Transactions on Fuzzy Sets and Systems (TFSS)
URL: https://tfss.journals.iau.ir/
Online ISSN: 2821-0131
Vol.1, No.2, (2022), 99-128
DOI: http://doi.org/10.30495/tfss.2022.1962834.1041

Article Type: Original Research Article (Invited Paper)

On Generalized Mixture Functions

Antonio Diego S. Farias, **Valdigleis S. Costa**, **Luiz Ranyer A. Lopes**, **Regivan H. N. Santiago**, **Benjamín Bedregal***

Abstract. In the literature it is very common to see problems in which it is necessary to aggregate a set of data into a single one. An important tool able to deal with these issues is the aggregation functions, which we can highlight as the OWA functions. However, there are other functions that are also capable of performing these tasks, such as the preaggregation function and mixture functions. In this paper we investigate two special types of functions, the Generalized Mixture functions and Bounded Generalized Mixture functions, which generalize both OWA and Mixture functions. We also prove some properties, constructions and examples of these functions. Both the Generalized and Bounded Generalized Mixture functions are developed in such a way that the weight vectors are variables that depend on the input vector, which generalizes the aggregation functions: *Minimum*, *Maximum*, *Arithmetic Mean* and *Median*, and are extensively used in image processing. Finally, we propose a Generalized Mixture function, denoted by **H**, and we show that **H** satisfies a series of properties in order to apply this function in an illustrative example of application: The image reduction process.

AMS Subject Classification 2020: 03E72; 47S40; 94A08
Keywords and Phrases: Aggregation functions, Preaggregation functions, OWA functions, Generalized Mixture functions, Image reduction.

1 Introduction

Some functions are able to transform a set of data into a single one, for example, aggregations functions [3, 6, 22] and mixture functions [6]. This type of function has applications in several areas; for example, we can cite [8, 17, 19, 43, 44]. Image processing used in medicine; for example, you can apply it to: detect tumors [26, 36, 40, 58]; support techniques in advancing dental treatments [14, 25, 52, 54], etc. Such images are not always obtained with suitable quality, and to detect the desired information, various methods have been developed in order to eliminate most of the noise contained in these images [29, 42, 50]. These functions can also be used to reduce the size of images (this process is called image reduction).

The methods of image reduction are used in order to decrease your resolutions, usually aiming the reduction of memory consumption required for its storage [23]. There are several techniques for image reduction to achieve this goal in the literature, among these techniques, we can cite Paternain *et al.* [45], that built a method of reduction using *weighted averaging aggregation functions*. The method proposed by Paternain *et al.* consists of: (1) Reducing a given image by using a reduction operator (based on weighted averaging

***Corresponding author:** Benjamín Bedregal, Email: bedregal@dimap.ufrn.br, ORCID: 0000-0002-6757-7934
Received: 8 July 2022; Revised: 22 July 2022; Accepted: 22 July 2022; Published Online: 7 November 2022.

How to cite: A. D. S. Farias, V. S. Costa, L. R. A. Lopes, R. H. N. Santiago and B. Bedregal, On Generalized Mixture Functions, *Trans. Fuzzy Sets Syst.*, 1(2) (2022), 99-128.

aggregation functions); (2) Building a new image from the reduced one, and (3) Analyzing the quality of the last image be using the measures *PSNR* and *MSSIM* defined in [23].

Because of its broad capacity of applications, many researchers have invested in aggregate functions and its extensions [34, 39, 46, 48, 61, 64]. In this sense, thinking about the problem of decision-making, Yager [60] introduced a special class of aggregate functions, called Ordered Weighted Averaging - OWA, and ever since several authors have proposed generalizations for these functions [12, 33, 37, 53, 61]. Mixture functions, presented in [6], and variants of Choquet integrals in [2, 10, 15, 35] are other important examples of generalization of the OWA. These functions are not aggregate functions, but also are efficient in converting various information into a single one.

In this paper we studied a class of functions introduced in [46] and called Generalized Mixture - GM. Since then many other papers on this class of functions have been found, for example [13, 20, 21, 47, 49]. GM also generalizes the notion of OWA and consequently, also encompass functions as: *Arithmetic Mean*, *Median*, *Maximum* and *Minimum*. Besides that, it is a generalized form of another important class of functions: The *Mixture functions* - MO, which as well as OWA functions, are determined from weights $w_1, w_2, \cdots, w_n \in [0,1]$, which generally satisfy the condition $\sum_{i=1}^{n} w_i = 1$. The GM functions,as well as the MO functions, are weighted averaging means with dynamic weights, i.e., the weights of these functions depend on the input variables. This characteristic of more flexible weights of OWA′ allows us to define functions whose weights are suited for each input, which does not occur in OWA's. However, we ended up losing the property of monotonicity, which can be replaced by directional monotonicity [9] in order to obtain preaggregation functions.

Later, in this work, we weaken the condition of the vector of weights $\left(\sum_{i=1}^{n} w_i = 1 \text{ to } \sum_{i=1}^{n} w_i \leq 1\right)$, thereby obtaining in another generalization of OWA, called the *Bounded Generalized Mixture* - BGM function, we propose a special GM function (denoted by $\mathbf{H}$). This way, we provide a wide range of their properties such as: idempotence, symmetry, homogeneity and directional monotonicity. To finalize this work, we apply $\mathbf{H}$ in a method of image reduction [4, 7, 44, 51, 56, 59] and we compare this function with *Minimum*, *Maximum*, *Arithmetic Mean*, *Median* and cOWA. The method adopted was the same as Paternain *et al.* [45].

This work is structured in the following way: The next section provides the basic concepts of Aggregation functions theory; In Section 3, we introduce the concepts of Generalized Mixture - GM and Bounded Generalized Mixture - BGM operators, we show properties, constructions, examples and propose a particular GM function (called $\mathbf{H}$). Also in Section 3, we show that $\mathbf{H}$ is idempotent, homogeneous, shift-invariant, symmetric, self dual and directionally monotonic, which is important to the image reduction field [45]. In Section 4, we provide an illustrative application for GM's. in image reduction and finally in Section 5 we close this paper with some final remarks.

2 Aggregation Functions

Aggregation functions are important mathematical tools for applications in various fields, such as: Information fuzzy [17, 19, 24, 32]; Decision making [8, 11, 41, 44, 64]; Image processing [4, 26, 45] and Engineering [31, 43]. In this section we introduce them together with examples and properties. We also present a special family of aggregation functions called *Ordered Weighted Averaging* (OWA), showing some of its features and the notion of *Mixture Operator* (MO), a generalized form of OWA.

2.1 Definition and Examples

Aggregation functions are n-ary operations on the unit interval $[0,1]$ which are able to summarize an n-dimensional information $\mathbf{x} = (x_1, \ldots, x_n) \in [0,1]^n$ into a unique data $\mathbf{x} \in [0,1]$. Formally, they are the following functions:

Definition 2.1. *An n-ary aggregation function is a mapping $A : [0,1]^n \to [0,1]$, which associates each n-dimensional vector $\mathbf{x} = (x_1, \ldots, x_n)$ to a single value $A(\mathbf{x})$ in the interval $[0,1]$ which satisfies the mononicity condition*[2] *and also the boundary condition*[3]*:*

Example 2.2. *Given $\mathbf{x} = (x_1, \ldots, x_n)$,*

(a) *Arithmetic Mean:* $Arith(\mathbf{x}) = \dfrac{1}{n}(x_1 + x_2 ... + x_n)$

(b) *Minimum:* $Min(\mathbf{x}) = min\{x_1, x_2, ..., x_n\}$;

(c) *Maximum:* $Max(\mathbf{x}) = max\{x_1, x_2, ..., x_n\}$;

(d) *Product:* $Prod(\mathbf{x}) = \prod_{i=1}^{n} x_i$;

(e) *Weighted Average: For* $(w_1, \cdots, w_n) \in [0,1]^n$, *with* $\sum_{i=1}^{n} w_i = 1, WAvg(\mathbf{x}) = \sum_{i=1}^{n} w_i \cdot x_i$.

Remark 2.3. *From now on we will use the short term "aggregation" instead of "n-ary aggregation function".*

Aggregations can be divided into four distinct classes: *Averaging, Conjunctive, Disjunctive* and *Mixed*. Since this paper focus on averaging aggregations, we will define only this class.

Definition 2.4. *A function $f : [0,1]^n \longrightarrow [0,1]$ satisfies the* **averaging property**, *if for all $\mathbf{x} \in [0,1]^n$ we have:*

$$Min(\mathbf{x}) \leq f(\mathbf{x}) \leq Max(\mathbf{x}).$$

When an aggregation f satisfies the averaging property we say that f is a **averaging function**. *Futhermore, if a aggregation that satisfies the averaging property is called of* **averaging aggregation function**. *As in this paper we are dedicated to studying only functions that satisfy the averaging property, we will not detail the Conjunctive, Disjuntive and Mixed functions. A wider approach in aggregation can be found in [1, 3, 6, 16, 22].*

Example 2.5. *The functons Min, Max, $Arith$ and $WAvg$ are averaging aggregations.*

In the definition below we describe a series of properties that the aggregations functions (like any other function) can satisfy.

Definition 2.6. *Let $f : [0,1]^n \to [0,1]$ be a function. We say that f*

(1) *is* **Idempotent** *if, and only if, $f(x, ..., x) = x$ for all $x \in [0,1]$.*

(2) *is* **Homogeneous** *of order k if, and only if, for all $\lambda \in [0,1]$ and $\mathbf{x} \in [0,1]^n$, $f(\lambda x_1, \lambda x_2, ..., \lambda x_n) = \lambda^k f(x_1,$
$x_2, ..., x_n)$. When f is homogeneous of order 1 we simply say that f is homogeneous.*

(3) *is* **Shift-invariant** *if, and only if, $f(x_1 + r, x_2 + r, .., x_n + r) = f(x_1, x_2, .., x_n) + r$, for all $r \in [-1,1]$, $\mathbf{x} \in [0,1]^n$, $(x_1 + r, x_2 + r, ..., x_n + r) \in [0,1]^n$ and $f(x_1, x_2, ..., x_n) + r \in [0,1]$.*

(4) *is* **Monotonic** *if, and only if, $f(\mathbf{x}) \leq f(\mathbf{y})$ whenever $x_i \leq y_i$, for all $i \in \{1, \cdots, n\}$.*

(5) *is* **Strictly Monotone** *if, and only if, $f(\mathbf{x}) < f(\mathbf{y})$ whenever $\mathbf{x} < \mathbf{y}$, i.e., $\mathbf{x} \leq \mathbf{y}$ and $\mathbf{x} \neq \mathbf{y}$.*

[2] If $\mathbf{x} \leq \mathbf{y}$, i.e., $x_i \leq y_i$, for all $i = 1, 2, ..., n$, then $A(\mathbf{x}) \leq A(\mathbf{y})$.

[3] $A(0, ..., 0) = 0$ and $A(1, ..., 1) = 1$.

(6) *has a* **Neutral Element** $e \in [0,1]$, *if for all* $t \in [0,1]$ *it has to be:*

$$f(e, ..., e, t, e, ..., e) = t.$$

(7) *is* **Symmetric** *if, and only if, its value is not changed under the permutations of coordinate for any input vector, i.e.:*

$$f(x_1, x_2, ..., x_n) = f(x_{p_{(1)}}, x_{p_{(2)}}, \cdots, x_{p_{(n)}})$$

for all vector $\mathbf{x} = (x_1, x_2, ..., x_n)$ *and any permutation* $p : \{1, 2, ..., n\} \to \{1, 2, ..., n\}$.

(8) *has an* **Absorbing Element** ***(Annihilator)*** $a \in [0,1]$, *if:*

$$f(x_1, x_2, ..., x_{i-1}, a, x_{i+1}, ..., x_n) = a.$$

(9) *has a* **Zero Divisor** $a \in \,]0,1[$, *if for all* $i \in \{1, 2, \cdots, n\}$ *there is some vector* $\mathbf{x} \in]0,1]^n$, *of the form* $(x_1, ..., x_{i-1},$
$a, x_{i+1}, ..., x_n)$, *such that* $f(\mathbf{x}) = 0$.

(10) *has a* **One Divisor** $a \in]0,1[$, *if for any* $i \in \{1, 2, \cdots, n\}$ *there is some vector* $\mathbf{x} \in [0,1[^n$, *of the form* $(x_1, ..., x_{i-1},$
$a, x_{i+1}, ..., x_n)$, *such that* $f(\mathbf{x}) = 1$.

Example 2.7.

(i) *The functions: $Arith, Min$ and Max are examples of idempotent, homogeneous, shift-invariant and symmetric aggregations.*

(ii) *Min and Max have the elements* 0 *and* 1 *as its respective annihilators, but Arith does not have annihiladors.*

(iii) *Min, Max and Arith does not have zero divisors and one divisors.*

2.2 Ordered Weighted Averaging - OWA Functions

In the field of aggregations there is a very important kind of function in which the aggregation behavior is provided parametrically; they are called: ***Ordered Weighted Averaging*** or simply OWA [60]. More precisely, they are average aggregation whose behavior is in function of a vector of weights. Observe the following definition.

Definition 2.8. *Let be an input vector* $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in [0,1]^n$ *and a vector of weights* $\mathbf{w} = (w_1, \ldots, w_n) \in [0,1]^n$, *such that* $\sum_{i=1}^{n} w_i = 1$. *Assuming the permutation of* $\mathbf{x}$*:*

$$Sort(\mathbf{x}) = (x_{(1)}, x_{(2)}, \ldots, x_{(n)})$$

such that $x_{(i)} \geq x_{(i+1)}$, *i.e.,* $x_{(1)} \geq x_{(2)} \geq \cdots \geq x_{(n)}$. *The Ordered Weighted Averaging (OWA) function with respect to* $\mathbf{w}$, *is the function* $OWA_{\mathbf{w}} : [0,1]^n \to [0,1]$ *such that:*

$$OWA_{\mathbf{w}}(\mathbf{x}) = \sum_{i=1}^{n} w_i \cdot x_{(i)}$$

Remark 2.9. *In what follows we remove* $\mathbf{w}$ *from* $OWA_{\mathbf{w}}(\mathbf{x})$ *and write only OWA.*

The main properties of OWA functions are:

(a) For any vector of weights $\mathbf{w}$, the function $\mathsf{OWA}_{\mathbf{w}}(\mathbf{x})$ is an idempotent aggregation function. Moreover, OWA's are strictly increasing if all weights $\mathbf{w}$ are positive;

(b) The dual of a $\mathsf{OWA}_{\mathbf{w}}$, denoted by $(\mathsf{OWA})^d$, is an OWA with the vector of weights dually ordered, i.e. $(\mathsf{OWA}_{\mathbf{w}})^d = \mathsf{OWA}_{\mathbf{w}^d}$, where $\mathbf{w}^d = (w_{p(n)}, w_{p(n-1)}, ..., w_{p(1)})$.

(c) OWA are continuous, symmetric and shift-invariant;

(d) They do not have neutral or absorption elements, on exception for the second and third case below.

Following is a series of examples of OWA functions

Example 2.10.

(1) If $\mathbf{w} = (0, 0, 0, ..., 1)$, then $\mathsf{OWA}(\mathbf{x}) = Min(\mathbf{x})$;

(2) If $\mathbf{w} = (1, 0, 0, ..., 0)$, then $\mathsf{OWA}(\mathbf{x}) = Max(\mathbf{x})$;

(3) If all weight vector components are equal to $\frac{1}{n}$, then $\mathsf{OWA}(\mathbf{x}) = Arith(\mathbf{x})$;

(4) if $w_i = 0$, for all i, with the exception of a k-th member, i.e, $w_k = 1$, then this OWA is called **static** *and $\mathsf{OWA}_{\mathbf{w}}(x) = x_{(k)}$;*

(5) Given a vector $\mathbf{x}$ and its ordered permutation $Sort(\mathbf{x}) = (x_{(1)}, \dots, x_{(n)})$, the Median function

$$Med(\mathbf{x}) = \begin{cases} \frac{1}{2}(x_{(k)} + x_{(k+1)}), & \text{if } n = 2k \\ x_{(k+1)}, & \text{if } n = 2k+1 \end{cases}$$

is an OWA function in which the vector of weights is defined by:

- *If n is odd, then $w_i = 0$ for all $i \neq \lceil \frac{n}{2} \rceil$ and $w_{\lceil n/2 \rceil} = 1$.*
- *If n is even, then $w_i = 0$ for all $i \neq \lfloor \frac{n+1}{2} \rfloor$ and $i \neq \lceil \frac{n+1}{2} \rceil$, and $w_{\lceil (n+1)/2 \rceil} = w_{\lfloor (n+1)/2 \rfloor} = \frac{1}{2}$.*

In addition to the above functions, another important example of OWA, which we will use later in this work, is the **centered OWA** or cOWA[61].

Example 2.11. *The n-dimensional cOWA function is the OWA operator, with weighted vector defined by:*

- *If n is even, then $w_j = \frac{2(2j-1)}{n^2}$, for $1 \leq j \leq \frac{n}{2}$, and $w_{n/2+i} = w_{n/2-i+1}$, for $1 \leq i \leq \frac{n}{2}$.*
- *If n is odd, then $w_j = \frac{2(2j-1)}{n^2}$, for $1 \leq j \leq \frac{n-1}{2}$, $w_{n/2+i} = w_{n/2-i+1}$, for $1 \leq i \leq \frac{n}{2}$, and $w_{(n+1)/2} = 1 - 2 \sum_{j=1}^{(n-1)/2} w_i$.*

The OWA functions are defined in terms of a predetermined vector of weights; namely this vector of wights is fixed previously by the user. In the next section present a generalized form of OWA in order to relax this situation. The vector of weights will be in function of the vector of inputs $\mathbf{x} = (x_1, \dots x_n)$. To achieve that we replace, in the OWA expression, the vector of weights by a family of functions, called **Weighted functions**.

3 Weighted functions

As mentioned, the OWA functions are means with previously fixed weights. In the literature we can find some kind of functions that overcome this situation, by providing variable weights. These functions are called here *weighted functions*. An important example of that is the Mean of Bajraktarevic, presented in [6].

Definition 3.1 (Mean of Bajraktarevic)**.** *Let* $\mathbf{w}(t) = (w_1(t), \cdots, w_n(t))$ *be a vector of weights functions* $w_i : [0,1] \to [0,+\infty)$*, and let* $g : [0,1] \to (-\infty,+\infty)$ *be a strictly monotone function. The* ***mean of Bajraktarevic*** *is the function:*

$$f(\mathbf{x}) = g^{-1}\left(\frac{\sum_{i=1}^{n} w_i(x_i)g(x_i)}{\sum_{i=1}^{n} w_i(x_i)}\right)$$

In the case of $g(t) = t$, the mean of Bajraktarevic is also called **Mixture function**, in other words, the mixture functions have the form:

$$M(\mathbf{x}) = \frac{\sum_{i=1}^{n} w_i(x_i) \cdot x_i}{\sum_{i=1}^{n} w_i(x_i)} \tag{1}$$

Generally, the mixture functions are not aggregation functions in general, since they do not always satisfy monotonicity, however [38, 39, 48] provides sufficient conditions to overcome this situation.

Remark 3.2. *Note in Equation (1) that each weight* $w_i(x_i)$ *is the value of a single variable function; namely the weight is the value of a function* w_i *applied to the* i*-th position of the input vector* $\mathbf{x} = (x_1, \ldots, x_n)$*. However, this restriction can be relaxed in order to obtain a weight* $w_i(\mathbf{x})$*, i.e. weight which is in function of the whole input. This generalization of mixture operators were done by Pereira [46, 47] and the resulting functions were called of* ***Generalized Mixture Functions (GMF)****.*

Although Pereira has introduced GMFs he did not provide a deep investigation about them. In what follows we provide some results about such functions; their relation with OWA's, Mixture Functions and Preaggregations. We finally generalize GMF's to the notion of ***Bounded Generalized Mixture Functions (BGMF)*** *and provide some relations of them with the notions of monotonicity, directional monotonicity, Weak-dual and Weak-conjugate functions.*

3.1 Weighted Averaging Functions

Before defining the notion of Weighted Averaging functions, we need to establish the notion of **weight-function**.

Definition 3.3. *A finite family of functions* $\Gamma = \{f_i : [0,1]^n \to [0,1] \mid 1 \leq i \leq n\}$ *such that* $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$ *is called family of* **weight-functions (FWF)**.

The **Generalized Mixture Function***, or simply GM, associated to a FWF* Γ *is the function* $GM_\Gamma : [0,1]^n \to [0,1]$ *given by:*

$$GM_\Gamma(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}) \cdot x_i$$

In the Examples 3.4-3.10 we present GM functions.

Example 3.4. *Let be* $\Gamma = \left\{ f_i(\mathbf{x}) = \frac{1}{n} \mid 1 \le i \le n \right\}$*. The* GM *operator associated to* Γ*,* $\mathsf{GM}_\Gamma(\mathbf{x})$*, is* $Arith(\mathbf{x})$*.*

Example 3.5. *The function Minimum can be obtained from* $\Gamma = \{f_i \mid 1 \le i \le n\}$*, where for all* $\mathbf{x} \in [0,1]^n$*,* $f_{(n)}(\mathbf{x}) = 1$ *and* $f_i(\mathbf{x}) = 0$*, if* $i \ne (n)$ *.*

Example 3.6. *Similarly, the function Maximum is also of type* GM *with* Γ *dually defined.*

Example 3.7. *For any vector of weights* $\mathbf{w} = (w_1, w_2, ..., w_n)$*, A function* $\mathsf{OWA}_{\mathbf{w}}(\mathbf{x})$ *is a* GM *in which the weight-function are given by:* $f_i(\mathbf{x}) = w_{p(i)}$*, where* $p : \{1,2,\cdots,n\} \longrightarrow \{1,2,\cdots,n\}$ *is the permutation, such that* $p(i) = j$ *with* $x_i = x_{(j)}$*. For example: If* $\mathbf{w} = (0.3, 0.4, 0.3)$*, then for* $\mathbf{x} = (0.1, 1.0, 0.9)$ *we have* $x_1 = x_{(3)}$*,* $x_2 = x_{(1)}$ *and* $x_3 = x_{(2)}$*. Thus,* $f_1(\mathbf{x}) = 0.3$*,* $f_2(\mathbf{x}) = 0.3$*,* $f_3(\mathbf{x}) = 0.4$*, and* $\mathsf{GM}(\mathbf{x}) = 0.3 \cdot 0.1 + 0.3 \cdot 1.0 + 0.4 \cdot 0.9 = 0.69$

Remark 3.8. *Example 3.7 shows that any* OWA *function is* GM*. However, there are* GM *functions which are not* OWA*:*

Example 3.9. *Let* $\Gamma = \{\sin(x) \cdot y, 1 - \sin(x) \cdot y\}$*. The respective* GM *function is*

$$\mathsf{GM}(x,y) = (\sin(x) \cdot y) \cdot x + (1 - \sin(x) \cdot y) \cdot y,$$

which is not an OWA *function.*

The following example shows that the *mixture functions* are also special types of GM function.

Example 3.10. *If* $\mathbf{w}(t) = (w_1(t), \cdots, w_n(t))$ *is a vector of weight functions* $w_i : [0,1] \to [0,+\infty)$*, and the mixture operator is* $M(\mathbf{x}) = \frac{\sum_{i=1}^{n} w_i(x_i) \cdot x_i}{\sum_{i=1}^{n} w_i(x_i)}$*, then* M *is also a* GM *function, with weight-functions given by* $f_i(\mathbf{x}) = \frac{w_i(x_i)}{\sum_{i=1}^{n} w_i(x_i)}$*.*

Remark 3.11. *Observe that the* GM *function at Example 3.9 can not be characterized as a mixture function, since* w_1 *is not a function that depends only of variable* x *and* w_2 *is not a function that depends only of variable* y*.*

At this point of paper, we relax the condition $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$ to $\sum_{i=1}^{n} f_i(\mathbf{x}) \le 1$, thus obtaining a new family of generalized mixture functions.

Definition 3.12. *Let* $\Gamma = \{f_i : [0,1]^n \to [0,1] \mid 1 \le i \le n\}$ *such that:*

(I) $\sum_{i=1}^{n} f_i(\mathbf{x}) \le 1$*, and*

(II) $\sum_{i=1}^{n} f_i(1,\cdots,1) = 1$*, for all* $i \in \{1,2,\cdots,n\}$*.*

A **Bounded Ganeralized Mixture** *(*BGM*) operator associated to a* Γ *is a function* $\mathsf{BGM}_\Gamma : [0,1]^n \to [0,1]$ *given by:*

$$\mathsf{BGM}_\Gamma(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}) \cdot x_i$$

Remark 3.13.

1. *Note that* GM *functions are* BGM *operators subject to the condition:*

   (III) $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$, *for any* $\mathbf{x} \in [0,1]^n$,

2. *Let* $\Gamma = \{f_i(x,y) = \frac{x}{n} : 1 \leq i \leq n\}$. *Then,* $BGM_\Gamma = \sum_{i=1}^{n} \frac{x_i^2}{n}$ *is not a* GM *operator, because, for example,* $\sum_{i=1}^{n} f_i(0,\cdots,0) = 0$.

3. *As* BGM *is a generalized form of* GM*, it follows that the functions defined in the Examples 3.4-3.10 are also* BGM *function. In this sense, is worth emphasizing that* BGM *generalize both:* OWA *and* GM *operators.*

Now, we establish several properties of GM and BGM functions.

3.2 Properties of GM and BGM Functions

As we have said previously, GM and BGM are generalized forms of OWA, which in turn belongs to the class of avegaring functions. However, we can not always guarantee that a BGM is an averaging function, while then GM functions are averaging function. The next proposition gives us a sufficient condition to achieve that.

Proposition 3.14. *If* Γ *is a FWF with* $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$, *then* GM_Γ *is an averaging function, i.e.:*

$$Min(\mathbf{x}) \leq GM_\Gamma(\mathbf{x}) \leq Max(\mathbf{x})$$

Proof. For all $\mathbf{x} = (x_1, ..., x_n)$,

$$Min(\mathbf{x}) \leq x_i \leq Max(\mathbf{x}), \ \forall i = 1, 2, ..., n.$$

So,

$$\sum_{i=1}^{n} f_i(\mathbf{x}) \cdot Min(\mathbf{x}) \leq \sum_{i=1}^{n} f_i(\mathbf{x}) \cdot x_i \leq \sum_{i=1}^{n} f_i(\mathbf{x}) \cdot Max(\mathbf{x}),$$

but as $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$, it follows that

$$Min(\mathbf{x}) \leq \sum_{i=1}^{n} f_i(\mathbf{x}) \cdot x_i \leq Max(\mathbf{x}).$$

□

Remark 3.15. *Observe that the restriction of condition* $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$ *can not be removed, i.e.,* BGM *not always are averaging functions, since for* $f_1(x,y) = \frac{x}{2}$ *and* $f_2(x,y) = \frac{y}{2}$, *we have* $BGM(0.5, 0.5) = 0.25 < Min(0.5, 0.5)$.

Proposition 3.16. *Let* Γ *be a FWF. Then, the* BGM_Γ *is idempotent if, and only, if* $\sum_{i=1}^{n} f_i(x,\cdots,x) = 1$ *for any* $x \in [0,1]$.

Proof. If $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$ and $\mathbf{x} = (x, ..., x)$, then:

$$\mathsf{BGM}_\Gamma(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}) \cdot x = x \cdot \sum_{i=1}^{n} f_i(\mathbf{x}) = x$$

Reciprocally, if BGM is an idempotent function and $\sum_{i=1}^{n} f_i(x, \cdots, x) < 1$ for some $x \in [0,1]$ we have to

$$\mathsf{BGM}_\Gamma(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}) \cdot x < x \cdot 1 = x.$$

Thus, the condition $\sum_{i=1}^{n} f_i(x, \cdots, x) = 1$ can not be removed. □

Corollary 3.17. *Any GM function is idempotent.*

Proof. Straightforward.

□

Example 3.18. *We can not always guarantee that a BGM is idempotent, because if we take $f_1(x, y) = \frac{x}{2}$ and $f_2(x, y) = \frac{y}{2}$, then $\mathsf{BGM}(0.5, 0.5) = 0.25 \neq 0.5$.*

Proposition 3.19. *If Γ is a FWF invariant under translations, i.e, $f_i(x_1+\lambda, x_2+\lambda, ..., x_n+\lambda) = f_i(x_1, x_2, ..., x_n)$ for any $\mathbf{x} \in [0,1]^n$, for $i \in \{1, 2, \cdots, n\}$, satisfying 1 and $\lambda \in [-1, 1]$, then BGM_Γ is shift-invariant.*

Proof. Let $\mathbf{x} = (x_1, ..., x_n) \in [0,1]^n$ and $\lambda \in [-1,1]$ such that $(x_1 + \lambda, x_2 + \lambda, ..., x_n + \lambda) \in [0,1]^n$. then,

$$\begin{aligned}
\mathsf{BGM}_\Gamma(x_1 + \lambda, ..., x_n + \lambda) &= \sum_{i=1}^{n} f_i(x_1 + \lambda, ..., x_n + \lambda) \cdot (x_i + \lambda) \\
&= \sum_{i=1}^{n} f_i(x_1 + \lambda, ..., x_n + \lambda) \cdot x_i + \sum_{i=1}^{n} f_i(x_1 + \lambda, ..., x_n + \lambda) \cdot \lambda \\
&= \sum_{i=1}^{n} f_i(x_1, ..., x_n) \cdot x_i + \lambda \\
&= \mathsf{BGM}_\Gamma(x_1, ..., x_n) + \lambda
\end{aligned}$$

□

Remark 3.20. *The condition 1 is also important to preserve shift-invariance, since if we define $f_1(x, y) = f_2(x, y) = \frac{|x-y|}{2}$, for $(x, y) \neq (1, 1)$, and $f_1(1, 1) = f_2(1, 1) = \frac{1}{2}$, then f_1 and f_2 are invariant under translations, but $\mathsf{BGM}(0, 0.1) = 0.005$ and $\mathsf{BGM}(0 + 0.1, 0.1 + 0.1) = 0.015 \neq 0.005 + 0.1$.*

Proposition 3.21. *If Γ is homogeneous of order k (i.e. if each f_i is homogeneous of order k), then $\mathsf{BGM}_\Gamma(\mathbf{x})$ is homogeneous of order $k + 1$.*

Proof. Of course that, if $\lambda = 0$, then $\mathsf{BGM}_\Gamma(\lambda x_1, ..., \lambda x_n) = \lambda f(x_1, ..., x_n)$. Now, to $\lambda \neq 0$ we have:

$$\begin{aligned}
\mathsf{BGM}_\Gamma(\lambda x_1, ..., \lambda x_n) &= \sum_{i=1}^{n} f_i(\lambda x_1, ..., \lambda x_n) \cdot \lambda x_i \\
&= \lambda \cdot \sum_{i=1}^{n} \lambda^k f_i(x_1, ..., x_n) x_i \\
&= \lambda^{k+1} \cdot \mathsf{BGM}_\Gamma(x_1, ..., x_n)
\end{aligned}$$

□

Remark 3.22. *Note that if* $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$*, then* f_i *cannot be homogeneous of order* $k > 0$*, since*

$$1 = \sum_{i=1}^{n} f_i(\lambda x_1, \cdots, \lambda x_n) = \lambda^k \sum_{i=1}^{n} f_i(\mathbf{x}) = \lambda^k,$$

i.e., there are no GM*'s homogeneous of order* $k > 1$*. However, if we remove this restriction, then we can have* Γ *with homogeneous* f_i*s of order* $k > 0$*. For example,* $f_i(\mathbf{x}) = \frac{x_i}{n}$ *is homogeneous of order* 1*, and so, according to Proposition 3.21,* BGM_Γ *is homogeneous of order* 2*.*

The next example shows a GM function which is not a mixture operator.

Example 3.23. *Let* Γ *be defined by*

$$f_i(x_1, ..., x_n) = \begin{cases} \frac{1}{n}, & \text{if } x_1 = \cdots = x_n = 0 \\ \frac{x_i}{\sum_{j=1}^{n} x_j}, & \text{otherwise} \end{cases}$$

Then,

$$GM_\Gamma(\mathbf{x}) = \begin{cases} 0, & \text{if } x_1, ..., x_n = 0 \\ \frac{\sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n} x_i}, & \text{otherwise} \end{cases}$$

Observe that this function, like that in Example 3.9, cannot be characterized as a mixture function, since f_i does not depend exclusively from x_i. This GM_Γ is idempotent, homogeneous and shift-invariant, but is not monotonic, since $\mathsf{GM}_\Gamma(0.5, 0.2, 0.1) = 0.375$ and $\mathsf{GM}_\Gamma(0.5, 0.22, 0.2) = 0.368$.

Proposition 3.24. *The* N*-dual*[4]*, with respect to stantard fuzzy negation*[5]*, of a* GM *function is also a* GM *function.*

Proof. If Γ is a FWF, then

$$\begin{aligned} \mathsf{GM}_\Gamma^N(x_1, \cdots, x_n) &= 1 - \sum_{i=1}^{n} f_i(1 - x_1, \cdots, 1 - x_n) \cdot (1 - x_i) \\ &= 1 - \sum_{i=1}^{n} f_i(1 - x_1, \cdots, 1 - x_n) + \sum_{i=1}^{n} f_i(1 - x_1, \cdots, 1 - x_n) \cdot x_i \\ &= \sum_{i=1}^{n} f_i(1 - x_1, \cdots, 1 - x_n) \cdot x_i \\ &= \sum_{i=1}^{n} g_i(x_1, \cdots, x_n) \cdot x_i, \end{aligned}$$

where $g_i(x_1, \cdots, x_n) = f_i(1 - x_1, \cdots, 1 - x_n)$. □

Proposition 3.25. *If* $\Gamma = \{f_1, \cdots, f_n\}$ *is a FWF, then* $\Gamma^R = \{f_n, \cdots, f_1\}$ *also is a FWF. Besides that,* $GM_\Gamma^R = GM_{\Gamma^R}$

[4]The N-dual of a function $F : [0,1]^n \longrightarrow [0,1]$ is $F^N(x_1, \cdots, x_n) = N(F(N(x_1), \cdots, N(x_n))$, where N is a fuzzy negation, i.e., a function decreasing function $N : [0,1] \longrightarrow [0,1]$ with $N(0) = 1$ and $N(1) = 0$.

[5]The standard fuzzy negation if $N(x) = 1 - x$.

Proof. Direct from the definition. □

Examples 3.9 and 3.10 show that GM functions encompass both: OWA and Mixture functions, and thus these functions are special cases GM proposed here. It is also important to note that GM and BGM functions, as well as Mixture functions, are not generally aggregations since it fails to satisfy the monotonicity condition. In examples 3.4, 3.5, 3.6, 3.7 and 3.9 the respective GM's are monotonic, but in Example 3.23 (that we bring forward) the function there is not monotonic. When the GM is monotonic, obviously, this function is an aggregation, since the boundary condition is trivially satisfied.

Some conditions for monotonicity of GM functions were studied by Pereira *et al.* in [46, 47, 48]. In this work we will not study monotonicity criteria, but a more weakened form, called **weak monotonicity** or **directional monotonicity**.

3.3 Directional Monotonicity

There are many n-ary functions that do not satisfy the monotonicity condition, but its restriction to certain directions are monotonic functions. In this sense, Wilkin and Beliakov in [57] introduce the concept of **weakly monotonicity** (see also [5]), which was generalized by Bustince *et al.* in [9], which defines the notion of **directional monotonicity**.

Definition 3.26. *Let* $\mathbf{r} = (r_1, \cdots, r_n)$ *a not null* n*-dimentional vector. A function* $F : [0,1]^n \longrightarrow [0,1]$ *is* **r-increasing** *if fo all* $\mathbf{x} = (x_1, \cdots, x_n)$ *and* $t > 0$ *such that* $(x_1 + tr_1, \cdots, x_n + tr_n) \in [0,1]^n$*, we have*

$$F(x_1, \cdots, x_n) \leq F(x_1 + tr_1, \cdots, x_n + tr_n),$$

that is, F *is increasing in the direction of vector* $\mathbf{r}$*.*

Definition 3.27. *A function* $F : [0,1]^n \longrightarrow [0,1]$ *is an* n*-ary* **preaggregation** *function (or simply preaggregation) if satisfies the boundary condition,* $F(0, \cdots, 0) = 0$ *and* $F(1, \cdots, 1) = 1$*, and is* $\mathbf{r}$*-increasing for some direction* $\mathbf{r} \in [0,1]^n$*.*

In [34], Lucca *et al.* was presented properties, constructions and application for preaggregations function. They show that the following functions are examples of preaggregations.

Example 3.28.

1. *Mode*$(x_1, \cdots, x_n)$*, that is* $(1,1)$*-increasing;*
2. $F(x,y) = x - (max\{0, x-y\})^2$*, tha is* $(0,1)$*-increasing;*
3. *The weighted Lehmer mean (with convention* $0/0 = 0$*)*

$$L_\lambda(x,y) = \frac{\lambda x^2 + (1-\lambda)y^2}{\lambda x + (1-\lambda)y}, \text{ where } 0 < \lambda < 1$$

   is $(1-\lambda, \lambda)$*-increasing;*

4. $$A(x,y) = \begin{cases} x(1-x), & \text{if } y \leq 3/4 \\ 1, & \text{otherwise} \end{cases}$$

   is $(0,a)$*-increasing for any* $a > 0$*, but for no other direction;*

5. $$B(x,y) = \begin{cases} y(1-y), & \text{if } x \leq 3/4 \\ 1, & \text{otherwise} \end{cases}$$

   is $(b,0)$*-increasing for any* $b > 0$*, but for no other direction.*

Remark 3.29. *Any aggregation functions is also a preaggregation function.*

Proposition 3.30. *If BGM_Γ is shift-invariant, then BGM_Γ is a preaggregation function $(k, k, \cdots, k)$-increasing.*

Proof. Just see that for all $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in [0,1]^n$ and any $t > 0$ such that $(x_1 + tk, x_2 + tk, \cdots, x_n + tk) \in [0,1]$ we have

$$\mathsf{BGM}_\Gamma(x_1 + tk, \cdots, x_n + tk) = \mathsf{BGM}_\Gamma(x_1, \cdots, x_n) + tk,$$

and so

$$\mathsf{BGM}_\Gamma(x_1, \cdots, x_n) \leq \mathsf{BGM}(x_1 + tk, \cdots, x_n + tk)$$

□

Corollary 3.31. *If Γ is a FWF invariant under translations, i.e, $f_i(x_1+\lambda, x_2+\lambda, ..., x_n+\lambda) = f_i(x_1, x_2, ..., x_n)$, for $i \in \{1, 2, \cdots, n\}$, for any $\mathbf{x} = (x_1, \cdots, x_n) \in [0,1]^n$ and $\lambda \in [0,1]$ such that $(x_1 + \lambda, x_2 + \lambda, ..., x_n + \lambda) \in [0,1]^n$ satisfying 1, BGM_Γ is a preaggregation function $(k, k, \cdots, k)$-increasing.*

Proof. By Proposition 3.19, BGM_Γ is shift-invariant, and so, by Proposition 3.30, BGM_Γ is a preaggregation function $(k, k, \cdots, k)$-increasing. □

In fact, the conditions required by Corollary 3.31 are very strong. In the following proposition, we relax these conditions:

Proposition 3.32. *If Γ is a FWF with $f_i(x_1, \cdots, x_n) \leq f_i(x_1 + \lambda, \cdots, x_i + \lambda)$, for $i \in \{1, 2, \cdots, n\}$, for any $\mathbf{x} = (x_1, \cdots, x_n) \in [0,1]^n$ and $\lambda \in [0,1]$ such that $(x_1 + \lambda, x_2 + \lambda, ..., x_n + \lambda) \in [0,1]^n$, then BGM_Γ is a preaggregation function $(k, k, \cdots, k)$-increasing.*

Proof. For any $\mathbf{x} = (x_1, \cdots, x_n) \in [0,1]^n$ and $\lambda \in [0,1]$ such that $(x_1 + \lambda, x_2 + \lambda, ..., x_n + \lambda) \in [0,1]^n$ we observe that

$$\begin{aligned}
\mathsf{BGM}_\Gamma(x_1 + \lambda, ..., x_n + \lambda) &= \sum_{i=1}^{n} f_i(x_1 + \lambda, ..., x_n + \lambda) \cdot (x_i + \lambda) \\
&= \sum_{i=1}^{n} f_i(x_1 + \lambda, ..., x_n + \lambda) \cdot x_i \; + \; \sum_{i=1}^{n} f_i(x_1 + \lambda, ..., x_n + \lambda) \cdot \lambda \\
&\geq \sum_{i=1}^{n} f_i(x_1, ..., x_n) \cdot x_i + \lambda \\
&\geq \mathsf{BGM}_\Gamma(x_1, ..., x_n)
\end{aligned}$$

□

Example 3.33. *Let Γ whose functions are given by*

$$f_i(x_1, \cdots, x_n) = \begin{cases} \frac{1}{n}, & \text{if } x_1 = \cdots = x_n \\ \frac{x_{(1)} - x_i}{\sum\limits_{j=1}^{n} (x_{(1)} - x_j)}, & \text{otherwise} \end{cases}.$$

We can easily prove that satisfies

$$f_i(x_1 + \lambda, x_2 + \lambda, \cdots, x_n + \lambda) = f_i(x_1, x_2, \cdots, x_n).$$

More generally, for any $\alpha \geq 1$

$$f_i(x_1, \cdots, x_n) = \begin{cases} \frac{1}{n}, & \text{if } x_1 = \cdots = x_n \\ \frac{x_{(1)} - x_i}{\sum\limits_{j=1}^{n} (x_{(1)} - x_j)^\alpha}, & \text{otherwise} \end{cases}$$

is such that

$$f_i(x_1, x_2, \cdots, x_n) \leq f_i(x_1 + \lambda, x_2 + \lambda, \cdots, x_n + \lambda).$$

Thus, the corresponding BGM is $(k, \cdots, k)$-increasing. In additon, note that, for $\alpha > 1$, $\Gamma = \{f_i\}$ does not satisfies $\sum_{i=1}^{n} f_i(\mathbf{x}) = \mathbf{1}$.

We can also establish a criterion analogous to the Proposition 3.32, substituting the vector $(k, \cdots, k)$ for any direction $\mathbf{r}$, as follow:

Proposition 3.34. *If Γ is a FWF such that there is a diretional vector $\mathbf{r} = (r_1, r_2, \cdots, r_n) \in [0,1]^n$ with $f_i(x_1, \cdots, x_n) \leq f_i(x_1 + \lambda \cdot r_1, \cdots, x_i + \lambda \cdot r_n)$, for $i \in \{1, 2, \cdots, n\}$, for any $\mathbf{x} = (x_1, \cdots, x_n) \in [0,1]^n$ and $\lambda \in [0,1]$ such that $(x_1 + \lambda \cdot r_1, x_2 + \lambda \cdot r_2, ..., x_n + \lambda \cdot r_n) \in [0,1]^n$, then BGM_Γ is a preaggregation function $\mathbf{r}$-increasing.*

Proof. Is similar to what was done in Proposition 3.32. □

Corollary 3.35. *If Γ is a FWF such that there is a diretional vector $\mathbf{r}$ with $\frac{\partial f_i}{\partial \mathbf{r}}(\mathbf{x}) \geq 0$ for any $f_i \in \Gamma$ and $\mathbf{x} \in [0,1]^n$, then BGM_Γ is a preaggregation function $\mathbf{r}$-increasing.*

Note that this condition can not be satisfied, in the case that $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$, for all $\mathbf{x} \in [0,1]^n$, unless that the functions f_i are constant in the direction of vector $\mathbf{r}$, because:

$$\sum_{i=1}^{n} f_i(\mathbf{x}) = 1 \implies \sum_{i=1}^{n} \frac{\partial f_i(\mathbf{x})}{\partial \mathbf{r}} = 0$$

and so,

$$\frac{\partial f_i(\mathbf{x})}{\partial \mathbf{r}} \geq 0 \implies \frac{\partial f_i(\mathbf{x})}{\partial \mathbf{r}} = 0$$

Example 3.36. *Obviously, if $f_i = w_i$ is constant, then BGM_Γ is $\mathbf{r}$-increaing for any direction $\mathbf{r}$. Now, given a direction $\mathbf{r} = (r_1, \cdots, r_n) \in [0,1]^n$ we can build a $\mathbf{r}$-increaing BGM function defining:*

$$f_i(x_1, \cdots, x_n) = \begin{cases} 0, & \text{if } min\{x_1, \cdots, x_n\} = 0 \\ \frac{min\left\{\frac{x_i}{r_i}, 1\right\}}{n}, & \text{otherwise} \end{cases},$$

we obtain a BGM $\mathbf{r}$-increasing.

As previously mentioned, both the Aggregation functions $(Min, Max, Med, Arith, \mathsf{OWA}, \cdots)$ and generalized mixture functions (and also bounded generalized mixture functions) can be used in many applications. To finalize this paper we bring an illustrative example of application, where we apply some functions in the scope of image processing. More precisely, we will use generalized mixture functions in the image reduction process.

Before presenting this example of application, we propouse a special GM function, which satisfies several interesting properties, as we will show in this paper, and will be used in the application.

Definition 3.37. *Consider the family Γ of functions*

$$f_i(\mathbf{x}) = \begin{cases} \frac{1}{n}, & \text{if } \mathbf{x} = (x, ..., x) \\ \frac{1}{n-1}\left(1 - \frac{|x_i - Med(\mathbf{x})|}{\sum_{j=1}^{n} |x_j - Med(\mathbf{x})|}\right), & \text{otherwise} \end{cases}$$

Γ is a FWF, with $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$ for all $\mathbf{x} \in [0,1]^n$, i.e., BGM_Γ is a GM, that will be denoted by $\mathbf{H}$. The computation of $\mathbf{H}$ can be performed using the following expressions:

$$\mathbf{H}(\mathbf{x}) = \begin{cases} x, & \text{if } \mathbf{x} = (x, ..., x) \\ \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \frac{x_i|x_i - Med(\mathbf{x})|}{\sum_{j=1}^{n}|x_j - Med(\mathbf{x})|}\right), & \text{otherwise} \end{cases}$$

Example 3.38. *Let be $n = 5$. So, for $\mathbf{x} = (0.1, 0.25, 0.3, 0, 1)$ we have*

$$f_1(\mathbf{x}) = 0.21875,\ \ f_2(\mathbf{x}) = 0.25,\ \ f_3(\mathbf{x}) = 0.2395, f_4(\mathbf{x}) = 0.198, f_5(\mathbf{x}) = 0.09375$$

And

$$\mathbf{H}(\mathbf{x}) = 0.249975.$$

Note that the larger weights occur in the coordinates closest to the median. Besides, if we take the fixed vector of weights $\mathbf{w} = (0.21875, 0.25, 0.2395, 0.198, 0.09375)$, then $\mathsf{OWA}_\mathbf{w}(0.1, 0.25, 0.3, 0, 1) = 0.249975 = \mathbf{H}(0.1, 0.25, 0.3, 0, 1)$. In other words, the function $\mathbf{H}$ can be seen as a function that transforms each input $\mathbf{x}$ into the output of an OWA. More precisely,

$$\mathbf{H}(\mathbf{x}) = \mathsf{OWA}_{(f_1(\mathbf{x}),\cdots,f_n(\mathbf{x}))}(\mathbf{x})$$

It is not difficult to see that the above equation holds for all $n \in \mathbb{N}$ and $\mathbf{x} \in [0,1]^n$.

In the next subsection we discuss others properties of the function $\mathbf{H}$.

3.4 Properties of H

In this part of paper we will discuss about the properties of operator $\mathbf{H}$. It is easy to check that $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$ for any $\mathbf{x} \in [0,1]^n$ and therefore, by Propositions 3.14 and 3.16, $\mathbf{H}$ is an averaging and idempotent function. Furthermore,

Proposition 3.39. *The weight-functions at Definition 3.37 are invariant under translations and is also homogeneous of order 0.*

Proof. Let $\mathbf{x} = (x_1, ..., x_n) \in [0,1]^n$ and $\lambda \in [0,1]$ such that $\mathbf{x}' = (x_1 + \lambda, ..., x_n + \lambda) \in [0,1]^n$. Then, since $Med(\mathbf{x}') = Med(\mathbf{x}) + \lambda$ we have, for $\mathbf{x} \neq (x, ..., x)$:

$$\begin{aligned} f_i(\mathbf{x}') &= \frac{1}{n-1}\left(1 - \frac{|x_i+\lambda-Med(\mathbf{x}')|}{\sum_{j=1}^{n}|x_j+\lambda-Med(\mathbf{x}')|}\right) \\ &= \frac{1}{n-1}\left(1 - \frac{|x_i+\lambda-(Med(\mathbf{x})+\lambda)|}{\sum_{j=1}^{n}|x_j+\lambda-(Med(\mathbf{x})+\lambda)|}\right) \\ &= \frac{1}{n-1}\left(1 - \frac{|x_i-Med(\mathbf{x})|}{\sum_{j=1}^{n}|x_j-Med(\mathbf{x})|}\right) \\ &= f_i(\mathbf{x}). \end{aligned}$$

Therefore, $(f_1(\mathbf{x}'), ..., f_n(\mathbf{x}')) = (f_1(\mathbf{x}), ..., f_n(\mathbf{x}))$. The case in which $\mathbf{x} = (x, ..., x)$ is immediate.

To check the second property, make $\mathbf{x}'' = (\lambda x_1, ..., \lambda x_n)$, note that $Med(\mathbf{x}'') = \lambda Med(\mathbf{x})$ and for $\mathbf{x} \neq (x, ..., x)$

$$\begin{aligned} f_i(\mathbf{x}'') &= \frac{1}{n-1}\left(1 - \frac{|\lambda x_i - Med(\lambda \mathbf{x})|}{\sum_{j=1}^{n} |\lambda x_j - Med(\lambda \mathbf{x})|}\right) \\ &= \frac{1}{n-1}\left(1 - \frac{|\lambda x_i - \lambda Med(\mathbf{x})|}{\sum_{j=1}^{n} |\lambda x_j - \lambda Med(\mathbf{x})|}\right) \\ &= \frac{1}{n-1}\left(1 - \frac{|\lambda| \cdot |x_i - Med(\mathbf{x})|}{|\lambda| \cdot \sum_{j=1}^{n} |x_j - Med(\mathbf{x})|}\right) \\ &= \frac{1}{n-1}\left(1 - \frac{|x_i - Med(\mathbf{x})|}{\sum_{j=1}^{n} |x_j - Med(\mathbf{x})|}\right) \\ &= f_i(\mathbf{x}) \end{aligned}$$

Hence, $(f_1(\mathbf{x}''), ..., f_n(\mathbf{x}'')) = (f_1(\mathbf{x}), ..., f_n(\mathbf{x})) = f(\mathbf{x})$. The case in which $\mathbf{x} = (x, ..., x)$ is also immediately. □

Corollary 3.40. **H** *is shift-invariant and homogeneous.*

Proof. Straightforward for Propositions 3.19 and 3.21. □

In addition to idempotency, homogeneity and shift-invariance **H** has the following proprerties.

Proposition 3.41. **H** *has no neutral element.*

Proof. Suppose **H** has a neutral element e, find the vector of weight for $\mathbf{x} = (e, ..., e, x, e, ..., e)$. Note that if $n \geq 3$, then $Med(\mathbf{x}) = e$ and therefore,

$$\begin{aligned} f_i(\mathbf{x}) &= \frac{1}{n-1}\left(1 - \frac{|x_i - Med(\mathbf{x})|}{\sum_{j=1}^{n} |x_j - Med(\mathbf{x})|}\right) \\ &= \frac{1}{n-1}\left(1 - \frac{|x_i - e|}{\sum_{j=1}^{n} |x_j - e|}\right) \\ &= \frac{1}{n-1}\left(1 - \frac{|x_i - e|}{|x - e|}\right). \end{aligned}$$

So,

$$f_i(\mathbf{x}) = \begin{cases} \frac{1}{n-1}, & \text{if } x_i = e \\ 0, & \text{if } x_i = x \end{cases}, \text{ to } n \geq 3$$

i.e.,

$$f(\mathbf{x}) = \left(\frac{1}{n-1}, ..., \frac{1}{n-1}, 0, \frac{1}{n-1}, ..., \frac{1}{n-1}\right)$$

and

$$\mathbf{H}(\mathbf{x}) = (n-1)\cdot\frac{e}{n-1} = e$$

But since e is a neutral element of $\mathbf{H}$, $\mathbf{H}(\mathbf{x}) = x$. Absurd, since we can always take $x \neq e$.

For $n = 2$, we have $Med(\mathbf{x}) = \frac{x+e}{2}$, where $\mathbf{x} = (x, e)$ or $\mathbf{x} = (e, x)$. In both cases it is not difficult to show that $f(\mathbf{x}) = (0.5, 0.5)$ and $\mathbf{H}(\mathbf{x}) = \frac{x+e}{2}$. Thus, taking $x \neq e$, again we have $\mathbf{H}(x, e) \neq x$. □

Proposition 3.42. **H** *has no absorbing elements.*

Proof. To $n = 2$, we have $\mathbf{H}(\mathbf{x}) = \frac{x_1+x_2}{2}$, which has no absorbing elements. Now for $n \geq 3$ we have to $\mathbf{x} = (a, 0, ..., 0)$ with $Med(\mathbf{x}) = 0$ therefore,

$$f_1(\mathbf{x}) = \frac{1}{n-1}\left(1 - \frac{a}{a}\right) = 0 \text{ and } f_i(\mathbf{x}) = \frac{1}{n-1}, \forall i = 2, ..., n.$$

therefore,

$$\mathbf{H}(a, 0, ..., 0) = 0\cdot a + \frac{1}{n-1}\cdot 0 + ... + \frac{1}{n-1}\cdot 0 = a \Rightarrow a = 0,$$

but to $\mathbf{x} = (a, 1, ..., 1)$ we have to $Med(\mathbf{x}) = 1$. Furthermore,

$$f_1(\mathbf{x}) = \frac{1}{n-1}\left(1 - \frac{1-a}{1} - a\right) = 0$$

and

$$f_i(\mathbf{x}) = \frac{1}{n-1} \text{ for } i = 2, 3, ..., n.$$

therefore,

$$\mathbf{H}(a, 1, ..., 1) = 0\cdot a + \frac{1}{n-1}\cdot 1 + ... + \frac{1}{n-1}\cdot 1 = a \Rightarrow a = 1.$$

With this we prove that **H** does note have annihiladors. □

Proposition 3.43. **H** *has no zero divisors.*

Proof. Let $a \in]0, 1[$ and consider $\mathbf{x} = (a, x_2, ..., x_n) \in]0, 1]^n$. In order to have $\mathbf{H}(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})\cdot x_i = 0$ we have $f_i(\mathbf{x})\cdot x_i = 0$ for all $i = 1, 2, ..., n$. But as $a \neq 0$ and we can always take $x_2, x_3, ..., x_n$ also different from zero, then for each $i = 1, 2, ..., n$ there remains only the possibility of terms:

$$f_i(\mathbf{x}) = 0 \text{ for } i = 1, 2, ..., n.$$

This is absurd, for $f_i(\mathbf{x}) \in [0, 1]$ and $\sum_{i=1}^{n} f_i(\mathbf{x}) = 1$. like this, **H** has no zero divisors. □

Proposition 3.44. **H** *does not have one divisors*

Proof. Just to see that $a \in]0, 1[$, we have to $\mathbf{H}(a, 0, ..., 0) = f_1(\mathbf{x}).a \leq a < 1$. □

Proposition 3.45. **H** *is symmetric.*

Proof. Let $P : \{1, 2, ..., n\} \to \{1, 2, ..., n\}$ be a permutation. So we can easily see that

$$Med(x_{P(1)}, x_{P(2)}, ..., x_{P(n)}) = Med(x_1, x_2, ..., x_n)$$

for all $\mathbf{x} = (x_1, x_2, ..., x_n) \in [0, 1]^n$. We also have to $\sum_{i=1}^{n} |x_{P(i)} - Med(x_{P(1)}, x_{P(2)}, ..., x_{P(n)})| = \sum_{i=1}^{n} |x_i - Med(\mathbf{x})|$. Thus, it suffices to consider the case where $(x_{P(1)}, x_{P(2)}, ..., x_{P(n)}) \neq (x, x, ..., x)$. But $(x_{P(1)}, x_{P(2)}, ..., x_{P(n)}) \neq (x, x, ..., x)$ we have to:

$$\begin{aligned}
\mathbf{H}(x_{P(1)}, x_{P(2)}, ..., x_{P(n)}) &= \frac{1}{n-1}\sum_{i=1}^{n}\left(x_{P(i)} - \frac{x_{P(i)}|x_{P(i)} - Med(x_{P(1)}, ..., x_{P(n)})|}{\sum_{j=1}^{n}|x_{P(i)} - Med(x_{P(1)}, ..., x_{P(n)})|}\right) \\
&= \frac{\sum_{i=1}^{n} x_{P(i)}}{n-1} - \frac{1}{n-1}\cdot\sum_{i=1}^{n}\frac{x_{P(i)}|x_{P(i)} - Med(x_1, ..., x_n)|}{\sum_{j=1}^{n}|x_{P(i)} - Med(x_1, ..., x_n)|} \\
&= \frac{\sum_{i=1}^{n} x_i}{n-1} - \frac{1}{n-1}\cdot\sum_{i=1}^{n}\frac{x_{P(i)}|x_{P(i)} - Med(x_1, ..., x_n)|}{\sum_{j=1}^{n}|x_i - Med(x_1, ..., x_n)|} \\
&= \frac{\sum_{i=1}^{n} x_i}{n-1} - \frac{1}{n-1}\cdot\sum_{i=1}^{n}\frac{x_i|x_i - Med(x_1, ..., x_n)|}{\sum_{j=1}^{n}|x_i - Med(x_1, ..., x_n)|} \\
&= \mathbf{H}(x_1, ..., x_n).
\end{aligned}$$

□

Proposition 3.46. *If $N : [0, 1] \longrightarrow [0, 1]$ is the standard fuzzy negation, then $\mathbf{H}^N = \mathbf{H}$.*

Proof. If $\mathbf{x} = (x, \cdots, x)$, then

$$\mathbf{H}^N(\mathbf{x}) = 1 - \mathbf{H}(1 - x, 1 - x, \cdots, 1 - x) = 1 - (1 - x) = x = \mathbf{H}(\mathbf{x})$$

For $\mathbf{x} \neq (x, \cdots, x)$, we have:

$$\begin{aligned}
\mathbf{H}^N(\mathbf{x}) &= 1 - \frac{1}{n-1}\sum_{i=1}^{n}\left(1 - x_i - \frac{(1-x_i)|1-x_i-Med(1-x_1, \cdots, 1-x_n)|}{\sum_{j=1}^{n}|1-x_i-Med(1-x_1, \cdots, 1-x_n)|}\right) \\
&= 1 - \frac{1}{n-1}\sum_{i=1}^{n}\left(1 - x_i - \frac{(1-x_i)|1-x_i-1+Med(x_1, \cdots, x_n)|}{\sum_{j=1}^{n}|1-x_i-1+Med(x_1, \cdots, x_n)|}\right) \\
&= 1 - \frac{1}{n-1}\sum_{i=1}^{n}\left(1 - x_i - \frac{(1-x_i)|-x_i+Med(x_1, \cdots, x_n)|}{\sum_{j=1}^{n}|-x_i+Med(x_1, \cdots, x_n)|}\right)
\end{aligned}$$

$$
\begin{aligned}
&= 1-\frac{1}{n-1}\sum_{i=1}^{n}\left(1-x_i-\frac{(1-x_i)|x_i-Med(x_1,\cdots,x_n)|}{\sum_{j=1}^{n}|x_i-Med(x_1,\cdots,x_n)|}\right)\\
&= 1-\frac{1}{n-1}\left[n-\sum_{i=1}^{n}\left(x_i-\frac{x_i|x_i-Med(x_1,\cdots,x_n)|}{\sum_{j=1}^{n}|x_i-Med(x_1,\cdots,x_n)|}\right)-\sum_{i=1}^{n}\frac{|x_i-Med(x_1,\cdots,x_n)|}{\sum_{j=1}^{n}|x_i-Med(x_1,\cdots,x_n)|}\right]\\
&= 1-\frac{1}{n-1}\left[n-1-\sum_{i=1}^{n}\left(x_i-\frac{x_i|x_i-Med(x_1,\cdots,x_n)|}{\sum_{j=1}^{n}|x_i-Med(x_1,\cdots,x_n)|}\right)\right]\\
&= \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i-\frac{x_i|x_i-Med(x_1,\cdots,x_n)|}{\sum_{j=1}^{n}|x_i-Med(x_1,\cdots,x_n)|}\right)\\
&= \mathbf{H}(\mathbf{x})
\end{aligned}
$$

□

Therefore, **H** satisfies the following properties:

- Idempotence
- Homogeneity
- Shift-invariance
- Symmetry.
- has no neutral element
- has no absorbing elements
- has no zero divisors
- does not have one divisors
- is self dual

Although we have not been able to demonstrate that **H** is an aggregation function, in the next proposition we show that **H** is $(k,\cdots,k)$-increasing (for $k>0$), so **H** is a preaggregation function.

Proposition 3.47. *If $k>0$, then* **H** *is $(k,\cdots,k)$-increasing.*

Proof. As **H** is shift-invariant, its follow of Proposition 3.30 that **H** is $(k,\cdots,k)$-increasing. □

Corollary 3.48. **H** *is a preaggregation function.*

The aggregation functions are very important for computing science, since in many applications the expected result is a single data, and therefore these applications use an aggregation function to convert this set of data into a unique output. In fact, a preaggregation can often be applied in place of aggregation. In this sense, we will apply the function **H** (which is a GM function) (in an illustrative example) to reduce images and then we compare the obtained results with the results obtained by some aggregations.

4 The Image Reduction by GM functions

In this part of our work we use the GM functions Min, Max, $Arith$, Med, cOWA and **H** to build image reduction operators and is an improvement of the done in [18]. But first, we will introduce some important concepts of image processing.

Definition 4.1. *An image is a matrix* $m \times n$*,* $M = A(i,j)$*, where each* $A(i,j) \in [0,1]$ *represents a pixel. More specifically, the value* $A(i,j)$ *is proportional to the light intensity at the considered point.*

In essence, a reduction operator reduces a given image $m \times n$ to another $m' \times n'$, such that $m' < m$ and $n' < n$. For example,

$$\begin{bmatrix} 0.1 & 0.2 & 0 & 0.5 \\ 0.3 & 0.3 & 0.2 & 0.8 \\ 1 & 0.5 & 0.6 & 0.4 \\ 0 & 0.3 & 0.5 & 0.7 \end{bmatrix} \longmapsto \begin{bmatrix} 0.1 & 0 \\ 1 & 0.6 \end{bmatrix}$$

There are several possible ways to reduce a given image, as shown in the following example:

Example 4.2. *The image*

$$M = \begin{bmatrix} 0.8 & 0.7 & 0.2 & 1 & 0.5 & 0.5 \\ 0.6 & 0.2 & 0.3 & 0.1 & 1 & 0 \\ 0 & 0 & 0.6 & 0.4 & 0.9 & 1 \\ 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 \end{bmatrix},$$

can be reduced to another 2×3 *by partitioning* M *in blocks* 2×2 *and applying to each block, for example, the function* $f(x,y,z,w) = Max(x,y,z,w)$*. In this case, we obtain the image:*

$$M_* = \begin{bmatrix} 0.8 & 1 & 1 \\ 0.2 & 0.6 & 1 \end{bmatrix}$$

The Figure 1 illustrates the reduction process of an image.

Figure 1: Example of image redction.

In fact, if we apply any other function, we get a new image, usually different from the previous one, but what is the best?

One possible answer to this question involves a method called **magnification** or **extension** (see [27, 62, 63]), which is a method which magnifies the reduced image to another with the same size of the original one. The magnified image is then compared with the original input image.

Example 4.3. *From M_* we can build a 4×6 image imply cloning each pixel (also known as nearest neighbor interpolation), as below:*

$$\left[\ x\ \right] \longmapsto \begin{bmatrix} x & x \\ x & x \end{bmatrix}$$

Thus, we obtain the following image:

$$M_1 = \begin{bmatrix} 0.8 & 0.8 & 1 & 1 & 1 & 1 \\ 0.8 & 0.8 & 1 & 1 & 1 & 1 \\ 0.2 & 0.2 & 0.6 & 0.6 & 1 & 1 \\ 0.2 & 0.2 & 0.6 & 0.6 & 1 & 1 \end{bmatrix}$$

This simple magnification method is also called of nearest neighbor interpolation. The Figure 2 illustrates the magnification process.

Figure 2: Example of magnification.

Given two different reductions of the same image (let's say M' and M^*), We compare the reductions following the steps: (1) Use a magnification method to magnify M' and M^* for the original size; (2) Compare each obtained image with the original one, using a some similarity measure.

There are several similarity measures, as for example, the measure PSNR (see [23]), that is calculated as follows:

$$PSNR(I,K) = 10 \cdot log_{10}\left(\frac{MAX_I^2}{MSE(I,K)}\right),$$

where $I = I(i,j)$ and $K = K(i,j)$ are two images, $MSE(I,K) = \frac{1}{nm}\sum_{i=1}^{m}\sum_{j=1}^{n}[I(i,j) - K(i,j)]^n$ and MAX_I is the maximum possible pixel value of pixel.

The degree of similarity between two images is proportional to the value of the PSNR, i.e., how much larger if the PSNR, more approximated are the analyzed images[6].

In what follows, we use the GM functions: Min, Mix, Med, $Arith$ cOWA and **H** to reduce images in grayscale[7], applying the following method:

[6] In particular, if the input image are equal, then the MSE value is zero and the PSNR will be infinity.

[7] The reduction of color images is similar.

Method 1

1. Reduce the input images using the Min, Max, $Arith$, Med, cOWA and **H**;
2. Magnify the reduced images to the original size using the nearest neighbor interpolation;
3. Compare the last image with the original one using the measure $PSNR$.

Remark 4.4. *This is a general method which can be applied to any kind of image. In this work we applied it to 10 images in grayscale of size* 512×512[8] *(as shown in Figure 3).*

Figure 3: Imput images

In the Tables 1 and 2 (see Appendix) we present the PSNR values between the output images provided by Method 1 and original inputs.

According to PSNR, $Arith$ provided the higher quality images. However, the reduction operators generated by **H** and cOWA provide with us quite similar images to those given by $Arith$.

Note that although the magnification method by cloning of pixels is a simple and quick method (in running time) it brings us some limitations. The results obtained by this method are not good, in addition, the method itself causes that the $Arith$ operator is better than other operators, since by reducing a set of pixels x_1, x_2, x_3, x_4 to a single pixel y, and then compare $MSE = (x_1, y)^2 + (x_2 - y)^2 + (x_3 - y)^2 + (x_4 - y)^2$ (because each pixel y is repeated 4 times in the process of magnification), so of course $y = \frac{1}{4}(x_1 + x_2 + x_3 + x_4)$ has the lowest measurement error.

For this reason we also analyze two other methods of magnification: (1) Bilinear interpolation and (2) Bicubic interpolation (see [23, 28, 30, 55]). Thus, we have two other methods: Method 2 and Method 3, respectively

In Tables 3, 4, 5 and 6 (see Appendix), we present the results obtained with the use of these others magnification methods.

Tables 1, 2, 3, 4, 5 and 6 (see Appendix) show us that among the analyzed GM, the averaging functions ($Arith$, Med, cOWA and **H**) are responsible for generating better quality images. However it is difficult to determine the most appropriate function to reduce images, since each particular function may be more suitable for a certain method of magnification, for example: $Arith$ is closer to magnifying by pixels cloning.

We can also observe that a more complex method of magnification, interpolation, are able to reconstruct images with higher quality. Obviously, the computational cost (running time) of these methods are also higher.

[8] In this paper we made two reductions: using 2×2 blocks and 4×4 blocks.

Ii is worth to emphasize that the reduction with H operator together with magnification by bicubic interpolation scored the highest quality among all analyzed methods (function together magnification) or both reduction: In scale as 2×2 and in scale 4×4.

This shows that in some applications, the use of a generating function of weights (i.e., a weight-function) in order to obtain a GM function may be more interesting than the use of a single weight vector.

This idea of replacing the weight vector by a weight function may also be used in others areas of computing, for example: In decision making and in artificial intelligence. These publications will be investigated in future work.

5 Final Remarks

In this paper we study two generalized forms of Ordered Weighted Averaging function and Mixture function, calls respectively of **Generalized Mixture** and **Bounded Generalized Mixture** functions. These functions are defined by weights, which are obtained dynamically from each input vector $\mathbf{x} \in [0,1]^n$. We demonstrated, among other results, that OWA and mixture functions are particular cases of GM and BGM functions, and thus we obtain that functions such as *Arithmetic Mean*, *Median*, *Maximum*, *Minimum* and cOWA are also examples of GM functions.

In the second part of this work, we present some properties as well as constructs and examples of GM functions. In particular we define a special GM function, called **H**, and show that **H** satisfies important properties for image applications: Idempotence, symmetry, homogeneity, shift-invariance, and moreover, it has no zero divisors and one divisors, and also does not have neutral elements. We further prove that **H** is a preaggregation function $(k, \cdots, k)$-increasing, and then we use GM functions ($Min, Max, Med, Arith$, cOWA and **H**) to verify the applicability of these functions, in this paper for image reduction.

To determine whether these functions are good reducers of images, we need a method of magnification. In Method 1, we magnify images by simply cloning the pixels. However this method brings some limitations, therefore also analyzes the other two magnification methods (bilinear and bicubic interpolation), giving rise to Methods 2 and 3. This other methods are more suitable, and we see that **H** is a fine function to perform this task, using Method 3.

Note that the generalized mixture functions can also be used in others fields of application, for example in data classification [13] and decision making [49]. In this paper, your focus is on just one of this possibility of applications. However, other applications will be investigated in future works.

Acknowledgements: This work is partially supported by the Brazilian National Council for Scientific and Technological Development CNPq under the Processes 311429/2020-3 and 312053/2018-5.

Conflict of Interest: The authors declare that there are no conflict of interest.

References

[1] M. Baczyński and B. Jayaram, Fuzzy Implications, Vol. 231 of Studies in Fuzziness and Soft Computing, *Springer, Berlin*, (2008). doi: 10.1007/978-3-540-69082-5.

[2] T. V. V. Batista, B. Bedregal and R. M. Moraes, Constructing multi-layer classifier ensembles using the Choquet integral based on overlap and quasi-overlap functions, *Neurocomputing*, 500 (2022), 413–421.

[3] G. Beliakov, A. Pradera and T. Calvo, Aggregation Functions: A Guide for Practitioners, Vol. 221 of Studies in Fuzziness and Soft Computing, *Springer, Berlin*, (2007).

[4] G. Beliakov, H. Bustince and D. Paternain, Image reduction using means on discrete product lattices, *IEEE Trans. Image Process.*, 21 (2012), 1070–1083.

[5] G. Beliakov, T. Calvo and T. Wilkin, On the weak monotonicity of Gini means and other mixture functions, *Inf. Sci.*, 300 (2015), 70–84.

[6] G. Beliakov, H. Bustince and T. Calvo, A Practical Guide to Averaging Functions, Vol. 329 of Studies in Fuzziness and Soft Computing, *Springer, Berlin*, (2016).

[7] G. Beliakov, G. Das, H. Q. Vu, T. Wilkin and Y. Xiang, Fuzzy connectives for efficient image reduction and speeding up image analysis, *IEEE Access*, 6 (2018), 68403–68414.

[8] H. Bustince, M. Galar, B. Bedregal, A. Kolesárová and R. Mesiar, A new approach to interval-valued Choquet integrals and the problem of ordering in interval-valued fuzzy set applications, *IEEE Trans. Fuzzy Syst.*, 21 (2013), 1150–1162.

[9] H. Bustince, J. Fernandez, A. Kolesárová and R. Mesiar, Directional monotonicity of fusion functions, *Eur. J. Oper. Res.*, 244 (2015), 300–308.

[10] H. Bustince, R. Mesiar, J. Fernández, M. Galar, D. Paternain, A. H. Altalhi, G. P. Dimuro, B. R. C. Bedregal and Z. Takác, d-Choquet integrals: Choquet integrals based on dissimilarities, *Fuzzy Sets Syst.*, 414 (2021), 1–27.

[11] H. Bustince, B. Bedregal, M.J. Campión, I.A. da Silva, J. Fernández, E. Induráin, A. Raventós-Pujol, R. H. N. Santiago, Aggregation of Individual Rankings Through Fusion Functions: Criticism and Optimality Analysis, *IEEE Trans. Fuzzy Syst.*, 30(3) (2022), 638–648.

[12] C.-H. Cheng and J.-R. Chang, MCDM aggregation model using situational ME-OWA and ME-OWGA operators, *Int. J. Uncertain. Fuzziness Knowl. Based Syst*, 14 (2006), 421–443.

[13] V. S. Costa, A. D. S. Farias, B. R. C. Bedregal, R. H. N. Santiago and A. M. de P. Canuto, Combining multiple algorithms in classifier ensembles using generalized mixture functions, *Neurocomputing*, 313 (2018), 402–414.

[14] S. C. Dighe and R. Shriram, Dental biometrics for human identification based on dental work and image properties in Periapical radiographs, *TENCON 2012 - 2012 IEEE Region 10 Conference*, (2012), 1–6. doi: 10.1109/TENCON.2012.6412216.

[15] G. P. Dimuro, J. Fernández, B. R. C. Bedregal, R. Mesiar, J. A. Sanz, G. Lucca and H. Bustince, The state-of-art of the generalizations of the Choquet integral: From aggregation and pre-aggregation to ordered directionally monotone functions, *Inf. Fusion*, 57 (2020), 27–43.

[16] D. Dubois and H. Prade, Fundamentals of Fuzzy Sets, Vol. 7 of The Handbooks of Fuzzy Sets, 1 ed., *Springer, New York*, (2000).

[17] D. Dubois and H. Prade, On the use of aggregation operations in information fusion processes, *Fuzzy Sets Syst.*, 142 (2004), 143–161.

[18] A. D. S. Farias, V. S. Costa, R. H. N. Santiago and B. R. C. Bedregal, The image reduction process based on generalized mixture functions, *2016 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS), 31 October 2016 – 04 November 2016, El Paso, TX, USA* , (2016), 1–6. doi: 10.1109/NAFIPS.2016.7851591.

[19] A. D. S. Farias, L. R. A. Lopes, B. C. Bedregal and R. H. N. Santiago, Closure properties for fuzzy recursively enumerable languages and fuzzy recursive languages, *J. Intell. Fuzzy Syst.*, 31 (2016), 1795–1806.

[20] A. D. S. Farias, R. H. N. Santiago and B. R. C. Bedregal, Some properties of generalized mixture functions, 2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE, July 24-29, 2016, Vancouver, BC, Canada, (2016), 288–293. doi: 10.1109/FUZZ-IEEE.2016.7737699.

[21] A. D. S. Farias, C. Callejas, R. H. N. Santiago and B. R. C. Bedregal, Directional and ordered directional monotonicity of generalized and bounded generalized mixture functions, *2018 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE, July 8-13, 2018, Rio de Janeiro, Brazil*, (2018), 1–7. doi: 10.1109/FUZZ-IEEE.2018.8491541.

[22] M. Grabisch, J. L. Marichal, R. Mesiar and E. Pap, Aggregation Functions, Vol. 127 of Encyclopedia of Mathematics and its Applications, *University Press Cambridge, Cambridge*, (2009).

[23] R. C. Gonzales and R. E. Woods, Digital Image Processing, 3rd ed., *Pearson, New Jersey*, (2008).

[24] E. Hancer, B. Xue, M. Zhang, D. Karaboga and B. Akay, A multi-objective artificial bee colony approach to feature selection using fuzzy mutual information, *2015 IEEE Congress on Evolutionary Computation (CEC)*, (2015), 2420–2427. doi: 10.1109/CEC.2015.7257185.

[25] G. Jaffino, A. Banumathi, U. Gurunathan, J. P. Jose, Dental work extraction for different radiographic images in human forensic identification, *2014 International Conference on Communication and Network Technologies (ICCNT)*, (2014), 52–56. doi: 10.1109/CNT.2014.7062724.

[26] R. P. Joseph, C. S. Singh and M. Manikandan, Brain tumor MRI image segmentation and detection in image processing, *Int. J. Res. Technol.*, 3 (2014), 1–5.

[27] A. Jurio, M. Pagola, R. Mesiar, G. Beliakov and H. Bustince, Image magnification using interval information, *IEEE Trans. Image Process.*, 20 (2011), 3112–3123.

[28] R. Keys, Cubic convolution interpolation for digital image processing, *IEEE Trans. Acoust. Speech Signal Process.*, 29 (1981).

[29] J. H. Kim, I. J. Ahn, W. H. Nam and J. B. Ra, An effective post-filtering framework for 3-D PET image denoising based on noise and sensitivity characteristics, *IEEE Trans. Nucl. Sci.*, 62 (2015), 137–147.

[30] T. M. Lehmann, C. Gonner and K. Spitzer, Survey: interpolation methods in medical image processing, *IEEE Trans. Medical Imaging*, 18 (1999), 1049–1075.

[31] X. Liang and W. Xu, Aggregation method for motor drive systems, *Electric Power Syst. Research*, 117 (2014), 27–35.

[32] L. Lingling, Z. Xian, H. Pengju and L. Zhigang, The research on the method of fuzzy information processing, *2012 3rd International Conference on System Science, Engineering Design and Manufacturing Informatization (ICSEM)*, 2 (2012), 47–50. doi: 10.1109/ICSSEM.2012.6340804.

[33] I. Lizasoain and C. Moreno, OWA operators defined on complete lattices, *Fuzzy Sets Syst.*, 224 (2013), 36 – 52.

[34] G. Lucca, J. A. Sanz, G. P. Dimuro, B. Bedregal, R. Mesiar, A. Kolesárová and H. Bustince, Preaggregation functions: Construction and an application, *IEEE Trans. Fuzzy Syst.*, 24 (2016), 260–272.

[35] G. Lucca, J. A. Sanz, G. P. Dimuro, B. R. C. Bedregal and H. Bustince, A proposal for tuning the α parameter in C_αC-integrals for application in fuzzy rule-based classification systems, *Nat. Comput.*, 19 (2020), 533–546.

[36] N. V. Manokar, V. Manokar, Rinesh, K. P. Sridhar and L. M. Patnaik, Wavelets based decomposition and classification of diseased fMRI brain images for inter racial disease types of Alzheimer's vs tumors using SOFM and enhancement by LVQ neural networks, *2012 2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC)*, (2012), 822–827. doi: 10.1109/PDGC.2012.6449929.

[37] J. M. Merigó and A. M. Gil-Lafuente, The induced generalized OWA operator, *Inf. Sci.*, 179 (2009), 729–741.

[38] R. Mesiar and J. Špirková, Weighted means and weighting functions, *Kybernetika*, 42 (2006), 151–160.

[39] R. Mesiar, J. Špirková and L. Vavríková, Weighted aggregation operators based on minimization, *Inf. Sci.*, 178 (2008), 1133–1140.

[40] J. Mihailović, A. Savić, J. Bogdanović-Pristov and K. Radotić, MRI brain tumors images by using independent component analysis, *2011 IEEE 9th International Symposium on Intelligent Systems and Informatics*, (2011), 433–435. doi: 10.1109/SISY.2011.6034366.

[41] T. Milfont, I. Mezzomo, B. Bedregal, E. Mansilla and H. Bustince, Aggregation functions on n-dimensional ordered vectors equipped with an admissible order and an application in multi-criteria group decision-making, *Int. J. Approx. Reason.*, 137 (2021), 34–50.

[42] N. Mustafa, S. A. Khan, J. P. Li, M. Khalil, K. Kumar and M. Giess, Medical image de-noising schemes using wavelet transform with fixed form thresholding, *2014 11th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, (2014), 397–402. doi: 10.1109/ICCWAMTIP.2014.7073435.

[43] D. Nolasco, F. Costa, E. Palmeira, D. Alves, B. Bedregal, T. Rocha, R. Ribeiro and J. Silva, Wavelet-fuzzy power quality diagnosis system with inference method based on overlap functions: Case study in an AC microgrid, *Eng. Appl. Artif. Intell.*, 85 (2019), 284–294.

[44] D. Paternain, A. Jurio, E. Barrenechea, H. Bustince, B. Bedregal and E. Szmidt, An alternative to fuzzy methods in decision-making problems, *Expert Syst. Appl.*, 39 (2012), 7729–7735.

[45] D. Paternain, J. Fernández, H. Bustince, R. Mesiar and G. Beliakov, Construction of image reduction operators using averaging aggregation functions, *Fuzzy Sets Syst.*, 261 (2015), 87–111.

[46] R. A. M. Pereira and G. Pasi, On non-monotonic aggregation: mixture operators, *Proc. 4th Meeting of the EURO Working Group on Fuzzy Sets (EUROFUSE'99) and 2nd Internat. Conf. on Soft and Intelingent Computing (SIC'99), Budapest, Hungary*, (1999).

[47] R. A. M. Pereira, The orness of mixture operators: the exponential case, *Proc. 8th Internat. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'2000), Madrid, Spain*, (2000).

[48] R. A. M. Pereira and R. A. Ribeiro, Aggregation with generalized mixture operators using weighting functions, *Fuzzy Sets Syst.*, 137 (2003), 43–58.

[49] R. A. Ribeiro and R. A. M. Pereira, Generalized mixture operators using weighting functions: A comparative study with WA and OWA, *Eur. J. Oper. Res.*, 145 (2003), 329–342.

[50] S. Sahu, N. Loya and A. G. Keskar, Restoration and enhancement of impulse noise image for human visual system, *2015 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, (2015), 1–6. doi: 10.1109/CONECCT.2015.7383913.

[51] B. Z. Shaick and L. Yaroslavsky, Image reduction for object recognition, *4th EURASIP-IEEE Region 8 International Symposium on Video/Image Processing and Multimedia Communications VIPromCom*, (2002), 333–338. doi: 10.1109/VIPROM.2002.1026678.

[52] A. J. Solanki, K. R. Jain and N. P. Desai, ISEF based identification of RCT/filling in dental caries of decayed tooth, *Int. J. Image Process.*, 7 (2013), 149–162.

[53] Z.-X. Su, G.-P. Xia, M.-Y. Chen and L. Wang, Induced generalized intuitionistic fuzzy OWA operator for multi-attribute group decision making, *Expert Syst. Appl.*, 39 (2012), 1902–1910.

[54] Suprijanto, Gianto, E. Juliastuti, Azhari and L. Epsilawati, Image contrast enhancement for film-based dental panoramic radiography, *2012 International Conference on System Engineering and Technology (ICSET)*, (2012), 1–5. doi:10.1109/ICSEngT.2012.6339321.

[55] P. Thevenaz, T. Blu and M. Unser, Interpolation revisited, *IEEE Trans. Medical Imaging*, 19 (2000), 739–758.

[56] T. Wilkin, Image reduction operators based on non-monotonic averaging functions, *2013 IEEE Int. Conf. on Fuzzy Systems, (FUZZ)*, (2013), 1–8. doi: 10.1109/FUZZ-IEEE.2013.6622458.

[57] T. Wilkin and G. Beliakov, Weakly monotonic averaging functions, *Int. J. Intell. Syst.*, 30(2) (2015), 144–169.

[58] S. K. Woo, K. M. Kim, T. S. Lee, J. H. Jung, J. G. Kim, J. S. Kim, T. H. Choi, G. I. An and G. J. Cheon, Registration method for the detection of tumors in lung and liver using multimodal small animal imaging, *IEEE Trans. Nucl. Sci.*, 56 (2009), 1454–1458.

[59] B. Y. Wu and X. Q. Sheng, A complex image reduction technique using genetic algorithm for the MoM solution of half-space MPIE, *IEEE Trans. Antennas Propag.*, 63 (2015), 3727–3731.

[60] R. R. Yager, On ordered weighted averaging aggregation operators in multicriteria decisionmaking, *IEEE Trans. Syst. Man Cybern.*, 18 (1988), 183–190.

[61] R. R. Yager, Centered OWA operators, *Soft Comput.*, 11 (2006), 631–639.

[62] J. Yang, J. Wright, T. Huang and Y. Ma, Image super-resolution as sparse representation of raw image patches, IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, (2008), 1–8. doi: 10.1109/CVPR.2008.4587647.

[63] J. Yang, J. Wright, T. S. Huang and Y. Ma, Image super-resolution via sparse representation, *IEEE Trans. Image Process.*, 19 (2010), 2861–2873.

[64] S.-M. Zhou, F. Chiclana, R. I. John and J. M. Garibaldi, Type-1 OWA operators for aggregating uncertain information with uncertain weights induced by type-2 linguistic quantifiers, *Fuzzy Sets Syst.*, 159 (2008), 3281–3296.

Antonio Diego Silva Farias
Department of Exact and Natural Sciences

Federal Rural University of Semi-Arid–UFERSA
Pau dos Ferros-RN, Brazil
E-mail: antonio.diego@ufersa.edu.br

Valdigleis da Silva Costa
Collegiate of Computer Science
Federal University of Vale do São Francisco–UNIVASF
Salgueiro-PE, Brazil
E-mail: valdigleis.costa@univasf.edu.br

Luiz Ranyer A. Lopes
Federal Institute of Rio Grande Norte–IFRN
Natal-RN, Brazil
E-mail: ranyer.lopes@gmail.com

Regivan Hugo Nunes Santiago
Department of Informatics and Applied Mathematics
Federal University of Rio Grande do Norte–UFRN
Natal-RN, Brazil
E-mail: regivan@dimap.ufrn.br

Benjamín Bedregal
Department of Informatics and Applied Mathematics
Federal University of Rio Grande do Norte–UFRN
Natal-RN, Brazil
E-mail: bedregal@dimap.ufrn.br

6 Appendix

Table 1: $PSNR$ values of reconstruction of imagens of Figure 3 by nearest neighbor interpolation. The underline value represents the second high quality image

USING 2 × 2 BLOCKS

	Min	Max	Med	$Arith$	cOWA	**H**
Img 01	26,68848	26,60371	30,66996	**30,89667**	30,73823	$\underline{30,75448}$
Img 02	33,50403	33,46846	37,51525	**37,64240**	37,57713	$\underline{37,58138}$
Img 03	26,80034	26,74460	30,47904	**30,55504**	$\underline{30,52128}$	30,51564
Img 04	28,90415	28,83284	32,88120	**33,01225**	$\underline{32,94828}$	32,94146
Img 05	25,04896	25,04438	28,75582	**28,85475**	$\underline{28,81506}$	28,79901
Img 06	38,10156	38,07248	42,08612	**42,13003**	$\underline{42,12316}$	42,11653
Img 07	24,48520	24,38872	28,31229	**28,45667**	28,35114	$\underline{28,37668}$
Img 08	23,69576	23,73464	27,41557	**27,51579**	$\underline{27,46383}$	27,45864
Img 09	26,19262	26,09448	30,06427	**30,22940**	30,11893	$\underline{30,13332}$
Img 10	21,48459	21,41350	25,37475	**25,58054**	25,43016	$\underline{25,45073}$
Avg	27,49057	27,43978	31,35543	**31,48735**	31,40872	$\underline{31,41279}$

Table 2: $PSNR$ values of reconstruction of imagens of Figure 1 by nearest neighbor interpolation. The underline value represents the second high quality image

USING 4 × 4 BLOCKS

	Min	Max	Med	$Arith$	cOWA	**H**
Img 01	21,37117	20,83960	26,73708	**27,07854**	27,01270	$\underline{27,07067}$
Img 02	19,70858	19,54290	23,92198	**24,07786**	24,05762	$\underline{24,07478}$
Img 03	20,46198	20,82576	25,64113	**26,16092**	26,08186	$\underline{26,14607}$
Img 04	22,59335	22,24354	27,94347	**28,26449**	28,19574	$\underline{28,25700}$
Img 05	18,86628	19,55278	24,12507	**24,68962**	24,58713	$\underline{24,67322}$
Img 06	29,48308	29,26559	34,89670	**35,11481**	35,09436	$\underline{35,11023}$
Img 07	18,95771	18,72670	24,18918	**24,55073**	24,48373	$\underline{24,54269}$
Img 08	17,71071	18,59348	23,11305	**23,54332**	23,43522	$\underline{23,53119}$
Img 09	20,97846	20,44416	26,23824	**26,53197**	26,42064	$\underline{26,52562}$
Img 10	16,47636	16,22205	21,89755	**22,22614**	22,10356	$\underline{22,21825}$
Avg	20,66077	20,62565	25,87034	**26,22384**	26,14726	$\underline{26,21497}$

Table 3: $PSNR$ values of reconstruction of imagens of Figure 3 by bilinear interpolation. The underline value represents the second high quality image

USING 2×2 BLOCKS

	Min	Max	Med	$Arith$	cOWA	**H**
Img 01	27,25658	27,41249	31, 70137	31,66148	31,64818	**31,70944**
Img 02	29,07393	29,09065	29,98667	**30,00618**	29, 99790	29,99295
Img 03	28,07377	27,53953	**31,96271**	31,87901	31,87085	31, 94673
Img 04	29,70934	29,78913	**34,39128**	34,28215	34,31414	34, 37504
Img 05	26,30684	25,74955	**30,17965**	30,08193	30,05530	30, 16533
Img 06	40,09734	39,94107	**48,99047**	48,55730	48,52986	48, 86710
Img 07	25,10689	25,04408	28, 93328	28,92340	28,89276	**28,94254**
Img 08	24,63619	24,10410	28, 19100	28,17758	28,16818	**28,19312**
Img 09	26,60297	26,71398	30,54028	**30,56126**	30,52693	30, 55733
Img 10	21,93973	21,90280	25,71329	**25,74295**	25,69402	25, 73353
Avg	27,88036	27,72874	**32,05900**	31,98732	31,96981	32, 04831

Table 4: $PSNR$ values of reconstruction of imagens of Figure 3 by bilinear interpolation. The underline value represents the second high quality image

USING 4×4 BLOCKS

	Min	Max	Med	$Arith$	cOWA	**H**
Img 01	21,84394	21,46624	28, 12885	28,03911	**28,13262**	28,08806
Img 02	20,22210	19,99324	24,09349	24,09114	24, 09696	**24,10058**
Img 03	21,36383	21,65788	27,34577	27,53279	**27,57114**	27, 56163
Img 04	23,23057	22,96007	**29,81717**	29,65596	29, 77096	29,71475
Img 05	19,54307	20,06159	25,32192	25,47922	25, 51400	**25,51442**
Img 06	30,92215	30,60188	**42,72668**	41,77064	41, 99358	41,97442
Img 07	19,43662	19,19604	24,96897	25,00413	**25,05911**	25, 02899
Img 08	18,28578	18,86696	23,87169	24, 09781	24,07356	**24,10310**
Img 09	21,32747	20,91360	27,09762	27,10526	**27,16280**	27, 13073
Img 10	16,77848	16,57833	22,58040	22,61488	22, 63949	**22,63987**
Avg	21,29540	21,22958	27, 59525	27,53909	**27,60142**	27,58566

Table 5: $PSNR$ values of reconstruction of imagens of Figure 3 by bicubic interpolation. The underline value represents the second high quality image

USING 2×2 BLOCKS

	Min	Max	Med	$Arith$	cOWA	**H**
Img 01	27,39667	27,45993	32,53367	**32,62657**	32,52946	32,58602
Img 02	30,06149	30,00816	31,28820	**31,31873**	31,30611	31,29877
Img 03	28,09952	27,62931	32,92967	32,90897	32,87767	**32,93859**
Img 04	29,92114	29,94430	35,70586	35,70361	35,68906	**35,73313**
Img 05	26,38597	25,93655	31,32017	31,30790	31,25508	**31,33640**
Img 06	40,05229	40,02173	**51,35284**	51,07478	51,01447	51,31081
Img 07	25,23188	25,16984	29,85564	**29,93609**	29,85733	29,89915
Img 08	24,72669	24,32047	29,10402	**29,15066**	29,11737	29,12822
Img 09	26,73252	26,79140	31,27454	**31,38274**	31,29368	31,32452
Img 10	22,04218	21,98136	26,39147	**26,52171**	26,41585	26,44659
Avg	28,06504	27,92630	33,17561	33,19318	33,13561	**33,20022**

Table 6: $PSNR$ values of reconstruction of imagens of Figure 3 by bicubic interpolation. The underline value represents the second high quality image

USING 4×4 BLOCKS

	Min	Max	Med	$Arith$	cOWA	**H**
Img 01	21,83423	21,39364	28,64265	28,74908	**28,80893**	28,78768
Img 02	20,20038	19,88701	24,49596	24,56989	24,56761	**24,57359**
Img 03	21,25132	21,55589	27,82091	28,31402	28,28961	**28,32229**
Img 04	23,22310	22,89860	30,47704	30,54773	**30,60332**	30,59348
Img 05	19,45423	20,06391	25,74518	26,18606	26,15139	**26,20092**
Img 06	30,81953	30,48357	**44,31891**	43,83439	44,03526	44,05492
Img 07	19,36949	19,11221	25,29211	25,49221	25,49999	**25,50641**
Img 08	18,21007	18,91559	24,17857	**24,57330**	24,49174	24,56575
Img 09	21,32252	20,85345	27,41366	27,56839	27,55860	**27,58354**
Img 10	16,76501	16,53815	22,82004	23,00025	22,96201	**23,01459**
Avg	21,24499	21,17020	28,12050	28,28353	28,29685	**28,32032**