

Fast Computation of Centroids for Constant-Width Interval-Valued Fuzzy Sets

Jerry M. Mendel and Hongwei Wu
Signal and Image Processing Institute
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089-2564, USA

Vladik Kreinovich and Gang Xiang
Pan-American Center for
Earth and Environmental Studies
University of Texas at El Paso
El Paso, TX 79968, USA
vladik@utep.edu

Abstract—Interval-valued fuzzy sets provide a more adequate description of uncertainty than traditional fuzzy sets; it is therefore important to use interval-valued fuzzy sets in applications. One of the main applications of fuzzy sets is fuzzy control, and one of the most computationally intensive part of fuzzy control is defuzzification. Since a transition to interval-valued fuzzy sets usually increases the amount of computations, it is vitally important to design faster algorithms for the corresponding defuzzification. In this paper, we provide such an algorithm for a practically important case of constant-width interval-valued fuzzy sets.

I. PRACTICAL NEED FOR FUZZY SETS: A BRIEF REMINDER

In many application areas, we have top-quality best experts who can provide the best recommendations: the best medical doctors can provide the best diagnostics, the best pilots can provide the best control of a plane, etc.

It is desirable to incorporate the knowledge of these top experts into automated systems that would, ideally, help other experts reach decisions which are (almost) as good as the decisions of the top experts. We would like to design a system that would provide medical doctors with the diagnostic advice coming from the best experts in the field; we would like to design a system that would, if necessary, provide pilots with the piloting advice coming from the best pilots, etc.

For that, we must be able to incorporate the knowledge of the top experts inside a computer. One of the main problems with this incorporation is that usually, experts cannot describe their knowledge in precise terms. For example, to describe a size of a certain type of a tumor, an expert would rarely use exact bounds like “the size is between 1.4 and 1.6 cm”; instead, an expert would use uncertain words from a natural language – e.g., that the size is “approximately 1.5, with an error about 0.1”.

Formalizing such expert statements is one of the main objectives of *fuzzy logic* designed by Lotfi A. Zadeh; see, e.g., [4], [11] and references therein.

In fuzzy logic, a statement like “approximately 1.5” is described by a *membership function* that assigns, to each real value x around 1.5, a degree $\mu(x)$ to which this value x fits the expert’s description.

II. WHERE DO MEMBERSHIP DEGREES COME FROM – A BRIEF REMINDER

As we have just mentioned, one of the main ideas of fuzzy logic is that to characterize a statement about some (unknown) value (or values), for each possible value x , we describe the degree $\mu(x)$ to which this value is consistent with the expert’s knowledge. An important practical question is: from where do we get these degrees (see, e.g., [1]).

Since these degrees describe the degree of certainty of an expert, it is natural to ask the expert to provide these degrees – and this is what is actually done in most applications of fuzzy techniques. To be more precise, an expert provides the degree of certainty for several values x , and then we use interpolation and extrapolation to estimate the values of $\mu(x)$ for all other x .

III. PRACTICALLY IMPORTANT CASE: TRIANGULAR MEMBERSHIP FUNCTIONS

Often, to characterize the expert’s uncertainty about the value of a quantity, we ask the expert to provide us with:

- the range $[\underline{x}, \bar{x}]$ of possible values – so that $\mu(x) = 0$ for all $x \notin [\underline{x}, \bar{x}]$ – and
- the “most probable” value \tilde{x} of the quantity x – for which the degree of possibility $\mu(\tilde{x})$ is the largest (this largest degree is usually assumed to be equal to 1).

In this case, we can use linear interpolation to describe the values of $\mu(x)$ for the subintervals $[\underline{x}, \tilde{x}]$ and $[\tilde{x}, \bar{x}]$. The resulting membership function has a *triangular* shape: it is equal to 0 outside the interval $[\underline{x}, \bar{x}]$, and its graph on this interval consists of two straight line segments rising from 0 to 1 and then going from 1 to 0.

IV. APPLICATIONS OF FUZZY SETS: IN BRIEF

It is well known that there are a lot of practical applications of fuzzy techniques.

In many applications, we start with fuzzy values – and we propagate the original fuzziness all the way to the answer, by transforming fuzzy rules and fuzzy inputs into fuzzy recommendations.

However, there is one well known exception to this general feature of fuzzy technique applications. One of the main applications of fuzzy techniques is *intelligent control*. In fuzzy

control, the objective is not so much to provide an advise to the expert, but rather to generate a single (crisp) control value u_c that will be automatically applied by the automated controller.

To get this value, we first use the fuzzy control rules to combine the membership functions of the inputs into a membership function $\mu(u)$ for the desired control u . This function would be a good output if all we wanted was an advise to a human expert. However, since we are interested in generating a crisp value for the automatical controller, we must transform the fuzzy membership function $\mu(u)$ into a single value u_c . This transformation from fuzzy to crisp is called *defuzzification*; see, e.g., a survey [6].

The most widely used defuzzification technique – so-called *centroid defuzzification* – is, in effect, based on minimizing the mean square difference between the actual (unknown) optimal control u and the generated control u_c produced as a result. In this least square optimization, we weigh each value u with the weight proportional to its degree of possibility $\mu(u)$. The resulting optimization problem

$$\int \mu(u) \cdot (u - u_c)^2 du \rightarrow \min_{u_c}$$

can be explicitly solved if we differentiate the corresponding objective function by u_c and equate the resulting value to 0. As a result, we get the formula

$$u_c = \frac{\int u \cdot \mu(u) du}{\int \mu(u) du}. \quad (1)$$

This formula is called centroid defuzzification because it describes the u -coordinate of the center of mass of the region bounded by the graph of the membership function $\mu(u)$.

V. LIMITATIONS OF TRADITIONAL FUZZY SETS AND A PRACTICAL NEED FOR INTERVAL-VALUED FUZZY SETS

In the traditional fuzzy approach, for each statement, the same expert who made this informal statement is also supposed to provide us with the membership degrees $\mu(x)$ that describe to what degree different values x fit this statement.

The problem with this approach is that, just like an expert is unable to provide the crisp values of the desired quantities, this same expert is usually unable to describe his or her degree of certainty by an exact number.

At best, an expert can provide us with an *approximate* degree $\tilde{\mu}(x)$ and a bound $\Delta(x)$ describing his uncertainty – i.e., in effect, with an *interval* of possible values of certainty: $[\underline{\mu}(x), \bar{\mu}(x)] = [\tilde{\mu}(x) - \Delta(x), \tilde{\mu}(x) + \Delta(x)]$.

The resulting *interval-valued* membership function is a more adequate description of expert's uncertainty than, e.g., the function $\tilde{\mu}(x)$ – since it also provides us with the expert's estimates of how accurately this expert estimates his/her own uncertainty.

VI. IMPORTANT CASE: CONSTANT-WIDTH INTERVAL-VALUED FUZZY SETS

Extracting the degrees $\mu(x)$ from an expert is already a difficult task. Extracting the uncertainty bounds on these degrees is even more difficult.

In most practical situations, the most we can do is to ask an expert for a crude overall estimate Δ of the uncertainty with which the expert estimates his or her degree of certainty. This means, in effect, that instead of different bounds $\Delta(x)$ for different values x , we get the same bounds Δ for all possible values of the quantity x .

In other words, in such situations, $\Delta(x)$ is simply a constant. In this case, we get interval-valued fuzzy sets of constant width $\bar{\mu}(x) - \underline{\mu}(x) = 2\Delta \equiv \text{const.}$

VII. INTERVAL-VALUED FUZZY SETS ARE USEFUL IN MANY PRACTICAL PROBLEMS

Interval-valued fuzzy sets have been successfully applied to many practical problems including intelligent control; see, e.g., [8], [10] and references therein. Since interval-valued fuzzy sets provide a more accurate representation of expert uncertainty, the resulting systems often lead to a better quality control than more traditional “first-order” applications of fuzzy techniques.

In view of the success of these applications, it seems reasonable to apply interval-valued fuzzy sets to other areas of application of traditional fuzzy techniques. However, we cannot always do that.

VIII. INCREASE IN COMPUTATIONAL COMPLEXITY – ONE OF THE MAIN REASONS PREVENTING US FROM WIDER USE OF INTERVAL-VALUED FUZZY SETS

One of the main obstacles to replacing traditional fuzzy sets with interval-valued fuzzy sets in many applications of fuzzy techniques is that this replacement often leads to an increase in computational complexity.

Indeed, to describe a traditional fuzzy set, for each possible value x of the corresponding quantity, we only need to store a *single* value $\mu(x)$. To describe an interval-valued fuzzy set, for each possible x , we need, in general, to store *two* values:

- either the endpoints $\underline{\mu}(x)$ and $\bar{\mu}(x)$ of the corresponding interval,
- or its midpoint $\tilde{\mu}(x)$ and half-width $\Delta(x)$.

So, when we replace the traditional fuzzy approach with the interval-valued fuzzy approach, we immediately double the amount of information about uncertainty that we need to process.

Even reading and storing this information, operations for which computation time grows linearly with the input, we thus need twice more computation time. For many efficient data processing algorithms (see, e.g., [2]), the computation time grows as a square $O(n^2)$ or a cube $O(n^3)$ of the size n of the input. For such algorithms, the doubling of the input size leads to a 4 or 8 times increase in computation time.

This increase in computation time is one of the main reasons why interval-valued fuzzy techniques are not as widely used – as they should be based on the improved quality of their existing applications.

IX. THIS INCREASE IN COMPUTATION TIME IS ESPECIALLY CRUCIAL IN REAL-TIME CONTROL APPLICATIONS

The drastic increase in computation time from traditional fuzzy to interval-valued fuzzy techniques may not be that crucial for more traditional expert system-time applications. Indeed, in such applications, the main objective of fuzzy techniques is to produce advice for the experts. In such applications, computation time is much smaller than the time that is needed for an expert to make a decision. As a result, even an 8-fold increase in computation time will not drastically slow down the decision making process.

However, as we have mentioned, one of the main applications of fuzzy techniques is intelligent control. In control applications, we need to produce results in real time, often really fast. In such cases, the 8-fold increase in computation time means that to avoid slowing down the controller, we must use an 8 times faster processor – which may be more expensive and more difficult to fit into this particular controller device.

X. CENTROID DEFUZZIFICATION: THE MOST TIME-CONSUMING PART OF INTELLIGENT CONTROL

Of all the computational steps required for fuzzy control, the most time-consuming step is centroid defuzzification.

According to the original defuzzification formula (1), to compute the desired value of the control u_c , we must perform two integrations and then divide the results. Integration can be done numerically, as a sum. So, if we have n values $\mu(x_i)$ of the membership function, we need a linear time $O(n)$ to compute both integrals. For large n , this may constitute a long time, but it is still much faster than $O(n^2)$ or $O(n^3)$ algorithms.

In many cases, we know the analytical expression for the membership function and thus, we can perform the integration analytically. This is the case, for example, when we have a triangular membership function. In this case, by integrating the corresponding expressions, we get explicit analytical expressions for u_c – and thus, avoid time-consuming integrations altogether.

XI. CENTROID DEFUZZIFICATION BECOMES EVEN MORE COMPUTATIONALLY INTENSIVE FOR INTERVAL-VALUED FUZZY SETS

When we move from a number-valued membership function $\mu(x)$ to the interval-valued membership function $[\underline{\mu}(x), \overline{\mu}(x)]$, the computational problem becomes even more difficult.

Having an interval-valued membership function means, in effect, that we do not know the exact membership function $\mu(x)$; we may therefore have different membership functions $\mu(x)$ – as long as for every x , we have $\mu(x) \in [\underline{\mu}(x), \overline{\mu}(x)]$.

Different membership functions $\mu(x)$ with this property lead, in general, to different values of the control u_c . Thus, instead of a single value u_c , we should produce an *interval* $[\underline{u}_c, \overline{u}_c]$ of possible values of u_c – corresponding to all possible membership functions $\mu(x) \in [\underline{\mu}(x), \overline{\mu}(x)]$.

A literal implementation of this idea would require an impossible amount of computation time. Indeed, let us assume that for each x , we check m different possible values of $\mu(x)$ within the corresponding interval. Then, for n different values x_i , we need to check m^n different membership functions – an amount which, for large n , exceeds the number of particles in the Universe.

XII. SOPHISTICATED ALGORITHMS MAKE CENTROIDS OF INTERVAL-VALUED FUZZY SETS COMPUTABLE – BUT STILL MORE COMPUTATIONALLY INTENSIVE THAN FOR TRADITIONAL FUZZY SETS

A (known) good news is that in the interval-valued fuzzy case, there is no need to actually perform all m^n computations. Namely, as proven in [3], [5], [7], we can find the desired endpoints \underline{u}_c and \overline{u}_c as follows:

$$\underline{u}_c = \min_{\ell} \frac{N(\ell)}{D(\ell)}, \quad (2)$$

where

$$D(\ell) \stackrel{\text{def}}{=} \int_{\underline{u}}^{\ell} \overline{\mu}(u) du + \int_{\ell}^{\overline{u}} \underline{\mu}(u) du; \quad (3)$$

$$N(\ell) \stackrel{\text{def}}{=} \int_{\underline{u}}^{\ell} u \cdot \overline{\mu}(u) du + \int_{\ell}^{\overline{u}} u \cdot \underline{\mu}(u) du; \quad (4)$$

and

$$\overline{u}_c = \max_{\ell} \frac{\overline{N}(\ell)}{\overline{D}(\ell)}, \quad (5)$$

where

$$\overline{D}(\ell) \stackrel{\text{def}}{=} \int_{\underline{u}}^{\ell} \underline{\mu}(u) du + \int_{\ell}^{\overline{u}} \overline{\mu}(u) du; \quad (6)$$

$$\overline{N}(\ell) \stackrel{\text{def}}{=} \int_{\underline{u}}^{\ell} u \cdot \underline{\mu}(u) du + \int_{\ell}^{\overline{u}} u \cdot \overline{\mu}(u) du. \quad (7)$$

One way to use these formulas is to compute these ratios for different values ℓ , and then to find the ratio which is the smallest (for \underline{u}_c) or the largest (for \overline{u}_c).

If we use n values of x_i , then computing each of these integrals, for every ℓ , requires $O(n)$ steps. Since we need to check n different values ℓ , we thus need $n \times O(n) = O(n^2)$ computational steps. This is drastically faster than m^n steps – and actually doable. However, it is still much slower than $O(n)$ computational steps that we need to compute a centroid of a traditional fuzzy set.

XIII. THERE ARE TECHNIQUES FOR COMPUTING THE CENTROID SOMEWHAT FASTER – BUT IN GENERAL, THESE FASTER COMPUTATIONS STILL REQUIRE TIME $O(n^2) \gg O(n)$.

One way to compute the centroid bounds yet faster – described in [9], [12] – is to take into account the known fact that the minimum and the maximum of a smooth function of the variable ℓ are attained when the derivative of this function relative to ℓ is equal to 0.

For \underline{u}_c , we must differentiate the ratio $\frac{N(\ell)}{D(\ell)}$. The derivative of this ratio is equal to 0 if and only if

$$\frac{\partial N(\ell)}{\partial \ell} \cdot \underline{D}(\ell) = \frac{\partial \underline{D}(\ell)}{\partial \ell} \cdot N(\ell). \quad (8)$$

Differentiating the above expression (3) and (4) for $N(\ell)$ and $\underline{D}(\ell)$, we conclude that

$$\frac{\partial N(\ell)}{\partial \ell} = \ell \cdot \bar{\mu}(\ell) - \ell \cdot \underline{\mu}(\ell) = \ell \cdot (\bar{\mu}(\ell) - \underline{\mu}(\ell))$$

and

$$\frac{\partial \underline{D}(\ell)}{\partial \ell} = \bar{\mu}(\ell) - \underline{\mu}(\ell).$$

Substituting these values into the equation (8), and dividing both sides by the common factor $\bar{\mu}(\ell) - \underline{\mu}(\ell)$, we conclude that $\underline{D}(\ell) \cdot \ell = N(\ell)$, i.e., that $\ell = \frac{N(\ell)}{\underline{D}(\ell)}$. In other words, we conclude that $\ell = \underline{u}_c$.

Thus, instead of computing \underline{u}_c as the value ℓ where the ratio (2) attains its minimal values, we can instead find \underline{u}_c as the value ℓ for which

$$\ell = \frac{N(\ell)}{\underline{D}(\ell)}. \quad (9)$$

The computational advantage of this approach in comparison with the optimization formula (2) is as follows:

- In general, to find the smallest value of a function, we must consider all possible values of the variable ℓ . In our case, this means that we need to consider n different values ℓ .
- If our objective is to find the value ℓ that satisfies an equation (e.g., equation (9)), then, once we have found this value, we do not need to search anymore. In principle, the desired value can be anywhere from the very first point to the very last (n -th) point. Thus, on average, we need to only search $n/2$ points – i.e., only half as many as for the optimization approach.

Similarly, to compute \bar{u}_c , instead of solving the optimization problem (5), we can find \bar{u}_c as the value ℓ that satisfies the following equality:

$$\ell = \frac{\bar{N}(\ell)}{\bar{D}(\ell)}. \quad (10)$$

In both cases, instead of $C \cdot n^2$ computational steps, we need only half as many, i.e., $(C/2) \cdot n^2$ computational steps. This is faster than before, but still, we need $O(n^2) \gg O(n)$ computational steps.

XIV. RESULTING PROBLEM: IN BRIEF

Since the existing algorithms for computing the endpoints \underline{u}_c and \bar{u}_c of the centroid interval are very time-consuming, it is desirable to come up with faster algorithms for computing these endpoints.

XV. WHAT WAS KNOWN: FAST COMPUTATION OF CENTROIDS FOR TRIANGULAR CONSTANT-WIDTH INTERVAL-VALUED FUZZY SETS

In [9], [12], it was shown that for a constant-width triangular membership function, we can use explicit formulas to compute \underline{u} and \bar{u} .

XVI. OUR NEW RESULT: FAST COMPUTATION OF CENTROIDS FOR ARBITRARILY SHAPED CONSTANT-WIDTH INTERVAL-VALUED FUZZY SETS

In this paper, we extend the fast algorithm from [9], [12] to arbitrary constant-width interval-valued fuzzy sets.

Let us derive the formulas that lead to this algorithm. Let us start with the formulas for computing \underline{u}_c . These formulas describe the desired value \underline{u}_c in terms of an equation that depends on the integrals $\underline{D}(\ell)$ and $N(\ell)$ – which, in turn, require integration of the endpoints $\underline{\mu}(u)$ and $\bar{\mu}(u)$ (in accordance with formulas (3) and (4)).

For a constant-width interval fuzzy set, for every u , we have $\bar{\mu}(u) = \underline{\mu}(u) + 2\Delta$. Substituting this expression into the formula (3), we conclude that

$$\underline{D}(\ell) = \int_{\underline{u}}^{\bar{u}} \underline{\mu}(u) du + \int_{\underline{u}}^{\ell} 2\Delta du = D_0 + 2\Delta \cdot (\ell - \underline{u}), \quad (11)$$

where we denoted

$$D_0 \stackrel{\text{def}}{=} \int_{\underline{u}}^{\bar{u}} \underline{\mu}(u) du. \quad (12)$$

Similarly, substituting this expression into the formula (4), we conclude that

$$\begin{aligned} N(\ell) &= \int_{\underline{u}}^{\bar{u}} u \cdot \underline{\mu}(u) du + \int_{\underline{u}}^{\ell} u \cdot 2\Delta du = \\ &= N_0 + \Delta \cdot (\ell^2 - \underline{u}^2), \end{aligned} \quad (13)$$

where we denoted

$$N_0 \stackrel{\text{def}}{=} \int_{\underline{u}}^{\bar{u}} u \cdot \underline{\mu}(u) du. \quad (14)$$

Thus, the condition that $\underline{D}(\ell) \cdot \ell = N(\ell)$ that determines the desired value \underline{u}_c can be rewritten as

$$\ell \cdot (D_0 + 2\Delta \cdot (\ell - \underline{u})) = N_0 + \Delta \cdot (\ell^2 - \underline{u}^2),$$

i.e., as a quadratic equation

$$\Delta \cdot \ell^2 + \ell \cdot (D_0 - 2\Delta \cdot \underline{u}) + (\Delta \cdot \underline{u}^2 - N_0) = 0 \quad (14)$$

This quadratic equation has an explicit (thus easy-to-compute) solution

$$\ell = \frac{-(D_0 - 2\Delta \cdot \underline{u}) \pm \sqrt{d}}{2\Delta}, \quad (15)$$

where

$$\begin{aligned} d &\stackrel{\text{def}}{=} (D_0 - 2\Delta \cdot \underline{u})^2 + 4(N_0 - \Delta \cdot \underline{u}^2) \cdot \Delta = \\ &= D_0^2 - 4 \cdot D_0 \cdot \Delta \cdot \underline{u} + 4 \cdot N_0 \cdot \Delta. \end{aligned}$$

Thus,

$$\underline{d} = D_0^2 + 4\Delta \cdot (N_0 - D_0 \cdot \underline{u}). \quad (16)$$

Similarly, for computing \bar{u}_c , we have

$$\begin{aligned} \bar{D}(\ell) &= \int_{\underline{u}}^{\bar{u}} \underline{\mu}(u) du + \int_{\ell}^{\bar{u}} 2\Delta du = \\ &D_0 + 2\Delta \cdot (\bar{u} - \ell), \end{aligned} \quad (17)$$

and

$$\begin{aligned} \bar{N}(\ell) &= \int_{\underline{u}}^{\bar{u}} u \cdot \underline{\mu}(u) du + \int_{\ell}^{\bar{u}} u \cdot 2\Delta du = \\ &N_0 + \Delta \cdot (\bar{u}^2 - \ell^2). \end{aligned} \quad (18)$$

Thus, the condition that $\bar{D}(\ell) \cdot \ell = \bar{N}(\ell)$ that determines the desired value \bar{u}_c can be rewritten as

$$\ell \cdot (D_0 + 2\Delta \cdot (\bar{u} - \ell)) = N_0 + \Delta \cdot (\bar{u}^2 - \ell^2),$$

i.e., as a quadratic equation

$$\Delta \cdot \ell^2 - \ell \cdot (D_0 + 2\Delta \cdot \bar{u}) + (\Delta \cdot \bar{u}^2 + N_0) = 0 \quad (19)$$

This quadratic equation has an explicit (thus easy-to-compute) solution

$$\ell = \frac{(D_0 + 2\Delta \cdot \bar{u}) \pm \sqrt{\bar{d}}}{2\Delta}, \quad (20)$$

where

$$\begin{aligned} \bar{d} &\stackrel{\text{def}}{=} (D_0 + 2\Delta \cdot \bar{u})^2 - 4(N_0 + \Delta \cdot \bar{u}^2) \cdot \Delta = \\ &D_0^2 + 4 \cdot D_0 \cdot \Delta \cdot \bar{u} - 4 \cdot N_0 \cdot \Delta. \end{aligned}$$

Thus,

$$\bar{d} = D_0^2 - 4\Delta \cdot (N_0 - D_0 \cdot \bar{u}). \quad (21)$$

XVII. RESULTING ALGORITHM

As a result, we arrive at the following algorithm for computing centroids of arbitrarily shaped constant-width interval-valued fuzzy sets:

- first, we compute the values D_0 and N_0 by using formulas (12) and (14);
- then, we compute the values \underline{d} and \bar{d} by using formulas (16) and (21);
- after this, we compute the values \underline{u}_c and \bar{u}_c by using formulas (15) and (20).

Once we have computed D_0 and N_0 , all the following computations are explicit and thus, require only a finite number of computational steps.

The computation of each of the integrals D_0 and N_0 requires $O(n)$ steps, just like for the traditional fuzzy sets. Thus, the resulting algorithm for computing a centroid of an interval-valued fuzzy set requires only slightly more computation time than for traditional fuzzy sets – this is exactly what we wanted.

We have mentioned that in the traditional fuzzy sets, we often do not even need $O(n)$ computational steps to compute the integrals D and N – if we have an explicit analytical expression for these integrals, and thus, spend only a constant number of computational steps. In the interval-valued fuzzy case, similarly, if we know an explicit analytical expression for $\underline{\mu}(u)$, then we need a constant number of computational steps to compute both integrals D_0 and N_0 – and thus, to compute the desired endpoints \underline{u}_c and \bar{u}_c .

XVIII. DISCUSSION: CAN WE EXTEND OUR ALGORITHMS TO A MORE GENERAL CASE?

In the above text, we considered the case when the interval-valued fuzzy set has a constant width $\bar{\mu}(u) - \underline{\mu}(u) = 2\Delta(u) \equiv 2\Delta$. This case corresponds to the situation when we only have a single parameter Δ to gauge how accurately an expert can quantify his or her degree of certainty.

In some practical situations, we can have a more accurate description of this accuracy – e.g., we can have a piece-wise linear expression for $\Delta(u)$. This expression can appear if we ask an expert to gauge his or her accuracy in quantifying the degree of certainty at different values u_i – and then extrapolate these values to arbitrary u by assuming that the dependence $\Delta(u)$ is linear on each interpolation interval $[u_i, u_{i+1}]$. In other words, the origin of piece-wise linear functions $\Delta(u)$ is the same as the above-described origin of piece-wise linear membership functions $\mu(x)$ – linear interpolation.

For a constant function $\Delta(u) \equiv \Delta$, integrations (3) and (4) lead to a linear expression for $\underline{D}(\ell)$ and to a quadratic expression for $\underline{N}(\ell)$ – thus, the desired equation $\underline{D}(\ell) \cdot \ell = \underline{N}(\ell)$ becomes a quadratic equation in terms of $\ell = \underline{u}_c$. Similarly, integrations lead to a linear expression for $\bar{D}(\ell)$ and to a quadratic expression for $\bar{N}(\ell)$ – thus, the desired equation $\bar{D}(\ell) \cdot \ell = \bar{N}(\ell)$ becomes a quadratic equation in terms of $\ell = \bar{u}_c$.

For a piece-wise linear function $\Delta(u)$, integrations (3) and (4) lead to a quadratic expression for $\underline{D}(\ell)$ and to a cubic expression for $\underline{N}(\ell)$ – thus, the desired equation $\underline{D}(\ell) \cdot \ell = \underline{N}(\ell)$ becomes a cubic equation in terms of $\ell = \underline{u}_c$. Similarly, the equation $\bar{D}(\ell) \cdot \ell = \bar{N}(\ell)$ becomes a cubic equation in terms of $\ell = \bar{u}_c$.

For cubic equations, there are also explicit formulas for solving them – although they are not so numerically simple and useful as for quadratic equations.

XIX. CASE OF NARROW INTERVALS

We have mentioned that in general, computing the centroid interval for an interval-valued fuzzy set

$$[\tilde{\mu}(u) - \Delta(u), \tilde{\mu}(u) + \Delta(u)]$$

can be very time consuming. The corresponding computation time is much larger than the time needed to compute the centroid of a number-valued fuzzy set $\tilde{\mu}(u)$. We can therefore make an informal conclusion that the computational difficulty of this interval-related problem is caused by the fact that experts cannot exactly describe their degrees of certainty $\mu(u)$. As a result, algorithms needed for general intervals require much more computation time than algorithms needed when the corresponding intervals have negligible width $\Delta(u) \ll \tilde{\mu}(u)$ – i.e., for the practical purposes, can be considered degenerate (one-point) intervals $[\tilde{\mu}(u), \tilde{\mu}(u)]$.

This conclusion leads to a natural hypothesis that the narrower the intervals, the faster we can compute the centroid. Let us therefore consider the corresponding case of narrow intervals, i.e., the case when the half-width $\Delta(u)$ is not

negligible but still much smaller than the membership value $\tilde{\mu}(u)$.

In this *narrow intervals* case, when computing different characteristics such as centroid endpoints, we can *linearize* these expressions – i.e., expand these characteristics in Taylor series and keep only linear terms in the resulting expansion. In this section, we show that by using linearization, we can indeed speed up the computation of the centroid intervals.

Let us first apply this idea of linearization to the equation (9) that describes \underline{u}_c . We start with the approximate centroid value

$$\tilde{u} = \frac{\tilde{N}}{\tilde{D}}, \quad (22)$$

where

$$\tilde{D} \stackrel{\text{def}}{=} \int_{\underline{u}} \tilde{\mu}(u) du \quad \tilde{N} \stackrel{\text{def}}{=} \int_{\underline{u}} u \cdot \tilde{\mu}(u) du. \quad (23)$$

We then assume that $\underline{u}_c = \ell = \tilde{u} - \delta u$, where δu is a small number, and in the resulting equation $\underline{D}(\ell) \cdot \ell = \underline{N}(\ell)$, we ignore terms which are quadratic in $\Delta(u)$ and δu . Thus:

$$\begin{aligned} \underline{D}(\ell) &= \int_{\underline{u}}^{\tilde{u}-\delta u} (\tilde{u} + \Delta(u)) du + \int_{\tilde{u}-\delta u}^{\tilde{u}} (\tilde{u} - \Delta(u)) du = \\ &\tilde{D} + \int_{\underline{u}}^{\tilde{u}} \Delta(u) du - \int_{\tilde{u}}^{\tilde{u}} \Delta(u) du - 2\Delta(\tilde{u}) \cdot \delta u. \end{aligned}$$

Similarly, for $\underline{N}(\ell)$, we get

$$\begin{aligned} \underline{N}(\ell) &= \int_{\underline{u}}^{\tilde{u}-\delta u} u \cdot (\tilde{u} + \Delta(u)) du + \int_{\tilde{u}-\delta u}^{\tilde{u}} u \cdot (\tilde{u} - \Delta(u)) du = \\ &\tilde{N} + \int_{\underline{u}}^{\tilde{u}} u \cdot \Delta(u) du - \int_{\tilde{u}}^{\tilde{u}} u \cdot \Delta(u) du - 2\tilde{u} \cdot \Delta(\tilde{u}) \cdot \delta u. \end{aligned}$$

Taking into account that $\ell = \tilde{u} - \delta u$, we conclude that the equation $\underline{N}(\ell) = \ell \cdot \underline{D}(\ell)$ takes the form

$$\begin{aligned} \left(\tilde{D} + \int_{\underline{u}}^{\tilde{u}} \Delta(u) du - \int_{\tilde{u}}^{\tilde{u}} \Delta(u) du - 2\Delta(\tilde{u}) \cdot \delta u \right) \cdot (\tilde{u} - \delta u) = \\ \tilde{N} + \int_{\underline{u}}^{\tilde{u}} u \cdot \Delta(u) du - \int_{\tilde{u}}^{\tilde{u}} u \cdot \Delta(u) du - 2\tilde{u} \cdot \Delta(\tilde{u}) \cdot \delta u. \end{aligned}$$

Opening parentheses, ignoring quadratic terms, and taking into account that by definition of \tilde{u} , we have $\tilde{u} \cdot \tilde{D} = \tilde{N}$, we conclude that

$$\delta u = \frac{\delta N}{\tilde{D}}, \quad (24)$$

where

$$\begin{aligned} \delta N \stackrel{\text{def}}{=} \int_{\tilde{u}}^{\tilde{u}} u \cdot \Delta(u) du - \int_{\underline{u}}^{\tilde{u}} u \cdot \Delta(u) du - \\ \tilde{u} \cdot \left(\int_{\tilde{u}}^{\tilde{u}} \Delta(u) du - \int_{\underline{u}}^{\tilde{u}} \Delta(u) du \right). \end{aligned} \quad (25)$$

A similar analysis of \bar{u}_c shows that $\bar{u}_c = \tilde{u} + \delta u$.

Thus, we arrive at the following algorithm for computing the centroid interval for narrow interval-values fuzzy sets:

- first, we compute the approximate centroid value \tilde{u} by using formulas (22)–(23);
- then, we compute δu by using formulas (24) and (25);
- the resulting centroid interval is $[\tilde{u} - \delta u, \tilde{u} + \delta u]$.

In this computation, instead of computing two integrals D and N (as for traditional number-valued fuzzy sets), we must now compute three intervals: \tilde{D} , \tilde{N} , and δN . Computing each integral requires $O(n)$ time, so we still have a linear-time algorithm for computing the centroid range – although this interval requires 50% more computation time.

Comment. A similar analysis can be done for the case when we cannot ignore quadratic terms but we can safely ignore cubic and higher order terms. In this case, we can similarly expand both sides of the equation (9) in terms of $\Delta(u)$ and δu , keep only quadratic terms in this expansion, and get an explicit quadratic-accuracy expression for δu . The computation of this expression also requires time $o(n)$ – although it will mean computing even more integrals than in the linear approximation.

ACKNOWLEDGMENTS

This work was supported in part by NASA under cooperative agreement NCC5-209, NSF grants EAR-0225670 and DMS-0532645, Star Award from the University of Texas System, and by the Texas Department of Transportation grant No. 0-5453.

REFERENCES

- [1] T. Bilgic and I. B. Turksen, “Measurement of membership functions: theoretical and empirical work”, In: D. Dubois and H. Prade (eds.), *Handbook of Fuzzy Systems, Col. 1: Foundations*, Kluwer, Boston, Massachusetts, 2000, pp. 195–228.
- [2] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.
- [3] N. N. Karnik and J. M. Mendel, “Centroid of a type-2 fuzzy set”, *Information Sciences*, 1001, Vol. 132, pp. 195–220.
- [4] G. Klir and B. Yuan, *Fuzzy sets and fuzzy logic: theory and applications*. Prentice Hall, Upper Saddle River, New Jersey, 1995.
- [5] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational complexity and feasibility of data processing and interval computations*, Kluwer Academic Press, 1998.
- [6] V. Kreinovich, G. C. Mouzouris, and H. T. Nguyen, “Fuzzy rule based modeling as a universal approximation tool”, In: H. T. Nguyen and M. Sugeno (eds.), *Fuzzy Systems: Modeling and Control*, Kluwer, Boston, MA, 1998, pp. 135–195.
- [7] R. Lea, V. Kreinovich, and R. Trejo, “Optimal interval enclosures for fractionally-linear functions, and their application to intelligent control”, *Reliable Computing*, 1996, Vol. 2, No. 3, pp. 265–286.
- [8] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice-Hall, Upper Saddle River, New Jersey, 2001.
- [9] J. M. Mendel and H. Wu, “Centroid uncertainty bounds for interval type-2 fuzzy sets: forward and inverse problem”, *Proceedings of the 2005 IEEE International Conference on Fuzzy Systems FUZZ-IEEE’2005*, Reno, Nevada, May 22–25, 2005.
- [10] H. T. Nguyen, V. Kreinovich, and Q. Zuo, “Interval-valued degrees of belief: applications of interval computations to expert systems and intelligent control”, *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, 1997, Vol. 5, No. 3, pp. 317–358.
- [11] H. T. Nguyen and E. A. Walker, *A first course in fuzzy logic*, CRC Press, Boca Raton, Florida, 1997.
- [12] H. Wu and J. M. Mendel, “Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems”, *IEEE Transactions on Fuzzy Systems*, 2002, Vol. 10, No. 5, pp. 622–639.