Group Decis Negot (2018) 27:215–221
https://doi.org/10.1007/s10726-018-9555-0

An Improvement to Determining Expert Weights in Group Multiple Attribute Decision Making Problem

S. Abootalebi[1] · A. Hadi-Vencheh[1] · A. Jamshidi[1]

Published online: 15 January 2018

Abstract In this paper we present an extended version of the XC-model (Xu and Cai in Group Decis Negot 21:863–875, 2012) for group multiple attribute decision making problem. The proposed model is a linear programming model based on deviation function to find the optimal expert weights. An illustrative example is given to compare our results with XC-model.

Keywords Expert weights · Group multiple attribute decision making · Linear programming model · Deviation function · Genetic algorithm

1 Introduction

In a recent paper in this journal Xu and Cai (2012) described a method to determine expert weights in GMADM problem, hereafter called the XC-model. They first normalized all individual decision matrices of experts. Then the collective decision matrix is constructed by the weighted arithmetic averaging of individual decision matrices and the nonlinear optimization model based on deviation function between individual decision matrices and collective decision matrix is defined. Based on the derived expert weights, they aggregated the individual decision matrices to collective matrix and the

✉ A. Hadi-Vencheh
ahadi@khuisf.ac.ir

S. Abootalebi
s.abootalebi@mau.ac.ir

A. Jamshidi
ali.jamshidi@khuisf.ac.ir

1 Department of Mathematics, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran

weighted additive of each alternative values considered as the score of that alternatives. The best alternative and rank of all alternatives computed with these scores.
The authors of XC-model employ genetic algorithm (GA) to find the solution. As we know the GA algorithms are typically used to provide approximate solutions to problems that cannot be solved easily using other analytical techniques. Many optimization problems fall into this category. It may be too computationally-intensive to find an exact solution but sometimes a near-optimal solution can be effective. Due to their random nature, GA algorithms are never guaranteed to find an optimal solution for any problem. On the other hand, the famous simplex method for linear programming (LP) assumes that the objective function and constraints are linear functions of the variables. A linear function expresses proportionality—its graph is a straight line. For such problems, the simplex method is highly accurate, very fast—often hundreds of times faster than other methods—and yields the globally optimal solution in virtually all cases.
The purpose of this short paper is to present an improved version of the XC-model to determine the expert weights in GMADM problem. For this purpose, we convert the proposed nonlinear programming model in XC-model to a linear one and apply the simplex method to solve the problem instead of using the GA algorithm.

2 XC-Model (Xu and Cai 2012)

Assume that the GMADM problem is composed of n alternatives $\{x_1, x_2, \ldots, x_n\}$, m attributes $\{u_1, u_2, \ldots, u_m\}$ which their weights are $\{w_1, w_2, \ldots, w_m\}$, where $w_i \geq 0,\ i = 1, 2, \ldots, m$, and $\sum_{i=1}^{m} w_i = 1$. Also suppose that there is a group of s experts $\{e_1, e_2, \ldots, e_s\}$ with the weights $\{\lambda_1, \lambda_2, \ldots, \lambda_s\}$, where $\lambda_k \geq 0,\ k = 1, 2, \ldots, s$, and $\sum_{k=1}^{s} \lambda_k = 1$. The preferences of the kth expert consider as the entries of kth matrix that shows with A^k. The (i, j)th entry of this matrix that shows with a_{ij}^k is the preference of kth expert about jth alternative with respect to ith attribute.
In GMADM problem we have two attribute types, benefit attributes and cost attributes. For comparing and use the values of attributes all values must be normalized such that they do not have dimensions and units. In the XC-model the authors use the following transformation to normalize data

$$
\begin{aligned}
r_{ij}^k &= a_{ij}^k \Big/ \sum_{h=1}^{n} a_{ih}^k, && \text{for benefit attribute } u_i,\ j = 1, 2, \ldots, n \\
r_{ij}^k &= (1/a_{ij}^k) \Big/ \sum_{h=1}^{n} (1/a_{ih}^k), && \text{for cost attribute } u_i, \quad j = 1, 2, \ldots, n
\end{aligned}
\tag{1}
$$

Next, the deviation variable e_{ij}^k is defined in XC-model as below:

$$
e_{ij}^k = \left| r_{ij}^k - \sum_{k=1}^{s} \lambda_k r_{ij}^k \right|, \quad \text{for all} \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, s
\tag{2}
$$

where $\sum_{k=1}^{s} \lambda_k r_{ij}^k$ is considered as collective decision matrix of all individual decision matrices. Now construct the following deviation function:

$$F(\lambda) = \sum_{k=1}^{s} \sum_{j=1}^{n} \sum_{i=1}^{m} w_i e_{ij}^k = \sum_{k=1}^{s} \sum_{j=1}^{n} \sum_{i=1}^{m} w_i \left| r_{ij}^k - \sum_{k=1}^{s} \lambda_k r_{ij}^k \right| \tag{3}$$

Clearly, the above deviation should be as small as possible. So the following non-linear model should be solved:

$$\begin{aligned} &\min F(\lambda) = \sum_{k=1}^{s} \sum_{j=1}^{n} \sum_{i=1}^{m} w_i \left| r_{ij}^k - \sum_{k=1}^{s} \lambda_k r_{ij}^k \right| \\ &s.t. \qquad \lambda_k > 0, \quad k = 1, 2, \ldots, s, \quad \sum_{k=1}^{s} \lambda_k = 1 \end{aligned} \tag{4}$$

To solve the above model, Xu and Cai adopt a GA that can be described as follows:

GA algorithm

Step 1 Predefine the maximum iteration number t*, and randomly generate an initial population $\Theta^{(t)} = \{\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(p)}\}$, where $t = 0$, and $\lambda^{(l)} = \{\lambda_1^{(1)}, \lambda_2^{(l)}, \ldots, \lambda_s^{(l)}\}$ $(l = 1, 2, \ldots, p)$ are the expert weight vectors (or chromosomes). Then input the attribute weights w_i $(i = 1, 2, \ldots, m)$ and all the normalized individual decision matrices $R^k = (r_{ij}^k)_{m\times n}$ $(k = 1, 2, \ldots, s)$.

Step 2 By the optimization model (4), define the fitness function as:

$$F(\lambda^{(l)}) = \sum_{k=1}^{s} \sum_{j=1}^{n} \sum_{i=1}^{m} w_i |r_{ij}^k - \sum_{k=1}^{s} \lambda_k^{(l)} r_{ij}^k| \tag{5}$$

and then compute the fitness value $F(\lambda^{(l)})$ of each $\lambda^{(l)}$ in the current population $\Theta^{(t)}$, where $\lambda_k \geq 0,\ k = 1, 2, \ldots, s$, and $\sum_{k=1}^{s} \lambda_k = 1$.

Step 3 Create new weight vectors (or chromosomes) by mating the current weight vectors, and apply mutation and recombination as the parent chromosomes mate.

Step 4 Delete members of the current population $\Theta^{(t)}$ to make room for the new weight vectors.

Step 5 Utilize (5) to compute the fitness values of the new weight vectors, and insert these vectors into the current population $\Theta^{(t)}$.

Step 6 If there is no further decrease of the minimum fitness value, or $t = t^*$, then go to Step 7; otherwise, let $t = t + 1$, and go to Step 3.

Step 7 Output the minimum fitness value $F(\lambda^*)$ and the corresponding weight vector λ^*.

Based on the optimal weight vector λ^*, they get the collective decision matrix $R = (r_{ij})_{m\times n}$:

$$r_{ij} = \sum_{k=1}^{s} \lambda_k^* r_{ij}^k, \text{ for all } i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n \tag{6}$$

Then utilize the weighted arithmetic averaging operator:

$$r_j = \sum_{i=1}^{m} w_i r_{ij}, \quad \text{for all } j = 1, 2, \ldots, n \tag{7}$$

to aggregate all the attribute values in jth column of R, and get the overall attribute value r_j corresponding to the alternative x_j. Afterthat, they rank all the alternatives x_j, $(j = 1, 2, \ldots, n)$ and select the best one according to r_j, $(j = 1, 2, \ldots, n)$.

3 The Proposed Method

As mentioned before Xu and Cai solve the model (4) using genetic algorithm. But as we know the main disadvantage of GA is that there is no guarantee of finding global optimal solution. Indeed in GA the optimal solution heavily depends on the fitness function, hence it must be determined accurately. There are no standard method to define a fitness function and it is the sole responsibility of the user to define it. Besides, sometimes premature convergence may occur. Thus the diversity in the population is lost, which is one of the major objectives of GA. Finally, note that in GA the termination criteria are also not standardized. Until now no effective single terminator criterion has been identified. To address these issues, we recommend another strategy to solve model (4). For this purpose, we convert model (4) to an LP model which can be solved easily using any LP solver. To show that the non-linear model can be linearized, let

$$\begin{aligned} a_{ij}^k &= \tfrac{1}{2}\left(\left|r_{ij}^k - \sum_{k=1}^{s} \lambda_k r_{ij}^k\right| + r_{ij}^k - \sum_{k=1}^{s} \lambda_k r_{ij}^k\right), \\ b_{ij}^k &= \tfrac{1}{2}\left(\left|r_{ij}^k - \sum_{k=1}^{s} \lambda_k r_{ij}^k\right| - \left(r_{ij}^k - \sum_{k=1}^{s} \lambda_k r_{ij}^k\right)\right) \end{aligned} \tag{8}$$

Then, the model (4) is transformed to the following LP model:

$$\begin{aligned} \min F(\lambda) &= \sum_{k=1}^{s} \sum_{j=1}^{n} \sum_{i=1}^{m} w_i (a_{ij}^k + b_{ij}^k) \\ s.t. \quad & \sum_{k=1}^{s} \lambda_k = 1 \\ & a_{ij}^k - b_{ij}^k = r_{ij}^k - \sum_{k=1}^{s} \lambda_k r_{ij}^k, \ i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, n \\ & a_{ij}^k \geq 0, \ b_{ij}^k \geq 0, \ \lambda_k > 0, \ k = 1, 2, \ldots, s, \ i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, n. \end{aligned} \tag{9}$$

The model (9) is linear and can be solved easily. Evidently, using an LP solver leads to the optimal and exact solution of this model.

Table 1 Decision matrix A_1

	x_1	x_2	x_3	x_4
u_1	6.5	11	6.5	11
u_2	6	7.5	5	6.5
u_3	6	7	4.5	5
u_4	0.7	1.7	0.5	1.4

Table 2 Decision matrix A_2

	x_1	x_2	x_3	x_4
u_1	7	10	7	9
u_2	5.5	7	5	6
u_3	6.5	6	4	5.5
u_4	0.6	1.5	0.4	1.3

Table 3 Decision matrix A_3

	x_1	x_2	x_3	x_4
u_1	6	12	7	10
u_2	5	6.5	4.5	6.5
u_3	6	7.5	5	4
u_4	0.6	1.3	0.5	1.5

4 Illustrative Example

We applied our method, to the same GMADM problem as discussed in Xu and Cai (2012). An investment company is planning to exploit a new model of cars and there are four feasible alternatives x_j, $(j = 1, 2, 3, 4)$. When making a decision, the attributes considered are as follows: u_1: investment amount ($100,000.000); u_2: expected net-profit amount ($100,000.000); u_3: venture profit amount ($100,000.000); and u_4: venture-loss amount ($100,000.000). Among these four attributes, u_2 and u_3 are of benefit type; u_1 and u_4 are of cost type. The weight vector of the attributes u_i, $(i = 1, 2, 3, 4)$ is $w = (0.3, 0.2, 0.2, 0.3)$. An expert group is formed which consists of three experts e_k $(k = 1, 2, 3)$. These experts evaluate the alternatives x_j $(j = 1, 2, 3, 4)$ with respect to the attributes u_i $(i = 1, 2, 3, 4)$, and construct the following three decision matrices (see Tables 1, 2, 3):
By (1), we first normalize the decision matrices A_k $(k = 1, 2, 3)$ into the normalized decision matrices R_k $(k = 1, 2, 3)$ (see Tables 4, 5, 6):
Based on the normalized decision matrices R_k $(k = 1, 2, 3)$, the weight vector obtained by XC-model is $\lambda^* = (0.445, 0.318, 0.237)$ while solving our proposed linear model provides the expert weights as $\lambda^* = (0.7006, 0.1437, 0.1557)$. As we see the first expert e_1 has the maximum weight with both methods. So e_1 plays an important role in decision making process. With XC-model e_2 and e_3 ranked second and third expert, respectively. But using proposed method they ranked third and second expert, respectively. Now we rank the alternatives based on the derived weights.

Table 4 Decision matrix R_1

	x_1	x_2	x_3	x_4
u_1	0.314	0.186	0.314	0.186
u_2	0.240	0.300	0.200	0.260
u_3	0.267	0.311	0.200	0.222
u_4	0.302	0.124	0.423	0.151

Table 5 Decision matrix R_2

	x_1	x_2	x_3	x_4
u_1	0.288	0.201	0.288	0.224
u_2	0.234	0.298	0.213	0.255
u_3	0.295	0.273	0.182	0.250
u_4	0.297	0.119	0.446	0.137

Table 6 Decision matrix R_3

	x_1	x_2	x_3	x_4
u_1	0.338	0.169	0.290	0.203
u_2	0.222	0.289	0.200	0.289
u_3	0.267	0.333	0.222	0.178
u_4	0.327	0.151	0.392	0.131

Table 7 Collective decision matrix R with proposed model

	x_1	x_2	x_3	x_4
u_1	0.314	0.186	0.307	0.194
u_2	0.236	0.298	0.202	0.264
u_3	0.271	0.309	0.201	0.219
u_4	0.305	0.127	0.421	0.146

The collective decision matrix based on our model is shown in Table 7. The overall attribute values r_j $(j = 1, 2, 3, 4)$ are $r_1 = 0.287$, $r_2 = 0.215$, $r_3 = 0.299$ and $r_4 = 0.199$. Based on which we get the ranking of the alternatives x_j $(j = 1, 2, 3, 4)$ as $x_3 \succ x_1 \succ x_2 \succ x_4$ which is same as XC-model.

5 Conclusion

In this paper we presented an improvement of the XC-model for group multiple attribute decision making problem. The contribution of this paper is to provide a model for deriving expert weights using a linear optimization which can be solved easily. The proposed model provide an exact optimal solution. An illustrative example is presented to compare our model with the XC-model.

Reference

Xu Z, Cai X (2012) Minimizing group discordance optimization model for deriving expert weights. Group Decis Negot 21:863–875