

## Article

# An Iterative Approach for the Solution of the Constrained OWA Aggregation Problem with Two Comonotone Constraints

Lucian Coroianu<sup>1</sup> and Robert Fullér<sup>2,\*</sup> <sup>1</sup> Department of Mathematics and Informatics, University of Oradea, 410087 Oradea, Romania<sup>2</sup> Department of Informatics, Széchenyi István University, 9026 Győr, Hungary

\* Correspondence: rfuller@sze.hu

**Abstract:** In this paper, first, we extend the analytical expression of the optimal solution of the constrained OWA aggregation problem with two comonotone constraints by also including the case when the OWA weights are arbitrary non-negative numbers. Then, we indicate an iterative algorithm that precisely indicates whether a constraint in an auxiliary problem is either bidding or strictly redundant. Actually, the bidding constraint (or two bidding constraints, as this case also may occur) are essential in expressing the solution of the initial constrained OWA aggregation problem.

**Keywords:** ordered weighted averaging; constrained OWA aggregation; comonotone constraints



**Citation:** Coroianu, L.; Fullér, R. An Iterative Approach for the Solution of the Constrained OWA Aggregation Problem with Two Comonotone Constraints. *Information* **2022**, *13*, 443. <https://doi.org/10.3390/info13100443>

Academic Editors: Luis Martínez López, Lesheng Jin, Zhen-Song Chen and Humberto Bustince

Received: 31 August 2022

Accepted: 18 September 2022

Published: 21 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Since the introduction of ordered weighted averaging operators (OWA operators) by Yager in [1], this topic has attracted huge interest in both theoretical and practical directions. For detailed accounts on the state of the art, we recommend, for example, the works [2–5]. In this paper, our goal is to continue the investigation of the so-called constrained OWA aggregation problem. Here, the goal is to optimize the OWA operator under linear constraints. Yager started this research in [6] by proposing an algorithm for the maximization problem that can solve the problem in some special cases. Then, in [7], the authors solve the problem in the special case when we have one restriction and all coefficients in the constraint are equal. In paper [8], the result is generalized; this time, the coefficients in the single constraint are arbitrary. Another approach for this case can be also found in the recent paper [9]. The minimization problem in the case of a single constraint is solved in [10]. Recently, in the paper [3], the authors found a way to solve the maximization and minimization problems in the case when we have two comonotone constraints. In this contribution, we continue the work started in [3]. We will discuss the maximization problem since the results for the minimization problem can be easily deduced using the patterns from the papers [3,10]. First, we find a simple way to generalize the main results in [3] for the case when the OWA weights are arbitrary non-negative numbers. In [3], we assumed these weights to be strictly positive to avoid division with zero in some cases. However, these cases can be eliminated as they will give some redundant constraints. In this way, the results apply for OWA operators having some weights that can be equal to zero, such as, for example the Olympic weights (see, e.g., [2]). We reiterate again as in others of our papers that for solving such problems, it seems that one effective approach is to use the dual of some linear programs derived from the initial constrained OWA aggregation problem. Other optimization problems also use the dual of linear programs (see [11,12]). In the papers mentioned earlier, the idea is to optimize the OWA operator under some linear constraints. Another problem which is of great interest among researchers is to optimize the OWA weights under some additional constraints (see, e.g., [2,5,13,14]). Finally, let us also mention that the study of OWA type operators and their generalizations is a dynamic process, and numerous interesting directions have opened in recent years (see, e.g., [15–19]).

In Section 2, we present the constrained OWA aggregation problem with comonotone constraints and, in the special case of two comonotone constraints, we extend the main results proved in [3] for the case when the OWA weights are arbitrary non-negative numbers. In Section 3, we present the iterative algorithm for an auxiliary problem associated with the initial problem that finds at every step a constraint that is either binding or strictly redundant. In Section 4, we test this algorithm on concrete examples, and we also discuss its proficiency. The paper ends with conclusions that sum up the main contributions as an important step to the general setting of an arbitrary number of constraints.

### 2. Constrained OWA Aggregation with Comonotone Constraints

In this section, we recall briefly the basics on the constrained OWA aggregation problem. These details can be found in numerous papers, and we use here similar arguments as in [3].

Suppose we have the non-negative weights  $w_1, \dots, w_n$  such that  $w_1 + \dots + w_n = 1$  and define a mapping  $F : \mathbb{R}^n \rightarrow \mathbb{R}$ ,

$$F(x_1, \dots, x_n) = \sum_{i=1}^n w_i y_i,$$

where  $y_i$  is the  $i$ -th largest element of the sample  $x_1, \dots, x_n$ . Then consider a matrix  $A$  of type  $(m, n)$  with real entries and a vector  $b \in \mathbb{R}^m$ . A constrained maximum OWA aggregation problem corresponding to the above data is the problem (see [6])

$$\max F(x_1, \dots, x_n) \text{ subject to } Ax \leq b, x \geq 0. \tag{1}$$

Let us recall now two particular problems where the coefficients in the constraints can be rearranged to satisfy certain monotonicity properties. The maximization problem is

$$\begin{cases} \max F(x_1, \dots, x_n), \\ \alpha_{i1}x_1 + \dots + \alpha_{in}x_n \leq 1, \text{ for all } i = 1, \dots, m, \\ x \geq 0, \alpha_{ij} > 0, \end{cases} \tag{2}$$

and there exists a permutation  $\sigma \in S_n$  such that

$$\alpha_{i\sigma_1} \leq \alpha_{i\sigma_2} \leq \dots \leq \alpha_{i\sigma_n}, i = \overline{1, m}. \tag{3}$$

Here,  $S_n$  denotes the set of all permutations of  $\{1, \dots, n\}$ , and for some  $\sigma \in S_n$ , we use the notation  $\sigma_k$  for the value  $\sigma(k)$  for any  $k \in \{1, \dots, n\}$ . From now on, we will say that the constraints in problem (2) are comonotone whenever condition (3) is satisfied. The minimization problem is

$$\begin{cases} \min F(x_1, \dots, x_n), \\ \alpha_{i1}x_1 + \dots + \alpha_{in}x_n \geq 1, \text{ for all } i = 1, \dots, m, \\ x \geq 0, \alpha_{ij} > 0, \end{cases} \tag{4}$$

and, again, there exists  $\sigma \in S_n$  such that

$$\alpha_{i\sigma_1} \geq \alpha_{i\sigma_2} \geq \dots \geq \alpha_{i\sigma_n}, i = \overline{1, m}. \tag{5}$$

Obviously, in the minimization problem above, the constraints are comonotone as well.

Considering the general problem (1), Yager used a method based on mixed integer linear programming to approach the optimal solution. The method is quite complex since it requires to introduce auxiliary variables, sometimes causing difficulties in calculations. When the single constraint is particularized to  $x_1 + \dots + x_n = 1$ , the problem was solved completely in [7] by providing an analytical solution as a function of the weights. Furthermore, considering arbitrary coefficients in the single constraint, in paper [8], the analytical

solution was obtained as a function depending on the weights and on the coefficients in the constraint. This problem can be formulated as

$$\max F(x_1, \dots, x_n) \text{ subject to } \alpha_1 x_1 + \dots + \alpha_n x_n \leq 1, x \geq 0. \tag{6}$$

The following theorem recalls the main result from [8]. In what follows,  $S_n$  denotes the set of permutations of the set  $\{1, \dots, n\}$ .

**Theorem 1** (see [8], Theorem 3). *Consider problem (6). Then:*

- (i) *If there exists  $i_0 \in \{1, \dots, n\}$  such that  $\alpha_{i_0} \leq 0$ , then  $F$  is unbounded on the feasible set, and its supremum over the feasible set is  $\infty$ ;*
- (ii) *If  $\alpha_i > 0, i \in \{1, \dots, n\}$ , then taking (any)  $\sigma \in S_n$  with the property that  $\alpha_{\sigma_1} \leq \alpha_{\sigma_2} \leq \dots \leq \alpha_{\sigma_n}$ , and  $k^* \in \{1, \dots, n\}$ , such that*

$$\frac{w_1 + \dots + w_{k^*}}{\alpha_{\sigma_1} + \dots + \alpha_{\sigma_{k^*}}} = \max \left\{ \frac{w_1 + \dots + w_k}{\alpha_{\sigma_1} + \dots + \alpha_{\sigma_k}} : k \in \{1, \dots, n\} \right\},$$

*then  $(x_1^*, \dots, x_n^*)$  is an optimal solution of problem (6), where*

$$\begin{aligned} x_{\sigma_1}^* &= \dots = x_{\sigma_{k^*}}^* = \frac{1}{\alpha_{\sigma_1} + \dots + \alpha_{\sigma_{k^*}}}, \\ x_{\sigma_{k^*+1}}^* &= \dots = x_{\sigma_n}^* = 0. \end{aligned}$$

*In particular, if  $0 < \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n$ , and  $k^* \in \{1, \dots, n\}$  is such that*

$$\frac{w_1 + \dots + w_{k^*}}{\alpha_1 + \dots + \alpha_{k^*}} = \max \left\{ \frac{w_1 + \dots + w_k}{\alpha_1 + \dots + \alpha_k} : k \in \{1, \dots, n\} \right\},$$

*then  $(x_1^*, \dots, x_n^*)$  is a solution of (6), where*

$$\begin{aligned} x_1^* &= \dots = x_{k^*}^* = \frac{1}{\alpha_1 + \dots + \alpha_{k^*}}, \\ x_{k^*+1}^* &= \dots = x_n^* = 0. \end{aligned}$$

An analogue to the minimization problem can be found in [10] (see Theorem 2 there).

Let us now discuss the case when we have two comonotone constraints. We will discuss the maximization problem here by briefly recalling the reasoning used in [3].

Consider the problem

$$\begin{cases} \max F(x_1, \dots, x_n), \\ \alpha_1 x_1 + \dots + \alpha_n x_n \leq 1, \\ \beta_1 x_1 + \dots + \beta_n x_n \leq 1, \\ x \geq 0, \end{cases} \tag{7}$$

in the case when  $0 < \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n$  and  $0 < \beta_1 \leq \beta_2 \leq \dots \leq \beta_n$ . In what follows, we consider the special case when the weights are positive. To obtain an analytical representation of the optimal solution, several linear programs are associated with this problem. By Theorem 1 in [3], any solution of problem

$$\begin{cases} \max(w_1 x_1 + \dots + w_n x_n), \\ \alpha_1 x_1 + \dots + \alpha_n x_n \leq 1, \\ \beta_1 x_1 + \dots + \beta_n x_n \leq 1 \\ x_1 \geq \dots \geq x_n \geq 0, \end{cases} \tag{8}$$

is a solution of problem (7). Note that both problems have nonempty solution sets.

To solve (8), we need its dual, which is

$$\left\{ \begin{array}{l} \min(t_1 + t_2), \\ \alpha_1 t_1 + \beta_1 t_2 - t_3 \geq w_1, \\ \alpha_2 t_1 + \beta_2 t_2 + t_3 - t_4 \geq w_2, \\ \alpha_3 t_1 + \beta_3 t_2 + t_4 - t_5 \geq w_3, \\ \vdots \\ \alpha_{n-1} t_1 + \beta_{n-1} t_2 + t_n - t_{n+1} \geq w_{n-1}, \\ \alpha_n t_1 + \beta_n t_2 + t_{n+1} \geq w_n \\ t_1 \geq 0, t_2 \geq 0, \dots, t_{n+1} \geq 0. \end{array} \right. \tag{9}$$

Furthermore, we can simplify this problem by introducing the problem

$$\left\{ \begin{array}{l} \min(t_1 + t_2), \\ \alpha_1 t_1 + \beta_1 t_2 \geq w_1, \\ (\alpha_1 + \alpha_2) t_1 + (\beta_1 + \beta_2) t_2 \geq w_1 + w_2, \\ \vdots \\ \left( \sum_{i=1}^k \alpha_i \right) \cdot t_1 + \left( \sum_{i=1}^k \beta_i \right) \cdot t_2 \geq \sum_{i=1}^k w_i, \\ \vdots \\ \left( \sum_{i=1}^n \alpha_i \right) \cdot t_1 + \left( \sum_{i=1}^n \beta_i \right) \cdot t_2 \geq \sum_{i=1}^n w_i, \\ t_1 \geq 0, t_2 \geq 0. \end{array} \right. \tag{10}$$

Here, we make the first improvement with respect to the reasoning used in [3]. Namely, if  $w_k = 0$  for some  $k \in \{1, \dots, n\}$ , then constraint number  $k$  is redundant in (10). Therefore, problem (10) is equivalent to problem

$$\left\{ \begin{array}{l} \min t_1 + t_2, \\ \left( \sum_{i=1}^k \alpha_i \right) \cdot t_1 + \left( \sum_{i=1}^k \beta_i \right) \cdot t_2 \geq \sum_{i=1}^k w_i, k \in I_n \\ t_1 \geq 0, t_2 \geq 0, \end{array} \right. \tag{11}$$

where  $I_n = \{k \in \{1, \dots, n\} : w_k > 0\}$ . This improvement will allow us to investigate problems where some of the weights can be equal to 0, such as, for example the Olympic weights (see, e.g., [2]). As we mentioned in [3], if  $(t_1^*, t_2^*, \dots, t_{n+1}^*)$  is a solution of problem (9), then  $(t_1^*, t_2^*)$  is a feasible solution for problem (10), and consequently, it is feasible for problem (11). Now, suppose that  $(\bar{t}_1, \bar{t}_2)$  is a solution of problem (11). One can easily prove (see again [3]) that  $(\bar{t}_1, \bar{t}_2)$  extends to a feasible solution  $(\bar{t}_1, \bar{t}_2, t_3^*, \dots, t_{n+1}^*)$  of problem (9). Thus, considering only the first two components of the feasible solutions of problem (9), we obtain the same set as for the feasible set of problem (11). Obviously, both problems will have the same minimal value, which in addition is finite.

In order to find a solution for problem (11), we need to investigate a problem given as

$$\left\{ \begin{array}{l} \min(t_1 + t_2), \\ a_k \cdot t_1 + b_k \cdot t_2 \geq 1, k \in I_n, \\ t_1 \geq 0, t_2 \geq 0. \end{array} \right. \tag{12}$$

It will suffice to consider only the case when  $a_k > 0$  and  $b_k > 0$  for all  $k \in \{1, \dots, n\}$ . Taking

$$a_k = \frac{\sum_{i=1}^k \alpha_i}{\sum_{i=1}^k w_i} \text{ and } b_k = \frac{\sum_{i=1}^k \beta_i}{\sum_{i=1}^k w_i}, k \in I_n, \tag{13}$$

problem (11) becomes exactly a problem of type (12). Therefore, solving problem (12) will result in solving problem (11) as well.

We need the following auxiliary result proved in [3].

**Lemma 1.** (see [3], Lemma 7) Consider problem (12), where  $a_k$  and  $b_k$  are given in (13),  $k \in I_n$ . Suppose that  $k_1, k_2 \in I_n$  are such that  $a_{k_1} \geq b_{k_1}$  and  $a_{k_2} \leq b_{k_2}$ . If  $(t_1^*, t_2^*)$  is a solution of the system

$$\begin{cases} a_{k_1} t_1 + b_{k_1} t_2 = 1, \\ a_{k_2} t_1 + b_{k_2} t_2 = 1, \end{cases}$$

and if  $(t_1^*, t_2^*)$  is feasible for problem (12), then  $(t_1^*, t_2^*)$  is an optimal solution for problem (12).

All the information above will be very useful in the next section, where we will propose an iterative algorithm to approach the solution of problem (7).

Actually, we can now characterize the optimal solution of problem (7) by slightly improving Theorem 8 in [3]. We omit the proof since one can easily deduce the necessary modifications comparing to the statement of Theorem 8 in [3] coupled with Lemma 1 from above, which stays at the bases of the next theorem.

**Theorem 2.** (see also Theorem 8 in [3] covering the case when the weights are strictly positive, that is,  $I_n = \{1, \dots, n\}$ ) Consider problem (7) in the special case when  $0 < \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n$  and  $0 < \beta_1 \leq \beta_2 \leq \dots \leq \beta_n$ . Then consider problem (12) where  $a_k$  and  $b_k$  are obtained using the substitutions given in (13),  $k \in I_n$ . Furthermore, take  $l = \arg \min\{a_k : k \in I_n\}$  and  $p = \arg \min\{b_k : k \in I_n\}$ . We have the following cases (not necessarily distinct but covering all possible scenarios) in obtaining the optimal solution and the optimal value of problem (7).

(i) If  $a_l \geq b_l$ , then  $(x_1^*, \dots, x_n^*)$  is a solution of problem (7), where

$$\begin{aligned} x_1^* &= \dots = x_l^* = \frac{1}{\alpha_1 + \dots + \alpha_l}, \\ x_{l+1}^* &= \dots = x_n^* = 0. \end{aligned}$$

In addition,  $\frac{w_1 + \dots + w_l}{\alpha_1 + \dots + \alpha_l}$  is the optimal value of problem (7).

(ii) If  $a_p \leq b_p$ , then  $(x_1^*, \dots, x_n^*)$  is a solution of problem (7), where

$$\begin{aligned} x_1^* &= \dots = x_p^* = \frac{1}{\beta_1 + \dots + \beta_p}, \\ x_{p+1}^* &= \dots = x_n^* = 0. \end{aligned}$$

In addition,  $\frac{w_1 + \dots + w_p}{\beta_1 + \dots + \beta_p}$  is the optimal value of problem (7).

(iii) If in problem (12) there exists a binding constraint  $C_{k_0}$  such that  $a_{k_0} = b_{k_0}$ , then  $(x_1^*, \dots, x_n^*)$  is a solution of problem (7), where

$$\begin{aligned} x_1^* &= \dots = x_{k_0}^* \\ &= \frac{1}{\alpha_1 + \dots + \alpha_{k_0}}, \\ x_{k_0+1}^* &= \dots = x_n^* = 0. \end{aligned}$$

In addition,  $\frac{w_1+\dots+w_{k_0}}{\alpha_1+\dots+\alpha_{k_0}}$  is the optimal value of problem (7).

(iv) If in problem (12) the optimal solution satisfies with equality the constraints  $C_{k_1}$  and  $C_{k_2}$ , where  $k_1 < k_2$  and  $(a_{k_1} - b_{k_1}) \cdot (a_{k_2} - b_{k_2}) < 0$ , then  $(x_1^*, \dots, x_n^*)$  is a solution of problem (7), where

$$\begin{aligned}
 x_i^* &= \frac{\sum_{i=k_1+1}^{k_2} (\beta_i - \alpha_i)}{\left(\sum_{i=1}^{k_1} \alpha_i\right) \cdot \left(\sum_{i=k_1+1}^{k_2} \beta_i\right) - \left(\sum_{i=1}^{k_1} \beta_i\right) \cdot \left(\sum_{i=k_1+1}^{k_2} \alpha_i\right)}, i = \overline{1, k_1} \\
 x_i^* &= \frac{\sum_{i=1}^{k_1} (\alpha_i - \beta_i)}{\left(\sum_{i=1}^{k_1} \alpha_i\right) \cdot \left(\sum_{i=k_1+1}^{k_2} \beta_i\right) - \left(\sum_{i=1}^{k_1} \beta_i\right) \cdot \left(\sum_{i=k_1+1}^{k_2} \alpha_i\right)}, i = \overline{k_1 + 1, k_2}, \\
 x_{k_2+1}^* &= \dots = x_n^* = 0.
 \end{aligned}$$

In addition, the optimal value of problem (7) is equal to

$$\frac{\left(\sum_{i=1}^{k_1} w_i\right) \cdot \left(\sum_{i=k_1+1}^{k_2} (\beta_i - \alpha_i)\right) + \left(\sum_{i=k_1+1}^{k_2} w_i\right) \cdot \left(\sum_{i=1}^{k_1} (\alpha_i - \beta_i)\right)}{\left(\sum_{i=1}^{k_1} \alpha_i\right) \cdot \left(\sum_{i=k_1+1}^{k_2} \beta_i\right) - \left(\sum_{i=k_1+1}^{k_2} \alpha_i\right) \cdot \left(\sum_{i=1}^{k_1} \beta_i\right)}.$$

**Remark 1.** Using the above theorem, we can also generalize the results given in Theorems 9 and 10, respectively, in paper [3]. In Theorem 9, we considered the case when there exists a permutation  $\sigma \in S_n$  such that  $0 < \alpha_{\sigma_1} \leq \alpha_{\sigma_2} \leq \dots \leq \alpha_{\sigma_n}$  and  $0 < \beta_{\sigma_1} \leq \beta_{\sigma_2} \leq \dots \leq \beta_{\sigma_n}$ . Then, in Theorem 10, we considered the case when  $0 < \alpha_1 = \alpha_2 = \dots = \alpha_n = \alpha$ . Obviously, in view of Theorem 3 from above, all these results can be extended for the more general case when the weights in problem (7) are only assumed to be non-negative. Then, of course, we can state similar refinements for the case of minimization problem. They are easily deduced from the corresponding results obtained in paper [3] (see Theorems 11–13, respectively, in [3]).

### 3. An Iterative Algorithm to Achieve the Optimal Solution

In this section we propose an iterative algorithm to obtain the optimal solution in Problem (7). Although the computer implementation of Theorem 2 can be done in a very convenient way based on the Simplex algorithm (see the examples in [3]), the following algorithm has an interesting particularity; namely, it can identify a constraint that is either binding or redundant. In this way, we can eliminate the constraints one by one until we obtain the optimal solution. We also hope this algorithm can be generalized for the case when we have more than two comonotone constraints; however, this remains an interesting open question in our opinion.

To construct our algorithm, we need to investigate Problem (12), where  $a_k$  and  $b_k$  are given in (13),  $k \in I_n$ . We also need some concepts and notations that are well-known in linear programming. Let us denote with  $C_k$  constraint number  $k$  of problem (12),  $k \in I_n$ . Then, we denote with  $U$  the feasible region of problem (10). Next, for some  $k \in I_n$ , let  $P_k = I_n \setminus \{k\}$  and

$$U_k = \{(t_1, t_2) \in [0, \infty) \times [0, \infty) : a_i \cdot t_1 + b_i \cdot t_2 \geq 1, i \in P_k\}.$$

In other words,  $U_k$  is the feasible region of any optimization problem which keeps all the constraints from Problem (12) except for constraint  $C_k$ . The constraint  $C_k$  is called redundant if  $U = U_k$ . In other words, the solution set of the optimization problem with feasible region  $U$  coincides with the solution set of the optimization problem that has the same objective function and all the constraints except for  $C_k$ . This means that constraint

$C_k$  can be removed when solving the given problem. The constraint  $C_k$  is called strongly redundant if it is redundant and

$$a_k t_1 + b_k t_2 > 1, \text{ for all } (t_1, t_2) \in U.$$

Therefore,  $C_k$  is strongly redundant if and only if the segment which corresponds to the solutions of the equation  $a_k t_1 + b_k t_2 = 1, t_1 \geq 0, t_2 \geq 0$ , does not intersect  $U$ . A redundant constraint that is not strongly redundant is called weakly redundant. The constraint  $C_k$  is called binding if there exists at least one optimal point which satisfies this constraint with equality. This means that the segment corresponding to the equation  $a_k t_1 + b_k t_2 = 1, t_1 \geq 0, t_2 \geq 0$ , contains an optimal point of the problem. Note that it is possible for a weakly redundant constraint to be binding as well. All these concepts were discussed with respect to our Problem (12), but of course, they can be defined accordingly for any kind of optimization problem.

Now, with these new tools, we can investigate Problem (12). As we said in the introduction, searching for binding constraints may be just as difficult as solving the program. However, we can easily spot some binding constraints in Problem (12). We also believe it is worthwhile to do that as otherwise all constraints are needed when performing the algorithm, and therefore, this calculation will be more complex than those used to eliminate the redundant constraints. In general, if  $k_1, k_2 \in I_n$  are such that  $a_{k_1} \leq a_{k_2}$  and  $b_{k_1} \leq b_{k_2}$ , then constraint  $C_{k_2}$  is redundant, and it can be eliminated. Using this fact, we propose a simple method to eliminate some redundant constraints. First, we set  $M_1 = I_n$  and let  $N_1$  be the subset of  $M_1$  such that  $a_l = \min\{a_k : k \in M_1\}$ , for all  $l \in N_1$ . Then, let  $p_1$  be the index with the minimum value (just to make a choice if more indices would satisfy the following property) in  $N_1$  such that  $b_{p_1} = \min\{b_k : k \in N_1\}$ . We keep constraint  $C_{p_1}$  and eliminate all the other constraints indexed in  $N_1$  since they all are redundant. Next, for any  $k \in M_1 \setminus N_1$ , we compute  $(a_{p_1} - a_k) \cdot (b_{p_1} - b_k)$ . If this value is strictly negative, then we keep constraint  $C_k$ , and if not, then we eliminate  $C_k$  because it is redundant. Let  $M_2$  be the set of indices that correspond only to the constraints that were not eliminated and which does not contain  $p_1$ . We continue with the same reasoning, with the only difference that now we take  $M_2$  instead of  $M_1$ , then define  $N_2$  with respect to  $M_2$  the same way we defined  $N_1$  with respect to  $M_1$ . We then define  $p_2$  the same way we defined  $p_1$  and so on,  $M_3, N_3$  and  $p_3$ , and so on, ..., until we get that  $M_k$  is the empty set. Note that  $k$  is at most equal to  $n - 1$ . At every step  $k$ , we select in a set that we denote with  $J$ , the index of each constraint that was not eliminated from  $N_k$ , that is,  $p_1, p_2$ , and so on. Thus, the constraints indexed by  $J$  will give the same feasible region as the initial set of  $n$  constraints. In addition, if  $k_1, k_2 \in J, k_1 \neq k_2$ , then  $(a_{k_1} - a_{k_2}) \cdot (b_{k_1} - b_{k_2}) < 0$ . Then, there exists at most one index  $k \in J$  such that  $a_k = b_k$ . There are other methods to obtain the set  $J$ , but in our opinion, this one is between the fastest when using the computer. In all that follows in this paper, the set  $J$  will be the one obtained with the above technique. Please note that it still may be possible to have redundant constraints among those indexed in  $J$ . Just before our first key result, we explain why it is not useful to search such constraints outside the proposed algorithm.

We are now in position to present a key result that will then give us a fast algorithm to solve Problem (12). What is really interesting in the following theorem is that we present a precise and simple method to search a constraint that will prove to be either a binding constraint or a strongly redundant constraint. This happens rarely in linear programming, and it also explains why it is not necessary to search for redundant constraints separately. This is indeed an important advantage mainly because the techniques to find redundant inequalities involve a lot of computation.

**Theorem 3.** Consider Problem (12), where  $a_k$  and  $b_k$  are given in (13),  $k \in I_n$ . Then, let  $J \subseteq I_n$  be the set of indices obtained by the technique described just before Lemma 1). Then, Let  $J_1 = \{k \in J : a_k \geq b_k\}$  and  $J_2 = \{k \in J : a_k \leq b_k\}$ . In addition, suppose that both  $J_1$  and  $J_2$  are nonempty. Then, let  $k_1 \in J_1$  be such that  $a_{k_1} = \min\{a_k : k \in J_1\}$  and  $b_{k_2} = \min\{b_k : k \in J_2\}$  (by the construction of  $J$ , it follows that  $a_{k_1}$  and  $b_{k_2}$  are unique minimizers). Then, constraint  $C_{k_1}$  is either binding, or it is strongly redundant. Similarly, constraint  $C_{k_2}$  is either binding, or it is strongly redundant.

**Proof.** Due to the absolutely similar reasoning of the assertions, we prove only the first one. Suppose that  $C_{k_1}$  is not strongly redundant. Again, let  $f(t_1, t_2) = t_1 + t_2$  be the objective function. We have two cases: (i) the point  $(1/a_{k_1}, 0)$  belongs to the feasible region and (ii)  $(1/a_{k_1}, 0)$  does not belong to the feasible region.

For case (i), since  $a_{k_1} \leq b_{k_1}$ , with reasoning as in the proof of Lemma 1, it follows that  $f(1/a_{k_1}, 0) \leq f(t_1, t_2) \leq f(0, 1/b_{k_1})$  for any feasible point  $(t_1, t_2)$  that satisfies constraint  $C_{k_1}$  with equality. Now, if  $(t_1, t_2)$  is an arbitrary feasible point, it is clear that the intersection of the segments  $[(0, 0), (t_1, t_2)]$  and  $[(1/a_{k_1}, 0), (0, 1/b_{k_1})]$  is nonempty. Let  $(u_1, u_2)$  be in this intersection. With reasoning as in the proof of Lemma 1, we obtain that  $f(t_1, t_2) \geq f(u_1, u_2) \geq f(1/a_{k_1}, 0)$ . This means that  $(1/a_{k_1}, 0)$  is an optimal solution of Problem (12).

For case (ii), from all the feasible points that satisfy constraint  $C_{k_1}$  with equality, we chose the one for which the first component has the maximum value. In other words, considering the intersection of  $[(1/a_{k_1}, 0), (0, 1/b_{k_1})]$  with the feasible region, we take the point which is nearest to  $(1/a_{k_1}, 0)$  with respect to the usual Euclidean metric in  $\mathbb{R}^2$ . Let us denote this point with  $(t_1^*, t_2^*)$ . Note that since the feasible region is a closed and convex subset of  $\mathbb{R}^2$ , this intersection is a closed segment; therefore, the construction of  $(t_1^*, t_2^*)$  is correct. Suppose that  $a_0 = \min\{a_k : k \in J\}$ . It is immediate that  $1/a_0$  is a feasible point of Problem (12). As  $1/a_{k_1}$  is not, it necessarily follows that  $a_{k_1} > a_0$ . From this property it results that there exists a constraint  $C_l$  such that  $(t_1^*, t_2^*)$  satisfies this constraint with equality and such that  $a_l < a_{k_1}$ . This property can be easily deduced using some elementary geometrical reasoning. For the sake of correctness, let us give a rigorous proof. Let us choose arbitrary  $k \in J$  such that  $a_k < a_{k_1}$  (such element exists since  $a_{k_1} > a_0$ ). In this case, the solution of the system  $a_{k_1}t_1 + b_{k_1}t_2 = 1, a_k t_1 + b_k t_2 = 1$ , must have its (unique) solution on the segment  $[(t_1^*, t_2^*), (1/a_{k_1}, 0)]$ . Otherwise, then  $(0, 0)$  and  $(t_1^*, t_2^*)$  would be on the same semispace with respect to the separating line  $a_k x + b_k y = 1$ ; hence,  $a_k t_1^* + b_k t_2^* < 1$ . This implies that  $(t_1^*, t_2^*)$  is not feasible, which is a contradiction. Now, let us choose an arbitrary  $k \in J$  such that  $a_k > a_{k_1}$ . In this case, the unique solution of system  $a_{k_1}t_1 + b_{k_1}t_2 = 1, a_k t_1 + b_k t_2 = 1, t_1 \geq 0, t_2 \geq 0$ , lies on the segment  $[(t_1^*, t_2^*), (0, 1/b_{k_1})]$ . Otherwise, we obtain the same contradiction as above. Now, by way of contradiction, suppose that for any  $k \in J$ , such that  $a_k < a_{k_1}$ , we have  $a_k t_1^* + b_k t_2^* \neq 1$ . Using the properties mentioned just above, we obtain that  $a_k t_1^* + b_k t_2^* > 1$ . Then, if  $k \in J$  is such that  $a_k > a_{k_1}$ , we obtain that  $a_k t_1 + b_k t_2 \geq 1$  for all  $(t_1, t_2) \in [(t_1^*, t_2^*), (1/a_{k_1}, 0)]$ . All these imply that there exists  $(\bar{t}_1, \bar{t}_2) \in [(t_1^*, t_2^*), (1/a_{k_1}, 0)]$  sufficiently close to  $(t_1^*, t_2^*)$ , such that  $\bar{t}_1 > t_1^*$  and such that  $a_k \bar{t}_1 + b_k \bar{t}_2 \geq 1$ , for all  $k \in J$ . This means that  $(\bar{t}_1, \bar{t}_2)$  is a feasible point for Problem (12). On the other hand,  $(\bar{t}_1, \bar{t}_2)$  satisfies constraint  $C_{k_1}$  with equality and  $\bar{t}_1 > t_1^*$ . This contradicts the construction of  $t_1^*$ . Therefore, there exists a constraint  $C_l$  such that  $(t_1^*, t_2^*)$  satisfies this constraint with equality and such that  $a_l < a_{k_1}$ . By the construction of  $a_{k_1}$ , it follows that  $a_l < b_l$ . Therefore,  $(t_1^*, t_2^*)$  is a feasible point which is a solution of the system

$$\begin{cases} a_{k_1} t_1 + b_{k_1} t_2 = 1, \\ a_l t_1 + b_l t_2 = 1, \end{cases}$$

where  $a_{k_1} \geq b_{k_1}$  and  $a_l < b_l$ . By Lemma 1, it follows that  $(t_1^*, t_2^*)$  is an optimal solution for Problem (12). The proof is complete now. In the case of constraint  $C_{k_2}$ , the reasoning is identical. In this case, if  $C_{k_2}$  is binding, then the optimal solution is the feasible point that satisfies  $C_{k_2}$  with equality and which is the nearest to  $(0, 1/b_{k_2})$ , which equivalently means that from all feasible points that satisfy  $C_{k_2}$  with equality, it has the the minimum value for the first component.  $\square$

From the above theorem, we can actually describe precisely the optimal point. We did that in the proof, but it is worthwhile to highlight this fact in the following corollary given without proof since it is nothing else but an analytical characterization of the optimal solution obtained in the previous theorem.

**Corollary 1.** Consider all hypotheses and notations from Theorem 3. If  $C_{k_1}$  is binding, then let  $[\alpha, \beta]$  be the solution set of variable  $t_1$  obtained after we solve the system of the constraints with the substitution  $t_2 = (1 - a_{k_1}t_1)/b_{k_1}$  (as  $C_{k_1}$  is binding, this solution set is always nonempty). Then an optimal solution of problem (12) is  $(\beta, (1 - a_{k_1}\beta)/b_{k_1})$  and the optimal value is  $(1 - a_{k_1}/b_{k_1})\beta + 1/b_{k_1}$ . Then, if  $C_{k_2}$  is binding, denote again with  $[\alpha, \beta]$  the solution set of variable  $t_1$  using the substitution  $t_2 = (1 - a_{k_2}t_1)/b_{k_2}$ . Then, an optimal solution of problem (12) is  $(\alpha, (1 - a_{k_2}\alpha)/b_{k_2})$ , and the optimal value is  $(1 - a_{k_2}/b_{k_2})\alpha + 1/b_{k_2}$ .

Now, we are in position to present an algorithm that will give us a solution of Problem (12). Obviously, there are two methods to approach the solution. Either we search the binding constraint in the set  $J_1$  or we search in the set  $J_2$ . If we would make simulations on very large numbers of such problems, we believe that we would obtain on average the same running time. In general, we will apply the algorithm using  $J_1$  if its cardinal is less than or equal to the cardinal of  $J_2$ ; otherwise, we will apply the algorithm using  $J_2$ . We can also propose an algorithm that takes into consideration both sets  $J_1$  and  $J_2$ . First, we search the binding constraint on  $J_1$  taking the constraint  $C_{k_1}$  as described in the statement of Theorem 3. If  $C_{k_1}$  is binding, then we find the optimal solution as described in the previous corollary. If  $C_{k_1}$  is redundant, then we check the constraint  $C_{k_2}$  in  $J_2$  selected exactly as in the statement of Theorem 3. If  $C_{k_1}$  is binding, then we find the optimal solution as described in the previous corollary. If not, then we update  $J_1 = J_1 \setminus \{k_1\}$  and search again for the constraint in  $J_1$  according to the construction in Theorem 3, and so on. We omit this second variant because we think it is slower in general.

In Algorithm 1, we search for the binding constraint from beginning to end either in  $J_1$  or in  $J_2$ . The first two steps are essential in our choice. For that, we need to compute  $a_{k^*} = \min\{a_k : k \in J\}$  and  $b_{k^{**}} = \min\{b_k : k \in J\}$ . Note that besides computing  $a_{k^*}$  and  $b_{k^{**}}$ , we will also need to identify the indexes  $a_{k^*}$  and  $b_{k^{**}}$ .

---

**Algorithm 1:** solution

---

**Step 1**

If  $a_{k^*} \geq b_{k^{**}}$ , then  $(1/a_{k^*}, 0)$  is an optimal solution of Problem (12), and  $1/a_{k^*}$  is the optimal value of Problem (12). If  $a_{k^*} < b_{k^{**}}$ , then go to step 2.

**Step 2**

If  $a_{k^{**}} \leq b_{k^{**}}$ , then  $(0, 1/b_{k^{**}})$  is the optimal solution of Problem (12) and  $1/b_{k^{**}}$  is the optimal value of Problem (12). If  $a_{k^{**}} > b_{k^{**}}$  then go to step 3.

**Step 3**

If we reached this step of the algorithm, it means that both  $J_1$  and  $J_2$  are nonempty. What is more, both of them contain at least one index corresponding to a binding constraint. Let us explain for  $J_1$ , since for  $J_2$ , the explanation is identical.

As  $a_{k^{**}} > b_{k^{**}}$ , it follows that  $k^{**}$  is in  $J_1$ . If  $C_k$  would be strongly redundant for any  $k \in J_1$  such that  $a_k < a_{k^{**}}$ , then by Theorem 3, it easily follows that constraint  $C_{k^{**}}$  is binding. Here, we need to decide if we search the binding constraint considering the set  $J_1$  or the set  $J_2$ . We can impose a selection criterion.

For example, we choose to go with  $J_1$  if its cardinal is less than or equal to the cardinal of  $J_2$  and with  $J_2$  otherwise. In what follows, we explain the algorithm when the option is  $J_1$ , and at the end of it, we explain in a remark the very small differences that occur in the case when the option is  $J_2$ .

Take  $a_{l_1} = \min\{a_k : k \in J_1\}$ . We solve the system of the constraints indexed in  $J$  with the substitution  $t_2 = (1 - a_{l_1}t_1)/b_{l_1}$ . If we obtain for variable  $t_1$  the solution  $[\alpha, \beta]$  then an optimal solution of problem (12) is  $(\beta, (1 - a_{l_1}\beta)/b_{l_1})$ , and the optimal value is  $(1 - a_{l_1}/b_{l_1})\beta + 1/b_{l_1}$ . If this system has no solution, then go to step 4.

**Step 4**

We set  $J := J \setminus \{l_1\}$  and  $J_1 := J_1 \setminus \{l_1\}$ , and we repeat all steps 1–3 for the newly obtained  $J$  and  $J_1$ .

---

We observe that step 3 is repeated until we get the first binding constraint, and we know that in the worst case this binding constraint is  $C_{k^{**}}$ . We also have to notice that we have a maximum of  $n - 1$  iterations in terms of repeatedly applying step 3. What is more, if we chose to go with  $J_1$  because its cardinal does not exceed the cardinal of  $J_2$ , then we have at most  $\lceil |J|/2 \rceil + 1$  iterations (here,  $\lceil \cdot \rceil$  stands for the integer part of a real number). The most important utility of Algorithm 1 is that it helps us to indicate the binding constraint or the binding constraints, respectively, in Theorem 2, corresponding to case (iii) or (iv), respectively.

**Remark 2.** *If we go with  $J_2$  instead of  $J_1$ , then we just need to adapt the calculations in step 3. First, we take  $b_{p_1} = \min\{b_k : k \in J_2\}$ . Then, we solve the system of the constraints with the substitution  $t_2 = (1 - a_{p_1}t_1)/b_{p_1}$ . If we obtain for variable  $t_1$  the solution  $[\alpha, \beta]$ , then the optimal solution of problem (12) is  $(\alpha, (1 - a_{p_1}\alpha)/b_{p_1})$  and the optimal value is  $(1 - a_{p_1}/b_{p_1})\alpha + 1/b_{p_1}$ . If this system has no solution, then go to step 4, where this time, we set  $J := J \setminus \{p_1\}$  and  $J_2 := J_2 \setminus \{p_1\}$ , and we repeat step 3 for the newly obtained  $J$  and  $J_2$ .*

**Remark 3.** *Another useful remark is that we can slightly accelerate the algorithm at step 3. Suppose again that we choose to go with  $J_1$ . Instead of solving the system of constraints with the substitution  $t_2 = (1 - a_{p_1}t_1)/b_{p_1}$ , we can use a different approach. For any  $k \in J_2$ , we solve the system  $a_{p_1}t_1 + b_{p_1}t_2 = 1, a_k t_1 + b_k t_2 = 1$ . From all solutions, we select the one for which the first component has the minimum value. Suppose this solution is  $(t_1^*, t_2^*)$ . If  $(t_1^*, t_2^*)$  satisfies all the constraints indexed in  $J_2$ , then  $(t_1^*, t_2^*)$  is the optimal solution of problem (12). Otherwise, if there exists a constraint indexed in  $J_2$  that is not satisfied by  $(t_1^*, t_2^*)$ , then constraint  $C_{p_1}$  is strongly redundant, and we move to step 4. It seems that, in general, this method is slightly faster than the one described in the algorithm in the cases when  $n$  is sufficiently large.*

#### 4. Some Concrete Examples

In what follows, first, we present some examples on which we test Algorithm 1. Finally, we apply the algorithm to solve a concrete constrained OWA aggregation problem.

**Example 1.** *Let us consider the problem*

$$t_1 + t_2 \rightarrow \min,$$

subject to

$$\begin{cases} 2t_1 + 4t_2 \geq 1, \\ 4t_1 + t_2 \geq 1, \\ 3t_1 + 2t_2 \geq 1, \\ 7t_1 + 5t_2 \geq 1, \\ t_1, t_2 \geq 0. \end{cases}$$

We observe that  $a_{k^*} = \min\{a_k : k \in J\} = 2$  and  $b_{k^{**}} = \min\{b_k : k \in J\} = 1$ . Since  $b_{k^*} = 4 > a_{k^*}$  and  $a_{k^{**}} = 4 > b_{k^{**}}$ , it follows that we can move to step three in Algorithm 1. It is easily seen that  $J_1 = \{2, 3, 4\}$  and  $J_2 = \{1\}$ . Clearly, the simplest way to reach the solution is to go with  $J_2$  since it has just one element. Practically, this means that constraint  $C_1$  is binding. According to the first algorithm, we solve the system of the constraints in the special case when  $t_2 = \frac{1-2t_1}{4}$ . Thus, we obtain the system of inequalities

$$\begin{cases} 4t_1 + \frac{1-2t_1}{4} \geq 1, \\ 3t_1 + 2 \cdot \frac{1-2t_1}{4} \geq 1, \\ 7t_1 + 5 \cdot \frac{1-2t_1}{4} \geq 1, \\ 1 - 2t_1 \geq 0, \\ t_1 \geq 0. \end{cases}$$

By simple calculations, the above system has the solution  $t_1 \in [\frac{1}{4}, \frac{1}{2}]$ . By using the first algorithm we get that  $t_1^* = \frac{1}{4}$  and by the substitutions used here, we get that  $t_2^* = \frac{1-2t_1^*}{4} = \frac{1}{8}$ . Therefore,  $(\frac{1}{4}, \frac{1}{8})$  is the optimal solution of our problem, and  $t_1^* + t_2^* = \frac{3}{8}$  is the optimal value. Even if  $J_1$  has three elements, we do not need more iterations. Indeed,  $a_{J_1} = \min\{a_k : k \in J_1\} = 3$ . Thus, we need to solve the system of the constraints under the substitution  $t_2 = \frac{1-3t_1}{2}$ . By simple calculations, we obtain  $t_1 \in [\frac{1}{5}, \frac{1}{4}]$ ; hence, we obtain the optimal solution at the first attempt. Note that with the second algorithm, no additional iterations are needed.

The following example shows that sometimes we may need step 4 of the first algorithm.

**Example 2.** Let us consider the problem

$$t_1 + t_2 \rightarrow \min,$$

subject to

$$\begin{cases} \frac{1}{2}t_1 + \frac{1}{6}t_2 \geq 1, \\ \frac{1}{3}t_1 + \frac{1}{12}t_2 \geq 1, \\ \frac{1}{4}t_1 + \frac{1}{4}t_2 \geq 1, \\ \frac{1}{5}t_1 + \frac{1}{3}t_2 \geq 1, \\ \frac{1}{6}t_1 + \frac{1}{2}t_2 \geq 1, \\ t_1, t_2 \geq 0. \end{cases}$$

Again, we cannot obtain the solution in the first two steps of Algorithm 1. Therefore, we move again to the third step. We have  $J_1 = \{1, 2, 3\}$  and  $J_2 = \{3, 4, 5\}$ . First, let us apply the algorithm for  $J_1$ . We observe that  $a_{k^*} = \min\{a_k : k \in J_1\} = a_3 = 4$ . Therefore, we have to solve the system of the constraints in the special case when  $t_2 = 4 - t_1$ . After simple calculations, we get that this system has no solution. This means that constraint  $C_3$  is redundant. This means that we have to go to step 4. We set  $J_1 := J_1 \setminus \{3\}$ . Now,  $\min\{a_k : k \in J_1\} = a_2 = 1/3$ . We have to solve the system of constraints in the special case when  $t_2 = 12 - 4t_1$ . We obtain the system

$$\begin{cases} \frac{1}{2}t_1 + \frac{1}{6} \cdot (12 - 4t_1) \geq 1, \\ \frac{1}{4}t_1 + \frac{1}{4} \cdot (12 - 4t_1) \geq 1, \\ \frac{1}{5}t_1 + \frac{1}{3} \cdot (12 - 4t_1) \geq 1, \\ \frac{1}{6}t_1 + \frac{1}{2} \cdot (12 - 4t_1) \geq 1, \\ 12 - 4t_1 \geq 0, \\ t_1 \geq 0. \end{cases}$$

**Example 3.** Let us now find the optimal solution of a constrained OWA aggregation problem with two comonotone constraints. We noticed that most often, we obtain a lot of redundant constraints in the auxiliary Problem (12). However, sometimes, we can obtain situations where Step 3 is needed in Algorithm 1. In our example, we consider the case  $n = 3$ , which we believe suffices to explain how the algorithm work. Since Algorithm 1 can easily be implemented on computer, one may consider higher values for  $n$ .

Consider the problem

$$\begin{cases} \max\left(\frac{1}{26}y_1 + \frac{5}{26}y_2 + \frac{10}{13}y_3\right), \\ \frac{1}{6}x_1 + \frac{17}{6}x_2 + \frac{7}{7}x_3 \leq 1, \\ \frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{11}{2}x_3 \leq 1, \\ x_1, x_2, x_3 \geq 0. \end{cases} \tag{14}$$

Clearly, we have two comonotone constraints here; therefore, we need the auxiliary problem (12), and by simple calculations this problem is

$$\begin{cases} \min(t_1 + t_2), \\ \frac{13}{3}t_1 + 13t_2 \geq 1, \\ 13t_1 + \frac{13}{3}t_2 \geq 1, \\ \frac{13}{2}t_1 + \frac{13}{2}t_2 \geq 1, \\ t_1, t_2 \geq 0. \end{cases} \tag{15}$$

We have

$$a_{k^*} = \min\{a_k : k \in \{1, 2, 3\}\} = a_1 = \frac{13}{3} < b_1$$

and

$$b_{k^{**}} = \min\{b_k : k \in \{1, 2, 3\}\} = b_2 = \frac{13}{3} < a_2.$$

This means that we need Step 3 in Algorithm 1. We have  $J_1 = \{2, 3\}$  and  $\min\{a_k : k \in J_1\} = a_3 = \frac{13}{2}$ . Therefore, we solve the system of constraints in (15) in the special case when

$$t_2 = \frac{1 - \frac{13}{2}t_1}{\frac{13}{2}} = \frac{2 - 13t_1}{13},$$

that is, when constraint 3 is satisfied with an equality. After simple calculations, we obtain the solution of this system as the interval  $[\frac{1}{26}, \frac{3}{26}]$ . This means that case (iii) in Theorem 2 is applicable, and by applying the formula for this case, we obtain that an optimal solution for problem (14) is

$$x_1 = x_2 = x_3 = \frac{1}{\alpha_1 + \alpha_2 + \alpha_3} = \frac{2}{13}$$

and the optimal value of this problem is  $\frac{2}{13}$ .

### 5. Conclusions

In this paper, we extended the solving of the constrained OWA aggregation problem with two comonotone constraints to the case when the OWA weights are arbitrary non-negative numbers. Moreover, we proposed an iterative algorithm to approach the optimal solution. This algorithm indicates a constraint that is either bidding or strictly redundant. We hope this is a first step towards the solving of constrained OWA aggregation problems with an arbitrary number of constraints.

**Author Contributions:** Conceptualization, formal analysis, writing—review and editing L.C.; Conceptualization, formal analysis, writing—review and editing R.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** Robert Fullér was partially supported by the ELKH-SZE Research Group for Cognitive Mapping of Decision Support Systems. Lucian Coroianu was supported by a grant awarded by the University of Oradea and titled “Approximation and optimization methods with applications”.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The study did not report any data.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

OWA	Ordered Weighted Averaging
MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	Linear dichroism

## References

1. Yager, R.R. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Syst. Cybern.* **1988**, *18*, 183–190. [[CrossRef](#)]
2. Coroianu, L.; Fullér, R.; Harmati, I.A. Best approximation of OWA Olympic weights under predefined level of orness. *Fuzzy Sets Syst.* **2022**, *448*, 127–144. [[CrossRef](#)]
3. Coroianu, L.; Fullér, R.; Gagolewski, M.; James, S. Constrained ordered weighted averaging aggregation with multiple comonotone constraints. *Fuzzy Sets Syst.* **2020**, *395*, 21–39. [[CrossRef](#)]
4. Emrouznejad, A.; Marra, M. Ordered weighted averaging operators 1988–2014: A citation-based literature survey. *Int. Intell. Syst.* **2014**, *29*, 994–1014. [[CrossRef](#)]
5. Nguyen, T.H. Simplifying the minimax disparity model for determining OWA weights in large-scale problems. In *New Trends in Emerging Complex Real Life Problems*; Daniele, P., Scrimali, L., Eds.; AIRO Springer Series; Springer: Cham, Switzerland, 2018; Volume 1. [[CrossRef](#)]
6. Yager, R.R. Constrained OWA aggregation. *Fuzzy Sets Syst.* **1996**, *81*, 89–101. [[CrossRef](#)]
7. Carlsson, C.; Fullér, R.; Majlender, P. A note on constrained OWA aggregation. *Fuzzy Sets Syst.* **2003**, *139*, 543–546. [[CrossRef](#)]
8. Coroianu, L.; Fullér, R. On the constrained OWA aggregation problem with single constraint. *Fuzzy Sets Syst.* **2018**, *332*, 37–43. [[CrossRef](#)]
9. Kim, E.Y.; Ahn, B.S. An Efficient Approach to Solve the Constrained OWA Aggregation Problem. *Symmetry* **2022**, *14*, 724. [[CrossRef](#)]
10. Coroianu, L.; Fullér, R. Minimum of constrained OWA aggregation problem with a single constraint. In *Fuzzy Logic and Applications. WILF 2018. Lecture Notes in Computer Science*; Fullér, R., Giove, S., Masulli, F., Eds.; Springer: Cham, Switzerland, 2019; Volume 11291, pp. 183–192.
11. Ogryczak, W.; Śliwiński, T. On efficient WOWA optimization for decision support under risk. *Int. J. Approx. Reason.* **2009**, *50*, 915–928. [[CrossRef](#)]
12. Ogryczak, W.; Śliwiński, T. On solving linear programs with the ordered weighted averaging objective. *Eur. J. Oper. Res.* **2003**, *148*, 80–91. [[CrossRef](#)]
13. Fullér, R.; Majlender, P. An analytic approach for obtaining maximal entropy OWA operator weights. *Fuzzy Sets Syst.* **2001**, *124*, 53–57. [[CrossRef](#)]
14. Sang, X.; Liu, X. An analytic approach to obtain the least squares deviation OWA operator weights. *Fuzzy Sets Syst.* **2014**, *240*, 103–116. [[CrossRef](#)]
15. Beliakov, G.; James, S. Choquet integral optimisation with constraints and the buoyancy property for fuzzy measures. *Inf. Sci.* **2021**, *578*, 22–36. [[CrossRef](#)]
16. Kang, B.; Deng, Y.; Hewage, K.; Sadiq, R. Generating Z-number based on OWA weights using maximum entropy. *Int. J. Intell. Syst.* **2018**, *38*, 1745–1755. [[CrossRef](#)]
17. Ochoa, G.; Lizașoian, I.; Paternain, D.; Bustince, H.; Pal, N.R. From quantitative to qualitative orness for lattice OWA operators. *Int. J. Gen. Syst.* **2017**, *46*, 640–669. [[CrossRef](#)]
18. Paternain, D.; Ochoa, G.; Lizașoian, I.; Bustince, H.; Mesiar, R. Quantitative orness for lattice OWA operators. *Inf. Fusion* **2016**, *30*, 27–35. [[CrossRef](#)]
19. Torra, V. Andness directedness for operators of the OWA and WOWA families. *Fuzzy Sets Syst.* **2021**, *414*, 28–37. [[CrossRef](#)]