Solving Mathematical Programming Problems with OWA Operators as Objective Functions

Ronald R. Yager
Machine Intelligence Institute
Iona College
New Rochelle, NY 10801
USA

ABSTRACT

The ordered weighted averaging operators are introduced and some of their properties described. Attention is then focused on the problem of maximizing an objective function which is of the form of an OWA aggregation of a group of variables that are interrelated and constrained by a collection of linear inequalities. It is shown how this problem can be modeled as a integer linear programming problem. Use is made of this procedure to provide a solution to fuzzy linear programming problems in which some linguistically proscribed number of the goals must be satisfied.

1. Introduction

The concept of **Ordered Weighted Averaging (OWA)** operators was introduced by Yager [1] as a way for providing aggregations which lie between the Max and Min operators. The structure of this operator involves a type of nonlinearity in the form of an ordering operation on the elements to be aggregated. As with the case of most nonlinearities this step introduces degree of complication into the procedure. In this work we are concerned with the problem of maximizing an OWA aggregation of a collection of variables. The variables are interrelated and are constrained by a collection of linear inequalities. This situation is representable as a mathematical programming problem which has linear constraints but a nonlinear objective function, the OWA aggregation. As we show in this work the nature of the nonlinearity, an ordering operation, allows us to convert this nonlinear mathematical programming problem into a linear programming problem but one in which we have to introduce some integer variables, actually binary variables. Thus it appears that the nature of the reordering type nonlinearity inherent in the OWA aggregation is in some sense equivalent to (modelable by) the introduction of some collection of binary variables.

In the following we review some basic ideas of the OWA aggregation operator [1, 2].

Definition: An ordered weighted averaging (OWA) operator of dimension n is a mapping

$$f: R^n \to R$$

that has an associated n vector W

$$W = [w_1, w_2, \ldots w_n]^T$$

such that

1. $w_i \in [0, 1]$
2. $\sum_i w_i = 1$

Furthermore, $f(a_1, \ldots a_n) = \sum_j w_j b_j$ where b_j is the j^{th} largest of the a_i.

A fundamental aspect of these operators is the re-ordering step. In particular, a weight w_i is not associated with a specific argument but with an ordered position of the aggregate. This ordering operation essentially provides a nonlinear aspect to this aggregation operation.

A number of properties can be associated with these operators [1, 2]. The OWA Operators are

1. commutative
2. monotonic
3. idempotent

The satisfaction of these three conditions assures these operators of being in the class of operators called mean operators. It can also be shown that the OWA aggregation is bounded by the Min and Max of the arguments. Thus, for any OWA aggregation f

0-7803-2461-7/95/$4.00

$$\text{Min}_i[a_i] \leq f(a_1, \ldots a_n) \leq \text{Max}_i[a_i].$$

The OWA operator provides for the implementation of a large class of mean operators. In [3] Yager discusses a number of families of these operators. The various different mean operators are implemented by appropriate selection of the weights in the associated weighting vector.

2. Mathematical Programming with OWA Objectives

In this section, we introduce and provide a basic solution to the problem of constrained OWA aggregation. In the constrained OWA aggregation problem we are interested in finding the maximal value for some objective function which is expressed as an OWA aggregation. Furthermore the variables appearing in this objective function are assumed to be constrained by some collection of linear equalities or inequalities. The use of an OWA objective function allows us to represent very complex and sophisticated objectives. As we shall see the price we pay for this increase in representational capacity when using OWA Objective functions is an increase in the difficulty of the resulting mathematical programming (MP) problem. In particular we must solve an integer programming problem.

Formally the constrained OWA aggregation problem can be expressed as the following mathematical programming problem:

Max: $F(x_1, \ldots x_n)$

Subject to:

$$a_{11}x_1 + a_{12}x_2 + \ldots a_{1n}x_n \leq b_1$$
$$a_{21}x_2 + a_{22}x_2 + \ldots a_{2n}x_n \leq b_2$$
$$\vdots$$
$$a_{m1}x_2 + a_{m2}x_2 + \ldots a_{mn}x_n \leq b_m$$
$$x_i \in R_i$$

In the above, it is assumed that F is a type of OWA aggregation of the x_i defined via a particular weighting vector W. We see that the above problem is a kind of mathematical programming problem in which only the objective function is nonlinear. We note that the nonlinearity in the objective function is of a type involving the ordering of the elements, the x_i's An example of the above formulation would be a situation in which we desire to find a solution to the constraints which has the Maximum Minimum value for the x_i's. In this case our objective would be to find Maximum of $\text{Min}_i[x_i]$ and our OWA aggregation function F would be characterized by the weighing vector $W*$ in which $w_n = 1$ and all other weights are equal to zero. On the other hand if our objective was to find a solution to the constraints which has the Maximum Maximum value for the x_i's then our OWA aggregation function F would be characterized by the weighing vector W^* in which $w_1 = 1$ and all other weights are equal to zero.

As we shall subsequently see the type of nonlinearity in the objective function, an ordering of the variables, will yield to a solution that is still a linear programming problem but one that requires the introduction of some integer, actually binary, variables. Thus the solution to the above constrained OWA aggregation but be of the form of mixed integer linear programming problem [4]. However the current state of the technology provides us with very efficient algorithms for solving these problems.

In the following, we shall let A denote the $m \times n$ matrix whose elements are the a_{ij}, let B be the m vector whose elements are b_i, and X be the n vector whose elements are x_i. Using this notation, we can express our basic problem as

Max: $F(X)$

Subject to:

$$AX \leq B$$
$$X \in R_i$$

A simple example of the type of problem we have in mind is

Max: $\text{Min}(x_1, x_2, x_3)$

Subject to:

$$x_1 + x_2 + x_3 \leq 1$$
$$0 \leq x_i \leq 1$$

Here we desire to find the values for x_1, x_2, and x_3 that are at most one and have the largest smallest value for the three variables.

Another example would be

Max: Median (x_1, x_2, x_3)

Subject to:

$$x_1 + x_2 + x_3 \leq 1$$
$$0 \leq x_i \leq 1$$

Here we desire to find values for x_1, x_2, and x_3 that have the largest median and sum to at least 1.

In order to find the optimal solution to this constrained OWA aggregation problem, we need to provide some mechanism for introducing the reordering operation required in the OWA operator that appears in the objective function which we desire to maximize. The ordering type nonlinearity introduced here can be captured in a linear type programming environment, but needs the introduction of a number of binary (0–1) integer variables. Nevertheless, we still retain the linear nature of the optimization problem and become faced with what is called a mixed integer linear programming problem. The term mixed alludes to the fact that some of the variables are required to be integer, while others can take values from a continuum.

Again consider our problem

Max: F(X) (I)'

Subject to:

$AX \leq B$ (II)

We now describe the procedure used to convert this nonlinear problem, the nonlinearity being the required ordering in the F, into a linear programming problem.

Our first step is to introduce an additional collection of n variables, the ordered variables, which we denote as $y_1, y_2, \ldots y_n$. Thus y_i is the ith largest of the x_j. We shall find it convenient at times to express this as a column vector Y in which y_i is the i^{th} element. Furthermore, since F is an OWA operator, it has an associated weighting vector W which affects the type of aggregation we are trying to optimize. We shall let w_i be the i^{th} weight in this vector. Using the new ordered variables, we can express our objective function (I)' in a purely linear form as

Max: $\sum_i w_i y_i$ (I)

where y_i is the i^{th} largest of the x_j. We can very succinctly express this objective function as **Max:** $W^T Y$.

Having introduced the vector Y into the objective function, we must now proceed to introduce additional constraints in the problem to assure ourselves that y_i is the i^{th} largest of the x_j.

The first step in this process is to inform the algorithm of the required ordering of the y_i, that is that $y_1 \geq y_2 \geq y_3 \geq \ldots y_n$. We can express this information in a collection of n – 1 constraints

$$\begin{aligned} y_2 - y_1 &\leq 0 \\ y_3 - y_2 &\leq 0 \\ y_4 - y_3 &\leq 0 \\ &\vdots \\ y_n - y_{n-1} &\leq 0 \end{aligned} \qquad \text{(III)}$$

We let G be the $n - 1 \times n$ vector whose elements are shown below

$$\begin{bmatrix} -1 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & -1 & 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & -1 & 1 \end{bmatrix}$$

Using this notation, we can express (III) as

$GY \leq 0$ (III)

With the introduction of (III), our optimization problem becomes

Max: $W^T Y$ (I)

Subject to: $AX \leq B$ (II)

$GY \leq 0$ (III)

We should also note that in the above that the x_i are still constrained to lie in the interval $[u_i, v_i]$ and the all the y_i are constrained to lie in the interval $[\hat{u}, \hat{v}]$ where $\hat{u} = \text{Min}_i[u_i]$ and $\hat{v} = \text{Max}_i[v_i]$.

We must now relate the y_i to the x_j in the appropriate manner. The first step in this direction is to impose the condition that y_n is equal to the smallest (n^{th} largest) of the x_i. To assure ourselves of the satisfaction of this condition, we impose the following collection of constraints:

$y_n - x_i \leq 0$ for $i = 1, \ldots n$

A positive coefficient for w_n will assure us

that y_n will try to get as big as possible, which is the smallest of the x_i. If $w_n = 0$, then the value of y_n is irrelevant, however because of the above constraints it will never be larger then the smallest x_i.

We can express the above constraint in a vector notation if we let I be the n column vector with one element then $y_n I - X \leq 0$.

Next, we desire to impose the condition that y_{n-1} is the second smallest of the x_i. To do this, we impose the following constraints:

$$y_{n-1} - x_1 - K z_{n-1,1} \leq 0$$
$$y_{n-1} - x_2 - K z_{n-1,2} \leq 0$$
$$\vdots$$
$$y_{n-1} - x_n - K z_{n-1,n} \leq 0 \quad (IV)_{n-1}$$
$$\sum_{j=1}^{n} z_{n-1,j} \leq 1$$
$$z_{n-1,j} \in \{0, 1\} \qquad j = 1, \ldots n$$

We emphasize that $z_{n-1,j}$ is restricted to be a binary integer variable, takes either 1 or 0 as a value. The need for introducing $z_{n-1,j}$ forces us to have an integer programming problem.

In the above, K is some constant much larger than any possible value that any of the x_j or y_i can assume. To see how this works, we note that if $z_{n-1,j} = 0$, then we effectively have

$$y_{n-1} - x_j \leq 0,$$

and thus y_{n-1} is restricted to be not greater than x_j. On the other hand, in the case when $z_{n-1,j} = 1$ we get $y_{n-1} - x_j \leq K$. Since K is some large number compared with the values for y_{n-1} and x_j, this effectively poses no constraint. Thus, the condition of $z_{n-1,\,i} = 1$ means no restriction is imposed. In addition, since $\sum_{j=1}^{n} z_{n-1,j} \leq 1$, this means that at most one of the $z_{n-1,j}$ can be one, and in turn only one of the restrictions, $y_{n-1} \leq x_j$ is withdrawn. In order to make y_{n-1} as large as possible, a condition implicitly imposed because of the non-negative nature of w_{n-1}, the algorithm will choose to relieve the most severe restriction which requires y_{n-1} to be less than the smallest of the x_i. The relief of this restriction allows y_{n-1} to become equal to the second smallest of x_i.

We can express the constraint just introduced as a concise vector notation

$$y_{n-1} I - X - K Z_{n-1} \leq 0$$
$$I^T Z_{n-1} \leq 1 \qquad (IV)_{n-1}$$
$$Z_{n-1} \in \{0, 1\}$$

In this notation, Z_{n-1} is an n column vector whose components are $z_{n-1,i}$.

We now turn to the situation of obtaining y_{n-2}, the third smallest of the x_i. In order to obtain this, we impose the constraints

$$y_{n-2} - x_1 - K z_{n-2,1} \leq 0$$
$$y_{n-2} - x_2 - K z_{n-2,2} \leq 0$$
$$\vdots$$
$$y_{n-2} - x_n - K z_{n-2,2} \leq 0 \quad (IV)_{n-2}$$
$$\sum_{j=1}^{n} z_{n-2,j} \leq 2$$
$$z_{n-2,j} \in \{0, 1\}$$

We note here that two of the constraints $y_{n-2} - x_j - K z_{n-2,j} \leq 0$ are allowed to be relaxed, and hence y_{n-2} can be evaluated to the third smallest value.

Expressing the above in vector notation we get

$$y_{n-2} I - X - K Z_{n-2} \leq 0$$
$$I^T Z_{n-2} \leq 2$$
$$Z_{n-2} \in (0, 1)$$

Again, Z_{n-2} is an n binary integer vector.

Generalizing the results, we see that for y_j we get the following:

$$y_j I - X - K Z_j \leq 0$$
$$I^T Z_j \leq n - j$$
$$Z_j \in \{0, 1\}$$

We now see the final structure of the mixed integer programming problem we must solve in order to solve the constrained OWA optimization problem.

Max: $W^T Y$ (I)

Subject to:

$AX \leq B$ (II)

$$GY \leq 0 \qquad \text{(III)}$$
$$y_j I - X - KZ_j \leq 0 \qquad \text{for } j = 1, n$$
$$I^T Z_j \leq n - j \qquad \text{for } j = 1, n$$
$$Z_j \in \{0, 1\} \qquad \text{for } j = 1, n$$

In the above, we recall that Z_j is an n vector; thus, we need to introduce $n \times n$ binary integer variables to accomplish our task. Actually, all we need are $n \times (n-1)$ binary integer variables since $Z_n = 0$ because

$$I^T Z_n \leq n - n \leq 0$$

and hence $z_{n,i} = 0$

We now look at some examples of this optimization of a constrained OWA aggregation problem.

Example: Consider

Max: $F(x_1, x_2, x_3)$

Subject to: $x_1 + x_2 + x_3 \leq 1$

$$x_i \geq 0$$

We express this in our formalism as

Max: $w_1 y_1 + w_2 y_2 + w_3 y_3 \qquad \text{(I)}$

Subject to:

$$x_1 + x_2 + x_3 \leq 1 \qquad \text{(II)}$$
$$y_2 - y_1 \leq 0$$
$$y_3 - y_2 \leq 0$$
$$y_3 - x_1 \leq 0$$
$$y_3 - x_2 \leq 0$$
$$y_3 - x_3 \leq 0$$
$$y_2 - x_1 - 20 z_1 \leq 0$$
$$y_2 - x_2 - 20 z_2 \leq 0$$
$$y_2 - x_3 - 20 z_3 \leq 0$$
$$z_1 + z_2 + z_3 \leq 1$$
$$y_1 - x_1 - 20 z_4 \leq 0$$
$$y_1 - x_1 - 20 z_4 \leq 0$$
$$y_1 - x_2 - 20 z_5 \leq 0$$
$$z_1 + z_2 + z_3 \leq 2$$
$$x_i \geq 0$$
$$z_i \text{ integer} \in [0, 1]$$

The solution to this problem depends upon the function F as manifested by the weights.

3. Constrained OWA Aggregation in Fuzzy Linear Programming

In [5], Zimmermann developed the concept of fuzzy linear programming. We shall briefly describe the basic idea; more details can be found in Zimmermann's text on fuzzy set theory [6].

Zimmermann suggests that the form of one type of fuzzy linear programming is to find the crisp solution vector x such that

$$Bx \lesssim d \qquad \text{(I)}$$

$$x \geq 0$$

In the above, x is an n vector, d is an m vector, and B is an $m \times n$ matrix. The symbol $\lesssim$ denotes a fuzzified version of the crisp less than or equal to operator $\leq$. The term $Bx \lesssim d$ contains both the constraints and objective function of a classic linear programming problem which has been fuzzified by softening the constraints and introducing a satisfying idea on the objective. We shall call each row of $Bx \lesssim d$ a goal and denote it $B_i x \lesssim d_i$. As we indicated, $\lesssim$ is a fuzzification or softening of the $\leq$ operator. In this spirit each row of $Bx \lesssim d$ can be represented as a fuzzy subset $G_i(X)$ indicating the degree to which the condition $B_i x \lesssim d_i$ is satisfied. In this spirit the fuzzy linear programming problem becomes one of finding the best solution vector x that satisfies all the goals. In this case we are faced with the problem of finding x such that Max (Min $G_i(x)$). Here Min $G_i(x)$ indicates the degree to which x satisfies all the goals.

We must now represent the fuzzy subsets, the $G_i(x)$. Zimmermann suggests introducing a tolerance level p_i for each goal and defining $G_i(x)$

$$= \begin{cases} 1 & B_i x \leq d_i \\ 1 - \dfrac{B_i x - d_i}{p_i} & \text{if } d_i < B_i x \lesssim d_i + p_i \\ 0 & \text{if } B_i x > d_i + p_i \end{cases}$$

The function is linear inside the tolerance range p_i.

A more useful formulation of the fuzzy subset G_i can be obtained [6] if we introduce an additional variable t_i measuring the degree of violation of the i^{th} goal, $G_i(x) = 1 - \frac{t_i}{p_i}$. Using this variable, we can express the fuzzy linear programming problem as

Max: [Min g_i]

$$\textbf{Such that: } g_i = 1 - \frac{t_i}{p_i}$$
$$B_i x - t_i \le d_i \qquad \text{(II)}$$
$$t_i \le p_i$$
$$x_i \ge 0 \text{ and } t_i \ge 0$$

Implicit in the formulation of the objective function in the above is the desire to satisfy all the goals, Min g_i. In some cases, a further softening of the problem can be had if we allow for the situation in which we don't require satisfaction of all the goals, but, for example, require that "most" or some other quantifier be used to provide the imperative of the aggregation use in the objective function.

Thus, if we let Q be a quantifier which we are going to use to guide our aggregation, we can calculate the weights associated with this quantifier as

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{m}\right)$$

where m is the number of goals in the linear programming problem. We then convert our linear programming problem to

Max: $F_W(g_1, \ldots g_n)$ (I)

$$\textbf{Such that: } g_i = 1 - \frac{t_i}{p_i}$$
$$B_i x - t_i \le d_i$$
$$t_i \le p_i \qquad \text{(II)}$$
$$x_i \ge 0 \text{ and } t_i \ge 0$$

where F_W is the OWA aggregation operator obtained using the weights w_i derived above.

In order to operationalize the ordering process required in the OWA aggregation in the objective function, we must introduce some other variables just as was done in the earlier section.

We first introduce the variables $y_1, \ldots y_m$ where y_i is indicating the i^{th} largest of the g_i. We can express the objective function as a simple linear aggregation $\sum_i w_i y_i$. We now introduce constraints to obtain the y_i. First we constrain the ordering of y_i here we need a collection of constraints $y_{i+1} - y_i \le 0$ for i = 1, m – 1 Furthermore, we introduce a collection of constraints for each y_i to assure us it is the i^{th} largest. We again need the following collection for each i = 0, . . . m – 1:

$$y_{n-i} I - g - K Z_{m-i} \le 0$$
$$I^T Z_{m-1} \le 1$$
$$Z_{m-i,j} \in \{0, 1\}$$

In the above, I, g, and Z_{m-i} are

$$I = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \; g = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix}, \; Z_{n-i} = \begin{bmatrix} z_{m-i,1} \\ z_{m-i,2} \\ \vdots \\ z_{m-i,2} \end{bmatrix}$$

4. References

[1]. 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.

[2]. Yager, R. R. and Filev, D. P., Essentials of Fuzzy Modeling and Control, John Wiley: New York 1994.

[3]. Yager, R. R., "Families of OWA operators," Fuzzy Sets and Systems 59, 125-148, 1993.

[4]. Plane, D. R. and McMillan, C., Discrete Optimization, Prentice-Hall, 1971.

[5]. Zimmermann, H. J., "Description and optimization of fuzzy systems," International Journal of General Systems 2, 209-215, 1976.

[6]. Zimmermann, H. J., Fuzzy Set Theory--and Its Applications, Kluwer-Nijhoff: Dordrecht, 1985.