \in \mathbb{R}^m$.

In [27], Ye has introduced the analytic center of a polyhedron \mathcal{P} as the point y^a of \mathcal{P}^0 that maximizes the value of the function $B(y^a)$:

$$B : \mathcal{P}^0 \rightarrow \mathcal{R} : \\ B(y^a) = \max \sum_{j=1}^m \ln(c_j - g_j.y^a) \quad (17)$$

Where $\mathcal{P}^0 = \{y \in \mathbb{R}^s : c - G.y > 0\}$ is the interior of \mathcal{P} .

The difference $c_j - g_j.y$ such as $j \leq m$ is a measure of the distance of y towards the boundary defined by the constraint $c_j > g_j.y$

Naturally, the analytic center of \mathcal{P} corresponds to the point y of \mathcal{P}^0 that maximizes the sum of distances between y and the boundaries of \mathcal{P} . In addition, the choice of a logarithmic function results from the high penalty around the boundaries.

To make this idea clearer, let us consider a square S of length of 1 where lies a point $M = (y_1, y_2) \in S$. Since $0 \leq y_1 \leq 1$ and $0 \leq y_2 \leq 1$, M is thus constrained by 4 boundaries and equation (16) can be written such as:

$$B(M) = \ln(y_1) + \ln(1 - y_1) + \ln(y_2) + \ln(1 - y_2) \quad (18)$$

Figure 14 shows the evolution of $B(M)$ according to the coordinates of M . We clearly observe that $B(M)$ hugely decreases as the point M gets closer to one of the four boundaries. We can also observe that the maximum of equation (16) is reached around the square center.

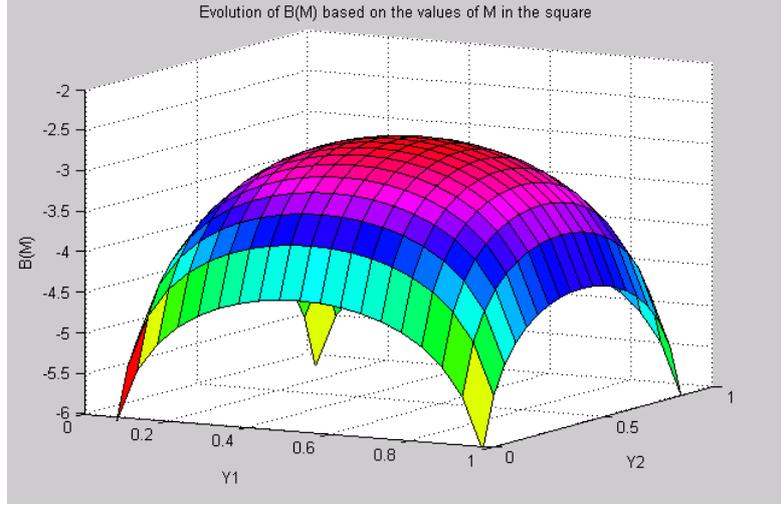


Figure 14: Evolution of $B(M)$ based on M coordinates

At this stage, using an appropriate algorithm (next section), we are able to find the analytic center y^a of the feasible region at iteration k . Before determining the closest hyperplane to y^a , the possible hyperplanes have to be listed.

As already stressed, a time series $\Phi(a_i)$ of an alternative a_i is composed of s instantaneous net flow scores $\phi(a_i, t_1), \dots, \phi(a_i, t_s)$. In fact, we assume that the DM does not know precisely the importance (weight) of each instant of the period T , but is able to evaluate a time series as a whole. For example, if the DM states that $\Phi(a_i)$ is preferred to $\Phi(a_j)$ (we can write for convenience, respectively, a_i and a_j), it means that:

$$\phi(a_i, t_1).w_1 + \dots + \phi(a_i, t_s).w_s > \phi(a_j, t_1).w_1 + \dots + \phi(a_j, t_s).w_s \quad (19)$$

we have: $(\phi(a_i, t_1) - \phi(a_j, t_1)).w_1 + \dots + (\phi(a_i, t_s) - \phi(a_j, t_s)).w_s > 0 \quad (20)$

Where w_1, w_2, \dots, w_s represent the weights of the different instants. These weights are unknown, and the terms $(\phi(a_i, t_\ell) - \phi(a_j, t_\ell))$ are constant and known. Geometrically, inequation (19) corresponds to the side of the hyperplane -whose equation is (20)- where the point $W = \{w_1, w_2, \dots, w_s\}$ lies.

$$(\phi(a_i, t_1) - \phi(a_j, t_1)).w_1 + \dots + (\phi(a_i, t_s) - \phi(a_j, t_s)).w_s = 0 \quad (21)$$

Equation (20) is the equation of the hyperplane denoted by H_{a_i, a_j} , formed by a_i and a_j and passing through the origin. Let us note that this equation also represents the situation where the DM is indifferent between a_i and a_j .

At this level, it is easy to find all possible hyperplanes by computing the differences within each pair of time series a_i and a_j . By doing so, we have $\frac{n^2-n}{2}$ different hyperplanes. After having listed all possible hyperplanes, we shall compute the distance $\mathcal{D}_{((a_i, a_j), y^a)}$ between each H_{a_i, a_j} and the analytic center y^a whose coordinates are $y_1^a, y_2^a, \dots, y_s^a$ at iteration k such as:

$$\mathcal{D}_{((a_i, a_j), y^a)} = \frac{|[\phi(a_i, t_1) - \phi(a_j, t_1)].y_1^a + \dots + [\phi(a_i, t_s) - \phi(a_j, t_s)].y_s^a|}{\sqrt{[\phi(a_i, t_1) - \phi(a_j, t_1)]^2 + \dots + [\phi(a_i, t_s) - \phi(a_j, t_s)]^2}} \quad (22)$$

Equation (21) is a generalisation of the distance in 3-dimensions of a point $P = (x_p, y_p, z_p)$ toward a plan π passing by the origin and whose Cartesian equation is $a.x + b.y + c.z = 0$ such as:

$$\mathcal{D}_{(\pi, P)} = \frac{|a.x_p + b.y_p + c.z_p|}{\sqrt{a^2 + b^2 + c^2}} \quad (23)$$

Thus, the selected hyperplane will be the one that minimizes $\mathcal{D}_{((a_i, a_j), y^a)}$. Finally, we ask the DM to give his preference over the two time series a_i and a_j whose hyperplane distance towards y^a presents the minimum value. At each iteration, the DM preference takes into account the whole period of T . The DM has, in fact, three possible answers:

$$\begin{aligned} a_i \succ a_j & \quad \text{means that } a_i \text{ is preferred to } a_j \\ a_i \prec a_j & \quad \text{means that } a_j \text{ is preferred to } a_i \\ a_i \equiv a_j & \quad \text{means that } a_i \text{ and } a_j \text{ are equivalent} \end{aligned}$$

From a geometrical point of view, adding a constraint in this way tends to cut the polyhedron in two almost equal parts. Then another iteration is started with the same operations but only based on the remaining region. The stopping condition of the iterative process has been set to a maximum number of iterations. Once this number is reached, the coordinates of the last computed analytic center are considered to be the best approximations of w_1, w_2, \dots, w_s .

5 Simulations about W determination

Of course, the determination process by querying should remain also feasible. This means getting the desired weighting vector in a reasonable number of queries addressed to the DM. To investigate this question, we proceed through simulations (see Algorithm I).

Algorithm I: Iterative querying process

- 1) **Input:** $n, k, s, q_h, p_h, f_h(a_i, t_\ell), W^*, iter_{max}$
 - 2) **For** $i = 1 \dots s$ **do**
 - Compute the i^{th} instantaneous net flow scores for n alternatives;**end**
 - 3) Initialize the time weighting constraints C_0 ;
 - 4) Compute all possible hyperplanes;
 - 5) Compute the analytic center y_0 of the polyhedron constrained by C_0 ;
 - 6) $C_{it} \longleftarrow C_0$;
 - 7) $y_{it} \longleftarrow y_0$;
 - 8) **For** $j = 1 \dots iter_{max}$ **do**
 - a) Find the closest hyperplane hyp to the last y_{it} ;
 - b) Ask the DM's preference C_{pr} about hyp ;
 - c) Update the constraints set $C_{it} = C_{it} \cup C_{pr}$;
 - d) Compute the analytic center y_{it} of the new constrained polyhedron;**end**
 - 9) $W_{estim} \longleftarrow y_{it}^T$;
-

Hereafter, we give more details about the algorithm:

1) We begin by creating a temporal MCDA problem such that we consider, among others:

- k criteria to be maximised and having equal weights. This choice is arbitrary. Indeed, the aim at this stage is only to create a problem that is able to generate instantaneous net flow scores for each alternative. This is required in the following steps.

- Each evaluation $f_h(a_i, t_\ell)$ is randomly generated by using a uniform distribution. For this purpose, we use the function $randint(1, s, [1, 10])$ (seen in subsection 4.3) to generate, for every criterion, a set of s values each one is between 1 and 10. Then, as shown in Algorithm II in appendix C, these evaluations are multiplied by a random value between 0.1 and 1. Figure 20 in appendix C shows 2 types of random evaluations generated with and without the multiplication. Thus, multiplying each set of these evaluations by a random value tends to differentiate alternatives, hence, makes the problem more credible.

- A static preference threshold has been chosen for this problem such as $p = 7$. Of course p can be lower or higher than 7, but the higher p the lower discrimination of alternatives. Since evaluations are between 1 and 10, $p = 7$ seems to be a reasonable preference threshold.

- W^* is a time weighting vector satisfying the time weighting constraints C_0 (section 3). In order to generate the weights $w_1^*, w_2^*, \dots, w_s^*$, we create a straight line whose

equation is $f(x) = \alpha \cdot x + \beta$ such as $0 \leq \alpha \leq 1/(s-1)$ and $\beta = 0.5 - \alpha \cdot (s/2)$. Thus the slope α is generated such as the straight line is between L_{min} and L_{max} (see figure 15):

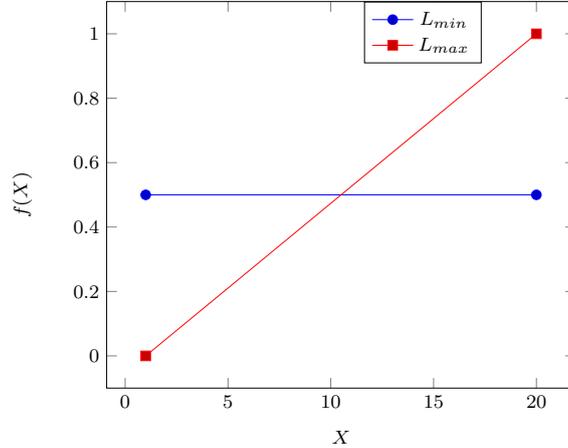


Figure 15: The minimum and maximum slopes for 20 instants

Then we uniformly reduce the values of $f(1), f(2), \dots, f(s)$ until the sum $f(1) + f(2) + \dots + f(s)$ is equal to $1 \pm \epsilon$. Since technically it is difficult to reach 1.000, we add $\epsilon = 0.001$ to it. Once this condition is met $f(1), f(2), \dots, f(s)$ are considered therefore to be respectively equal to w_1, w_2, \dots, w_s . Algorithm III in appendix C gives more details about this procedure. The range of the slope evolution has been chosen in order to cover a large set of weights distributions. The set $W = \{w_1, w_2, \dots, w_s\}$ is supposed to be the DM's time weighting vector and thus we should converge to it by the end of the algorithm.

Steps 2), 3) and 4) are already cited in the previous section.

Step 5) As introduced in the previous section, we compute the first analytic center y_0 of the s -dimensional polyhedron constrained by \mathcal{C}_0 . For this purpose, we use Frank-Wolfe algorithm [37] which finds the minimum of a non linear function $f(x)$ under the constraints of a domain D . Algorithm IV shows the main steps of Frank-Wolfe algorithm:

Algorithm IV: Frank-Wolfe algorithm [37]

- A) **Input:** A starting time weighting vector y_0 and $k = 0$
- B) **While** $\|\nabla f(y_k)\|_2 \geq \epsilon$ **and** $k \leq iter_{max}$ **do**
- a) find r_k that minimises $y^T \nabla f(y_k) \quad \forall y \in D$;
 - b) Let $\gamma = \frac{2}{k+2}$ or alternatively find γ that minimises $f(y_k + \gamma(r_k - y_k))$ subject to $0 \leq \gamma \leq 1$;
 - c) Update $y^{(k+1)} = y_k + \gamma(r_k - y_k)$;
 - d) $k = k + 1$;
- end**
-

In step *A.a*, the starting time weighting vector is a point y_0 in the s -dimensional polyhedron \mathcal{P} such as y_0 is far from the boundaries because of the logarithmic function penalty (subsection 4.6). One method we can use to find y_0 is the center of Chebyshev [38]. It is the center of the largest inscribed ball in a polyhedron. Beside the fact that this center is far from the polyhedron boundaries, this method requires simply to solve a linear programming. Chebyshev center algorithm is given in appendix C.

Step *B.a* of algorithm IV is simply a minimisation of the gradient of $f(y_k)$ which is nothing else than a sum of s products of w_ℓ and a constant. A linear programming algorithm can be used. In our case, we use the Simplex algorithm.

In step *B.c*, the value $(z_k - y_k)$ acts as the minimisation direction. We have considered $\epsilon = 0.1$ and $iter_{max} = 350$. Steps 6), 7), 8), 9) and 10) of algorithm I are supposed to be clear.

In order to compute statistics, simulations have been repeated 100 times for each pair of (A, T) (alternatives and instants sets).

The value of $iter_{max}$ can be adapted to the size of both alternatives and instants. According to table 1, a few queries ($iter_{max} < 10$) are, most of the time, sufficient to reach a Kendall correlation rate greater than 95%. The higher $iter_{max}$ the better the convergence rate.

	Instants					
Alt.*	5	10	15	20	25	30
5	2.3 +- 1.8	2.3 +- 1.9	2.4 +- 2.1	2.7 +- 2.2	3.2 +- 2.7	2.9 +- 2.3
10	2.4 +- 1.8	2.7 +- 2.2	3.4 +- 3.2	3.7+- 3.2	4.2 +- 3.7	4.3 +- 4.1
15	2.4 +- 1.9	3.5 +- 2.8	4.5 +- 3.5	5.2 +- 4.5	5.3 +- 4.0	5.3 +- 3.9
20	2.7 +- 1.9	3.9 +- 3.2	4.3 +- 3.6	5.2 +- 3.8	6.4+- 4.3	7.0 +- 4.7
25	2.7 +- 2.0	3.9 +- 2.9	5.6 +- 4.1	6.7 +- 5.1	6.9 +- 5.2	7.0 +- 5.0
30	3.0 +- 1.9	4.2 +- 2.9	5.9 +- 4.0	6.6 +- 4.8	7.4 +- 4.5	7.7 +- 5.3

Table 1: Average number of queries needed for a 95% convergence rate and corresponding standard deviations (in gray).

*: Alternatives

These results are obtained using *MATLAB*, version 7.5.0.342 (*R2007b*) on a *Pentium Dual – Core T2310 – 1.47GHz – 32bits* computer with a 1014Mo RAM.

Besides, the computation of the analytic center is the task which takes the most time among all other tasks of algorithm I. Based on 100 simulations, figure 16 gives an overview on the average of querying process duration based on the size of both alternatives and instants sets. This does not consider the reaction time of the DM:

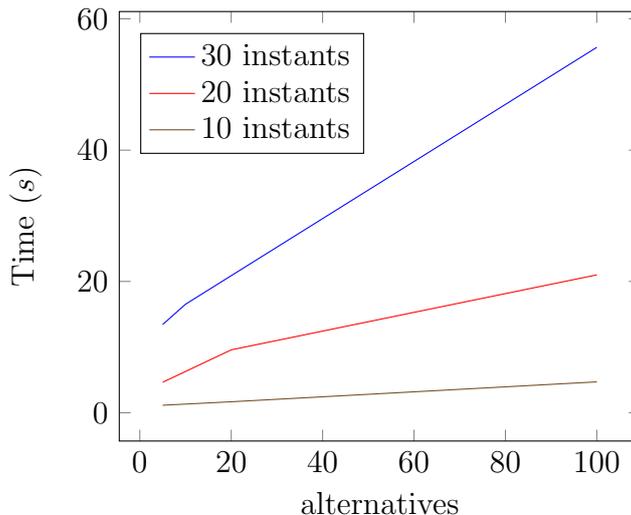


Figure 16: Average duration of the querying process based on (A, T) ,

At this stage, one may wonder whether the approach presented in this paper is the same as the one introduced by Eppe et al. in [15]. This point deserves to be clarified. Indeed, Eppe et al. [15] have developed an iterative querying approach close to the one of Iyengar et al. [26] in order to find the criteria weights in the case of PROMETHEE II. The added value of their method relies on three different types of queries. The best one is identified according to its expected impact on the reduction of the admissible weights area [15].

The model we propose involves a methodical difference with the one of Eppe et al. This concerns the increasing weights constraint where $w_1 \leq w_2 \leq \dots \leq w_\ell \leq \dots \leq w_s$. Moreover, we have used Frank-Wolfe algorithm instead of Newton's algorithm used by Eppe et al. Indeed, Frank-Wolfe replaces a differentiable function $f(y)$ with its first-order Taylor expansion. However, Newton's method requires the second-order Taylor expansion. This implies the computation of the inverse of the Hessian matrix (second derivative) at each iteration. This is prohibitively heavy in high dimensions [39]. In fact, the Hessian may become close to a singular matrix (non invertible) which induces computation errors [39]. As a result, since the PROMETHEE temporal model involves working in high dimensions ($s = 10, 20, 30, \dots$) the Hessian computation may be unstable. The interested reader can refer to [40] for much more information.

By using Frank-Wolfe method to compute the analytic center, we managed to satisfy equation 16 (in 4.6) with, by far, less computation problems than by using Newton's method.

In addition, for the sake of simplicity, the query in the model we propose is restricted to two alternatives.

6 Illustration

We consider the global ranking of ten emerging economies with respect to the Human Development Index (HDI). This index assumes that economic growth should not be the unique criterion while evaluating people well-being in a given country. Indeed, the HDI is an aggregated measure related to three basic aspects: health, education and standard of living [33]. They are assessed as follows:

- Life expectancy at birth

This criterion is defined in the United Nations Development Programme (UNDP) 2010 Report [34] as “the number of years a new-born infant could expect to live if prevailing patterns of age-specific mortality rates at the time of birth were to stay the same throughout the child’s life”.

- Education

This criterion is measured by two indicators which are defined in UNDP Website ([41], [42]) as:

-“Average number of years of education received by people aged of 25 and older, converted from education attainment levels using official durations of each level”.

-“Number of years of schooling that a child of school entrance age can expect to receive if prevailing patterns of age-specific enrolment rates persist throughout the child’s life”.

The arithmetic mean of these two indicators gives the score on education criterion.

- Gross National Income per capita

This UNDP 2010 Report [34] defines this criterion as “the sum of value added by all resident producers in the economy plus any product taxes (less subsidies) not included in the valuation of output plus net receipts of primary income (compensation of employees and property income) from abroad, divided by midyear population”.

The global scores of the HDI are annually published by UNDP [33]. We draw the attention of the reader to the fact that UNDP uses the geometric mean aggregation to compute those scores. More information is given in [32].

The data we use has been extracted from the Human Development Reports of 2015 (ex. [35]). The evaluations frequency is 5 years and covers the period between 1990 and 2015. Based on the evaluations of 2010, 2011, 2013, 2014, values corresponding to 2015 have been extrapolated by linear regression.

The instantaneous evaluations of all countries over the set F are given in appendix B (table 5). Each country is evaluated 6 times over each criterion. We consider:

$$A = \{ \text{Brazil, China, India, Indonesia, Malaysia, Mexico, Philippines, Russia,} \\ \text{South Africa, Turkey} \}$$
$$F = \{ \text{Life expectancy at birth, Education, Gross National Income per Capita} \}$$

$$T = \{t_{1990}, t_{1995}, \dots, t_{2015}\}$$

For convenience, we denote by C_1, C_2 and C_3 , respectively, the criteria of the set F . They are to be maximized. Their weights are chosen equal. The evaluations over C_1 and C_2 are expressed in years whereas those over C_3 are expressed in constant 2011 international dollars converted using purchasing power parity (PPP) rates [35].

A preference function of type III has been associated to each criterion. Tables 2,3 and 4 represent the evaluations disparity variation over T for, respectively, C_1, C_2 and C_3 . This helps us to answer the question whether p will be dynamic or static:

-Life expectancy at birth C_1 :

For this criterion we can propose that p has to be an increasing threshold. This is justified by the fact that it is easier to live longer in 2015 than in 1990, furthermore, table 2 shows almost a growing disparity in average during T :

	t_{1990}	t_{1995}	t_{2000}	t_{2005}	t_{2010}	t_{2015}
Evaluations Sd	3.860	3.818	5.222	6.624	6.087	5.480

Table 2: Standard deviation of evaluations for C_1

According to 2-segments technique, we need three values of dynamic thresholds, the intermediate one p_i should be around $3/5$ of T as seen in subsection 4.3 (case of 10 instants). Let us consider then: $p_1 = 7, p_i = 8$ and $p_s = 10$. In fact, we consider that 7 years are sufficient to make a significant difference between two life expectancies in 1990. We rise this threshold to 10 for 2015. Of course, other choices are possible.

- Education C_2 :

We can propose a decreasing threshold. In fact, since the disparity of the evaluations are slightly declining, a decreasing p will tend to make countries discrimination more perceptible by the last instants:

	t_{1990}	t_{1995}	t_{2000}	t_{2005}	t_{2010}	t_{2015}
Evaluations Sd	1.468	1.496	1.495	1.341	1.231	1.190

Table 3: Standard deviation of evaluations for C_2

Therefore, we need three values of dynamic thresholds. We propose: $p_1 = 3, p_i = 2.6$ (around $3/5$ of T) and $p_s = 2.2$. Here, we consider 3 as sufficient to make a significant difference between 2 countries in 1990 on education criterion. Since the countries become less disparate on this criterion, a decreasing threshold such as $p_s = 2.2$ seems to be suitable for 2015 (table 6 appendix B).

- Gross National Income per capita C_3 :

As stated previously, evaluations over C_3 are expressed in constant 2011 international dollars. This means no need to consider inflation rate during T . Table 4 shows a dropping evaluations disparity from t_{1995} followed by a slow recovery. Hence, there is no specific trend over T .

	t_{1990}	t_{1995}	t_{2000}	t_{2005}	t_{2010}	t_{2015}
Evaluations Sd	5244.8	4093.8	4410.2	4979.2	5194.7	5538.6

Table 4: Standard deviation of evaluations C_3

For this criterion, we shall rather keep a static preference threshold. Let us consider $p = 10000$. According to the corresponding evaluations in table 6 appendix B, a preference threshold of 10000 is a reasonable value. Once again, other choices are possible.

Taking into account the corresponding evaluations (table 6 in appendix B) and given a time weighting vector $W_5 = \{0.1168, 0.1368, 0.1568, 0.1768, 0.1968, 0.2168\}$ (seen in subsection 4.3), we applied the temporal extension of PROMETHEE. Hereafter the results of the global ranking of the countries of A with respect to the HDI after assessment during the period 1990 – 2015:

- 1-Russia
- 2-Malaysia
- 3-Mexico
- 4-Brazil
- 5-Turkey
- 6-China
- 7-South Africa
- 8-Philippines
- 9-Indonesia
- 10-India

Due to the small size of this example (10 alternatives over 6 instants), one query has been sufficient to reach 95% of Kendall correlation tau (section 5). This is consistent with simulation results in table 1. Yet, to reach 99% of correlation tau 3 iterations have been required.

Since this overall ranking represents an aggregated information over T , one may need to visualise the evolution of these countries during the same period. Figure 17 shows the temporal GAIA plan corresponding to this problem, only the criteria axes belonging to the last instant (t_{2015}) are kept:

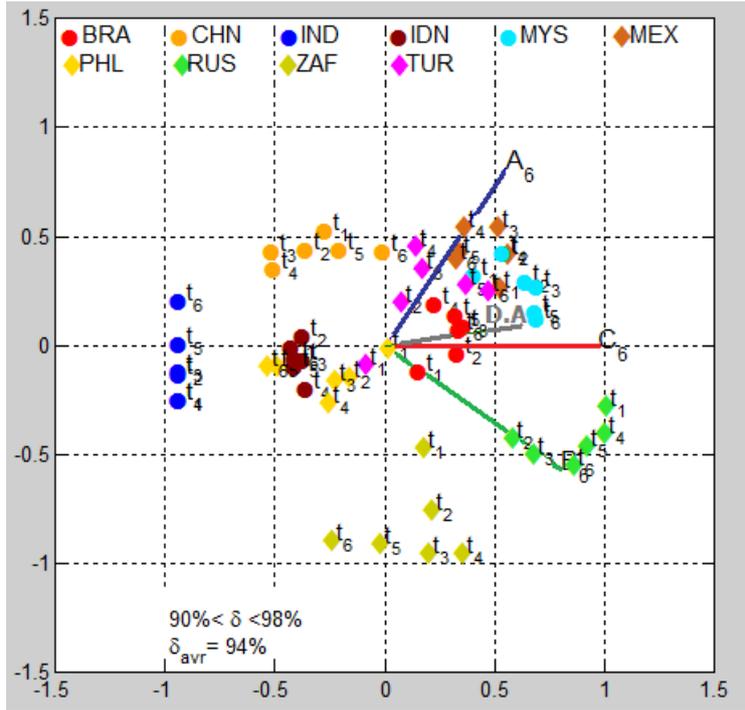


Figure 17: Temporal GAIA Plan of HDI problem (19 countries)

The shortest axis (D.A) in this figure represents the decision axis. In figure 18 we stress out the evolving of China and Turkey over T . In addition, we observe the stable behaviour of India during the same period. This plan also confirms the leading rank of Russia.

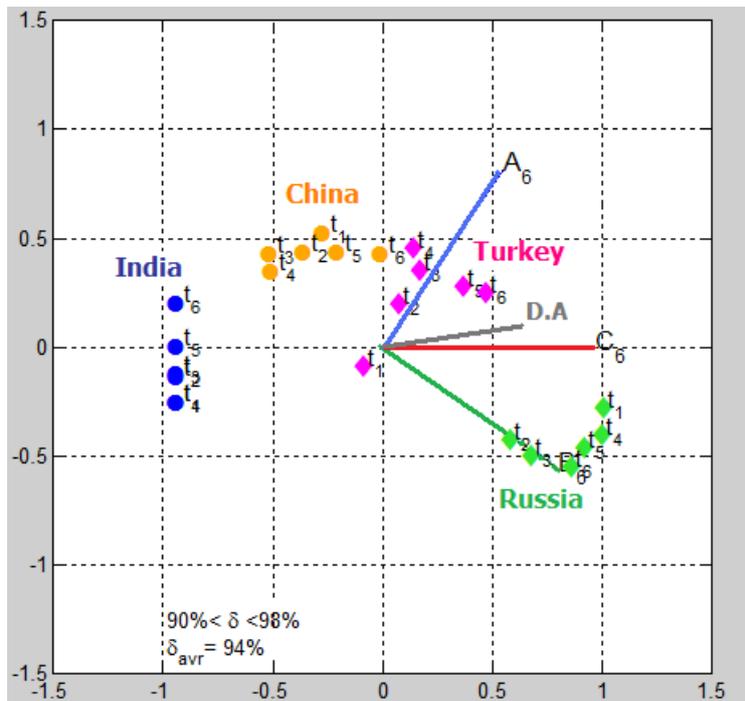


Figure 18: Temporal GAIA Plan of HDI problem (19 countries)

According to this figure, Turkey and China are also likely to evolve in the future in order to improve their respective ranks with respect to HDI.

Based on these results, the temporal model proposed in this paper provides consistent outputs. However, we shall not discuss modelling issues into details as the aim of this section is to illustrate the application of the method.

7 Conclusion and directions for future research

In this paper we have introduced an extension of PROMETHEE II to temporal evaluations. This includes two contributions related to the setting of dynamic thresholds and the time weighting vector elicitation. In addition to the effectiveness of the 2-segment technique, simulations have shown that, in most cases, a few queries addressed to the DM were sufficient to set instants weights.

Moreover, the temporal extension of GAIA allows to monitor the instantaneous behaviour of all alternatives during the evaluation period. Yet, as already stressed, this deserves further research.

With regard to whether the temporal extension of PROMETHEE II can be applied to other MCDA methods, the answer would be nuanced. Indeed, except for the time weighting vector elicitation, we note that the setting of dynamic preference (and/or indifference) thresholds could only be applied to the methods allowing this type of thresholds (ex. ELECTRE methods). Additionally, since GAIA plan is based on the Principle Components Analysis technique, which can be applied to each multi-criteria problem, a temporal extension of GAIA could also be transposable in other MCDA methods.

Let us note that the time weighting vector elicitation can be alleviated by asking the DM to choose the best instants weights evolution among, for instance, ten possible shapes. Besides, one may investigate the relevance of approximating these weights by an interpolation technique such as 2-segments technique.

The proposed model is intended to be suitable for a wide range of temporal MCDA problems. However, further researches are needed to adapt it to particular cases such as evaluations under uncertainty, asynchronous evaluations, dynamic alternatives,...

Finally, it is worth nothing that forecasting the alternatives behaviours is an interesting future research direction.

Appendix A

Algorithm VI: Generating random uniformly distributed evaluations
with an increasing standard deviation

- 1) **Input:** n, s
 - 2) **For:** $\ell = 1 : s$
 $Eval(\ell, 1 : n) = randint(1, n, [1, 10 + \ell])$;
end
-

In order to generate random evaluations with an increasing disparity over T for n alternatives, we use *randint()* *Matlab* function (uniform distribution) as given in algorithm VI above.

The principle is that $\forall a_i \in A$ at instant $t_\ell \in T$, *randint*(1, n , $[1, 10 + \ell]$) generates a vector of n values within the range $[1, 10 + \ell]$. Thus, the higher ℓ , the higher the evaluations disparity.

In the following, algorithm VII shows how to randomly generate the values of P_{real} according to a uniform distribution.

Algorithm VII: Generating a random set of P_{real} by a uniform distribution.

- 1) **Input:** $n, s, Eval$
 - 2) $maxEval =$ the largest evaluation at the last instant;
 - 3) $minEval =$ the smallest evaluation at the last instant;
 - 4) $slack = maxEval - minEval$;
 - 5) $P_{real}(1, 1) = 3$;
 - 6) **For:** $\ell = 2 : s$
 - 7) **If:** $P_{real}(1, \ell - 1) + 2 < slack$
 - a) $P_{real}(1, \ell) = randint(1, 1, [P_{real}(1, \ell - 1), P_{real}(1, \ell - 1) + 2])$;
 - b) **else** $P_{real}(1, \ell) = slack$;
 - end**
-

In steps 2) and 3) evaluations are considered to be the most disparate at the last instant.

In step 5) we have $p_1 = 3$, this choice is random. Since the preference threshold is assumed to increase over T , p_1 should be set as smaller (with respect to *slack*) as possible.

A consequence of step 7)a) is that $P_{real}(1, \ell - 1) \leq P_{real}(1, \ell) \leq P_{real}(1, \ell - 1) + 2$.

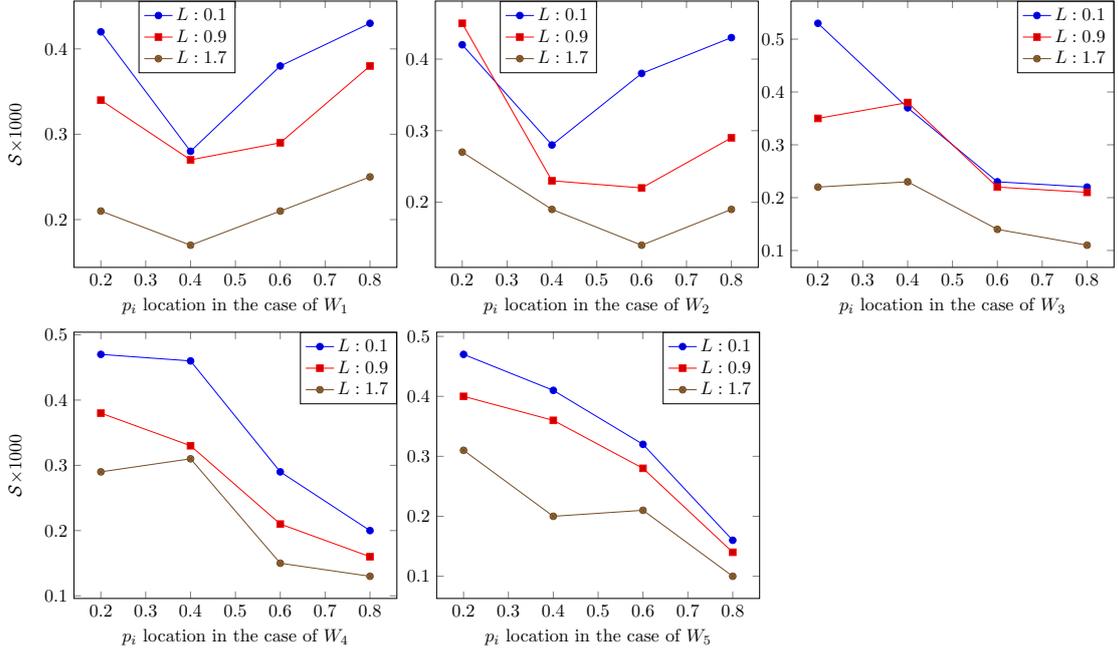


Figure 19: Average of fit quality of 2-segments technique for five different time weighting vectors and different values of L (λ) at 20 instants (negative exponential distribution)

If we consider the cumulative distribution function according to a truncated negative exponential distribution:

$$F(x) = \frac{e^{-\lambda a} - e^{-\lambda x}}{e^{-\lambda a} - e^{-\lambda b}} \quad (24)$$

Then, we can find the distribution x by setting $F(x)$ equal to a random uniformly distributed value $U \in [0, 1]$:

$$x = \frac{-\ln(e^{-\lambda a} - (e^{-\lambda a} - e^{-\lambda b}) \times U)}{\lambda} \quad (25)$$

This is the equation we used to generate a random value within $[a, b]$ by a negative exponential distribution.

In order to generate both $Eval$ and P_{real} by a negative exponential distribution, we use equation (24) above instead of $randint()$ function in algorithm VI and VII.

Appendix B

Criteria	30m Sprint									Anaerobic Capacity									Peak Power								
	t1	t2	t3	t4	t5	t6	t7	t8	t9	t1	t2	t3	t4	t5	t6	t7	t8	t9	t1	t2	t3	t4	t5	t6	t7	t8	t9
Weights	0,3																										
Instants	0,4																										
Max +1 Min -1	1																										
Pref. Function	III																										
p Threshold	1,139	1,134	1,154	1,186	1,113	1,016	0,905	0,83	0,7	8,283	5,529	6,696	6,846	7,126	6,846	6,529	6,056	6	0,671	0,602	0,627	0,68	0,746	0,845	0,904	0,916	1
q Threshold	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P1	5,5	5,4	5,2	5	4,8	4,7	4,5	4,5	4,4	38	44	45	45	46	48	48	49	50	13,6	13,9	14,1	14,4	14,7	15,1	15,4	15,7	16,3
P2	6,3	6,2	6,1	6	5,8	5,6	5,4	5,2	5	39	41	40	42	44	45	46	48	48	13,7	13,9	14	14,1	14,2	14,3	14,3	14,4	14,6
P3	5,7	5,5	5,3	5,1	5	4,9	4,8	4,8	4,8	49	50	51	52	54	54	55	55	15,2	15,3	15,5	15,8	16,1	16,4	16,6	16,7	17	
P4	5	4,9	4,8	4,7	4,6	4,5	4,5	4,3	4,3	48	47	48	51	53	55	56	57	57	14	14,3	14,4	14,6	14,7	14,6	14,8	14,9	15,1
Std. Deviation	4,657	4,637	4,717	4,848	4,548	4,153	3,7	3,391	2,861	5,025	3,354	4,062	4,153	4,323	4,153	3,961	3,674	3,64	6,379	5,723	5,958	6,457	7,084	8,031	8,584	8,7	9,5

Table 5: Evaluations Table for Football example cited in the introduction

Criteria	Life Expectancy at birth					Education					Gross National Income per Capita							
	1990	1995	2000	2010	2015	1990	1995	2000	2005	2010	2015	1990	1995	2000	2005	2010	2015	
Weights	0,333																	
Years	1																	
Max: +1 Min: -1	III																	
Pref. Function	III																	
p Threshold	7	7,333	7,66	8	9	10	3	2,87	2,74	2,6	2,4	2,2	10000	10000	10000	10000	10000	10000
q Threshold	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Brazil	65,3	67,6	70,1	71,9	73,3	74,8	8	8,95	9,95	10,15	11,05	11,6	10065	10959	11161	12032	14420	15062
China	69	69,9	71,7	73,7	75	76	6,8	7,25	7,85	8,75	9,85	10,3	1520	2508	3632	5632	9387	13347
India	57,9	60,4	62,6	64,5	66,5	68,4	5,35	5,9	6,45	7,35	8,25	8,55	1754	2046	2522	3239	4499	5814
Indonesia	63,3	65	66,3	67,2	68,1	69,1	6,75	7,2	8,7	9,3	9,95	10,3	4337	5930	5308	6547	8267	10130
Malaysia	70,7	71,8	72,8	73,6	74,1	74,8	8,1	8,9	10,25	10,15	11,35	9,72	13439	14500	17157	19725	23712	
Mexico	70,8	72,8	74,4	75,3	76,1	77	8,05	8,55	9,15	9,85	10,5	10,8	12074	12028	14388	14693	15395	16249
Philippines	65,3	66,1	66,7	67,2	67,7	68,3	8,7	8,95	9,5	9,75	9,75	10,2	3962	4111	4994	6058	7478	8232
Russia	68	66	65,1	65,8	68,6	70,3	10,95	10,85	11,85	12,6	13,1	13,35	19461	12011	12933	17797	21075	22094
South Africa	62,1	61,4	55,9	51,6	54,5	57,9	8,95	10,65	11	11,15	11,55	11,75	9987	9566	9719	10935	11833	12110
Turkey	64,3	67	70	72,5	74,2	75,6	6,7	7,2	8,3	8,95	10,55	11,05	10494	11317	12807	14987	16506	18976
Std. Deviation	3,86	3,81	5,22	6,62	6,08	5,48	1,46	1,49	1,49	1,34	1,23	1,19	5245	4094	4410	4979	5195	5539

Table 6: Evaluations Table for Emergent Economy problem in illustration section

Appendix C

Let us consider E as the set of evaluations of an alternative a_i on criterion f_h during s instants:

Algorithm II: Generating a random set of E (uniform distribution)

- 1) **Input:** s
 - 2) $E = \text{randint}(1, s, [1, 10]) \times \frac{\text{randint}(1, 1, [1, 10])}{10}$;
-

The term $\frac{\text{randint}(1, 1, [1, 10])}{10}$ generates a random value between 0.1 and 1.

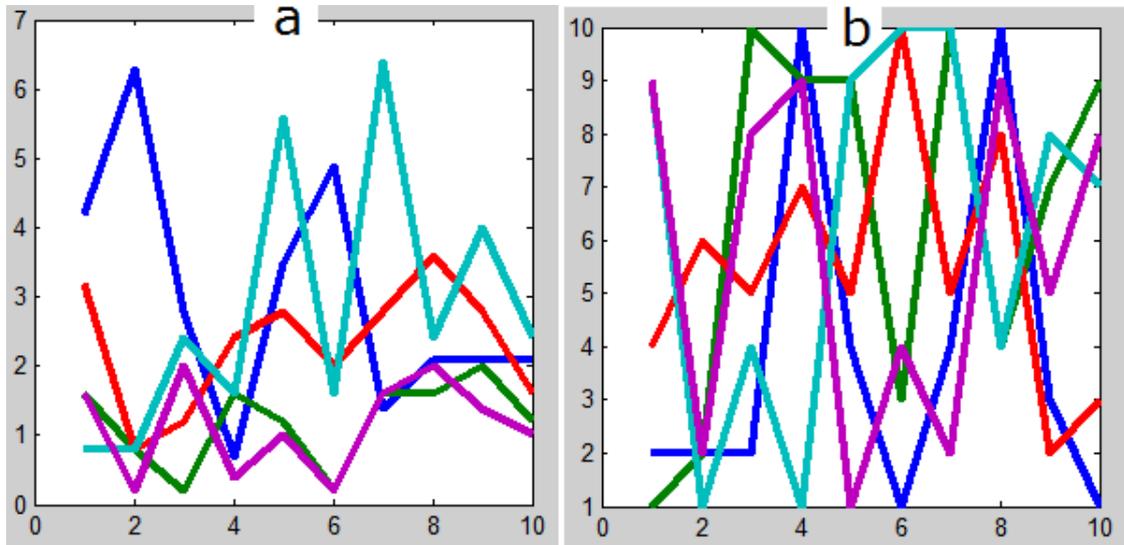


Figure 20: Random evaluations generation for 5 alternatives during 10 instants with (a) and without (b) the multiplication by the term $\frac{\text{randint}(1, 1, [1, 10])}{10}$

Algorithm III: Generating a random time weighting vector W

- 1) **Input:** s
- 2) Randomly generate α between 0 and $1/(s-1)$;
- 3) Compute $\beta = 0.5 - \alpha \cdot (s/2)$;
- 4) $stop = 0$
- 5) **While** $stop = 0$ **do**
 - a) **For** $i = 1 \dots s$ **do**
 $f(i) = f(i) - 0.00001$;
 end
 - b) **If** $0.999 < f(1) + f(2) + \dots + f(s) < 1.001$ **do**
 $stop = 1$;
 end

end

Let us consider A as a $m \times s$ matrix and b as a m -dimensional vector. \mathcal{P} is a polyhedron subject to $A.x < b$ and whose largest inscribed ball is: $\mathcal{B}(x_c, r)$.

Algorithm V: Chebyshev Center of a Polyhedron [38]

max r

Such that $a_i^T x_c + r \|a_i\|_2 \leq b_i$ where $i = 1, \dots, m$;

[38] gives more details about this algorithm.

One may wonder if the Chebyshev center can act as an analytic center of the polyhedron \mathcal{P} . In fact, if we consider a 3-dimensional polyhedron, the larger w_3 with respect to w_1 the smaller inscribed ball in this polyhedron. It results that the Chebyshev center can not match with the analytic center as defined by equation 16.

One also may argue if a starting point y_0 should not be generated by algorithm III above. In fact, this may be possible when we look for the first analytic center of \mathcal{P} . However, as we progress toward the analytic centers of more reduced areas, we found that a method based on algorithm III is a time-consuming task (several minutes) compared to Chebyshev method. The reason is that algorithm III tends to check the feasibility of every single possible weights combination.

References

- [1] Behzadian, M., Kazemzadh, A., Albadvi, D. and Aghdasi, M. “*PROMETHEE: A comprehensive literature review on methodologies and applications*”. EJOR,200, (2010), 198-215.
- [2] Brans, J.P. and Mareschal, B. “*Promethee Methods, in Multiple Criteria Decision Analysis: State of the Art*”. edited by Figueira, J. , Greco, S. and Ehrgott, M., Springer, (2005).
- [3] BRANS, J.P. “*L’Ingénierie de la Décision. Elaboration dinstruments d’aide à la décision. La Méthode PROMETHEE*”. Université LAVAL, Colloque d’aide à la Décision, QUEBEC, (1982), 183-213.
- [4] BRANS, J.P., MARESCHAL, B. and VINCKE, Ph. “*PROMETHEE: A new family of outranking methods*”. Operational research 84, Proceedings of the Tenth International Conference on Operational Research, (1984), 477-490.
- [5] BRANS, J.P. and VINCKE, Ph. “*A preference ranking organisation method. PROMETHEE for MCDM*”. Management Sciences, (1985), 31(6): 647-656.

- [6] BRANS, J.P. and MARESCHAL, B. “PROMETHEE-GAIA. Une méthodologie d’aide à la Décision en présence de critères multiples”. ULB, SMA, Ellipses, Paris, (2002).
- [7] BRANS, J.P. and MARESCHAL, B. “*An Introduction to Multicriteria Decision Aid: The PROMETHEE and GAIA Methods*”. Springer series, (2002), 163-195.
- [8] De Smet, Y. and Lidouh, K. “*PROMETHEE Methods in Multiple criteria decision making-State of the Art-Surveys*”. Springer-Verlag Berlin Heidelberg 2013
- [9] United Nations Development Programme Website
“www.undp.org/content/undp/en/home/”
- [10] Chunwei,L. Gang, K. “*A Time Series PROMETHEE Model for Sovereign Credit Default Risk Evaluation*”International Journal of Advancements in Computing Technology(IJACT). Volume4, Number17,September 2012. doi:10.4156/ijact.vol4.issue17.7
- [11] Fullér, R., Majlender, P. “*An analytic approach for obtaining maximal entropy OWA operator weights*”.Fuzzy Sets and Systems, 124(2001) 53-57.
- [12] Fullér, R., Majlender, P. “*On obtaining minimal variability OWA operator weights*”. Fuzzy Sets and Systems 136 (2003) 203215.
- [13] Wang,Y. , Parkan, C. “*A minimax disparity approach for obtaining OWA operator weights*”. Inform. Sci. 175 (2005) 2029.
- [14] Onder,E. , Tas, N., Hepsen, A. “*Performance Evaluation of Turkish Banks Using Analytical Hierarchy Process and Topsis Methods*”. Journal of International Scientific Publication: Economy and Business, Volume 7, Part 1.
- [15] Eppe, S., De Smet, Y. “*An adaptive questioning procedure for eliciting PROMETHEE IIs weight parameters*”.Published by: CoDE-SMG, CP 210/01, Université Libre de Bruxelles, Bvd du Triomphe, 1050 Ixelles, Belgium. Technical report number TR/SMG/2012-002
- [16] Bous, G., Fortemps, P., Glineur, F., Pirlot, M. “*ACUTA: A novel method for eliciting additive value functions on the basis of holistic preference statements*”. European Journal of Operational Research 206 (2010) 435444
- [17] Anwarul K., Md., Dey, S. “*Performance Analysis through CAMEL Rating: A Comparative Study of Selected Private Commercial Banks in Bangladesh*”. Journal of Politics and Governance, Vol. 1, No. 2/3, September 2012
- [18] Podvezko, V. “*The Comparative Analysis of MCDA Methods SAW and COPRAS*”. Inzinerine Ekonomika-Engineering Economics, 2011, 22(2), 134-146
- [19] Kaliszewski, I. , Zionts, S. “*An interactive programming method for solving the multiple criteria problem*”. Control and Cybernetics. vol. 33 (2004) No. 3

- [20] Benayoun, R., de Montgolfier, J., Tergny, J. “*Linear programming with multiple objective functions: Step Method (STEM)*”. Mathematical Programming 1 (1971) 366-375. North-Holland Publishing Company
- [21] V. d. Freitas, L., B. R. d. Freitas, A. P., V. Veraszto, V., A. S. Marins, F., B. Silva, M. “*Decision-Making with Multiple Criteria Using AHP and MAUT: An Industrial Application*”. European International Journal of Science and Technology. Vol. 2 No. 9. November, 2013.
- [22] Figueira, J., Greco, S., Roy, B., Slowinski, R. “*Electre Methods: Main Features and Recent Developments*”. 2010. hal-00876980
- [23] S. Vaidya, O., Kumar, S. “*Analytic hierarchy process: An overview of applications*”. European Journal of Operational Research. Volume 169, Issue 1, 16 February 2006, Pages 129
- [24] H, Q., D. Smet, Y., Bonney, J., “*D-Sight: A New Decision Making Software to Address Multi-Criteria Problems*”. International Journal of Decision Support System Technology 10/2012; 4(4):1-23. DOI: 10.4018/jdsst.2012100101
- [25] Mareschal, B. “*Visual PROMETHEE*”. Copyright 2012-2013 by VPSolutions. All Rights Reserved.
- [26] Iyengar, V. S., Lee, J., Cambell, M. “*Q-Eval: Evaluating Multiple Attribute Items Using Queries*”. Copyright 2001 ACM 1-58113-387-1/01/0010
- [27] Ye, Y., “*Interior Point Algorithms: Theory and Analysis*”. Wiley, New York
- [28] Davis, R., Roscoe, J., Phillips, R. “*Physical Education and the Study of Sport: Text with CD-ROM*”. 5th Edition. 06 October 2004
- [29] Abernethy, B. “*The Biophysical Foundations of Human Movement*”. 2005. p167
- [30] Cazorla, G. “*Evaluation des capacités physiologiques et physiques, I Evaluation des capacités dites ”Anaérobies”*”. Cours Masters ”Ingénierie de l’entraînement sportif”. p5 2003-2004
- [31] “www.brianmac.co.uk/sgtjump.htm#ref”. Visited Jan 20th 2016.
- [32] UNDP Human Development Report Office. OCCASIONAL PAPER. “*Training Material for Producing National Human Development Reports*”. March 2015
- [33] Human Development Reports Website. “<http://hdr.undp.org/en/content/human-development-index-hdi>”. Visited 7th July 2016.
- [34] “*Human Development Report 2010*”. 20th Anniversary Edition. Page 224.
- [35] “*Human Development Report 2015. Briefing note for countries on the 2015 Human Development Report. Malaysia*”
- [36] Calders, T., Van Assche, D. “*PROMETHEE is Not Quadratic: An $O(qn \log(n))$ Algorithm*”. Submitted on March 2016 to Elsevier

- [37] Jaggi, M. “*Frank-Wolfe Optimization Algorithms. A Brief Tutorial*”. Optimization and Big Data 2015, May 7 2015, Edinburgh
- [38] Wolkowicz, H. “*CO367/CM442 Nonlinear Optimization Lecture 1*”. January 04, 2010
- [39] Hyvarinen, A., Karhunen, J., Oja, E., “*Independent Component Analysis*”. Page 66. John Wiley and Sons. 5 avril 2004.
- [40] Gill, J., King, G. “*Numerical Issues in Statistical Computing for the Social Scientist*”. Chapter 6: Numerical Issues Involved in Inverting Hessian Matrices
- [41] “*I- United Nations Development Programme Website: <http://hdr.undp.org/fr/content/mean-years-schooling-adults-years>*”. Visited 7th July 2016.
- [42] “*II- United Nations Development Programme Website: <http://hdr.undp.org/fr/content/expected-years-schooling-children-years>*”. Visited 7th July 2016.