

Linguistic Weighted Power Means

Comparison with the Linguistic Weighted Average

John T. Rickard

Distributed Infinity, Inc.
Larkspur, CO, USA
terry.rickard@gmail.com

Janet Aisbett

School of Science and IT
University of Newcastle
Newcastle, NSW, Australia
janet.aisbett@newcastle.edu.au

Ronald R. Yager

Machine Intelligence
Institute, Iona College
New Rochelle, NY, USA
yager@panix.com

Greg Gibbon

School of Science and IT
University of Newcastle
Newcastle, NSW, Australia
greg.gibbon@newcastle.edu.au

Abstract – We have introduced the linguistic weighted power mean and its computation via a generalization of the Karnick-Mendel algorithm. This new family of aggregation operators admits interval type-2 fuzzy membership functions for both its inputs and its weights to account for imprecise knowledge of these quantities. The simplest member of this family, the linguistic weighted average, is the primary instantiation of a “perceptual computing” engine in Mendel and Wu’s recent book by this title. This paper presents a comparison of the linguistic weighted power mean and the linguistic weighted average on one of the examples they describe in detail—the “investment judgment advisor”. Our results illustrate the flexibility and range of logical inference provided by this very versatile aggregation operator for computing with words applications.

Keywords—aggregation operators; type-2 fuzzy logic; computing with words; perceptual computing

I. INTRODUCTION

The recent book by Mendel and Wu entitled *Perceptual Computing* [1] presents an approach to “computing with words” (CWW), a problem area originally suggested by Zadeh [2] that remains of high interest. (See, for example, the recent special section of the *IEEE Transactions on Fuzzy Systems* on CWW [3] and publications [4]–[7].)

Mendel and Wu’s approach to CWW is motivated by the premise that “words mean different things to different people,” and therefore the inputs, computational engine and outputs of a perceptual computer (Per-C) must explicitly take account of this inherent imprecision. Inputs to the Per-C are modeled as interval type-2 (IT2) fuzzy membership functions (MFs) spanning an appropriate range (e.g., a 0-10 scale), which is perhaps the simplest method for representing imprecise knowledge of membership values.

The Per-C is proposed to have three components: an *encoder*, a *computational engine* and a *decoder*. Mendel and Wu’s strategy for the encoder is to develop a code book of input and output *words* that are also represented by IT2 MFs. Inputs to the Per-C computational engine are selected from this vocabulary of input words. The IT2 fuzzy input variables and importance weights are mapped to code book words via similarity comparisons. The computational engine logically aggregates the imprecision represented in these word MFs into corresponding output IT2 MFs. These output IT2 MFs are

decoded by an analogous similarity mapping into corresponding word outputs.

Two architectures for the Per-C computational engine are proposed in [1]: 1) a bank of parallel *if-then rules*, and 2) a “*linguistic weighted average*” (LWA). The first of these relies upon results summarized in Mendel’s earlier book [8] on type-2 fuzzy rule-bases systems, where input variables are represented by IT2 MFs and the implication operation performed by the rules results in an output IT2 MF. In fuzzy control applications, this output MF is generally type-reduced to an interval whose midpoint describes the defuzzified scalar output of the computational engine. In CWW applications, the output IT2 MF is typically mapped into an output word in the code book vocabulary using the decoder described above.

The LWA Per-C architecture is employed in three of the four chapters in [1] presenting detailed examples of Per-C applications. It is a generalization of the traditional weighted average, first to a type-1 *fuzzy weighted average* (FWA) and then to a IT2 fuzzy weighted average in which all inputs and importance weights are represented by IT2 MFs. The latter generalization applies the celebrated Karnick-Mendel algorithm [9] for computing the centroid of an IT2 MF to the α -cuts of the upper and lower MFs (UMF and LMF, respectively) of the input variables and weights, resulting in the corresponding α -cuts of the LWA UMF and LMF. The output UMF and LMF are then constructed (in approximation) using the well-known α -cut Representation Theorem [10].

The LWA is perhaps the simplest aggregation operator $L(\tilde{\mathbf{x}}, \tilde{\mathbf{w}})$ mapping IT2 MFs of scores $\tilde{\mathbf{x}} = (\tilde{x}_1, \dots, \tilde{x}_n)$ and associated importance weights $\tilde{\mathbf{w}} = (\tilde{w}_1, \dots, \tilde{w}_n)$ into an output IT2 MF. These operators represent a form of data fusion in which both scores and weights are generally taken to lie within a common interval range and are characterized by IT2 MFs. Although the LWA is the primary computational method featured for computing with words in [1], it is limited in its ability to represent how humans aggregate preference scores. For example, it cannot model a mandatory requirement. Therefore it is highly desirable to develop more powerful and computationally feasible aggregation operators that can be applied to IT2 inputs.

The LWA is but one member of the class of aggregation operators generally referred to as the p-norm [11]. When the power r in the p-norm is allowed to vary over the entire real line, it is called a weighted power mean (WPM), defined by

$$L_r(\tilde{\mathbf{x}}, \tilde{\mathbf{w}}) \triangleq \lim_{q \rightarrow r} \left(\frac{\sum_{i=1}^n \tilde{w}_i \tilde{x}_i^q}{\sum_{i=1}^n \tilde{w}_i} \right)^{1/q}. \quad (1)$$

While it is no longer always a norm, the WPM is a flexible operator that is used in information retrieval and other ranked decision applications [12]. As r ranges over the real line, L_r ranges from logical conjunction $x_1 \wedge \dots \wedge x_n$ of the inputs (in the limit as $r \rightarrow -\infty$) to logical disjunction $x_1 \vee \dots \vee x_n$ (in the limit as $r \rightarrow \infty$). L_r is thus an *orand* operator. Other special cases of the weighted power mean include the LWA ($r = 1$), geometric mean ($r = 0$), harmonic mean ($r = -1$) and Euclidean mean ($r = 2$). Dujmovi [13] suggests (in a type-1 setting) that the WPM performs best of a general class of parameterised aggregation operators that range continuously between the extremes of conjunction and disjunction.

In a current paper [14], we have generalized the Karnick-Mendel algorithm to compute the output interval of the WPM given input intervals for $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{w}}$, for arbitrary values of the power r . We denote this as the Weighted Power Mean Enhanced Karnick-Mendel (WPMEKM) algorithm, since it makes use of performance enhancements over the original Karnick-Mendel algorithm as described in [15]-[16]. By employing the α -cut Representation Theorem, we use the WPMEKM algorithm to compute the output IT2 MF of $L_r(\tilde{\mathbf{x}}, \tilde{\mathbf{w}})$, given IT2 MFs for the inputs and weights. We denote this new aggregation operator as the *linguistic weighted power mean* (LWPM).

The primary contribution of this paper is to provide a comparison of Per-C computational engines using the LWPM aggregation operator $L_r(\tilde{\mathbf{x}}, \tilde{\mathbf{w}})$ with the LWA Per-C engine described by Mendel and Wu (which is $L_1(\tilde{\mathbf{x}}, \tilde{\mathbf{w}})$), using the detailed example of an Investment Judgment Advisor presented in [1] and described below. This example illustrates the flexibility and range of logical inferences available using the LWPM as compared to the LWA. Thus the LWPM represents a significant generalization and extension of the choices of Per-C computational engines available for CWW applications.

The remainder of this paper is organized as follows: first, we summarize the WPMEKM algorithm of [14]. Then we review the Investment Judgment Advisor (IJA) example of [1]. Next, we compare the IJA outputs using the LWPM versus the LWA, and discuss the advantages of the LWPM. Finally, we describe how even more useful logical aggregation operators can be constructed hierarchically using combinations of LWPM components.

II. SUMMARY OF WPMEKM ALGORITHM

Suppose initially that we have interval-valued scores $[\underline{x}_i, \bar{x}_i]$ and weights $[\underline{w}_i, \bar{w}_i]$, where $\bar{x}_i, \underline{x}_i \in [0, 1]$, $\bar{w}_i, \underline{w}_i \geq 0$, $i \in \{1, \dots, n\}$ are inputs to the LWPM $L_r(\tilde{\mathbf{x}}, \tilde{\mathbf{w}})$. We summarize below the WPMEKM algorithm for computing corresponding output intervals of the LWPM [14].

First we consider the exceptional cases $r = \pm\infty$, for which $L_r(\mathbf{x}, \mathbf{w})$ is independent of w_i . The interval endpoints then are determined solely by the x_i interval boundaries, with

$$\begin{aligned} \underline{L}_{-\infty}(\mathbf{x}, \mathbf{w}) &= \min_{i: w_i > 0} \{\underline{x}_i\}, \quad \bar{L}_{-\infty} = \min_{i: w_i > 0} \{\bar{x}_i\} \\ \underline{L}_{\infty}(\mathbf{x}, \mathbf{w}) &= \max_{i: w_i > 0} \{\underline{x}_i\}, \quad \bar{L}_{\infty} = \max_{i: w_i > 0} \{\bar{x}_i\}. \end{aligned} \quad (2)$$

Next, consider the weighted power mean for a finite power r . Then (1) can be rewritten as

$$L_r(\mathbf{x}, \mathbf{w}) = h_r \left(\sum_{i=1}^n w_i h_r^{-1}(x_i) / \sum_{i=1}^n w_i \right) = h_r(y_r(\mathbf{x}, \mathbf{w})) \quad (3)$$

where

$$h_r(z) = \begin{cases} z^{1/r}, & r \neq 0 \\ \exp(z), & r = 0 \end{cases}, \quad (4)$$

$$y_r(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^n w_i h_r^{-1}(x_i) / \sum_{i=1}^n w_i. \quad (5)$$

Note that h_r and h_r^{-1} are strictly increasing on the non-negative reals if $r \geq 0$ and are strictly decreasing if $r < 0$.

With (3), the value of the weighted power mean applied to intervals $[\underline{x}_i, \bar{x}_i]$, $[\underline{w}_i, \bar{w}_i]$ can be written as the interval

$$\begin{aligned} [\underline{L}_r, \bar{L}_r] &= \\ & \left[\min h_r \left(\sum_{i=1}^n w_i h_r^{-1}(x_i) / \sum_{i=1}^n w_i \right), \max h_r \left(\sum_{i=1}^n w_i h_r^{-1}(x_i) / \sum_{i=1}^n w_i \right) \right] \end{aligned} \quad (6)$$

where $\min$ and $\max$ are taken over $\{x_1, \dots, x_n, w_1, \dots, w_n : x_i \in [\underline{x}_i, \bar{x}_i], w_i \in [\underline{w}_i, \bar{w}_i], i = 1, \dots, n\}$.

From (6), we have

$$[\underline{L}_r, \bar{L}_r] = \begin{cases} [h_r(y_r), h_r(\bar{y}_r)] & r \geq 0 \\ [h_r(\bar{y}_r), h_r(y_r)] & r < 0 \end{cases}, \quad (7)$$

where

$$[\underline{y}_r, \bar{y}_r] = \left[\min \left\{ \frac{\sum_{i=1}^n w_i h_r^{-1}(x_i)}{\sum_{i=1}^n w_i} \right\}, \max \left\{ \frac{\sum_{i=1}^n w_i h_r^{-1}(x_i)}{\sum_{i=1}^n w_i} \right\} \right]. \quad (8)$$

Now for any $i \in \{1, \dots, n\}$, $h_r^{-1}(\underline{x}_i) \leq h_r^{-1}(\bar{x}_i)$ if h_r is strictly increasing and $h_r^{-1}(\underline{x}_i) \geq h_r^{-1}(\bar{x}_i)$ if h_r is strictly decreasing. So to compute (8), we compute the LWA of input intervals $[h_r^{-1}(\underline{x}_i), h_r^{-1}(\bar{x}_i)]$ when $r \geq 0$ and $[h_r^{-1}(\bar{x}_i), h_r^{-1}(\underline{x}_i)]$ when $r < 0$. Then we apply h_r to the resulting interval LWA, where the upper and lower bounds are transposed at this step if h_r is strictly decreasing. Thus in all cases, the bounds in (7) are given by:

$$\begin{aligned} \underline{L}_r &= h_r \left(\frac{\sum_{i=1}^n \underline{z}_i h_r^{-1}(\underline{x}_i)}{\sum_{i=1}^n \underline{z}_i} \right), \\ \underline{z}_i &= \begin{cases} \bar{w}_i, & \underline{x}_i < h_r(y_r(\underline{\mathbf{x}}, \underline{\mathbf{z}})) = \underline{L}_r \\ \underline{w}_i, & \underline{x}_i > h_r(y_r(\underline{\mathbf{x}}, \underline{\mathbf{z}})) = \underline{L}_r \end{cases} \\ \bar{L}_r &= h_r \left(\frac{\sum_{i=1}^n \bar{z}_i h_r^{-1}(\bar{x}_i)}{\sum_{i=1}^n \bar{z}_i} \right), \\ \bar{z}_i &= \begin{cases} \underline{w}_i, & \bar{x}_i < h_r(y_r(\bar{\mathbf{x}}, \bar{\mathbf{z}})) = \bar{L}_r \\ \bar{w}_i, & \bar{x}_i > h_r(y_r(\bar{\mathbf{x}}, \bar{\mathbf{z}})) = \bar{L}_r \end{cases} \end{aligned} \quad (9)$$

The EKM algorithms [15]-[16] can be adapted to compute

$$[\underline{y}, \bar{y}] = \left[\min h \left(\frac{\sum_{i=1}^n w_i h^{-1}(x_i)}{\sum_{i=1}^n w_i} \right), \max h \left(\frac{\sum_{i=1}^n w_i h^{-1}(x_i)}{\sum_{i=1}^n w_i} \right) \right] \text{ for}$$

any strictly monotonic function h , which in the present case we take to be h_r . We call the resulting algorithm WPMEKM for the special case of weighted power means, while recognizing the more general application.

WPMEKM Algorithm

- 1) If the minimum $L_r(\mathbf{x}, \mathbf{w}) = \underline{L}_r$ is to be computed, set $k = \left\lceil \frac{n}{2.4} \right\rceil$ where $\lceil x \rceil$ is the nearest integer to x . Set $x_i = \underline{x}_i$, $u_i = \underline{w}_i$ and $v_i = \bar{w}_i$ for each $i = 1, \dots, n$. If the maximum $L_r(\mathbf{x}, \mathbf{w}) = \bar{L}_r$ is to be computed, set $k = \left\lceil \frac{n}{1.7} \right\rceil$, $x_i = \bar{x}_i$, $u_i = \bar{w}_i$ and $v_i = \underline{w}_i$ for $i = 1, \dots, n$.
- 2) Sort the x_i for $i = 1, \dots, n$ in ascending order so that $x_{i_1} \leq x_{i_2} \leq \dots \leq x_{i_n}$ where $i_j, j = 1, \dots, n$ represents the

original index values. Rearrange the weight interval endpoints u_i and v_i using the same permutation of index values $i_j, j = 1, \dots, n$ that result from the x_i sort. Now refer to $x_i, u_i, v_i, i = 1, \dots, n$ as the sorted sets of inputs and weights.

3) Compute

$$a = \sum_{i=1}^k h^{-1}(x_i) v_i + \sum_{i=k+1}^n h^{-1}(x_i) u_i, \quad (10)$$

$$b = \sum_{i=1}^k v_i + \sum_{i=k+1}^n u_i, \quad y = \left(\frac{a}{b} \right). \quad (11)$$

4) Find $k' \in \{0, 1, \dots, n\}$ such that $x_{k'} \leq h(y) < x_{k'+1}$. Check if $k' = k$. If yes, set $L_r(\mathbf{x}, \mathbf{w}) = h(y)$, stop. If no, continue.

5) Set $s = \text{sign}(k' - k)$, and compute

$$a' = a + s \sum_{i=\min(k, k')+1}^{i=\max(k, k')} h^{-1}(x_i) (v_i - u_i), \quad (12)$$

$$b' = b + s \sum_{i=\min(k, k')+1}^{i=\max(k, k')} (v_i - u_i), \quad y = \left(\frac{a'}{b'} \right). \quad (13)$$

6) Set $a = a'$, $b = b'$ and $k = k'$. Go to step 4).

Note that because h is strictly monotonic, the sort on x_i results in either an ascending or descending sort on $h^{-1}(x_i)$ depending on whether h is increasing or decreasing.

With the WPMEKM algorithm, the weighted power mean interval boundaries can be computed for any value of r . These interval-valued results enable us directly to compute arbitrarily accurate approximations to the weighted power mean membership function for scores and weights that are fuzzy sets, using the α -cut representation theorem [10]. In this case the interval computation is applied to α -cuts of the input variables to determine the corresponding α -cuts of the output variable. In the IT2 case, this computation is applied to both the upper and lower bounding functions for the scores and weights to determine the corresponding upper and lower bounding functions of the global score.

III. INVESTMENT JUDGMENT ADVISOR REVIEW

The IJA example in [1] originated with Tong and Bonissone [17], who considered the decision confronting an individual with a moderately large amount of capital who wishes to select among five investment sector alternatives, using four investment criteria. The five alternatives are commodities (a_1), stocks (a_2), gold (a_3), real estate (a_4) and long-term bonds (a_5). The four investment criteria are:

risk of losing capital (c_1), inflation vulnerability (c_2), anticipated profit (c_3) and liquidity (c_4). The investor's goal is to rate each investment with respect to the four criteria and then aggregate these ratings into a decision statistic that can be used for ranking (or for capital allocation among the different alternatives to achieve an appropriately diversified portfolio.)

Through a process of interval analysis [18] using inputs from real subjects, Mendel and Wu derive three descriptive vocabularies of eight, seven and four words that are represented using IT2 MFs. The UMF and LMF of each vocabulary word MF is a trapezoid disposed on the domain interval $[0,10]$, with the extreme words in each vocabulary typically represented by left- or right-shoulder MFs, while the intermediate words are represented by mid-range MFs. (For details, see [1], Tables 7.13, 7.14 and 7.15.) The eight-word vocabulary expresses *amount*, characterized by None to Very Little (NVL), Very Low (VL), Low (L), More or Less Low (MLL), Fair to More or Less High (FMLH), More or Less High (MLH), High (H) and Extremely High (EH). This vocabulary is used to grade each alternative with respect to the first three criteria, where in keeping with [1] a simple *antonym* $\mu_{10-\bar{A}}(x)$ of the respective IT2 MF is used for criteria 1 and 2 in the aggregation, i.e., $\mu_{10-\bar{A}}(x) = \mu_{\bar{A}}(10-x)$. Alternative forms for the membership functions antonyms which exhibit more intuitive behaviour are discussed in [19].

The seven-word vocabulary expresses *quality*, characterized by Very Bad (VB), Bad (B), Somewhat Bad (SB), Fair (F), Somewhat Good (SG), Good (G) and Very Good (VG). This vocabulary is used in grading the alternatives with respect to criterion 4. The four-word vocabulary expresses *importance*, characterized by Unimportant (U), More or Less Unimportant (MLU), More or Less Important (MLI) and Very Important (VI). Table 1 (replicated from [1]) presents three hypothetical word evaluation matrices grading each investment alternative from the perspectives of speculative, conservative and in-between investors, respectively.

TABLE 1 LINGUISTIC RATINGS AND WEIGHTS OF INVESTMENT ALTERNATIVES WRT CRITERIA

(a) Speculative Investor

Ratings	c_1	c_2	c_3	c_4
a_1	FMLH	FMLH	EH	G
a_2	MLH	FMLH	H	SG
a_3	VL	L	VL	SB
a_4	MLL	MLH	MLH	SG
a_5	NVL	MLL	VL	VB
Weight	U	MLU	VI	MLU

(b) Conservative Investor

Ratings	c_1	c_2	c_3	c_4
a_1	EH	MLH	MLL	F
a_2	H	H	MLH	SG
a_3	MLL	FMLH	FMLH	F
a_4	MLH	MLH	H	F
a_5	NVL	MLL	MLH	SG
Weight	VI	MLI	VI	MLI

(c) In-between Investor

Ratings	c_1	c_2	c_3	c_4
a_1	H	H	MLH	SB
a_2	FMLH	FMLH	H	VG
a_3	L	MLL	MLL	SB
a_4	FMLH	FMLH	H	SG
a_5	L	L	L	F
Weight	MLI	MLI	VI	MLI

Mendel and Wu use the LWA to aggregate the criteria evaluations and importance evaluations for each alternative into corresponding output IT2 MFs, which are then ranked to determine the order of appropriateness of the investment categories for each of the three types of investors. In the next section we generalize these results using the LWPM with different power exponents on the same input data, which will illustrate the versatility of this aggregation operator as a function of the power exponent.

IV. COMPARISON OF AGGREGATION OPERATORS

When a parameterized aggregation operator such as the LWPM ranges between conjunction and disjunction, we define its similarity to conjunction as the degree of *andness* and its similarity to disjunction as the degree of *orness*, where andness is the complement to orness. The weighted arithmetic mean is equi-similar to conjunction and disjunction, with both andness and orness equal to 1/2. Andness of an aggregator L for which $x_1 \wedge \cdots \wedge x_n \leq L(\mathbf{x}, \mathbf{w}) \leq x_1 \vee \cdots \vee x_n$ everywhere is defined in [13] to be:

$$\alpha_g = \frac{\overline{x_1 \vee \cdots \vee x_n} - \overline{L(\mathbf{x}, \mathbf{w})}}{\overline{x_1 \vee \cdots \vee x_n} - \overline{x_1 \wedge \cdots \wedge x_n}} = \frac{n - (n+1) \left(\overline{L(\mathbf{x}, \mathbf{w})} \right)}{n-1}, \quad (14)$$

where the overbar denotes averaging over the hypercube $[0,1]^n$ in which $\mathbf{x}$ lies. An aggregator L that is more similar to conjunction than disjunction is referred to as *partial conjunction*, and if vice versa, as *partial disjunction* [13]. Properties are partially substitutable if L is a partial

disjunction, and must be partially simultaneous if L is a partial conjunction.

When L is the weighted power mean with fixed weights and input scores, $L_r(\mathbf{x}, \mathbf{w})$ is a continuous and increasing function of r . Therefore from (14), the andness $\alpha_g^{(r)}$ of $L_r(\mathbf{x}, \mathbf{w})$ is a monotonically decreasing function of r . When $-\infty < r < 1$, $L_r(\mathbf{x}, \mathbf{w})$ represents partial conjunction (i.e., $\alpha_g^{(r)} > 0.5$) whereas for $1 < r < +\infty$, $L_r(\mathbf{x}, \mathbf{w})$ represents partial disjunction. For $r \leq 0$, $L_r(\mathbf{x}, \mathbf{w}) = 0$ if $x_i = 0$ for any $i \in \{1, \dots, n\}$. Therefore $r = 0$ is the upper bound on r when all inputs are mandatory.

When the inputs are represented by IT2 MFs, the notions of substitutable and mandatory are no longer crisp, but the qualitative aspects of partial disjunction/conjunction are still preserved. To illustrate the range of behaviors for the LWPM as compared with the LWA, we plot in the following sequences of figures the output IT2 MF footprint of uncertainty (FOU) of the LWPM for different values of the power exponent r . In each figure we also plot the FOU of the LWA where $r = 1$, with andness $\alpha_g^{(1)} = 0.5$. The “+” crosshatch corresponds to the LWA MF, while the “x” crosshatch corresponds to the LWPM MF; the UMF is shown as a solid line, while the LMF is shown as a dashed line. The solid vertical line indicates the defuzzified centroid of the LWPM IT2 MF, and the two dashed vertical lines indicate the left and right endpoints of the LWPM centroid interval. Note that our purpose in the choices of alternatives shown below is to illustrate the cases where the LWPM is qualitatively the *most* similar and the *least* similar to the LWA results for a given value of the parameter r .

Consider first the extreme cases of $r = \pm\infty$, where the andness of the LWPM is 1 ($r = -\infty$) or 0 ($r = +\infty$). Figs. 1-6 are for $r = +\infty$, with andness $\alpha_g^{(\infty)} = 0$ (pure disjunction), where the LWPM MF lies generally to the right of the LWA MF. Odd numbered figures correspond to the alternatives for which the LWPM MF FOU is most similar qualitatively to that of the LWA, and even numbered figures correspond to alternatives for which it is most dissimilar to the LWA.

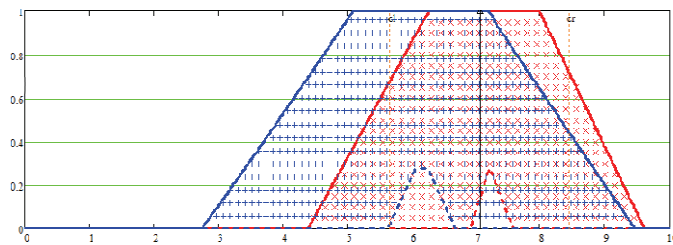


Figure 1 Speculative investor FOUs for a_4 , $r = +\infty$, $\alpha_g^{(\infty)} = 0$

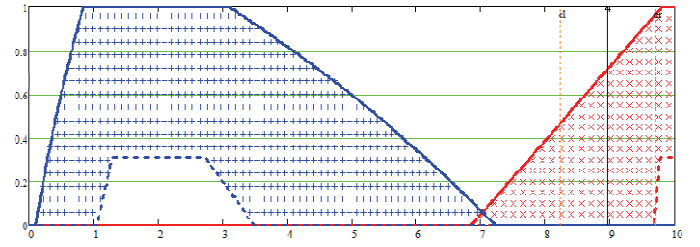


Figure 2 Speculative investor FOUs for a_5 , $r = +\infty$, $\alpha_g^{(\infty)} = 0$

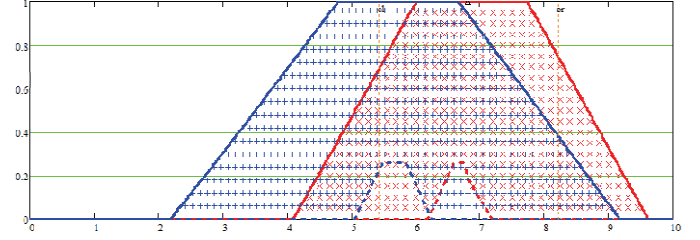


Figure 3 Conservative investor FOUs for a_3 , $r = +\infty$, $\alpha_g^{(\infty)} = 0$

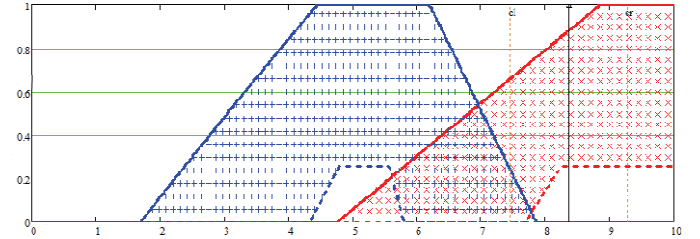


Figure 4 Conservative investor FOUs for a_4 , $r = +\infty$, $\alpha_g^{(\infty)} = 0$

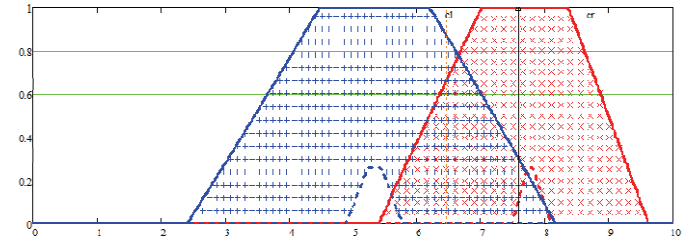


Figure 5 In-between investor FOUs for a_5 , $r = +\infty$, $\alpha_g^{(\infty)} = 0$

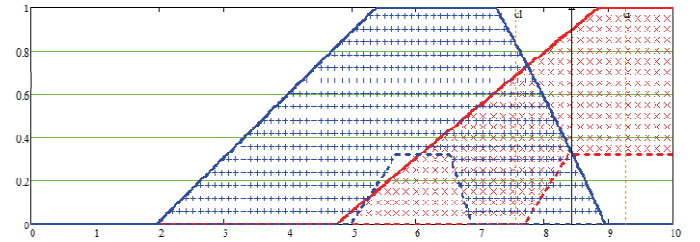


Figure 6 In-between investor FOUs for a_4 , $r = +\infty$, $\alpha_g^{(\infty)} = 0$

We observe a dramatic range of differences between the LWPM and LWA over different alternatives, which is a function of the *dispersion* of the evaluation words' IT2 MFs over the 0-10 scale with respect to the various criteria. The LWA smooths these differences, resulting in an interior

trapezoidal IT2 MF in all cases, while the LWPM for $r = +\infty$ often results in a right-shoulder trapezoidal IT2 MF.

Figs. 7-12 show analogous results for the case $r = -\infty$ (pure conjunction), which corresponds to andness $\alpha_g^{(-\infty)} = 1$.

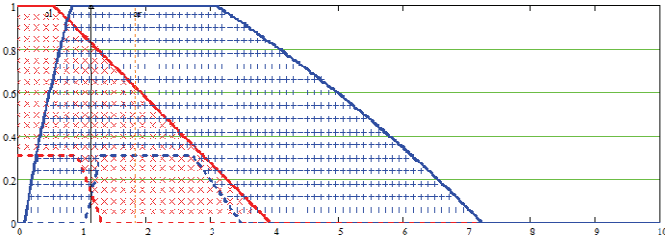


Figure 7 Speculative investor FOUs for a_5 , $r = -\infty$, $\alpha_g^{(-\infty)} = 1$

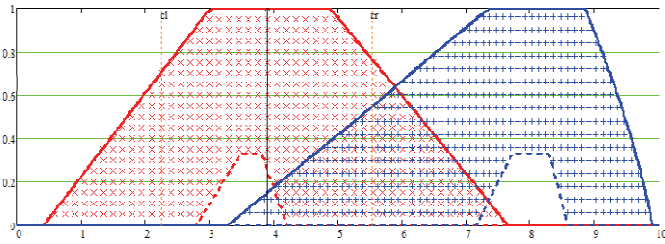


Figure 8 Speculative investor FOUs for a_1 , $r = -\infty$, $\alpha_g^{(-\infty)} = 1$

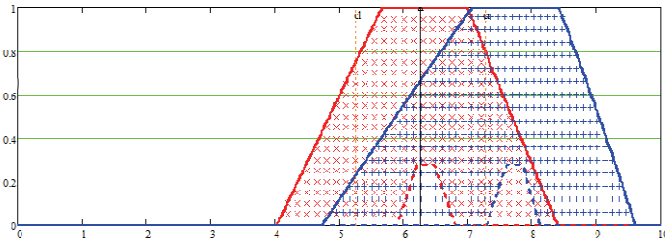


Figure 9 Conservative investor FOUs for a_5 , $r = -\infty$, $\alpha_g^{(-\infty)} = 1$

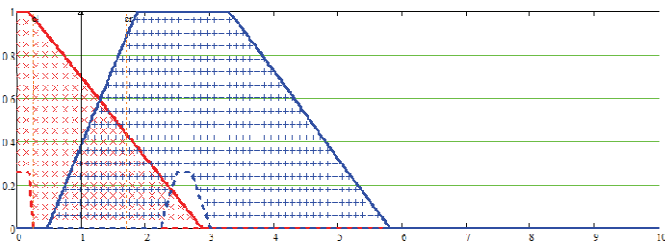


Figure 10 Conservative investor FOUs for a_1 , $r = -\infty$, $\alpha_g^{(-\infty)} = 1$

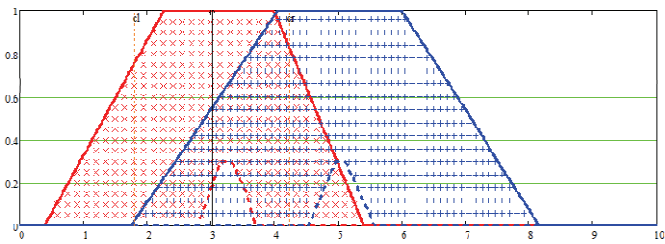


Figure 11 In-between investor FOUs for a_3 , $r = -\infty$, $\alpha_g^{(-\infty)} = 1$

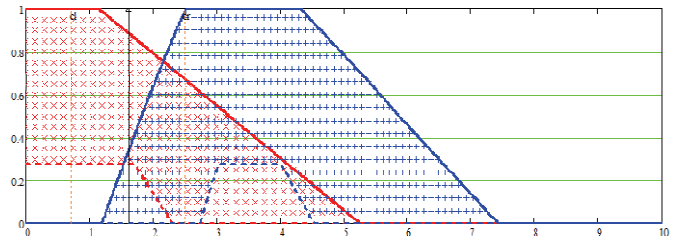


Figure 12 In-between investor FOUs for a_1 , $r = -\infty$, $\alpha_g^{(-\infty)} = 1$

Here, the LWPM MF lies generally to the left of the LWA MF, since the LWPM for $r = -\infty$ selects the *minimum* left and right endpoints over the input MFs for each α -cut. Again we observe a substantial range of differences between the LWPM and LWA IT2 MFs, with the LWPM often transitioning to a left-shoulder trapezoidal IT2 MF.

These examples serve to illustrate the substantially greater aggregation versatility of the LWPM over the LWA. By the choice of r , we impart different perspectives to the fuzzy IT2 decision process, ranging from the most optimistic view having an orness of 1, to the most pessimistic view having an andness of 1, with a continuous range of andness/orness between these extremes. This compares with the single value of andness = orness = 0.5 available with the LWA, which averages out the more extreme inputs and thus might be termed a “consensus” view.

Figs. 13-18 show the analogous results for the case $r = -1$ (i.e., the harmonic mean), for which $\alpha_g^{(-1)} = 0.773$. Here, we observe considerably more overlap between the LWPM and LWA MFs, since the power exponents of the two aggregation operators are closer in value. However, even in this case, we observe instances (i.e., Figs. 16 and 18) where the LWPM transitions into a left-shoulder trapezoidal IT2 MF, whereas the LWA has a midrange trapezoidal IT2 MF in all cases.

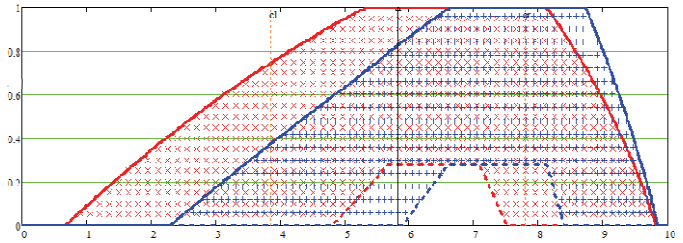


Figure 13 Speculative investor FOUs for a_2 , $r = -1$, $\alpha_g^{(-1)} = 0.773$

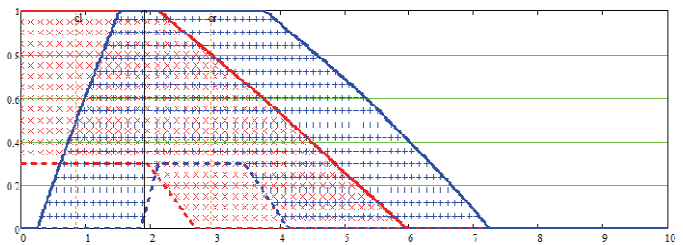


Figure 14 Speculative investor FOUs for a_3 , $r = -1$, $\alpha_g^{(-1)} = 0.773$

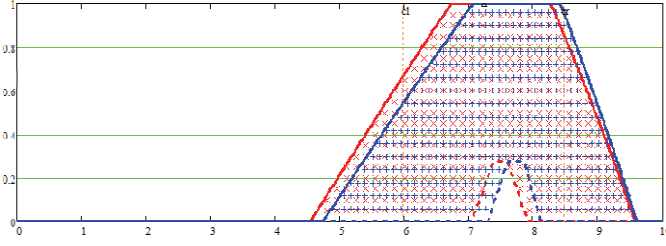


Figure 15 Conservative investor FOUs for a_5 , $r = -1$, $\alpha_g^{(-1)} = 0.773$

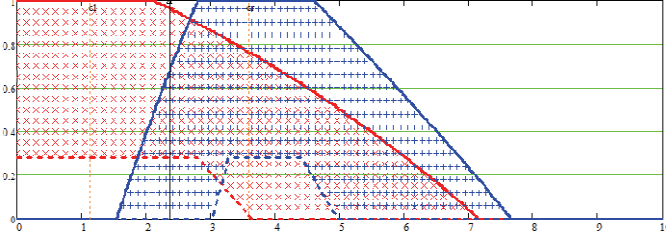


Figure 16 Conservative investor FOUs for a_2 , $r = -1$, $\alpha_g^{(-1)} = 0.773$

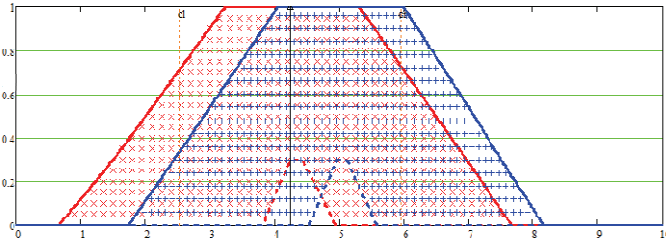


Figure 17 In-between investor FOUs for a_3 , $r = -1$, $\alpha_g^{(-1)} = 0.773$

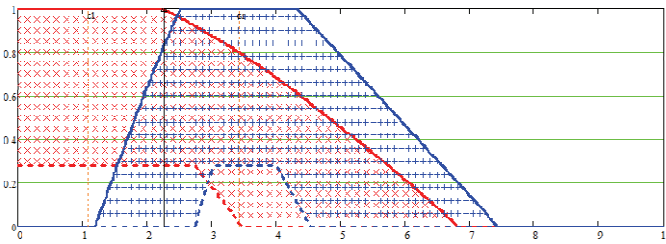


Figure 18 In-between investor FOUs for a_1 , $r = -1$, $\alpha_g^{(-1)} = 0.773$

In other examples in [1], Mendel and Wu employ hierarchical LWAs in a decision tree to aggregate subcategories of criteria scores into higher-level criteria. Use of the LWPM enables us further to generalize these hierarchical aggregations. For example, to aggregate a mandatory fuzzy input x_1 with a desired fuzzy input x_2 , Dujmovi [20] composes bivariate operators denoted by $x_1 \Delta (x_1 \nabla x_2)$ and called *conjunctive partial absorption* (CPA) aggregators (see also [13]). CPA aggregators apply a WPM partial disjunction (denoted ∇ , where $r \geq 1$) to the two inputs, and then take a partial conjunction (denoted Δ , where $r \leq 0$) of that output with the mandatory input.

In a type-1 fuzzy context, $x_1 \Delta (x_1 \nabla x_2)$ necessarily is zero if x_1 is zero, and $x_1 \Delta (x_1 \nabla x_2) - x_1$ is positive (resp. negative) when x_2 is greater (resp. less) than x_1 . The absolute difference $|x_1 \Delta (x_1 \nabla x_2) - x_1|$ is called the *reward* when x_2 is greater than x_1 , and is otherwise called a *penalty* [13]. We construct the operator $x_1 \Delta (x_1 \nabla x_2)$ from weighted power means as

$$CPA_2(\mathbf{x}, \mathbf{w}', \mathbf{w}) = L_r((x_1, L_s(\mathbf{x}, \mathbf{w})), \mathbf{w}'), \quad (15)$$

with $r \leq 0, 1 \leq s < \infty$. Here, $(x_1, L_s(\mathbf{x}, \mathbf{w}))$ are 2-dimensional scores, and $\mathbf{w}'$ is a 2-dimensional weight vector. However $\mathbf{x}$ and $\mathbf{w}$ may be of higher dimension if we have multiple partially substitutable inputs.

As an illustration of the application of IT2 CPA operators, suppose the speculative investor places a mandatory (in a type-1 context) requirement on high investment returns (criterion c_3 in Table 1), making it analogous to x_1 in (15), with a LWPM aggregation of the remaining criteria scores constituting x_2 using a power exponent of -1, as in Figs. 13-18. Suppose further that we specify a maximum reward of +15%, with a maximum penalty of -25%, and set $r = -10$ and $s = 1.449$ (the latter corresponding to an andness of 7/16) in (15). Using the techniques of [20], we compute corresponding type-1 weights $\mathbf{w} = (0.347, 0.653)$ and $\mathbf{w}' = (0.935, 0.065)$ in (15) that achieve these reward/penalty values, and then calculate the LWPMs for ∇ and Δ on the IT2 input variables, with singleton weights.

Fig. 19 shows the CPA IT2 FOU (X crosshatch) and the LWA FOU (+ crosshatch) for investment alternative a_1 (commodities). Comparing this with Fig. 20, where we used a LWPM on all four inputs with $r = -1$, we observe that the CPA FOU is substantially narrower, and is disposed over considerably higher values, than the LWPM FOU. This is understandable from inspection of Table 1, where we observe that the potential for high investment returns in commodities is graded as “extremely high” by a speculative investor. Thus we would expect an enhanced score for commodities when this criterion is elevated to mandatory.

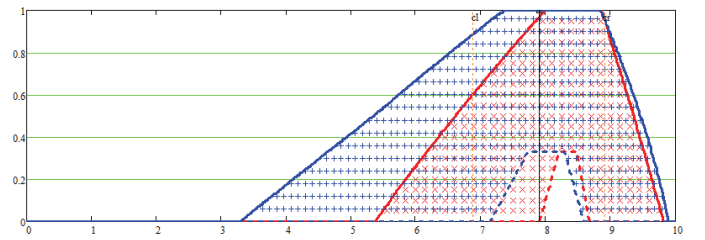


Figure 19 Speculative investor CPA for a_1 , with c_3 mandatory.

On the other hand, Figs. 21-22 show the same comparison for investment alternative a_5 (long-term bonds), whose potential for high returns is graded “very low” by the speculative investor. The CPA FOU differs relatively little from the comparable LWPM FOU, since alternative a_5 is in fact scored relatively low with respect to *all* criteria by the speculative investor.

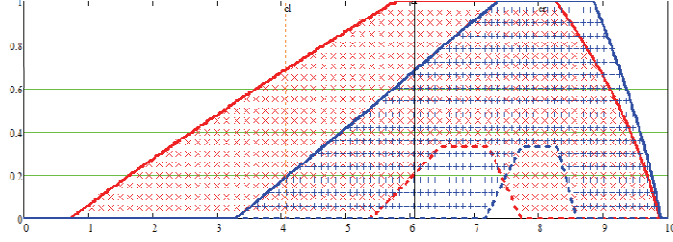


Figure 20 Speculative investor LWPM for a_1 , $r = -1$, $\alpha_g^{(-1)} = 0.773$.

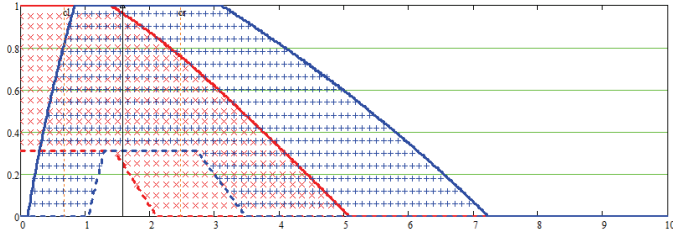


Figure 21 Speculative investor CPA for a_5 , with C_3 mandatory.

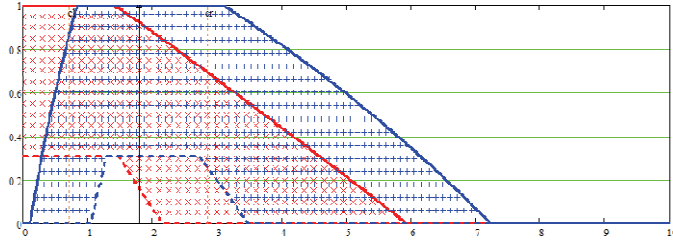


Figure 22 Speculative investor LWPM for a_5 , $r = -1$, $\alpha_g^{(-1)} = 0.773$.

V. CONCLUSION

We introduce a new family of Per-C computational engines based on the LWPM, and compare it with the LWA using an extended example described in [1]. The WPMEKM algorithm is presented for the computation of LWPM aggregations with interval input and weight variables. This can be extended to inputs and weights described by IT2 MFs using the α -cut Representation Theorem. The LWPM is shown to be a much more versatile aggregation operator, and thus it provides a significant extension to the suite of tools available for CWW applications. We also show how LWPM modules can be used to produce conjunctive partial absorption operators, and illustrate their differences with respect to a straightforward LWPM. These operators enable far more general hierarchical logic functions to be constructed than are available with simple hierarchies of LWAs.

- [1] Mendel, J.M. and D. Wu, *Perceptual Computing*, Hoboken, N.J.: John Wiley & Sons, Inc., 2010.
- [2] Zadeh, L.A., “Fuzzy logic = computing with words,” *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 103-111, 1996.
- [3] Special Section on Computing with Words, *IEEE Trans. Fuzzy Syst.*, vol. 18, June 2010.
- [4] Mendel, J.M., “Computing with words: Zadeh, Turing, Popper and Occam,” *IEEE Computational Intelligence Magazine*, vol. 2, pp. 10-17, Nov. 2007.
- [5] Yager, R.R., “Approximate reasoning as a basis for computing with words, in L.A. Zadeh and J. Kacprzyk (Eds.) *Computing with Words in Information/Intelligent Systems I: Foundations*, Heidelberg: Physica-Verlag, pp. 50-77, 1999.
- [6] Yager, R.R., “On the retranslation process in Zadeh’s paradigm of computing with words,” *IEEE Trans. On Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 34, pp. 1184-1195, March 2004.
- [7] Zadeh, L.A., “Toward human-level machine intelligence—Is it achievable? The need for a new paradigm shift,” *IEEE Computational Intelligence Magazine*, vol. 3, pp. 11-22, August 2008.
- [8] Mendel, J.M., *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Upper Saddle River, NJ: Prentice-Hall, 2001.
- [9] Karnick, N. and J.M. Mendel, “Centroid of a type-2 fuzzy set,” *Information Sciences*, vol. 132, pp. 195-220, 2001.
- [10] Klir, G. and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Upper Saddle River, NJ: Prentice-Hall, 1995.
- [11] Salton, G., E.A. Fox and H. Wu, “Extended Boolean information retrieval,” *Communications of the ACM*, vol. 26, pp. 1022-1036, Dec. 1983.
- [12] Hong W.S., Chen S.J. and Wang L.H., et al. “A new approach for fuzzy information retrieval based on weighted power-mean averaging operators,” *Computers & Mathematics with Applications*, vol. 53, pp.1800-1819, June 2007.
- [13] Dujmović, J., “Continuous preference logic for system evaluation,” *IEEE Trans. on Fuzzy Systems*, vol. 15, pp. 1082-1099, Dec. 2007.
- [14] Rickard, J.T., J. Aisbett, R.R. Yager and G. Gibbon, “Fuzzy weighted power means in evaluation decisions,” to be presented at *World Conf. on Soft Computing*, San Francisco, CA, May 2011.
- [15] Wu, D. and J.M. Mendel, “Aggregation using the linguistic weighted average and interval type-2 fuzzy sets,” *IEEE Trans. on Fuzzy Systems*, vol. 15, pp. 1145-1161, Dec. 2007.
- [16] Wu, D. and J.M. Mendel, “Corrections to ‘Aggregation using the linguistic weighted average and interval type-2 fuzzy sets’,” *IEEE Trans. on Fuzzy Systems*, vol. 16, pp. 1664-1666, Dec. 2008.
- [17] Tong, R.M. and P.P. Bonissone, “A linguistic approach to decision making with fuzzy sets,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-10, pp.716-723, Nov. 1980.
- [18] Liu F. and J.M. Mendel, “Aggregation using the fuzzy weighted average as computed using the Karnick-Mendel algorithms,” *IEEE Trans. Fuzzy Syst.*, vol.17, Dec. 2009.
- [19] Trillas E., C. Moraga, S. Guadarrama, S. Cubillo and E. Castiñeira, “Computing with Antonyms”, in M. Nikraves et al., (eds.), *Forging New Frontiers: Fuzzy Pioneers I. Studies in Fuzziness and Soft Computing*, vol. 217, pp. 133-153, 2007.
- [20] Dujmović, J., “Partial absorption functions,” *J. Univ. Belgrade, EE Dept. ser. Math & Physics*, no. 659, pp. 156-163, 1979.