Lexicographically Prioritized Multi-criteria Decisions Using Scoring Function

Ronald R. Yager
Machine Intelligence Institute, Iona College
New Rochelle, NY 10801
yager@panix.com

Abstract

We consider multi-criteria decision problems where there is a lexicographically induced prioritization relationship over the criteria. We suggest that the prioritization between criteria can be modeled by making the weights associated with a criteria dependent upon the satisfaction of the higher priority criteria. We implement this using a prioritized scoring operator. We show how the lack of satisfaction to higher order criteria block the possibility of compensation by lower priority criteria. We show that in the special case where the prioritization relationship among the criteria satisfies a linear ordering we can use a prioritized averaging operator.

Keywords: Multi-Criteria Decision-Making, aggregation, priority, compensation.

1. Introduction

Decisions based on the satisfaction of multiple criteria are pervasive in many domains. In these problems we have a collection of criteria $C = \{C_1, ..., C_n\}$ and a set of alternatives $X = \{x_1, ..., x_m\}$. Here we must choose between these alternatives based on their satisfaction to the criteria. In the following we shall assume that we have a measure of the satisfaction of criteria C_i by each alternative, $C_i(x)$, as a value in the unit interval. One commonly used approach is to calculate for each alternative a score C(x) as an aggregation of its satisfaction to the individual criteria $C(x) = \sum_{i=1}^{n} w_i C_i(x)$. The w_i are importance weights associated with the criteria which typically satisfy $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$

It is easy to see that this type of aggregation is monotonic in the sense that C(x) does not decrease if any of the $C_i(x)$ increases. It is also bounded, $\text{Min}_i[C_i(x)] \leq C(x) \leq \text{Max}_i[C_i(x)]$. It is also idempotent, if all $C_i(x) = a$ then $C(x) = a$. Because of these properties this is an averaging operator. Closely related to this is what we shall call a **scoring** (or precisely a weighted scoring) operator. The difference between a scoring operator and an averaging $\sum_{ij} w_{ij}C_{ij}(x)$ operator is that the scoring operator does not require that $\sum_{i=1}^{n} w_i = 1$. We note that while a scoring operator is monotonic it not necessarily bounded nor idempotent. Essentially a averaging operator is a special case of scoring operator. Both these operators can be used in the alternative selection problem.

These types of aggregation operators allow tradeoffs between criteria. In this type of aggregation we can compensate for a decrease of Δ in satisfaction to criteria C_i by gain $\frac{w_k}{w_i}\Delta$ in satisfaction to criteria C_k.

In some applications we may have a **lexicographic** ordering of the criteria and do not want to allow this kind of compensation between criteria. Consider the situation in which we are selecting a bicycle for our child based upon the criteria of safety and cost. However any bicycle we select must be safe. In this situation we do not want poor safety to be compensated for very low cost. Before even considering cost we must be sure the bicycle is safe. Here we have a **lexicographic** ordering induced prioritization of the criteria. Safety has a higher priority. Consider a problem of document retrieval in which we are looking for documents about the American revolution and prefer if they are from an academic website and written after 2003. Here again we have a lexicographic ordering on the criteria. In this case the property of it being about the American Revolution has a priority, if it is not about this topic we are not interested. In organizational decision making criteria desired by superiors generally, have a higher priority then those of their subordinates. The subordinate must select from among the solutions acceptable to the superior. Generally air traffic controller decisions involve a prioritization of considerations with passenger safety usually at the top.

In this work we shall suggest scoring operators that allow for the inclusion of a lexicographically induced priority between the criteria. Central to our approach will be the modeling of priority by using importance weights in which the importance of a lower priority criteria will be based on its satisfaction to the higher priority criteria [1]. As we shall see this result in a situation in which importance weights will not be the same across the alternatives. In this case for a given alternative the importance weights associated with a criterion will depend on the alternative's satisfaction to higher priority criteria. As we shall subsequently see when determining the score for an alternative this formulation will effectively prevent the alternative's satisfaction to lower priority criteria from compensating for its poor satisfaction to higher priority criteria. Lower priority criteria will only be able to contribute to the score of alternatives that have good satisfaction to the higher priority criteria.

2. Prioritized Scoring Functions

In the following we assume that we have a collection of criteria partitioned into q distinct categories, $H_1, H_2, ..., H_q$ such that $H_i = \{C_{i1}, C_{i2}, ..., C_{in_i}\}$, Here C_{ij} are the criteria in category H_i. We assume a prioritization between these categories $H_1 > H_2, ... > H_q$. The criteria in the class H_i have a higher priority than those in H_k if i < k. The total set of criteria is $C = \bigcup_{i=1}^{q} H_i$. We assume $n = \sum_{i=1}^{q} n_i$ the total number of criteria. Criteria in the same category have the same priority.

In figure #1 we show the positioning of the criteria

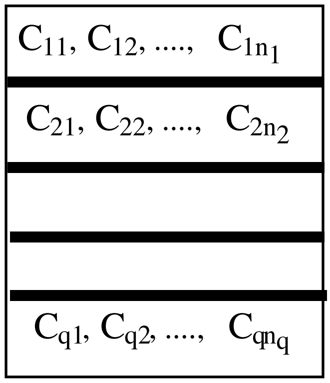

Figure #1. Prioritization of Criteria

We assume that for any alternative $x \in X$ we have for each criteria C_{ij}, a value $C_{ij}(x) \in [0, 1]$ indicating its satisfaction to criteria C_{ij}.

In the following we introduce an aggregation operator $F: [0, 1]^n \to [0, 1]$ such that $F((a_{11}, ..., a_{1n_1}), ..., (a_{q1}, ..., a_{qn_q})) = \sum_{i=1}^{q} (\sum_{j=1}^{n_i} w_{ij} a_{ij})$. We shall refer to as the **P**rioritized **S**coring (PS) operator. This aggregation operator allows us to calculate C(x) for any alternative as

$$C(x) = F(C_{ij}(x)) = \sum_{i=1}^{q} (\sum_{j=1}^{n_i} w_{ij} C_{ij}(x)).$$

However here the weights which will be dependent on x will be used to enforce the priority relationship. In order to obtain the weights for a given alternative x we proceed as follows.

For each category H_i we calculate

$$S_i = Min_j[C_{ij}(x)]$$

Here S_i is the value of the least satisfied criteria in category H_i under alternative x. Using this we will associate with each criteria C_{ij} a value u_{ij}. In particular for those criteria in category

H_1 we have $u_{1j} = 1$. For those criteria in category H_2 we have $u_{2j} = S_1$. For those criteria in category H_2 we have $u_{3j} = S_1S_2$. For those criteria in category H_4 we have $u_{4j} = S_1S_2S_3$. More generally u_{ij} is the product of the least satisfied criteria in all categories with higher priority than H_i.

We can more succinctly and more generally express $u_{ij} = T_i$, where $T_i = \prod_{k=1}^{i} S_{k-1}$ with the understanding that $S_0 = 1$ by default. We note that we can also express T_i as

$$T_i = S_{i-1}\, T_{i-1}$$

This equation along with the fact that $T_1 = S_0 = 1$ gives a recursive definition at T_i.

We now see that for all $C_{ij} \in H_i$ we have $u_{ij} = T_i$. Using this we obtain for each C_{ij} a weight w_{ij} with respect to alternative x such that $w_{ij} = u_{ij}$. We see that each $w_{ij} \in [0,1]$. We further observe that $T_i \geq T_k$ for $i < k$. From this it follows that if $i \leq j$ then $w_{ij} \geq w_{ke}$ for all j and e.

Using these weights we then can get an aggregated score x under these prioritized criteria as

$$C(x) = \sum_{i,\, j} w_{ij}\, C_{ij}(x) = \sum_{i,\, j} T_i\, C_{ij}(x) = \sum_{i=1}^{q} T_i(\sum_{j=1}^{n_i} C_{ij}(x))$$

We note that this operator is monotonic, if $C_{kj}(x)$ increases then C(x) can't decrease. We see this as follows:

$$\frac{\partial C(x)}{\partial C_{kj}(x)} = T_k + \sum_{i=k+1}^{q} \frac{\partial T_i}{\partial C_{kj}(x)} \sum_{j=1}^{n_i} C_{ij}(x)).$$

If $S_k \neq C_{kj}(x)$ then $\frac{\partial T_i}{\partial C_{kj}(x)} = 0$ for $i \geq k + 1$ and hence $\frac{\partial C(x)}{\partial C_{kj}(x)} = T_k \geq 0$. If $S_k = C_{kj}(x)$ then for $i \geq k + 1$ we have $\frac{\partial T_i}{\partial C_{kj}(x)} = \prod_{\substack{r = 1 \text{ to } i-1 \\ r \neq k}}^{i-1} S_r \geq 0$

and hence again $\frac{\partial C(x)}{\partial C_{kj}(x)} \geq 0$.

Following is an example using this PS operator.

Example: Consider the following prioritized collection of criteria:

$$H_1 = \{C_{11}, C_{12}\},\ H_2 = \{C_{21}\},$$
$$H_3 = \{C_{31}, C_{32}, C_{33}\},\ H_4 = \{C_{41}, C_{42}\}$$

Assume for alternative x we have: $C_{11}(x) = 0.7$, $C_{12}(x) = 1$, $C_{21}(x) = 0.9$, $C_{31}(x) = 0.8$, $C_{32}(x) = 1$, $C_{33}(x) = 0.2$, $C_{41}(x) = 1$, $C_{42}(x) = 0.9$

We first calculate:

$S_1 = \text{Min}[C_{11}(x), C_{12}(x)] = 0.7$

$S_2 = \text{Min}[C_{21}(x)] = 0.9,$

$S_3 = \text{Min}[C_{31}(x), C_{32}(x), C_{33}(x)] = 0.2$

$S_4 = \text{Min}[C_{41}(x), C_{42}(x)] = 0.9$

Using this we get:

$T_1 = 1$, $T_2 = S_1T_1 = 0.7$, $T_3 = S_2T_2 = 0.63$ and $T_4 = S_3T_3 = 0.12$.

From this we obtain: $u_{11} = u_{12} = T_1 = 1$, $u_{21} = T_2 = 0.7$. $u_{31} = u_{32} = u_{33} = T_3 = 0.63$. $u_{41} = u_{42} = T_4 = 0.12$.

In this case then we have

$w_{11} = w_{12} = 1$, $w_{21} = 0.7$. $w_{31} = w_{32} = w_{33} = 0.63$, $w_{41} = w_{42} = 0.12$

We now calculate $C(x) = \sum_{ij} w_{ij} C_{ij}(x) \quad = 3.82$

We now look at some further properties of the proposed aggregation method. We recall $H_i = \{C_{ij} \mid j = 1 \text{ to } n_i\}$ where the criteria in category H_i have priority over those in H_k if i_i i < k. Again letting $a_{ij} = C_{ij}(x)$ we have $S_i = \text{Min}_j[a_{ij}]$ and $S_o = 1$ and $T_i = \prod_{k=1}^{i} S_{k-1}$. Here with $u_{ij} = T_i$ we use as our weights in this prioritized scoring operator $w_{ij} = u_{ij} = T_i$ and hence

$$C(x) = \sum_{i=1}^{q} (\sum_{j=1}^{n_i} w_{ij}\, a_{ij}) = \sum_{i=1}^{q} T_i(\sum_{j=1}^{n_i} a_{ij})$$

Letting $A_i = \sum_{j=1}^{n_i} a_{ij}$ we have

$C(x) = \sum_{i=1}^{q} T_i A_i$.

We see that the weight associated with the elements in the i^{th} category is $T_i = \prod_{k=1}^{i} S_{k-1}$. Thus the criteria in H_i contribute proportionally to the product of the satisfaction of the higher order criteria. Thus poor satisfaction to any higher criteria reduces the ability for compensation by lower priority criteria. This is of course the fundamental feature of the prioritization relationship.

We also observe that if there exists some category H_r such that $C_{rj}(x) = 0$ for some criteria in H_r then $S_r = 0$ and $T_i = 0$ for $i > r$ and hence $C(x) = \sum_{i=1}^{r} T_i A_i$.

Note: While in the preceding we assumed $C_{ij}(x) \in [0, 1]$ this is not necessarily required. If we let $F_{ij} : \mathcal{R} \rightarrow [0, 1]$ be some function from the real numbers into the unit intervals such that $F_{ij}(C_{ij}(x))$ is some measure of how satisfied we are with a score $C_{ij}(x)$ for criteria C_{ij} then we allow the values of $C_{ij}(x)$ be any number if we calculate

$$S_i = \mathrm{Min}_j[F_{ij}(C_{ij}(x))]$$

Here we just transfer the $C_{ij}(x)$ into numbers in the unit interval for calculating S_i.

3. Non-Monotonicity under Normalization

A natural question that arises is why have we chosen this scoring type operator rather then an averaging operator which requires that the $\sum_{ij} w_{ij} = 1$. We see this can be easily accomplished by a simple normalization. In particular if instead of using $w_{ij} = u_{ij}$ we use $w_{ij} = \frac{u_{ij}}{\sum_{i=1}^{q} \sum_{j=1}^{n_i} u_{ij}}$ and since $u_{ij} = T_i$ this simplifies to $w_{ij} = \frac{T_i}{\sum_{i=1}^{q} n_i T_i}$. As the following example illustrates performing this normalization does not always guarantee a monotonic aggregation.

Example: Assume $H_1 = \{C_{11}, C_{12}, C_{13}, C_{14}\}$ and $H_2 = \{C_{21}, C_{22}, C_{23}\}$. Assume for x we have $C_{11}(x) = C_{12}(x) = C_{13}(x) = 1$, $C_{14}(x) = 0$ and $C_{21}(x) = C_{22}(x) = C_{23}(x) = 0$. In this case $S_1 = 0$ and hence $T_1 = 1$ and $T_2 = 0$. Thus we get $u_{1j} = 1$ and $u_{2j} = 0$ and hence $\sum_{i=1}^{q} \sum_{j=1}^{n_i} u_{ij} = 4$. From this we get $w_{1j} = \frac{1}{4}$ for j = 1 to 4 and $w_{2j} = 0$ for j = 1 to 3 and therefore

$$C(x) = \frac{1}{4}(C_{11}(x) + C_{12}(x) + C_{13}(x) + C_{14}(x))$$
$$= 0.75$$

Consider alternative y for which we have $C_{11}(y) = C_{12}(y) = C_{13}(y) = 1$, $C_{14}(y) = 1$ and $C_{21}(y) = C_{22}(y) = C_{23}(y) = 0$. The only difference between x and y is that we have increased the satisfaction of C_{14}, $C_{14}(y) = 1$ while $C_{14}(x) = 0$. Monotonicity requires that $C(y) \geq C(x)$. Let us calculate C(y). In this case $S_1 = 1$ and therefore $T_1 = 1$ and $T_2 = 1$. In this case all $u_{ij} = 1$ and hence $\sum_{ij} u_{ij} = 7$ and therefore all $w_{ij} = \frac{1}{7}$. From this we get that

$$C(y) = \frac{1}{7} \sum_{ij} C_{ij}(y) = \frac{4}{7} = 0.57 < 0.75.$$

Thus we see that $C(y) < C(x)$ and the monotonicity condition has not been satisfied.

We note the use of a scoring type aggregation operator does indeed respect the monotonicity. In this case $w_{ij} = u_{ij}$. Hence for x we have $w_{1j} = u_{1j} = 1$ and $w_{2j} = u_{2j} = 0$ From this we get $C(x) = 3$. For the case of y we get $w_{1j} = u_{1j} = 1$ and $w_{2j} = u_{2j} = 1$. From this we get $C(y) = 4$ and hence the monotonicity is respected.

4. Averaging Operators for Linear Ordered Criteria

In the preceding the priority relationship between the criteria was a weak ordering, we allowed ties as was the case for criteria in the same category. As we shall subsequently show if the priority relationship between the criteria is a linear ordering, no ties allowed, then we can obtain a prioritized averaging (PA) operator.

He we also assume we have a collection of criteria partitioned into q distinct categories, H_1, H_2, ..., H_q and we assume a prioritization between these categories $H_1 > H_2 > ... > H_q$. However here we assume each category has just one member $H_i = \{C_i\}$. Thus here there is a linear ordering among the criteria $C_1 > C_2 > ... > C_q$. We have used only one index, as we have no need for the second index. Our objective is to get a collection of weights w_i that respect the prioritization and use these to calculate $C(x) = \sum_{i=1}^{q} w_j C_j(x)$ Since we want this to be a prioritized averaging operator we require that $w_i \in [0, 1]$ and $\sum_{i=1}^{q} w_i = 1$. In order to obtain these weights we shall essentially follow the procedure used in the preceding with the addition of a normalization step.

For each priority category H_i we calculate S_i as the value of the least satisfied criteria in H_i, in this case we simply get $S_i = C_i(x)$. Again here we let $T_1 = 1$ and for $i > 1$ we let $T_i = \prod_{k=1}^{i-1} S_k$. If we let $S_0 = 1$ we can more succinctly express this as $T_i = \prod_{k=1}^{i} S_{k-1}$ for all i. Denoting $u_i = T_i$ as the un-normalized weights we can obtain normalized weights $w_i = \frac{u_i}{T}$ where $T = \sum_{i=1}^{q} u_i = \sum_{i=1}^{q} T_i$.

It is clear that the w_i lie in the unit interval and sum to one. To assure that $C(x) = \sum_{i=1}^{q} w_i C_i(x)$ is an averaging operator we must show that it is bounded and monotonic.

We now show that this PA aggregation method is bounded and monotonic. First we see that the value of this aggregation is bounded by the maximum and minimum of the arguments and hence it is also idempotent. For simplicity let us denote $a_i = C_i(x)$. Using this we have $C(x) = \sum_{i=1}^{q} w_i a_i$.

Consider now boundedness. Assume $a = \text{Min}_i[a_i]$ and $b = \text{Max}_i[a_j]$ then $C(x) = \sum_{i=1}^{q} w_i a_i \geq a$ and $C(x) = \sum_{i=1}^{q} w_i a_i \leq b$. Now consider the case where all the a_i are the same, $a_i = d$. In this case since $\sum_{i=1}^{q} w_i = 1$ we get $C(x) = \sum_{i=1}^{q} w_i d = d$ and hence the operation is idempotent.

We now consider the issue of monotonicity. We shall denote the satisfaction of each criteria to x as $a_i = C_i(x)$. We note that in this case with one criteria at each level, $S_i = a_i$. Here then $T_1 = 1$, $T_2 = a_1$ and more generally $T_i = \prod_{k=1}^{i} S_{k-1}$. Using this we have

$$C(x) = \frac{\sum_{i=1}^{q} T_i a_i}{T}$$

Let us denote $C(x) = \frac{M}{T}$ where $M = \sum_{i=1}^{q} T_i a_i$ and $T = \sum_{i=1}^{q} T_i$. For monotonicity to hold we have to show that $\frac{\partial C(x)}{\partial a_j} \geq 0$ for any j. This requires that $\frac{T \frac{\partial M}{\partial a_j} - M \frac{\partial T}{\partial a_j}}{(T)^2} \geq 0$. Hence we must show that the numerator is non-negative,

$$T \frac{\partial M}{\partial a_j} - M \frac{\partial T}{\partial a_j} \geq 0.$$

Before preceding we note that $\frac{\partial T_i}{\partial a_j} = 0$ for $i \leq j$ and $\frac{\partial T_i}{\partial a_j} = \frac{T_i}{a_j}$ for $i > j$. We also note that

$$M = \sum_{i=1}^{q} T_i a_i = \sum_{i=1}^{q} T_{i+1} \text{ since } T_i a_i = T_{i+1}.$$

However we shall find it more useful to express

$$M = \sum_{i=2}^{q+1} T_i.$$

We shall denote $A = \frac{\partial M}{\partial a_j} = \frac{1}{a_j} \sum_{i=j+1}^{q+1} T_i$.

We shall also let $B = \frac{\partial T}{\partial a_j}$ hence since $T = \sum_{i=1}^{q} T_i$ we have $B = \frac{\partial T}{\partial a_j} = \frac{1}{a_j} \sum_{i=j+1}^{q} T_i$.

From this we observe that $A \geq B$. In the following we shall find it convenient to denote $E = \sum_{i=2}^{j} T_i$.

Consider now the term $T\frac{\partial M}{\partial a_j} - M\frac{\partial T}{\partial a_j} = AT - BM$. We now observe that

$$T = \sum_{i=1}^{q} T_i = \sum_{i=1}^{j} T_i + a_j B.$$

Since $T_1 = 1$ then $T = 1 + E + a_j B$. We further observe that

$$M = \sum_{i=2}^{q+1} T_i. = E + a_j A.$$

Using the relations we see that

$$AT - BM = A(1 + E + a_j B) - B(E + a_j A) = A + EA + a_j BA - BE - a_j BA$$

$$AT - BM = A + E(A - B)$$

Since $A \geq B$ it follows that $AT - BM \geq 0$.

Thus for linear ordered criteria we can obtain a prioritized averaging aggregation operator.

5. Alternative Determination of Weights

In the preceding we introduced the prioritized scoring operator as a method for multi-criteria aggregation for the case in which our criteria where partitioned into q categories, $H_i = \{C_{ij} : j = 1, ..., n_i\}$ where category H_i had priority over H_k if $i < k$. For a given alternative x we shall find it convenient in the following to denote $C_{ij}(x) = a_{ij}$. Using this notation then we defined

$$S_o = 1$$

$$S_i = \text{Min}_i[a_{ij}] \text{ for } i = 1 \text{ to } qI$$

$$T_i = \prod_{k=1}^{i} S_{k-1} \text{ for } 1 = \text{to } q$$

With $w_{ij} = T_i$ we obtained as our aggregated value

$$C(x) = \sum_{i=1}^{q} \sum_{j=1}^{n_i} w_{ij}\, a_{ij} = \sum_{i=1}^{q} \sum_{j=1}^{n_i} T_i\, a_{ij}$$

Letting $A_i = \sum_{j=1}^{n_i} a_{ij}$ we can express this as $C(x) = \sum_{i=1}^{n} T_i A_i$.

In the preceding we assumed that the satisfaction to the priority class $H_i = \{C_{i1}, ..., C_{in_i}\}$ under alternative x was determined by the least satisfied criteria in H_i, $S_i = \text{Min}_j[C_{ij}(x)]$. Here we shall suggest some alternative methods for calculating S_i.

One method we shall consider will be based on the OWA aggregation operator [2]. Here we associate with each priority class H_i a vector $\mathbf{V_i}$ of dimension n_i called the OWA weighting vector. The components V_{ik} of $\mathbf{V_i}$ are such that $V_{ik} \in [0, 1]$ and $\sum_{k=1}^{n_i} V_{ik} = 1$. Additionally we let $\text{ind}_i(k)$ be an index of function so that $b_{ik}(x) = C_{\text{ind}_i(k)}(x)$ is the k^{th} largest of $C_{ij}(x)$. Using this we now calculate

$$S_i = \sum_{k=1}^{n_i} V_{ik}\, b_{ik}(x)$$

We see that if $V_{in_i} = 1$ and $V_{ik} = 0$ for $k \neq n_i$ then we get $S_i = \text{Min}_j[C_{ij}(x)]$, the original method. An important special case is where $V_{ik} = 1/n_i$ for all k. In this case $S_i = \frac{1}{n_i} \sum_{j=1}^{n_i} C_{ij}(x)$. Here we take as S_i the average of the satisfactions of the criteria in category H_i.

Another special case is when $V_{i1} = 1$ and $V_{ik} = 0$ for $k \neq 1$. In this case $S_i = \text{Max}_j[C_{ij}(x)]$. Here we take S_i as the score of the most satisfied criteria in category H_i. Many other weight vectors are possible for example if $V_{iq} = 1$ for some q S_i simply becomes the q^{th} largest of the $C_{ij}(x)$.

In this framework we can associate with each weighing vector $\mathbf{V_i}$ a measure called its attitudinal character denoted, A-C($\mathbf{V_i}$) [3]. We define this as

$$\text{A-C}(\mathbf{V_i}) = \frac{1}{n_i - 1} \sum_{k=1}^{n_i} V_{ik}\,(n_i - k)$$

It can easily be shown [2] that for the case where $V_{in_i} = 1$ we get A-C($\mathbf{V_i}$) = 0. For the case where $V_{ik} = \frac{1}{n_i}$ for all k then A-C($\mathbf{V_i}$) = 0.5 and for the case where $V_{i1} = 1$ we have A-C($\mathbf{V_i}$) = 1.

If we denote A-C(V_c) = α_i then we see in figure #2 the relationship between the value of α_i and the form for the calculation of S_i. Here then α_ι can be seen as a measure of the ***tolerance*** in determining the satisfaction of the category. While it is not necessary, it would be seen that the default situation is to assume α_ι is the same for all H_i.

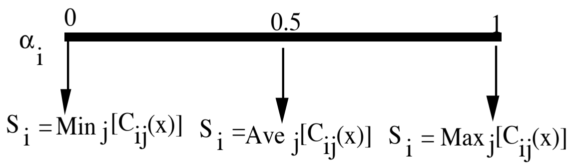

Figure #2. Relationship between α_i and the form of S_i.

Many of the techniques available for calculating the OWA weights [4, 5] can be tailored for this particular application. A particularly interesting possibility is to use a variation of the method originally suggested by O'Hagan [6-8]. In this case we would supply a desired level of tolerance α_i and solve the following mathematical programming problem for the V_{ik}

$$\text{Min} \sum_{k=1}^{n_i} (V_{ik})^2$$

$$\text{Such that: } \frac{1}{n_i - 1} \sum_{k=1}^{n_i} V_{ik}\,(n_i - k) = \alpha_i$$

$$\sum_{k=1}^{n_i} V_{ik} = 1$$

$$V_{ik} \geq 0$$

We provide an example of the preceding variation using the earlier example

Example: $H_1 = \{C_{11}, C_{12}\}$, $H_2 = \{C_{21}\}$, $H_3 = \{C_{31}, C_{32}, C_{33}\}$, $H_4 = \{C_{41}, C_{42}\}$

For alternative x we have

$C_{11}(x) = 0.7$, $C_{12}(x) = 1$

$C_{21}(x) = 0.9$

$C_{31}(x) =0 .8$, $C_{32}(x) = 1$, $C_{33}(x) = 0.2$

$C_{41}(x) = 1$, $C_{42}(x) = 0.9$

Consider the case where $S_i = \text{Max}_j[C_{ij}(x)]$. Here then

$$S_1 = 1, S_2 = 0.9, S_3 = 1. S_4 = 1$$

From this we get :

$$T_1 = 1, T_2 = S_1\,T_1 = 1, T_3 = S_2\,T_2 = 0.9, \quad T_4 = S_3\,T_3 = 0.9$$

With $C(x) = \sum_{i=1}^{4} A_i\,T_i$ where $A_i = \sum_{j=1}^{n_i} C_{ij}(x)$ we have

$$C(x) = (1)(1.7) + (1)(.9) + (0.9)(2) + (0.9)(1.9) = 6.11$$

Another approach for calculating the S_i involves associating with each criteria in H_i an additional local weight. In this case our form for H_i is

$$H_i = \{(C_{ij}, g_{ij}) \mid j = 1, \ldots.., n_i\}$$

where the g_{ij} indicates the importance of C_{ij} in calculating S_i. Here we assume that $g_{ij} \in [0, 1]$

and $\sum_{j=1}^{n_i} g_{ij} = 1$. Using these weights we can calculate $S_i = \sum_{j=1}^{n_i} g_{ij}C_{ij}(x)$.

An interesting special case of this is where some criteria C_{ij} has $g_{ij} = 0$. In this case the criteria plays no role in the determination of S_i but still is able to contribute to the overall calculation of C(x).

Another available method for calculating the S_i involves the idea of combining these local weights with a tolerance level. Here we assume for each H_i we have $H_i = \{(C_{ij}, g_{ij}), j = 1, ..., n_i\}$, $g_{ij} \in [0, 1]$ and $\sum_{j=1}^{n_i} g_{ij} = 1$, where again g_{ij} is the indication at the importance of C_{ij} in calculating S_i. In addition we assume a tolerance level $\alpha_i \in [0, 1]$ associated with H_i. Using one of the methods for generating OWA weights we can obtain a set of OWA weights, V_{ik}, for k = 1 to n_i. Let nd_i be an index such $nd_i(k)$ is the index of the k largest of the $C_{ij}(x)$. That is $b_{ik} = C_{i,nd_i(k)}(x)$ is the value of the k most satisfied criteria in H_i. With $d_{ik} = g_{i,nd_{i(k)}}$ being the importance weight associated with this kth most satisfied criteria on H_i we calculate

$$h_{ik} = \frac{d_{ik} \cdot V_{ik}}{\sum_{k=1}^{n_i} d_{ik} \cdot V_{ik}}$$

Using this we calculate

$$S_i = \sum_{k=1}^{n_i} h_{ik}\, b_{ik}.$$

In the special case when $V_{ik} = \frac{1}{n_i}$ for all k this reduces to the weighted average introduced earlier, $S_i = \sum_{j=1}^{n_i} g_{ij} \cdot C_{ij}(x)$.

.

6. Conclusions

We considered multi-criteria decision problems where there is a lexicographically induced prioritization relationship over the criteria. We suggested that prioritization between criteria can be modeled by making the weights associated with a criteria dependent upon the satisfaction of the higher priority criteria. This resulted in a situation in which the weights associated with the criteria depended upon the alternative being evaluated. We implemented this using a prioritized scoring operator. We noted that this scoring operator didn't require a normalization of the weights. We showed that in the special case where the prioritization relationship among the criteria satisfies a linear ordering we can use a prioritized averaging operator.

References

[1]. Yager, R. R., "Modeling prioritized multi-criteria decision making," IEEE Transactions on Systems, Man and Cybernetics Part B 34, 2396-2404, 2004.

[2]. Yager, R. R., "On ordered weighted averaging aggregation operators in multi-criteria decision making," IEEE Transactions on Systems, Man and Cybernetics 18, 183-190, 1988.

[3]. Yager, R. R., "On the cardinality index and attitudinal character of fuzzy measures," International Journal of General Systems 31, 303-329, 2002.

[4]. Xu, Z., "An overview of methods for determining OWA weights," International Journal of Intelligent Systems 20, 843-865, 2005.

[5]. Torra, V. and Narukawa, Y., Modeling Decisions: Information Fusion and Aggregation Operators, Springer: Berlin, 2007.

[6]. O'Hagan, M., "A fuzzy neuron based upon maximum entropy-ordered weighted averaging," in Uncertainty in Knowledge Bases, edited by Bouchon-Meunier, B., Yager, R. R. and Zadeh, L. A., Springer-Verlag: Berlin, 598-609, 1990.

[7]. O'Hagan, M., "Using maximum entropy-ordered weighted averaging to construct a fuzzy neuron," Proceedings 24th Annual IEEE Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, Ca, 618-623, 1990.

[8]. Fuller, R. and Majlender, P., "On obtaining minimal variability OWA weights," Fuzzy Sets and Systems 136, 203-215, 2003.