MEASURING COMPATIBILITY (CLOSENESS) IN WEIGHTED ENVIRONMENTS.
WHEN CLOSE REALLY MEANS CLOSE?

Claudio Garuti
General Manager Fulcrum Engineering
Santiago Chile
Claudiogaruti@fulcrum.cl

Keywords: Compatibility, Closeness, Singularities, Weighted Environments, Patterns.

1.- Introduction

The relevance of evaluating how close a vector is to any other, is a crucial issue in metric topology consideration and patterns behavior. The question of how to measure the distance / compatibility and the definition of an applicable threshold are the key factors to establish when vectors are close and when they are not.

In decision making environment, this question is also valid, even if the evaluation process becomes quite different, since the order topology, (used in decision making), differs considerably from the idea of closeness in metric topology used in engineering and physics. We have to consider that we are dealing with priority vectors, which means that we are inside a weighted environment or, a weighted space. In this non isotropic space, the coordinates numbers are not equivalent to each other this mean that the directions are not equivalent. The coordinates are not just numbers, they represent preferences, so they do impact distance calculation.

One have to ask what is better, to be close to a big coordinate number (representing a high priority) or to a small one?. A possible geometric explanation of this issue is showed in figure 1, First we have a flat space where the coordinate to assess the compatibility among the two persons are reflected by Vectors 1 and 2, and calculated the distance (as a similitude of closeness or compatibility among them. Then, a big coordinate value (heavy weight) appear changing the space geometry, this event reshape the space and make the measure of closeness also change; in consequence, what is considered close (or far) in a normal space may not be close (or far) in a weighted one. Even more, this reshaping effect in extreme situations may produce a singularity, the last happen when the ratio among the vector priorities (the coordinates), is high.

This basic but crucial difference with the normal Cartesian space, where all coordinates and directions are equivalent "equal weighted", forces us to review the way that we are measuring closeness and therefore, we need to reevaluate the way that we are calculating the compatibility index among priority vectors. A proposal for a new compatibility index that avoids the singularity effects in weighted environment is introduced.

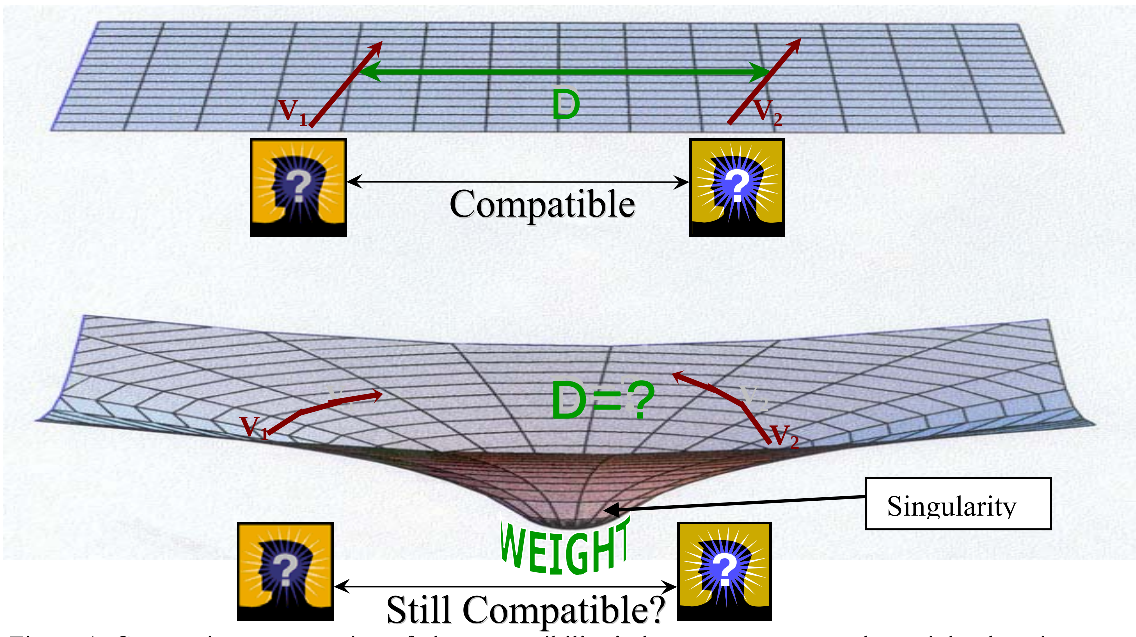

Figure 1: Geometric representation of the compatibility index measurement under weighted environment

Looking to figure 1, one might assign the structure of the decision (some AHP or ANP decision model) to the geometry of the decision space we are creating, the alternatives the matter that has to obey and follow those geometry, but at the same time telling to the structure how to reshape, or in other words how it has to crisp or curl due to the decision's judgments and the forthcoming synthesis.

The next logical step, is try to build a new compatibility index that take into account the explained situation and then prove numerically how it work. To do this, we have established a step by step process that allow to approximate to the final formula comparing the output results and their sensitive to extreme situations (singularity).

2.- Step by Step, the Building Process for a new Compatibility Index

There are different ways and formulas to evaluate the compatibility between two vectors, for example:

- Hilbert formula (Hilbert´s index) $D(A, B) = \text{Log}\ \{\text{Max}_i(a_i/b_i)\ /\text{Min}_i(a_i/b_i)\}$
- Inner vector product (IVP) $D(A, B) = \{A\}°\{B\}\ /n = (\Sigma_i\ a_i \text{ x } 1/b_i\ )\ /n$
- Weighted inner vector product (WIVP) $D(A,B) = \{A\}\ \{B\}\ \{W\} = \Sigma_i\ (a_i \text{ x } 1/b_i \text{ x } w_i)$
- Hadamard product (Saaty`s Index) $D(A, B) = [A]°[B]^t\ /n^2 = [\Sigma_i\Sigma_j\ a_{ij} \text{ x } 1/b_{ij}]\ /n^2$

Most of them work well in a unweighted environment, or in a "soft" weighted environment, which mean a relative flat geometry. But, when the weights are different, then the solution might be altered, and if the difference are big enough (big ratio for one or more coordinates of the vectors), then a singularity appear.

This paper present a way to overcome this situation and make possible to evaluate compatibility among any pair of vectors taking in consideration that we are dealing with weighted environments. The natural notion used here is: "It is more relevant to be close to the heavy coordinates than the lights one". So, they weights do care. But, observing the formulas none of them use explicitly the weights as data to evaluate the distance between vectors A and B (D(A, B)).

To introduce this new index in a more clear way, the construction process was divided in 3 main steps:

- First step:
  - The inner vector product (inverted).
- Second Step:
  - The weighted inner vector product.
- Third Step:
  - The Min-Max weighted inner vector product and its generalization using the gravity center principle.

First step: The inner vector product (inverted)

The ideas in the first step are two, one to show how to produce an inner product that can be useful for the calculation of the compatibility (or incompatibility) index. The second idea is to establish what should be the inferior and superior bonder of that index.

If we take the inner vector product of two equal vectors, inverting the coordinates of the second vector and then divide by n (the vector dimension) then we get back with the unit. In General:

AºB*/n = (Σ ai*1/bi)/n (1)

If A=B (complete flat geometry) then,
A={0.5; 0.5}
B={0.5; 0.5} ➔ B*={1/0.5; 1/0.5}
AºB*/n = (Ratio1 + Ratio2)/n = (1+1)/2 = 1
That mean: 100% of compatibility or 1-1 = 0% of incompatibility

By the other side, if we use another A, B vectors where we produce a strongly weighted space or crisp geometry, like
A={0.10 ; 0.90}
B={0.90 ; 0.10}
Please, note that A, B are both homogeneous priority vectors (their coordinate ratios are less than one order of magnitude) and more relevant for this step they are almost perpendicular vectors.

Applying the formula:
AºB*/n = (Ratio1 + Ratio2)/n = (0.1/0.9 + 0.9/0.1}/2 = 4.555 which is far from one, and that is right, (they are unlike vectors) but, they are “too” far, this is up of the 100% of unlike.
4.555, means 4.555-1=355.6% of incompatibility and this value has no sense, (or at least a hard physical interpretation).
Two equal vectors (like the first situation), has to be 100% close and one can not have a value over 100% (two vectors can not be more close than be the same vector). In the inverse way, if one think in terms of incompatibility instead of compatibility, you can not get an incompatibility over 100%, two vectors can not be more separate than having no projection one over the other (perpendicularity vectors condition). Hence, any number over 100% bond is a non sense value for incompatibility.

Graphically (figure 2):

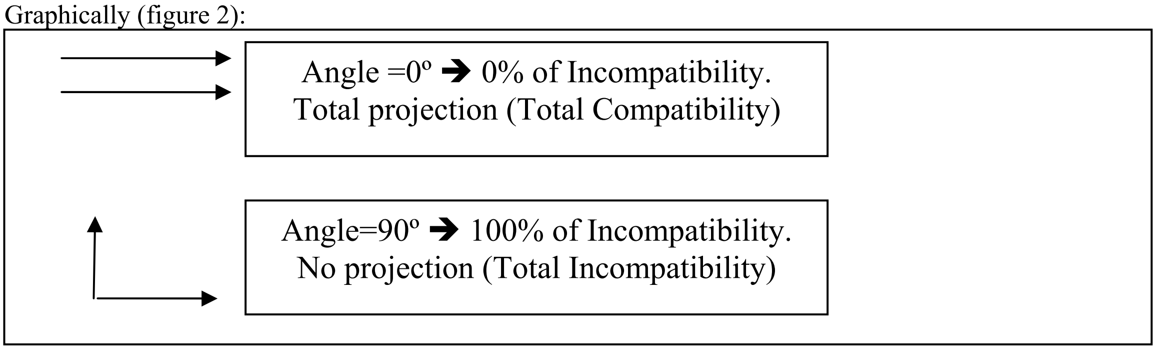

Figure 2

Second Step: The weighted inner vector product
The vector shall be weighted before to evaluate his index, since the coordinates are weights (they are not just numbers), so its important to capture in what coordinate the vector is being close or far. But, in this case, as the vectors are being weighted, then became necessary to take off the “n” value from the equation (due to the weighting process the n value is not required anymore), and that is very good thing, because the n value is just an “average weight” which could be very wrong if the coordinates of the vector (the weights) came apart from each other. By the way, this is the main reason that may appear the “singularity effect”, caused by extreme differences in the weights value.

Applying this weighting concept in **(1)** :
Weighted AºB= Ratio1*w1+Ratio2*w2=(0.1/0.9)*0.1+(0.9/0.1)*0.9= 8.111
8.11, means 8.11-1=711.1% of incompatibility. Also this value has no sense and in certain way is still worse than the one obtained from the “clean” dot product got it in the first step.
But, if you look the big picture this is a very logical result, since if you weight the wrong ratios, you can get still worse results. (making worse what is already bad).

So, what is happening?
The main issue here is not only in the weighting process (which of course is totally necessary), what real matter here is the vectors projections. It is necessary to establish the correct projections between the vectors before to build the ratios and weight anything. Establish the right projections will allow to establish the right ratios and then (and only then) weighting it in the correct way.

In order to find the right projections we can use the other way of represent the dot product (the geometric representation of dot product). AºB = ABcosine(<(AB)).
Considering that we always work in the first quadrant, (the coordinate numbers will be always positive by definition), then the angle will always lays between 0 – 90º.
If it is 0º we have full projection, represented by the same parallel vector, (getting total compatibility), and if its 90º we have no projection, getting total incompatibility. (As showed it in figure 1).
So, the bottom question here is how to capture the projections among vectors A and B.

Considering also that the norm of the vectors A and B are equal to one (they are normalized vectors), then is possible to write directly: AºB = cosine(<(AB))
So, now the question is how to establish the projection (the cosine function) in terms of the coordinates of the vectors and then (and only then) weighting them.

Third Step: The Min-Max weighted inner vector product and its generalization using the gravity center principle:

This third step is divided into two sub-steps. The first one (3.1) is about the Min-Max weighted inner vector particular solution. The second one (3.2) is the Min-Max general solution using the gravity center principle.

3.1.- The Min-Max Weighted inner vector product.

As we saw in the second step, the main question is about the projections of the vectors which can be captured by the cosine function.
The definition of the cosine function is: adjacent side /hypotenuse = shorter side /larger side.
In the figure 3, we can see its geometric interpretation.

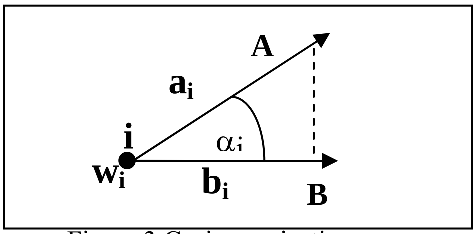

Figure 3 Cosine projection

Cosine $\alpha_i = b_i/a_i$; which represent the projection of A over B in that axis.

The cosine function is bonded between 0-1 so, we always got that the ratio among the coordinates vectors must be less-equal one. One way to reproduce this behavior is selecting the coordinates in a way that always have in the numerator the smallest coordinate and in the denominator the bigger one.
It is possible to represent this using the Min and Max functions like this:

Min{a1; b1}= Min{0.1; 0.9}= 0.1
Max{a1; b1}= Max {0.1; 0.9}= 0.9
Min / Max = 0.1 / 0.9 = 0.1111 (this is the right ratio R1 that has to be weighted).
In the same way for coordinate 2:
Min{a2; b2}= Min{0.9; 0.1}= 0.1
Max{a2; b2}= Max {0.9; 0.1}= 0.9
Min / Max = 0.1 / 0.9 = 0.1111 (this is the right ratio R2 that has to be weighted).

So, now we are able to weight both coordinates since we already have the right ratios, (given by the Min-Max function that is reproducing the cosine projection calculation):

Weighted AºB= Ratio1*w1+Ratio2*w2= (0.111)*0.1+(0.111)*0.9= 0.111.

Now, we can say that the compatibility is: 11.1% or the incompatibility is:
100% - 11.1%= 88.9%. Which represent (as expected), a very incompatible vectors but, with a value less than 100%.

Of course, to complete the process, is necessary to weight the projections in both ways (A over B and B over A), and take the average of both results. But that is just the same concept repeated.

Hence, the final formula take the form:

$$\mathbf{D(A,B) = [\Sigma_i\, Min_i\,(a_i,b_i)\,/Max_i\,(a_i,b_i\,) * a_i + \Sigma_i\, Min_i\,(a_i,b_i)\,/Max_i\,(a_i,b_i\,) * b_i]\,/2.} \qquad \mathbf{(2)}$$

We can also write (2) in its condensed form as:

$$\mathbf{D(A,B) = [\Sigma_i\,(Min_i\,(a_i,b_i)\,/Max_i\,(a_i,b_i\,)) * (a_i + b_i)/2]} \qquad \mathbf{(3)}$$

Readable as: point to point, add the arithmetic average of the i-coordinate weighted by its projection.

Let try it for different cases in a 2-Dimension couple of vectors (fig 4):

CASE	Vector coord.		Hadamard	Min/MaxCG	Hilbert	IVP
0	0,500	0,500	1,010%	9,523%	8,715%	1,010%
	0,450	0,550				
1	0,533	0,467	0,058%	2,371%	2,097%	0,218%
	0,545	0,455				
2	0,633	0,367	0,068%	2,370%	2,259%	0,760%
	0,645	0,355				
3	0,733	0,267	0,097%	2,363%	2,702%	1,548%
	0,745	0,255				
4	0,833	0,167	0,198%	2,348%	3,860%	3,161%
	0,845	0,155				
5	0,933	0,067	1,108%	2,285%	9,126%	10,274%
	0,945	0,055				
6	0,987	0,013	280,851%	1,839%	111,919%	599,399%
	0,999	0,001				

Figure 4: Different cases of Min/Max Index (Sensibility)

From results of figure 4, the Min-Max formula, is the only one that keep stable the tendency to "lowing" the incompatibility value as we change the coordinate values (as expected to be).

This is because the Min-Max formula is the only one that take into account two crucial issues:
1.- the weights of the coordinates
2.- the projection of the vectors.

This good behavior include also a logical and acceptable bounding condition situation, that is a range for incompatibility index starting from 0%, (total compatibility or parallel vector situation) to 100% (total incompatibility or perpendicular vector situation).
Incompatibility Index lays between [0% - 100%], no matter what coordinates value, or vector dimension one is dealing.
Dem:
$\Sigma_i a_i = 1, \Sigma_i b_i = 1$ ➔ $\Sigma_i (a_i + b_i)/2 = 1$
$(\text{Min/Max})_i$ .LE.1 for every i ➔ $\Sigma_i (\text{Min/Max})_i \times (a_i + b_i)/2$ .LE. 1 (100%)

Its possible to test the formula in different situations (normal or extreme), and one should get close results to Hadamard or Hilbert formulas if the coordinates are more or less equal among them (homogeneous coordinates). But, as you start to make differences for the coordinates values, or the coordinates between the vectors are very different ($a_i <> b_i$), even if they are homogeneous among themselves, then the issues of weight and projection will became more and more relevant, making important differences in the index final value (the singularity effect). In fact, the index value of incompatibility for the initial coordinates of A, B using Hadamard operator is equal to: 1,975.3% >> 100%, which is not a interpretable value. This is an interesting result, since prove that is not the homogeneity the main issue here, in fact, the vectors A and B are homogeneous but, the weights that are in play and the vector projections, are making the real difference to the final index value calculation. If those effects are not taking into account then, the results may yield out of the 0 – 100% bonds, which is not an expected result.

Is important to point out that the Min-Max formula also represent a real possibility to define a stable index threshold, which would be around 10% for any condition of coordinates or vector dimension. This threshold if tested for similar couple of vectors return with values less or equal 10% for Hilbert and Min-Max formulas and less or equal 1% for Hadamard and inner vector product, showing that in normal conditions, (no singularity present) the 10% is a good threshold value and still more important, for the Min-Max formula this threshold value stay unaltered even in presence of singularities (strong weighted space).

As a last example, we will show a simple application of the compatibility calculation, first over a not homogeneous set and then make it homogeneous deleting the alternative with the smallest weight.

Next in figures 5 and 6, is show a comparison matrix of energy consumption objects:

Annual Electricity	Elec-tric Range	Refrigerator	TV	Dish-Wash	Iron	Hair Dryer	Radio	AHP Eigen-vector	*Actual Relative Weights*
Elec.Range	1	2	5	8	7	9	9	.393	*.392*
Refrig.	1/2	1	4	5	5	7	9	.261	*.242*
TV	1/5	1/4	1	2	5	6	8	.131	*.167*
Dishwash.	1/8	1/5	1/2	1	4	9	9	.110	*.120*
Iron	1/7	1/5	1/5	1/4	1	5	9	.061	*.047*
Hair-dryer	1/9	1/7	1/6	1/9	1/5	1	5	.028	*.028*
Radio	*1/9*	*1/9*	*1/8*	*1/9*	*1/9*	*1/5*	*1*	*.016*	*.003*

Figure 5: Comparison Matrix for Electricity Annual Consumption

Hadamard Incompatibility Index = 45.5% (Compatibility of 54.5%)
Min-Max Incompatibility Index = 8.2% (Compatibility of 91.8%)

If we now delete the Radio alternative, since is the alternative that produce the bigger loss of homogeneity (0.393 is 24.6 times greater than 0.016).

Then:

Annual Electricity	Elec-tric Range	Refrigerator	TV	Dish-Wash	Iron	Hair Dryer	Radio	AHP Eigen-vector	*Actual Relative Weights*
Elec.Range	1	2	5	8	7	9	9	.393	*.392*
Refrig.	1/2	1	4	5	5	7	9	.261	*.242*
TV	1/5	1/4	1	2	5	6	8	.131	*.167*
Dishwash.	1/8	1/5	1/2	1	4	9	9	.110	*.120*
Iron	1/7	1/5	1/5	1/4	1	5	9	.061	*.047*
Hair-dryer	1/9	1/7	1/6	1/9	1/5	1	5	.028	*.028*

Figure 6: Comparison Matrix Without Radio Alternative

Renormalizing:

0.399	0.265	0.133	0.112	0.062	0.028
0.393	0.243	0.168	0.120	0.047	0.028
8.08%	Incompatibility with min-max				
2.40%	Incompatibility with hadamard				

The output shows that Incompatibility evaluated with Hadamard formula is 2.4%, representing a change in a large 94.7% over the 45.5% initial value. For a weighting way of thinking, this is a unexpected changing in the compatibility index value, considering that we are deleting an irrelevant alternative with only a 1.6% of the total weight.
By the other hands we have that the Incompatibility index using Min-Max formula is 8.08% (compatibility of 91.9%) which represent a little change of only 1% over the initial value (8.2%), which is an expected changing value.

This example shows clearly that the homogeneity should not be an issue for compatibility (as it is for consistency indeed). Since compatibility have to deal with reality and the reality doesn't need to be homogeneous. The radio exist and has to be considered in the final set of alternatives.

Homogeneity is necessary to reach a good consistency in the matrix comparisons, this could be done clustering the alternatives in two homogeneous sets, duplicating one alternative (for example the iron alternative) and use it as a pivot.

Conclusions

A big analogy can be made over this study of compatibility (as showed in figure 1 & 7). We can establish that the AHP or ANP model without the alternatives (the decision structure), represent the geometry of the decision space we are creating. Each criterion and weight represent a curl or a change in the slope of this space (bigger the weight, bigger the slope). So, any change we make over this space geometry (the model), must be reflected by the compatibility index value, in order to preserve an index that really can measure the compatibility in this two different structures.

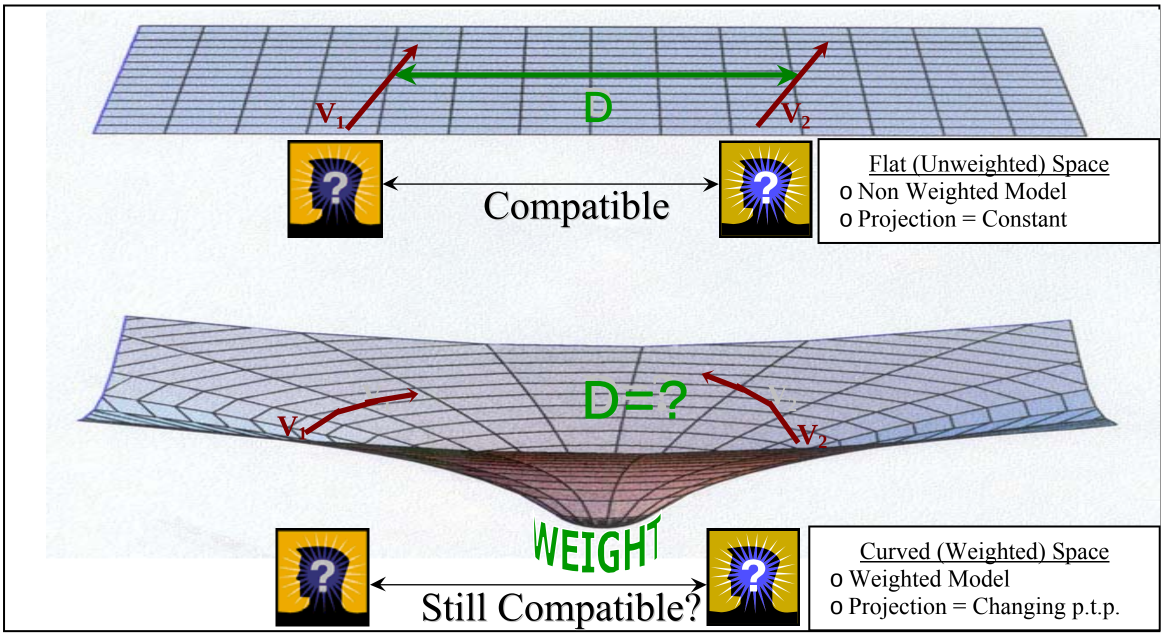

Figure 7: Model representation for weighted compatibility index assessment

Next in figure 8, is show a table where is clearly synthesized what is considered and what not for the different knew indexes. Of course, each of this four criteria checked ok, represent a better behavior for the corresponding index.

INDEX	SYMMETRY	WEIGHTS	PROJECTION	BOUNDARY [0-100]
HILBER	check OK	not check	not check	not check
STANDARD IVP	not check	not check	not check	not check
WEIGHTED IVP	not check	check OK	not check	not check
HADAMARD	check OK	not check	not check	not check
MIN-MAX.CG	check OK	check OK	check OK	check OK

Figure 8: Comparison table for different Compatibility index calculation

Where:

Symmetry = this condition is related to get the same index value independently of what vector we use as starting point when calculate the compatibility index.

Weights = this condition is checking if the index calculation formula is considering the weights of the model in an explicit way (avoiding cancellations).

Projection = this condition is checking if the index calculation formula is considering the changing projections (point to point), among the vectors.

Boundary = this condition is checking if the final index value belong to a bounded scale [0 – 100]%. (If it is bounded, then is possible to build a significant threshold inside the scale).

The final conclusion of this paper is to show a new formula able to assess the compatibility index in a weighted environment, for any dimensional vector (small or large) no matter if its homogeneous or not, presenting also a reliable threshold value that rest the same for any situation, no matter the dimension of the vectors or if it is present any singularity.