Soft Computing
https://doi.org/10.1007/s00500-021-06470-1

APPLICATION OF SOFT COMPUTING

An ordering of fuzzy numbers based on the Zadeh extension principle

Ana Aguilera[1] · Soraya Carrasquel[2] · David Coronado[2] · Ricardo Monascal[2] · Rosseline Rodríguez[2] · Leonid Tineo[2]

Accepted: 21 October 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Order appears in all areas related to mathematics and computer science. Given a set in many cases it is desirable to establish a precedence relation between the elements of the set either total or partial. In this paper, we explore an ordering operator for fuzzy numbers that is based on the Zadeh extension principle. This proposal takes into account the intuition of users of Database Management Systems. Our analysis includes a formal proof of these operator's properties and examples of the applicability of the operator in representative cases that show its suitability for intuition management. Also, an operator implementation in the Haskell and SQL languages is presented. This allows its evaluation in different contexts.

Keywords Ordering · Fuzzy numbers · Extension principle · DBMS

1 Introduction

The fuzzy number concept was introduced in (Zadeh 1965) with the purpose of manipulating approximate numerical values and to represent uncertain, imprecise and null values. Fuzzy numbers usually are represented by possibility distributions. They can often overlap each other in many practical situations, as it is the case in this work. When two fuzzy numbers overlap with each other, a fuzzy number may not be considered absolutely larger than the other (Azman and Abdullah 2012).

Ranking fuzzy numbers are an increasingly important research issue, which is a base in many multi-criteria decision making (MCDM) problems (Azman and Abdullah 2012; Chi and Yu 2018). In the fuzzy MCDM (FMCDM), evaluation ratings and criteria weights are assessed on imprecision, subjectivity or vagueness (Wang 2020a). In FMCDM problems, some evaluation ratings and criteria weights are displayed by linguistic terms and transferred into fuzzy numbers (Wang 2020b) . The rating of each alternative with respect to each criterion can be then described by fuzzy numbers and in particular by Ordered Fuzzy Numbers (OFNs) (Rudnik and Kacprzak 2017).

From the application view point, the ordering or ranking of fuzzy numbers plays a main role in real-life problems involving decision-making, clustering, optimization, transportation problems, etc. Problems in various fields such as social and economic systems, forecasting, optimization and risk analysis can be solved by ordering fuzzy numbers (Akyar 2018; Singh 2015). Another application of ranking fuzzy numbers can be found in hypothesis testing when doing statistical inference (Alizadeh et al. 2013). Thus, ordering or ranking fuzzy numbers is one of the important research topics in fuzzy set theory.

Ranking fuzzy numbers have been widely studied, and many different methods have been proposed (Vincent et al. 2017). Canfora and Troiano (2004) give up on the idea of creating a universal ordering. They assume that each permutation of a set of fuzzy numbers is a direct-type ordering,

✉ Ana Aguilera
ana.aguilera@uv.cl

Soraya Carrasquel
scarrasquel@usb.ve

David Coronado
dcoronado@usb.ve

Ricardo Monascal
rmonascal@usb.ve

Rosseline Rodríguez
crodrig@usb.ve

Leonid Tineo
leonid@usb.ve

1 Escuela de Ingeniería Informática, Facultad de Ingeniería, Universidad de Valparaíso, Valparaiso, Chile

2 Departamento de Computación y Tecnología de la Información, Universidad Simón Bolívar, Caracas, Venezuela

Published online: 10 November 2021

which is associated with a possibility degree. Thus, it results in a fuzzy ordering. If this degree is zero, then the ordering is impossible.

An efficient approach for ordering the fuzzy numbers is by the use of a ranking function $R : F(\mathbb{R}) \to \mathbb{R}$, where $F(\mathbb{R})$ is a set of fuzzy numbers defined on real line, which maps each fuzzy number into the real line, where a natural order exists (Kumar et al. 2011). This kind of method is called *indirect* order. The functions mostly used are the centroid, area, distance, etc. (Wang et al. 2006; Rao and Shankar 2011; Azman and Abdullah 2012; Ganesh and Suresh 2017). Most of these methods require membership functions of the fuzzy numbers to be ranked.

The variety of orderings proposed is due to the fact that there is not a method that gives a satisfactory result to all proposed ordering problems (Singh 2015). Most of the methods presented so far are only partial orderings; other methods do not agree with human intuition, and others cannot compare fuzzy numbers with precise numbers. In (Vincent et al. 2017) is presented a good resume of several methods for ranking fuzzy numbers. The reader can refer to the monograph Basar (2012) for the sequences of fuzzy numbers and related topics.

In this work, we present an ordering based on the Zadeh extension principle. This ordering is shown to be a total ordering. Additionally, we have analyzed various study cases to observe the suitability of the ordering relative to intuition. Also, an application for a database problem implemented in Haskel and PostgreSQL is presented.

This work is structured as follows: Sect. 2 presents the related works for proposed orderings that have been previously carried out. In Sect. 3, the theoretical concepts that sustain this work are exposed. In Sect. 4, the ordering based on the Zadeh extension principle and the proof of its properties are shown. In Sect. 5, we present the study case for the intuition analysis. In Sect. 6, a practical implementation experience in Haskell and PostgreSQL is shown. Finally, in Sect. 8, conclusions and future work are discussed.

2 Related works

In recent decades, numerous methods have been proposed for ordering fuzzy numbers (Singh 2015). In (Figueroa-García et al. 2018), it is pointed out that the methods for ranking a special type, very often used, of fuzzy numbers are mostly iterative procedures, but real-world problems (control and combinatorial problems, micro-controller design and implementation) cannot use iterative methods, so a new ranking method of this type of fuzzy number is shown, based on the Yager index rank.

In (Ziemba 2018), the authors relate how the (ranking) fuzzy numbers are used in MCDM (Multiple Criteria Decision Making) methods. Also some implementation is shown.

In (Seiti et al. 2019) the authors show risk modeling of fuzzy numbers and present a new modeling approach, which can be used either to explain or to justify the errors and risks associated with fuzzy numbers that are used in decision-making problems. Such a modeling is called *R*-numbers. In soft-computing and in the study of large size problems the fuzzy set theory (Biswas 2016) is also very important.

Yuan (1991) argues that there is no single ordering method that meets all needs. Singh (2015) presented a summary with some methods and emphasizes that it is not enough to consider a single criterion for ordering fuzzy numbers.

There are several classifications of fuzzy number ordering procedures. These include Mitchell and Schaefer's proposal (Mitchell and Schaefer 2000) that extends the idea of direct and indirect ordering applied to precise numbers in statistics. *Direct order* is the result of ordering a set according to its own values. *Indirect order* is the result of ordering a set using to another ordered set, usually, a subset of real numbers.

Within the indirect orderings, in (Nasseri et al. 2012), a method for ordering fuzzy numbers is presented. This method consists of assigning each fuzzy number a real value and then ordering them according to these numbers. Another indirect method is proposed by Yager (1980) which defines a maximum, then orders according to the Hamming distance to the maximum. The greater number will be that with a smaller distance. Kerre (1982) redefines the maximum of Yager's method to adapt it to more cases.

Rao and Shankar (2011) established an interesting procedure that generates an indirect ordering. Given a trapezoidal fuzzy number, first they divide it into two triangles and a rectangle. Second, the centroid of each figure is calculated. Third, the circumcenter of the triangle formed by the three centroids is found. Finally, the order is calculated according to the abscissa of the circumcenter. This method is easy to implement and can even be used with precise numbers if the same number is taken as the centroid. A disadvantage is that it applies only to fuzzy numbers defined in a trapezoidal form whereby the method cannot be used on fuzzy numbers defined by extension.

The use of the centroid can be extended to an indirect order with the calculation of the abscissa of fuzzy numbers centroid that is applicable to both trapezoidal and by extension definitions. That, consider the trapezoidal fuzzy number as a planar piece of an uniform material, e.g., iron, and considering a fuzzy number given by extension as a set of iron balls placed in the corresponding value in a long iron bar. In both cases, the centroid corresponds to the equilibrium point of the system. After that, the fuzzy numbers are compared by the result of the comparison of the corresponding abscissa.

The advantage over the previous method (Rao and Shankar 2011) is that it allows comparing combinations of trapezoidal definitions with definitions by extension of fuzzy numbers, as well as with precise numbers.

In (Kerre et al. 2000) is presented an ordering method based on Zadeh extension principle (Zadeh 1975b) using α-cuts. The method here presented also uses the Zadeh extension principle, but in other way been more easy to apply.

After the exposition of these proposals for fuzzy number ordering, we conclude that the choice of which is best depending on the application domain where it will be used. For the particular case of extended DBMS to allow attributes in databases whose values can be fuzzy numbers, the rule is the user's preference. Generally, this preference is given by the user's perception of the situation observed, which can be understood as intuition. For this reason, we analyze a method that uses a measure of intuition to compare two fuzzy numbers.

3 Theoretical framework

In this section, the main concepts that support this work are defined.

3.1 Fuzzy sets

Fuzzy sets (Zadeh 1965) are characterized by a membership function whose range is in the real interval [0, 1]. The closer to 1 the membership degree of an element is, the more possibly (or more accurately) it is included in the set. Thus, 0 is the measure of complete exclusion and 1 is the measure of complete inclusion.

The set of partially included elements is known as the *border* of the fuzzy set F, formally

$$\text{border}(F) = \{x \in X \mid 0 < \mu_F(x) < 1\}.$$

The set of completely included elements is called the *core*,

$$\text{core}(F) = \{x \in X \mid \mu_F(x) = 1\}.$$

The set of not completely excluded elements is make up the support, that is,

$$\text{support}(F) = \{x \in X \mid \mu_F(x) > 0\}.$$

It is said that an element has total membership when

$$\mu_F(x) = 1.$$

The membership function of a fuzzy set can be represented in different ways. In the case of an ordered numerical universe, the most usual representation is the trapezoidal one, specified with a quadruple (x_1, x_2, x_3, x_4) of ordered elements of the domain ($x_1 \leq x_2 \leq x_3 \leq x_4$). This quadruple defines the vertices of trapezoid $(x_1, 0)$, $(x_2, 1)$, $(x_3, 1)$, $(x_4, 0)$. In discrete cases, it is also usual to define the function F by extension

$$F = \{\mu_F(x_1)/x_1, \mu_F(x_2)/x_2, \ldots, \mu_F(x_n)/x_n\}$$

with $\{x_1, x_2, \ldots, x_n\} \subset X$ and $\mu_F(x) = 0, \forall x \in X \setminus \{x_1, x_2, \ldots, x_n\}$.

3.2 Possibility distributions

Based on the fuzzy sets concept, Zadeh (1965) defined *possibility distributions*. A possibility distribution π_V represents a restriction on a fuzzy variable V. This restriction is determined by the fuzzy set F of possible values of V, thus $V = \mu_F$. For unifying notions, Prade and Testemale (1989) apply the concept of possibility distributions to represent precise and imprecise values as well as nulls in databases.

3.3 Fuzzy number

The concept of fuzzy number was initially introduced by Zadeh (1975a) to manipulate approximate numerical values. Then, Dubois and Prade (1987) refined the concept. A fuzzy number is defined as a fuzzy subset of the real numbers such that its membership function is convex, semicontinuous superior and its support is limited (Galindo et al. 2006). Therefore, a fuzzy number is a particular case of a possibility distribution.

Fuzzy numbers can be represented in different ways. In this work, the two most common forms will be used: definition by extension and trapezoidal definition. In the case where a fuzzy number approximates a whole value, a definition by extension can be used. A fuzzy number A defined by extension is represented as a set of pairs of the form $A = \{\mu_A(x_1)/x_1, \mu_A(x_2)/x_2, \ldots, \mu_A(x_n)/x_n\}$.

A widely used fuzzy number representation is the trapezoidal membership function. In this case, membership is specified with a quatern (x_1, x_2, x_3, x_4) of elements of the $\mathbb{R}$ domain, ordered by $x_1 \leq x_2 \leq x_3 \leq x_4$ that define the vertices $(x_1, 0)$, $(x_2, 1)$, $(x_3, 1)$, $(x_4, 0)$ of a trapezoid, as can be seen in Fig. 1. Fuzzy numbers can be calculated by applying the extension principle.

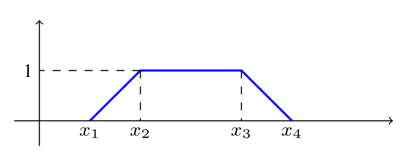


Fig. 1 Trapezoidal shape

3.4 Extension principle

Zadeh (1975a) postulated the generalized extension principle: Given an application $f : X_1 \times X_2 \times \ldots \times X_n \to Y$ and the fuzzy sets $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ where each $\mathcal{A}_i$ is defined on X_i, the application f allows definition of the fuzzy set $B = f(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n) \subseteq Y$ with the membership function

$$\mu_B(y) = \bigvee_{y=f(x_1,x_2,\ldots,x_n)} (\mu_{A_1}(x_1) \wedge \ldots \wedge \mu_{A_n}(x_n)).$$

where $\vee$ y $\wedge$ are triangular conorms, usually *maximum* and *minimum*, respectively.

3.5 Order

Order appears in all areas related to mathematics and computer science. Given a set in many cases it is desirable to establish a precedence relation between the elements of the set either total or partial (Rosen 2012).

In this sense, we can distinguish between two types of ordered sets. The partially ordered sets are those in which a precedence relation can be established between some of its elements. The totally ordered sets are those where a precedence relation can be established for all elements.

Definition 1 Given a set A and the relation R over A, the pair $\langle A, R\rangle$ is called a *partially ordered set* if R is a relation with the properties

– *Reflexive* $(\forall a \in A)\ aRa$
– *Antisymmetric* $(\forall a, b \in A)\ aRb \wedge bRa \Rightarrow a = b$
– *Transitive* $(\forall a, b, c \in A)\ aRb \wedge bRc \Rightarrow aRc$

A *totally ordered set* A is a partially ordered set in which all pair of elements x, y in A are comparables, i.e., $xRy \vee yRx$.

3.6 The ordering method based on the centroid

Many structures and mechanical systems act as if their masses were concentrated at a single point, called *the center of mass*. The location of this point depends on two main factors, the shape of the structure and the material with which it is built. In the case that the structure has a constant mass density (it is made of a single material) this center of mass is called *the centroid* (Thomas and Weir 2010).

Some methods studied are based on the centroid of the fuzzy numbers. There are several ways of using the centroid to order. We will study the following method:

– For an extension fuzzy number:

$$A = \{\mu_1/x_1, \mu_2/x_2, \ldots, \mu_n/x_n\}$$

it can be think as if there were masses $\mu_1, \mu_2, \ldots, \mu_n$, on a rigid horizontal axis x with a fulcrum at the origin and if $x_1, x_2, \ldots, x_n$ represent its location on the real line, then the centroid abscissa is given by (Thomas and Weir 2010):

$$\overline{x}_A = \frac{M}{m}$$

where M is the horizontal momentum, m is the mass of the system and are calculated with

$$M = \sum_{k=1}^{n} x_k \mu_k \qquad \text{and} \qquad m = \sum_{k=1}^{n} m_k.$$

– For a trapezoidal fuzzy number $A = (x_1, x_2, x_3, x_4)$, with membership function μ_A given by

$$\mu_A(x) = \begin{cases} 0, & \text{if } x < x_1 \\ (x - x_1)/(x_2 - x_1), & \text{if } x_1 \leq x < x_2 \\ 1, & \text{if } x_2 \leq x \leq x_3 \\ (x_4 - x)/(x_4 - x_3), & \text{if } x_3 < x \leq x_4 \\ 0, & \text{if } x_4 < x \end{cases}$$

it can be think as flat sheet of uniform material with trapezoidal shape, then, the centroid abscissa $\overline{x}$ is given by

$$\overline{x} = \frac{1}{3}\left(x_1 + x_2 + x_3 + x_4 + \frac{x_1x_2 - x_3x_4}{x_4 - x_1 + x_3 - x_2}\right).$$

Then, the order of the given fuzzy numbers is the order, in the real axe, of the corresponding abscissa. i.e., given two fuzzy numbers A and B with centroids $\overline{x}_A$ and $\overline{x}_B$, respectively, is defined

$$A \leq_c B \text{ if only if } \overline{x}_A \leq \overline{x}_B \tag{1}$$

We will compare the result based on this method and the result obtained with the new method here presented.

4 An order based on the Zadeh extension principle

Since possibility distributions define the fuzzy set of the possible values of a datum, the extension principle (Zadeh 1975b) is applicable for functions over possibilistic fuzzy data. In the formula of the extension principle, we changed the notation of membership function to the possibility distribution, and the fuzzy sets A_i by possibilistic data a_i, getting the following formula

$$\pi_{f(a_1,\ldots,a_n)}(y) = \bigvee_{y=f(x_1,\ldots,x_n)} (\pi_{a_1}(x_1) \wedge \cdots \wedge \pi_{a_n}(x_n)).$$

We calculate the application of the extension principle for the case where the function f takes exactly two arguments in the ordered universe X, i.e., provided with a total order relation $<$. The result produced will be a Boolean value $Y \in \{\text{true}, \text{false}\}$. In symbols

$$f : X \times X \to Y$$
$$f(x_1, x_2) = \begin{cases} \text{true}, & \text{if } x_1 < x_2 \\ \text{false}, & \text{in the other case} \end{cases}$$

Applying the extension principle, taking $\wedge$ as the minimum and $\vee$ the maximum, the following possibility distribution is defined:

$$\pi_{(a_1 < a_2)} = \left\{ \bigvee_{\substack{f(x_1,x_2)=y \\ x_1,x_2 \in X}} \left(\pi_{a_1}(x_1) \wedge \pi_{a_2}(x_2)\right)/y : y \in Y \right\}$$
$$= \left\{ \bigvee_{\substack{x_1 < x_2 \\ x_1,x_2 \in X}} \left(\pi_{a_1}(x_1) \wedge \pi_{a_2}(x_2)\right)/\text{true}, \right.$$
$$\left. \bigvee_{\substack{x_1 \geq x_2 \\ x_1,x_2 \in X}} \left(\pi_{a_1}(x_1) \wedge \pi_{a_2}(x_2)\right)/\text{false} \right\}$$

With this possibility distribution, the operators *possibly minor*, $<_p$ and *possibly equal*, $=_p$, are defined as:

$$a <_p b \leftrightarrow \pi_{a<b}(\text{true}) > \pi_{b<a}(\text{true})$$
$$a =_p b \leftrightarrow \pi_{a<b}(\text{true}) = \pi_{b<a}(\text{true}).$$

Together with these two definitions, the possible operator less or equal is defined as

$$a \leq_p b \leftrightarrow a <_p b \vee a =_p b.$$

In an equivalent way:

$$a \leq_p b \leftrightarrow \pi_{a<b}(\text{true}) \geq \pi_{b<a}(\text{true}) \quad (2)$$
$$a =_p b \leftrightarrow a \leq_P b \text{ and } b \leq_P b. \quad (3)$$

From this definition, and the definition of fuzzy number, it can be proved that, when an inequality is satisfied, for instance $a <_P b$, it is because

$$\max\{x \in \text{core}(a)\} < \min\{y \in \text{core}(b)\}.$$

This will be proved in the following proposition.

Theorem 1 *Let a and b be two fuzzy numbers, then $a <_P b$ if only if*

$$\max\{x \in \text{core}(a)\} < \min\{y \in \text{core}(b)\}.$$

Even more, if there exist $x, y \in \text{core}(a) \cap \text{core}(b)$ with $x \neq y$ then $a =_P b$.

Proof ($\Rightarrow$) By counter reciprocus, suppose

$$\max\{x \in \text{core}(a)\} \geq \min\{y \in \text{core}(b)\}.$$

Let us see that $b \leq_P a$. The hypothesis implies there exists $x_1 \in \text{core}(a)$, $y_1 \in \text{core}(b)$ with $y_1 \leq x_1$. By core definition, $\pi_a(x_1) = 1$ and $\pi_b(y_1) = 1$. Then

$$\pi_a(x_1) \wedge \pi_b(y_1) = 1.$$

As $y_1 < x_1$ and 1 is the maximum possible value of π, the following holds

$$\pi_{b<a}(\text{true}) = \bigvee_{y<x} (\pi_b(y) \wedge \pi_a(x))$$
$$= \pi_a(x_1) \wedge \pi_b(y_1) = 1$$

Thus $b \leq_P a$.
($\Leftarrow$) Suppose

$$\max\{x \in \text{core}(a)\} < \min\{y \in \text{core}(b)\}.$$

Let $\overline{x} = \max\{x \in \text{core}(a)\}$ and $\overline{y} = \min\{y \in \text{core}(b)\}$ Then, $\overline{x} < \overline{y}$ and

$$\pi_{a<b}(\text{true}) = \bigvee_{x<y} (\pi_a(x) \wedge \pi_b(y))$$
$$= \pi_a(\overline{x}) \wedge \pi_b(\overline{y}) = 1. \quad (4)$$

It remains to be shown that the inequality is strict. For this, we can demonstrate $\pi_{b<a}(\text{true}) < 1$. Calculating

$$\pi_{b<a}(\text{true}) = \bigvee_{y<x} (\pi_a(x) \wedge \pi_b(y)).$$

If there exist $x \in \text{support}(a)$ and $y \in \text{support}(b)$ with $y < x$, the condition $y < x$ implies that $\pi_a(x)$ and $\pi_b(y)$ cannot both be equal to 1, then

$$\pi_a(x) \wedge \pi_b(y) < 1$$
$$\Rightarrow \pi_{b<a}(\text{true}) < 1 \quad (5)$$

Hence, (4) and (5) imply $a <_P b$.

Suppose now there exist $x, y \in \text{core}(a) \cap \text{core}(b)$ with $x \neq y$. w.l.o.g. suppose $x < y$, then

$$\pi_a(x) \wedge \pi_b(y) = 1 \Rightarrow a \leq_P b$$
$$\pi_a(y) \wedge \pi_b(x) = 1 \Rightarrow b \leq_P a$$

And, in conclusion $a =_P b$. □

Below are proofs of some properties of the operators $<_P$ and $\leq_P$. First, the operator $\leq_P$ is reflexive and antisymmetric, and second, the operator $<_P$ is transitive; thus, it defines a strict order.

Proposition 1 *The operator $\leq_P$ is reflexive and antisymmetric, that is*

1. *If a is a fuzzy number, then $a \leq_P a$.*
2. *If a, b are two fuzzy numbers with $a \leq_P b$ and $b \leq_P a$, then $a =_P b$.*

Proof 1. $\leq_p$ is reflexive because $a \leq_p a$ is fulfilled, i.e., $a \leq_p a$ iff $a <_p a \vee a =_p a$ iff

$$(\pi_{a<a}(\text{true}) > \pi_{a<a}(\text{true}))$$
$$\vee$$
$$(\pi_{a<a}(\text{true}) = \pi_{a<a}(\text{true}))$$

which is true because $\pi_{a<a}(\text{true}) = \pi_{a<a}(\text{true})$ is true.

2. It is antisymmetric since

$$a \leq_p b \rightarrow \pi_{a<b}(\text{true}) \geq \pi_{b<a}(\text{true})$$
$$b \leq_p a \rightarrow \pi_{b<a}(\text{true}) \geq \pi_{a<b}(\text{true})$$

This implies

$$\pi_{a<b}(\text{true}) = \pi_{b<a}(\text{true})$$

by equation (3) $a =_p b$. □

Proposition 2 *The operator $<_P$ defines a strict order. That is, $<_P$ is non-reflexive and transitive. In other words,*

1. *For any fuzzy number, $a \not<_P a$.*
2. *If a, b and c are fuzzy numbers with $a <_P b$ and $b <_P c$ then $a <_P c$.*

Proof 1. It is clear, if a is a fuzzy number, $a =_P a$, this implies $a \not<_P a$.

2. By hypothesis and Theorem 1, the following holds

$$\max\{x \in \text{core}(a)\} < \min\{y \in \text{core}(b)\}$$
$$\max\{y \in \text{core}(b)\} < \min\{z \in \text{core}(c)\}$$

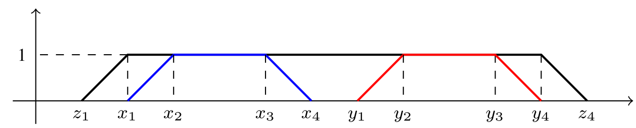


Fig. 2 Cases missing of transitivity of $<_P$

$$\Rightarrow$$
$$\max\{x \in \text{core}(a)\} < \min\{z \in \text{core}(c)\}$$

Again, by Theorem 1, $a <_P c$. □

The operator $(\leq_p)$ does not define an ordered relation (total or partial) because there are cases in which transitivity does not work. The cases are listed below.

1. $a =_P b$ and $b <_P c$.
2. $a <_P b$ and $b =_P c$.
3. $a =_P b$ and $b =_P c$.

The graph of the first case (or the second one) is presented in Fig. 2. In figure, a, b, c are fuzzy numbers with $a = (x_1, x_2, x_3, x_4)$, in blue, $b = (y_1, y_2, y_3, y_4)$, in red, and $c = (z_1, z_2, z_3, z_4)$, in black. Note that, according to Theorem 1, $a \leq_P b$, $b =_P c$ and $a =_P c$.

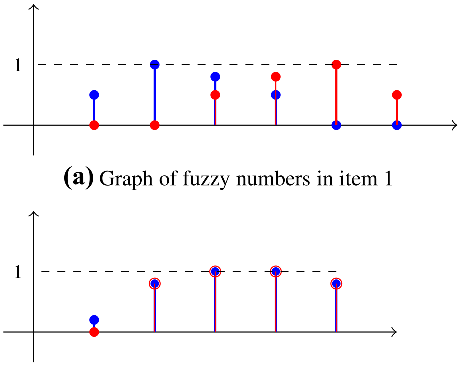
Is necessary to understand the concept of the *possible equality* to complete the demonstrations. Note that, for example, that $a =_P b$ does not imply $a = b$. Theorem 1 establishes that $a =_P b$ iff $\text{core}(a)$ and $\text{core}(b)$ intersect in more than one point. Then, if $a =_P b$, can happen any of the following situations

$$\text{core}(a) \subset \text{core}(b)$$
$$\text{core}(b) \subset \text{core}(a)$$
$$\text{core}(a) \not\subset \text{core}(b) \wedge \text{core}(b) \not\subset \text{core}(b).$$

Each of these cases needs to be studied in detail to prove the above list.

The operator $\leq_p$ almost defines a total order relationship for all combinations of fuzzy numbers pairs. Given two fuzzy numbers a and b, always we can calculate $\pi_{a<b}$ and $\pi_{b<a}$. By the Trichotomy principle of the reals, they are equal or one is greater than the other. But, if you try to compare three fuzzy numbers a, b and c, if, for instance, $a =_P b$, you must be very careful what is happening with c, which can possibly be equal to a, but possible less than b.

For the case in which the fuzzy numbers are given by extension, the order is calculated, verifying each pair of values x_1, x_2 such that they satisfy $x_1 < x_2$, where x_1 belongs to the set of possible values of the fuzzy number a and x_2 belongs to the set of possible values of the fuzzy number b. Theorem 1 reduces the pairs to compare. The uses of theorem are shown in Example 4.1.



(a) Graph of fuzzy numbers in item 1

(b) Graph of fuzzy numbers in item 2

Fig. 3 Graph of fuzzy numbers in Example 4.1

Example 4.1 In each case, Theorem 1 will be used to establish the order of the fuzzy numbers a (in blue) and b (in red), defined by extension for the given distribution. In all the cases, the expected result, intuitive, is $a <_P b$. The corresponding graphs are shown in Fig. 3.

1. Given

   $$a = \{0.5/1, 1/2, 0.8/3, 0.5/4\} \text{ and}$$
   $$b = \{0.3/3, 0.8/4, 1/5, 0.5/6\}$$

   whose graphics is shown in Fig. 3a. Applying Theorem 1:

   $$\text{core}(a) = \{2\} \qquad \text{core}(b) = \{5\}$$

   Then

   $$\max\{x \in \text{core}(a)\} = 2 < 5 = \min\{x \in \text{core}(b)\}$$

   Thus $a <_P b$ holds.
2. Given

   $$a = \{0.2/1, 0.8/2, 1/3, 1/4, 0.8/5\} \text{ and}$$
   $$b = \{0.8/2, 1/3, 1/4, 0.8/5\}$$

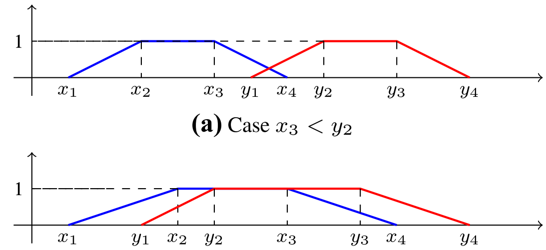
   whose graphics is shown in Fig. 3a. Applying Theorem 1:

   $$\text{core}(a) = \{3, 4\} \qquad \text{core}(b) = \{3, 4\}$$

   Then, as $3 \neq 4$ and

   $$3, 4 \in \text{core}(a) \cap \text{core}(b)$$

   Thus $a =_P b$ holds.



(a) Case $x_3 < y_2$

(b) Case $x_2 < y_2 < x_3$

Fig. 4 Figures of Corollary 1

For the case in which the fuzzy number is given in the trapezoidal form, the list of data pairs is not possible to enumerate. Therefore, the following corollary is used for cases that both trapezoids are not triangular, that is, a fuzzy number defined by (x_1, x_2, x_3, x_4) with $x_2 < x_3$. The cases $x_2 = x_3$ will be studied later.

Corollary 1 *Let a, b be two fuzzy numbers with trapezoidal definition $a = (x_1, x_2, x_3, x_4)$ and $b = (y_1, y_2, y_3, y_4)$, such us $x_2 < x_3$ and $y_2 < y_3$. Then, we have:*

1. *If $x_3 < y_2$, then $a <_P b$. Equivalently, if $y_3 < x_2$, then $b <_P a$.*
2. *If $x_2 < y_2 < x_3$, then $a =_P b$.*

Proof Before begin the demonstration, note that

$$\text{core}(a) = [x_2, x_3]$$
$$\text{core}(b) = [y_2, y_3].$$

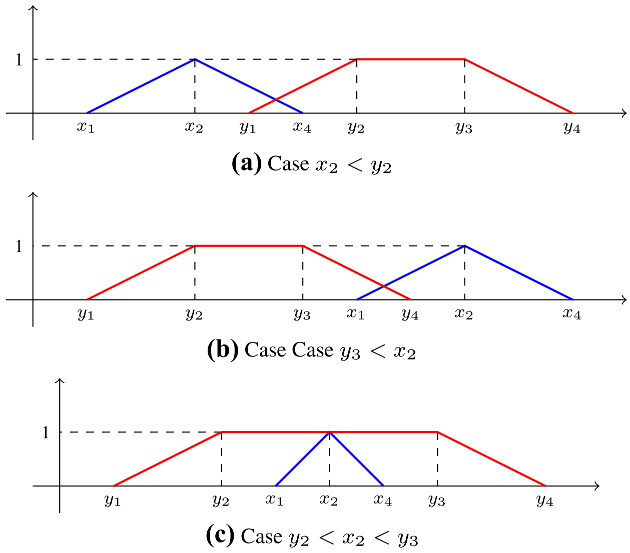
1. If $x_3 < y_2$, as shown in Fig. 4a, then

   $$\max\{x \in \text{core}(a)\} = x_3 < y_2 = \min\{y \in \text{core}(b)\}.$$

   By Theorem 1, $a <_P b$. The equivalent condition is obtained by changing the names of the vertex set x_i by y_i and applying this case.
2. If $x_2 < y_2 < x_3$, the situation is shown in Fig. 4b. Then note that $y_2 \in \text{core}(a) \cap \text{core}(b)$ and if x_3 is not, there exists $y \in (y_2, x_3)$ in the intersection. So, by Theorem 1, $a =_p b$. Complete the demonstration. □

Summarizing, it is shown that if both trapezoids intersect in the horizontal line (the core), the fuzzy numbers are possible equal; in the other case, the fuzzy number corresponding to the trapezoid on the left side will be less.

The next corollary presents the case in which at least one of the fuzzy numbers is triangular. w.l.o.g., a is assumed triangular. The fuzzy number b can be triangular or not.



(a) Case $x_2 < y_2$

(b) Case Case $y_3 < x_2$

(c) Case $y_2 < x_2 < y_3$

Fig. 5 Figures of Corollary 2

Corollary 2 *Let a, b be two fuzzy numbers with trapezoidal definition* $a = (x_1, x_2, x_2, x_4)$ *and* $b = (y_1, y_2, y_3, y_4)$. *Then*

1. *If* $x_2 < y_2$ *then* $a <_P b$.
2. *If* $y_3 < x_2$ *then* $b <_P a$.
3. *If* $y_2 < x_2 < y_3$ *then* $a =_P b$.

Proof Note that $\text{core}(a) = \{x_2\}$ and $\text{core}(b) = [y_2, y_3]$.

1. If $x_2 < y_2$ then $\max\{x \in \text{core}(a)\} = x_2 < y_2 = \min\{y \in \text{core}(b)\}$. By Theorem 1 $a <_P b$. This situation is shown in Fig. 5a.
2. If $y_3 < x_2$ then $\max\{y \in \text{core}(b)\} = y_3 < x_2 = \min\{x \in \text{core}(a)\}$. By Theorem 1 $b <_P a$. This case is graphed in Fig. 5b.
3. If $y_2 < x_2 < y_3$ then we cannot apply Theorem 1 because

   $$\text{core}(a) \cap \text{core}(b) = \{x_2\}.$$

   Note that $\pi_b(y_2) \wedge \pi_a(x_2) = 1$ and $\pi_a(x_2) \wedge \pi_b(y_3) = 1$ and hence

   $$\pi_{a<b}(\text{true}) = \pi_{b<a}(\text{true}) = 1.$$

   Conclude $a =_P b$. This case is shown in Fig. 5c. □

Note that Conditions 1 and 2 on the above corollary can be applied even if b is a triangular fuzzy number ($y_2 = y_3$). Condition 3 requires $y_2 < y_3$. Then, one remaining case is $y_2 = y_3 = x_2$. This case, graphed in Fig. 6, cannot be studied as a general case because the result depends of the slope of the sides of the triangles. Others remaining cases, with $y_2 < y_3$, are $x_2 = y_2$ and $x_2 = y_3$. These cases are studied in the next proposition. In these cases, Theorem 1 cannot be applied because the intersection of both cores is $\{x_2\}$.

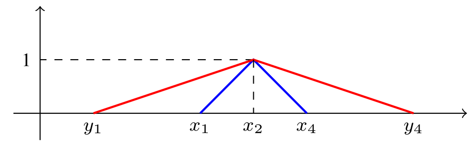
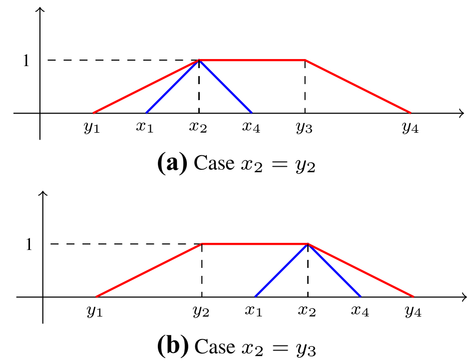


Fig. 6 The remain case $y_2 = y_3 = x_2$



(a) Case $x_2 = y_2$

(b) Case $x_2 = y_3$

Fig. 7 Cases studied in proposition 3

Proposition 3 *Let* a, b *be two trapezoidal fuzzy numbers, with a triangular, that is*

$$a = (x_1, x_2, x_2, x_4), \quad b = (y_1, y_2, y_3, y_4)$$

and $y_2 < y_3$. *Then*

1. *If* $x_2 = y_2$ *then* $a \leq_P b$.
2. *If* $x_2 = y_3$ *then* $b \leq_P a$.

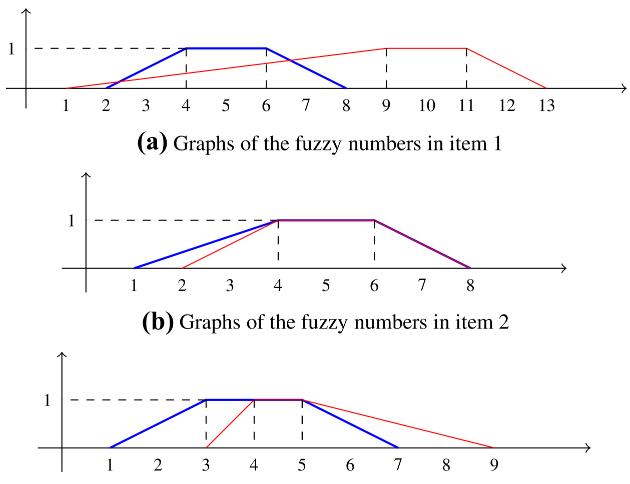
Proof 1. If $x_2 = y_2$ we have $x_2 < y_3$ and $\pi_a(x_2) \wedge \pi_b(y_3) = 1$, hence $a \leq_P b$.
2. If $x_2 = y_3$ we have $y_2 < x_2$ and $\pi_b(y_2) \wedge \pi_b(x_2) = 1$, hence $b \leq_P a$.
   Both cases are shown in Fig. 7. □

Let us see an example where these corollaries apply.

Example 4.2 In each case, Corollary 1 will be used to establish the order of the fuzzy numbers a (in blue) and b (in red), when the corresponding trapezoidal functions are given. It is noteworthy that, in the three cases shown, it is expected to be shown as an intuitive result, $a <_p b$. The corresponding graphs are shown in Fig. 8.

1. Given $a = (2, 4, 6, 8)$ and $b = (1, 9, 11,\ 13)$, whose graph is shown in Fig. 8a. Applying Corollary 1

   $$6 = x_3 < y_2 = 9 \Rightarrow a <_p b$$



(a) Graphs of the fuzzy numbers in item 1

(b) Graphs of the fuzzy numbers in item 2

(c) Graphs of the fuzzy numbers in item 3

Fig. 8 Graphs of the fuzzy numbers in Example 4.2

2. $a = \{1, 4, 6, 8\}$ y $b = \{2, 4, 6, 8\}$, whose graph is shown in Fig. 8b. Applying Corollary 1

$$4 = y_2 \leq x_3 = 6 \Rightarrow a =_p b$$

3. $a = \{2, 4, 6, 8\}$ y $b = \{4, 5, 6, 10\}$, whose graph is shown in Fig. 8c. Applying Corollary 1

$$5 = y_2 \leq x_3 = 6 \Rightarrow a =_p b$$

When comparing the graphs of two fuzzy numbers, we can take an intuitive measure to say which is less than the other. If the graph of the first number is displaced to the left of the second, we can conclude that the first number is smaller than the second. In the examples presented, it is then observed that, with the ordering based on Zadeh's extension principle, the expected intuitive result is not always obtained.

5 Study cases

As mentioned above, the user's preference will be taken as a rule. This preference is usually related to intuition, that is, the perception that the user has of the situation that he observes. In our case, we are looking at two fuzzy numbers to compare and decide which is smaller than the other and thus establish an order.

In this sense, we considered it pertinent to suggest some representative cases of possible comparisons between fuzzy numbers pairs. Before applying the order relation, we decided which is the smallest number of each pair based on intuition. Then, we proceeded to calculate the order using the method based on the Zadeh extension principle. In this way, we could evaluate the method to show whether or not this proposal is meaningful in terms of intuition.

The cases show examples where only trapezoidal fuzzy numbers are compared. Then, other examples compare only fuzzy numbers defined by extension. Finally, some cases with combinations of both types of definitions are presented. In all the cases showed, the user's preference assumed is that the fuzzy number a (blue) is possibly less than the fuzzy number b (red). That is, $a <_p b$.

Thus, the examples were organized into three groups: ten cases of comparisons between trapezoidal fuzzy numbers, nine cases of comparisons between fuzzy numbers by extension and six cases of mixed comparisons.

5.1 Examples with trapezoidal representation

In Table 1, the nine pairs of fuzzy numbers with trapezoidal definition to be compared are shown. The assumed intuition indicates that the fuzzy number A (in blue) will be less than the fuzzy number B (in red). When applying the ordering method based on the extension principle, the results shown in the same table were obtained. We observe that in case 6 the method fails. Cases 2, 3, 5 and 7 show that the compared numbers are possibly equal. The other cases give the expected result. These results could indicate that the intuition suggested by users who were consulted is not adequate for some cases. Another alternative could be that the method does not completely adjust to user preferences, so another method must be explored.

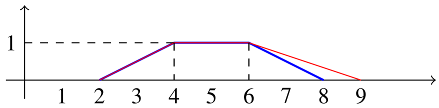
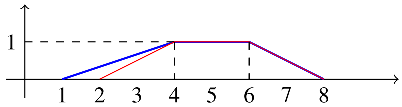
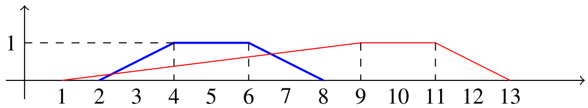
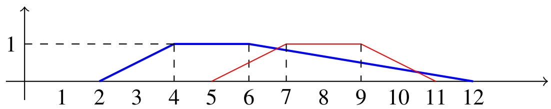
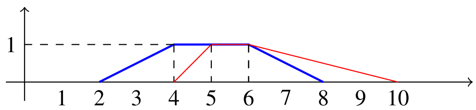
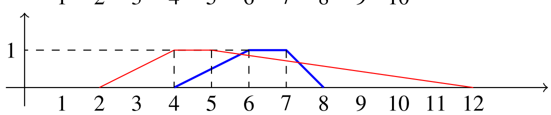
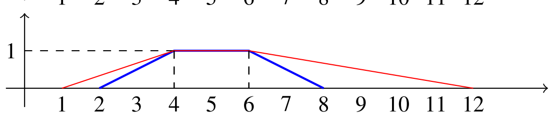
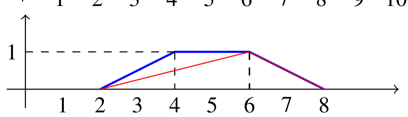
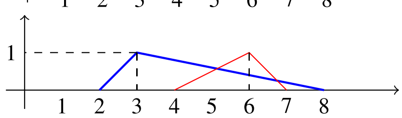
5.2 Examples with representation by extension

In Table 2, the nine pairs of fuzzy numbers with trapezoidal definition to be compared are shown. The assumed intuition indicates that the fuzzy number A (in blue) will be less than the fuzzy number B (in red). When applying the ordering method based on the extension principle, the results shown in columns fourth, fifth and sixth of the same table were obtained. We observe that in case 5 the method fails. Cases 3, 6 and 9 show that the compared numbers are possibly equal. The other cases give the expected result. Again, these results could indicate that the intuition suggested by the users consulted is not adequate for some cases. Another alternative could be the method does not completely adjust to the user preference, so another method must be explored.

5.3 Examples with mixed representation

In this section, we make a comparison between five pairs of fuzzy numbers. One of the fuzzy numbers is given by extension, and the other has a trapezoidal definition. In Table 3, the six pairs of fuzzy numbers to be compared are shown. The user preference in all cases is $A <_p B$. When applying

Table 1 Pairs of fuzzy numbers with trapezoidal definition to be compared

No.	FN	Graph	$\pi(\mathbf{a} < \mathbf{b})$	$\pi(\mathbf{b} < \mathbf{a})$	Conclusion
1	$a = (2, 4, 6, 8)$, $b = (2, 4, 6, 9)$		1	0	$a <_P b$
2	$a = (1, 4, 6, 8)$, $b = (2, 4, 6, 8)$		1	1	$a =_P b$
3	$a = (2, 4, 6, 8)$, $b = (1, 9, 11, 13)$		1	1	$a =_P b$
4	$a = (2, 4, 6, 12)$, $b = (5, 7, 9, 11)$		1	0.66	$a <_P b$
5	$a = (2, 4, 6, 8)$, $b = (4, 5, 6, 10)$		1	1	$a =_P b$
6	$a = (4, 6, 7, 8)$, $b = (2, 4, 5, 12)$		0.99	1	$b <_P a$
7	$a = (2, 4, 6, 8)$, $b = (1, 4, 6, 12)$		1	1	$a =_P b$
8	$a = (2, 4, 6, 8)$, $b = (2, 6, 6, 8)$		1	0.75	$a <_P b$
9	$a = (2, 3, 3, 8)$, $b = (4, 6, 6, 7)$		1	0.39	$a <_P b$

the ordering method based on the Zadeh Principle to pairs in this table, the results shown in columns fourth, fifth and sixth of the same table were obtained. We observe that in cases 2, 4 and 5 the compared numbers are possibly equal. The other cases give the expected result. Again, these results could indicate that the intuition suggested by the users consulted is not adequate for some cases. Another alternative could be that the method does not completely adjust to user preferences, so another method must be explored.

6 Practical experience of implementation

The ordering proposal was implemented in a first functional version using the Haskell programming language and later in the PostgreSQL Database Management System. In this section, parts of both experiences will be reported.

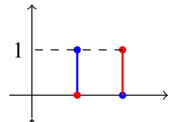
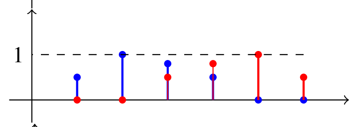
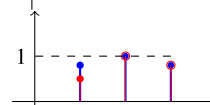
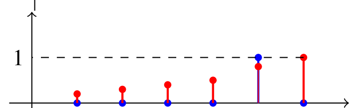
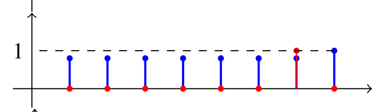
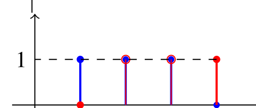
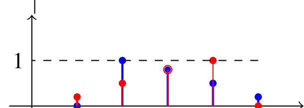
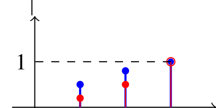
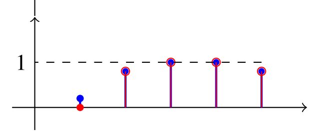
6.1 Haskell version

The first version was intended to test the definition of the ordering $\leq_C$, given in (1) on the set of study cases raised in the previous section. In addition, we wanted to verify whether the ordering complied with the intuition established in the user's preferences. This could be observed in the results obtained when executing the presented study cases in Haskell.

Haskell programming language was selected because it is very easy to use. As well as, it is very appropriate for mathematical definitions because of its functional paradigm. Below are some details of this implementation.

In the first place, the types associated with fuzzy numbers were defined: `Trapezoid`, `Extension` and `FuzzyNum`. The first represents the trapezoidal definition of a fuzzy number through a tuple with the four inflection points of the trapezoid. A fuzzy number by extension is defined as a list of pairs (**Double, Double**) where the first element corresponds

Table 2 Pairs of fuzzy numbers with definition by extension to be compared

No.	FN	Graph	$\pi(\mathbf{a} < \mathbf{b})$	$\pi(\mathbf{b} < \mathbf{a})$	Conclusion
1	$a = \{1/1\}$, $b = \{1/2\}$		1	0	$a <_P b$
2	$a = \{0.5/1, 1/2, 0.8/3, 0.5/4\}$, $b = \{0.5/3, 0.8/4, 1/5, 0.5/6\}$		1	0.5	$a <_P b$
3	$a = \{0.8/1, 1/2, 0.8/3\}$, $b = \{0.5/1, 1/2, 0.8/3\}$		0.8	0.8	$a =_P b$
4	$a = \{1/5\}$, $b = \{0.2/1, 0.3/2, 0.4/3, 0.5/4, 0.8/5, 1/6\}$		1	0.5	$a <_P b$
5	$a = \{0.8/1, 0.8/2, 0.8/3, 0.8/4, 0.8/5, 0.8/6, 0.8/7, 1/8\}$, $b = \{1/7\}$		0.8	1	$b <_P a$
6	$a = \{1/1, 1/2, 1/3\}$, $b = \{1/2, 1/3, 1/4\}$		1	1	$a =_P b$
7	$a = \{1/2, 0.8/3, 0.5/4, 0.2/5\}$, $b = \{0.2/1, 0.5/2, 0.8/3, 1/4\}$		1	0.5	$a <_P b$
8	$a = \{0.5/1, 0.8/2, 1/3\}$, $b = \{0.2/1, 0.5/2, 1/3\}$		0.8	0.5	$a <_P b$
9	$a = \{0.2/1, 0.8/2, 1/3, 1/4, 0.8/5\}$, $b = \{0.8/2, 1/3, 1/4, 0.8/5\}$		1	1	$a =_P b$

to the membership function and the second corresponds to a possible value of the number. `FuzzyNum` represents that a fuzzy number can be defined in two forms: trapezoid and by extension. The definitions in Haskell are

```
type Trapezoid = (Double,Double,Double,Double)
-- a trapezoid is a tuple of four inflection points

type Extension = [(Double,Double)]
-- a fuzzy number by extension is a set of pairs (membership
   , value)

type FuzzyNum = Either Trapecio Extension
-- a fuzzy number is a trapezoid or given by extension
```

For example, if you want to define the trapezoid (2, 4, 6, 8), the following syntax would be used:

Left (2.0, 4.0, 6.0, 8.0).

While to define the fuzzy number by extension {0.5/1, 1/2, 0.8/3, 0.5/4} the syntax would be used:

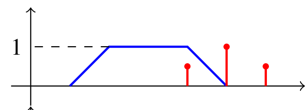
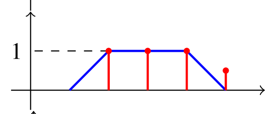
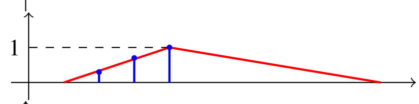
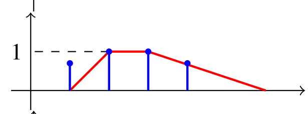
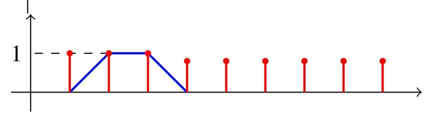
Right [(0.5,1.0), (1.0,2.0), (0.8,3.0), (0.5,4.0)].

The reserved words **Right** and **Left** indicate the type of definition that was chosen within the union based on the position it occupies: on the left for the trapezoids and on the right for the definitions by extension. Then, the comparison operator was implemented based on the extension principle.

Two important functions for the ordering calculation are defined below. The `pLess` function calculates the possibility that fuzzy number A is less than B, where A and B are defined by extension, i. e., $\pi_{a<b}(\text{true})$. For this, the extension principle is used.

```
pLess :: Extension -> Extension ->Double
pLess a b = maximum (0:[ ax `min` bx | ( ax , ay ) <- a ,
     ( bx , by ) <- b , ay < by ])
```

Table 3 Pairs of fuzzy numbers in mixed formats

No.	FN	Graph	$\pi(\mathbf{a} < \mathbf{b})$	$\pi(\mathbf{b} < \mathbf{a})$	Conclusion
1	$a = (1, 2, 4, 5)$, $b = \{0.5/4,\ 1/5,\ 0.5/6\}$		1	0	$a <_P b$
2	$a = (1, 2, 4, 5)$, $b = \{1/2,\ 1/3,\ 1/4,\ 0.5/5\}$		1	1	$a =_P b$
3	$a = \{0.5/2,\ 0.8/3,\ 1/4\}$, $b = (1, 4, 4, 10)$		0.83	0.66	$a <_P b$
4	$a = \{0.8/1,\ 1/2,\ 1/3,\ 0.8/4\}$, $b = (1, 2, 3, 6)$		1	1	$a =_P b$
5	$a = (1, 2, 3, 4)$, $b = \{1/1,\ 1/2,\ 1/3,\ 0.8/4, 0.8/5,$ $0.8/6,\ 0.8/7,\ 0.8/8,\ 0.8/9\}$		1	1	$a =_P b$

The `possibleLess` function decides if the first fuzzy number is possibly less than the second fuzzy number. When both numbers have a trapezoidal definition, Lemma 1 is used. If both numbers are given by extension, the pLess function is used to calculate the value of the possibilities $\pi_{a<b}(true)$ and $\pi_{b<a}(true)$. If the first value is greater than the second, it is concluded that $a <_P b$.

When you have a number defined by extension and the other in trapezoidal form, the latter approximates a definition by extension through the `trapToexp` function. This function takes each line of the trapezoid and gets 10 equidistant points from this line along with the value of the membership function for each point. In this way, a set of 30 *membership / value* pairs is obtained, which is represented by extension the trapezoid. Using the `pLess` function, the number given by extension and the fuzzy number by extension obtained from the `trapToext` function are compared. In this way, the value of the possibilities $\pi_{a<b}$(true) and $\pi_{b<a}$(true) is calculated for to decide if $a <_P b$.

```
possibleLess :: FuzzyNum -> FuzzyNum ->Bool
possibleLess (Left (x1,x2,x3,x4)) (Left (y1,y2,y3,y4)) = (x3
      < y2)
possibleLess (Right a) (Right b) = (pLess a b) > (pLess b a)
possibleLess (Left a) (Right b) = (pLess (trapToext a) b) >
      (pLess b (trapToext a))
possibleLess (Right a) (Left b) = (pLess a (trapToext b)) >
      (pLess (trapToext b) a)
```

A similar process is used for the "possibly equal" operator implementation through the `possibleEqual` function.

```
possibleEqual :: FuzzyNum -> FuzzyNum ->Bool
possibleEqual (Left (x1,x2,x3,x4)) (Left (y1,y2,y3,y4)) = (
      x2 <= y2 && y2 <= x3)
possibleEqual (Right a) (Right b) = (pLess a b) == (pLess b
      a)
possibleEqual (Left a) (Right b) = (pLess (trapToext a) b)
      == (pLess b (trapToext a))
possibleEqual (Right a) (Left b) = (pLess a (trapToext b))
      == (pLess (trapToext b) a)
```

The functions used to represent a trapezoid with a set of membership/value pairs are listed below. `discLine1`, `discLine2` and `discLine3` get 10 points from each line of the trapezoid. `myround` rounds real values to only 2 decimal places. Finally, `trapToext` concatenates the three lists of points obtained in a single set.

```
myround :: Double -> Integer -> Double
myround value n = (fromIntegral $ round $ value * (10^n)) /
      (10.0^n)

discLine1 :: Double -> Double -> Double -> Double ->
      Extension
discLine1 x1 x step x2
| step == 0 = [(1, x)]
| (step > 0 && x < x2) = ((myround ((x-x1)/(x2-x1)) 2), x) :
       discLine1 x1 (myround (x+step) 2) step x2
| (step > 0 && x >= x2) = []

discLine2 :: Double -> Double -> Double -> Extension
discLine2 x step x3
| step == 0 = []
| (step > 0 && x <= x3) = (1, x) : discLine2 (myround (x+
      step) 2) step x3
| (step > 0 && x > x3) = []

discLine3 :: Double -> Double -> Double -> Double ->
      Extension
discLine3 x3 x step x4
| step == 0 = [(1, x)]
| (step > 0 && x <= x4) = ((myround ((x4-x)/(x4-x3)) 2), x)
       : discLine3 x3 (myround (x+step) 2) step x4
| (step > 0 && x > x4) = []

trapToext :: Trapezoid -> Extension
```

```
trapToext (x1,x2,x3,x4) = (discLine1 x1 x1 ((x2–x1)/10) x2)
    ++ (discLine2 x2 ((x3–x2)/10) x3) ++ (discLine3 x3 x3
    ((x4–x3)/10) x4)
```

Then, we proceeded to construct the tables and use these operators to calculate the comparisons. The results obtained were shown in the previous section. For reasons of space, these definitions are not included.

6.2 PostgreSQL version

The original motivation for the classification of Type 2 fuzzy data was to include them in a Database Management System (DBMS). For this reason, we decided to carry out an implementation experience of the proposal in the PostgreSQL DBMS. This DBMS provides sufficient capabilities for the development of a new fuzzy type, as well as its ordering. In particular, the language offers the possibility of creating comparison operators ($<, >, =, \leq, \geq$) for custom data types.

So, the Database Catalog was extended with information to store the Type 2 fuzzy domains. Then, the syntax to express both the fuzzy numbers by extension and the trapezoidal fuzzy numbers in the DBMS language was defined. For creating a Type 2 fuzzy domain, the SQL language was extended with the following syntax

```
CREATE FUZZY DOMAIN <nombre> AS POSSIBILITY DISTRIBUTION ON
    <ordered_domain>
```

where `<name>` is an identifier and `<ordered_domain>` is a data type (integer, real, string or a range). The SQL language was also extended to represent Type 2 fuzzy literals. The SQL clause **ORDER BY** was extended so that it could use these comparators.

For example, the following is a possible definition of a fuzzy domain for age and another for salary:

```
CREATE FUZZY DOMAIN AgeDom
AS POSSIBILITY DISTRIBUTION ON 0..100
CREATE FUZZY DOMAIN SalaryDom
AS POSSIBILITY DISTRIBUTION ON NUMBER
```

Then, you can create an employee table in which the age and salary columns using `AgeDom` and `SalaryDom` Type 2 fuzzy domains previously defined through the statement

```
CREATE TABLE EMPLOYEES (
name VARCHAR,
lastname VARCHAR,
age AgeDom,
salary SalaryDom
)
```

So, you can make an insertion using the EMPLOYEE table previously defined with the age and salary Type 2 attributes, as follows

```
INSERT INTO EMPLOYEES VALUES (
'Rose','Wilson', {0.75/22, 1/23, 0.8/24}, (10000, 20000,
    30000, 40000)
)
```

Here, the age attribute takes the value 0.75/22, 1/23, 0.8/24, a possibility distribution defined by extension; and the salary attribute takes the value (10,000, 20,000, 30,000, 40,000), a possibility distribution with trapezoidal definition.

The comparison operator was written in the PL/pgSQL language. Type 2 fuzzy data are represented by records that store three attributes. The first attribute is an array of reals that store the membership value of each pair (membership/value) in a fuzzy number. The second attribute is an array of some valid data type that store the value. The data type can be an integer, char, real or Boolean. For every element saved in the membership array, there is an element in the values array. This is due to the representation of pairs, which is given by the position in the array. Finally, the third attribute stores a Boolean that represents whether the fuzzy value is by extension or whether it is a trapezoid. If the fuzzy number has a trapezoidal definition, the values in the membership array are [0.0, 1.0, 1.0, 0.0] representing the four inflection point of trapezoid. In this case, the values array contains the four abscissa corresponding to these inflection points.

Each comparison operator in SQL ($>$, $>=$, $=$, $<$, $<=$) is implemented as a function that receives two elements and return a Boolean. The elements can be any of the four data types mentioned (char, integer, real or Boolean) or a fuzzy data type (trapezoidal or by extension). For example, the function signature that implements the greater or equal operator is

```
CREATE OR REPLACE FUNCTION
fuzzy_scheme.fuzzy_greater_eq(elem1 anyelement, elem2
    anyelement)
RETURNS boolean
```

All implemented functions are stored in the fuzzy extension of the DBMS scheme.

7 Comparison with other methods

In the third column in Table 4 is the conclusion of the order obtained by centroid method ($\leq_c$) and in fourth column is the conclusion of order obtained by the method here developed ($\leq_z$), for trapezoidal fuzzy numbers. Only are shown in table the results that do not match with Zadeh method. In Tables 5 and 6 are the result for fuzzy numbers given by extension and the mixed cases. Remember that the expected result is always $a \leq b$.

Analyzing the results above, in particular in Table 4, rows 2 and 3, the expected result is accomplished by the centroid method, while Zadeh method just gives equality, but in row 4 the expected result is accomplished by the Zadeh method and not by centroid one.

In the rows 3, 6 and 9 in Table 5 and rows 2, 4 and 5 in Table 6 is repited the situation where the centroid method meets expectations, while Zadeh method just gives equality.

Table 4 Comparison between Zadeh and centroid method for trapezoidal fuzzy numbers

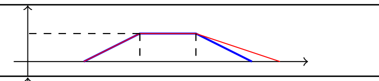
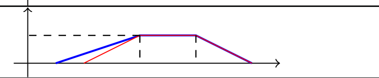
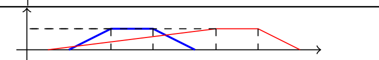
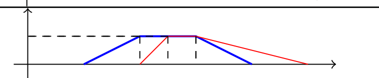
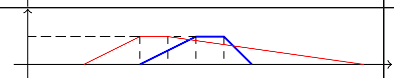
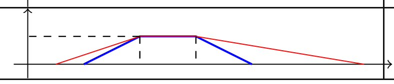
	a (blue) b (red)	$\leq_c$	$\leq_z$
1		$a \leq_c b$	$a =_z b$
2		$a \leq_c b$	$a =_z b$
3		$b \leq_c a$	$a <_z b$
5		$a \leq_c b$	$a =_z b$
6		$a \leq_c b$	$b <_z a$
7		$a \leq_c b$	$a =_z b$

Table 5 Comparison between Zadeh and centroid method for fuzzy numbers defined by extension

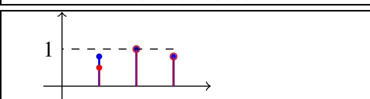
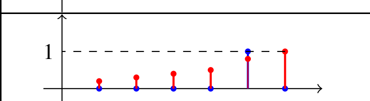
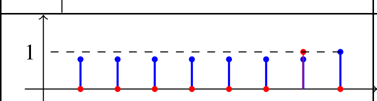
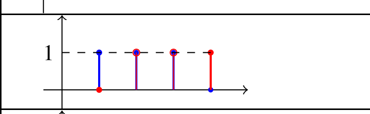
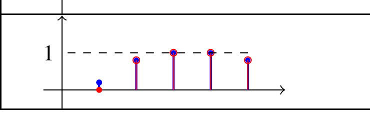
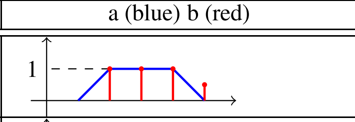
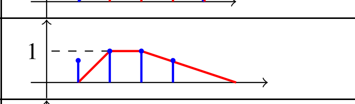
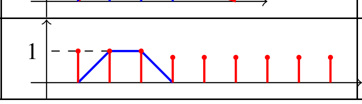
	a (blue) b (red)	$\leq_c$	$\leq_z$
3		$a \leq_c b$	$a =_z b$
4		$b \leq_c a$	$a <_z b$
5		$a \leq_c b$	$b <_z a$
6		$a \leq_c b$	$a =_z b$
9		$a \leq_c b$	$a =_z b$

Table 6 Comparison between Zadeh and centroid method for fuzzy numbers with mixed representation

	a (blue) b (red)	$\leq_c$	$\leq_z$
2		$a \leq_c b$	$a =_z b$
4		$a \leq_c b$	$a =_z b$
5		$a \leq_c b$	$a =_z b$

Is remember that the Zadeh method is reasonable more fast to apply that the centroid method. So, the Zadeh Method can be used in order to obtain fast result and a second method, e.g., the centroid, can be used in cases when Zadeh method gives equality.

8 Conclusions and future works

In order to give a suitable semantics to queries in databases with attributes whose values are fuzzy numbers, different proposals for ordering these numbers were studied. Ordering of data is a recurring activity when presenting query results in databases. The proposals are diverse, so the best choice is to use the one that best suits the application domain.

In this paper, an ordering proposal using the Zadeh extension principle is analyzed. It is a direct ordering that defines a comparison operator for fuzzy numbers that calculates the possibility that one number is less than the other. Given two fuzzy numbers A and B, both the possibility value A is less than B and the possibility value B is less than A calculated. If the first possibility is strictly greater than the second possibility, it is concluded that A is *possibly less than* B. If both possibilities have the same value, it is concluded that A is *possibly equal to* B. Thus, the operator $possiblylessthanorequalto$ is defined. This operator establishes a total ordering which is formally demonstrated in this work.

In the particular case of databases where the process of ordering query results is a recurring activity, the preferred way is marked by users' intuitions. For this reason, a set of study cases was designed that could be presented in attributes whose values are fuzzy numbers. For their design, different definitions and combinations of these numbers were considered. For each data pair, the order between them was chosen based on user's intuitions.

In this work, we performed an evaluation of a method based on the Zadeh extension principle using these study cases. We wanted to see if the application of the method produces the desired result according to our intuition. From this analysis, it can be concluded that in a good number of cases the method fails in terms of the expected intuition. So, if the user's preference is the rule to follow, it is recommended to use another method.

This paper also presents an implementation instance of the ordering method based on the Zadeh extension principle, using the Haskell and PostgreSQL programming languages. Due to its functional paradigm, Haskell is a simple language, easy to use and very suitable for mathematical definitions . Therefore, its use is recommended in the implementation of other ordering methods when studies similar to the one carried out in the present work are performed. In the PostgreSQL case, the implementation was more laborious, although it has the advantage that it can be used for real application examples and large databases.

It is noteworthy that the study cases presented in this article have an abstract definition. It would be advisable in future works to search real application domains and data from these domains. This way, new contextualized comparisons can be made in which the data semantics and the intuition of comparison are more clear.

In this work the ordering was studied formally and in a practical way. It was conceived for its use in database queries with the purpose of ordering the query results by attributes that contain fuzzy numbers. In future works, this ordering should be included as part of the extension of a database manager, such as PostgreSQL. This functionality can be included in the ORDER BY and GROUP BY clauses.

Acknowledgements This work is part of the project "Challenges of the Fuzzy Relational Model," with the support of UNEXPO "Antonio José de Sucre," Barquisimeto, Venezuela. We thank Professor Carlos Lameda, associate coordinator of this project, and Professor Ralph Grove, a friend in Norfolk, VA, who helped us with the editing of this paper. We also want to thank Him who can order everything, even the fuzziest lives, to whom, together with the Psalmist, we express: "Oh that my ways were ordered to keep thy statutes!" (Psalm 119:5).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

Akyar H (2018) A new approach for ordering fuzzy numbers and its application to fuzzy matrix games. J Sci Arts 18(3):547–564

Alizadeh HM, Khamseh AA, Ghomi SF (2013) Fuzzy hypothesis testing with vague data using likelihood ratio test. Soft Comput 17(9):1629–1641

Azman FN, Abdullah L (2012) Ranking fuzzy numbers by centroid method. Malays J Fundam Appl Sci 8(3)

Basar F (2012) Summability theory and its applications. Bentham Science Publishers

Biswas R (2016) Is fuzzy theory an appropriate tool for large size decision problems? Springer International Publishing, Cham, pp 93–118. https://doi.org/10.1007/978-3-319-26302-1_8

Canfora G, Troiano L (2004) Fuzzy ordering of fuzzy numbers. IEEE. https://doi.org/10.1109/FUZZY.2004.1375478

Chi HTX, Yu VF (2018) Ranking generalized fuzzy numbers based on centroid and rank index. Appl Soft Comput 68:283–292. https://doi.org/10.1016/j.asoc.2018.03.050

Dubois D, Prade H (1987) The mean value of a fuzzy number. Fuzzy Sets Syst 24(3):279–300. https://doi.org/10.1016/0165-0114(87)90028-5

Figueroa-García JC, Chalco-Cano Y, Román-Flores H (2018) Yager index and ranking for interval type-2 fuzzy numbers. IEEE Trans Fuzzy Sys 26(5):2709–2718. https://doi.org/10.1109/TFUZZ.2017.2788884

Galindo J, Urrutia A, Piattini M (2006) Fuzzy database modeling. Idea Group Publishing, Design and Implementation. https://doi.org/10.4018/978-1-59140-324-1

Ganesh AH, Suresh M (2017) Ordering of generalised trapezoidal fuzzy numbers based on area method using euler line of centroids. Adv Fuzzy Math 12(4):783–791

Kerre E (1982) The use of fuzzy set theory in electrocardiological diagnostics. In: M. Gupta and E. Sanchez (Eds) Aproximate reasoning in decision analysis 20: pp. 277–282

Kerre EE, Mareš M, Mesiar R (2000) Generated fuzzy quantities and their orderings. Springer, US, Boston, MA, pp 119–129. https://doi.org/10.1007/978-1-4615-5209-3_9

Kumar A, Singh P, Kaur P, Kaur A (2011) RM approach for ranking of L–R type generalized fuzzy numbers. Soft Comput 15(7):1373–1381

Mitchell HB, Schaefer PA (2000) On ordering fuzzy numbers. Int J Intel Sys 15(11):981–993. https://doi.org/10.1002/1098-111X(200011)15:11<981::AID-INT1>3.0.CO;2-Z

Nasseri SH, Taleshian F, Alizadeh Z, Vahidi J (2012) A new method for ordering LR fuzzy number. J Math Comput Sci 4(3):283–294. https://doi.org/10.22436/jmcs.04.03.01

Prade H, Testemale C (1989) The possibilistic approach to the handling of imprecison in database systems. IEEE Data Eng Bull 12(2):4–10

Rao PPB, Shankar NR (2011) Ranking fuzzy numbers with a distance method using circumcenter of centroids and an index of modality. Adv Fuzzy Syst 2011:3. https://doi.org/10.1155/2011/178308

Rosen KH (2012) Discrete Mathematics and its Applications, 7th edn. Mc Graw Hill, https://doi.org/10.1093/teamat/hrq007

Rudnik K, Kacprzak D (2017) Fuzzy topsis method with ordered fuzzy numbers for flow control in a manufacturing system. Appl Soft Comput 52:1020–1041

Seiti H, Hafezalkotob A, Martínez L (2019) R-numbers, a new risk modeling associated with fuzzy numbers and its application to decision making. Inform Sci 483:206–231. https://doi.org/10.1016/j.ins.2019.01.006

Singh P (2015) A novel method for ranking generalized fuzzy numbers. JISE J Inf Sci Eng 31(4):1373–1385. https://doi.org/10.6688/JISE.2015.31.4.13

Thomas GB, Weir MD (2010) Cálculo: una variable, 12th edn. Pearson Educación

Vincent FY, Van LH, Dat LQ, Chi HTX, Chou SY, Duong TTT (2017) Analyzing the ranking method for fuzzy numbers in fuzzy decision making based on the magnitude concepts. Int J Fuzzy Syst 19(5):1279–1289

Wang Y (2020) Combining technique for order preference by similarity to ideal solution with relative preference relation for interval-valued fuzzy multi-criteria decision-making. Soft Comput 24:11347–11364. https://doi.org/10.1007/s00500-019-04599-8

Wang Y (2020) Utilization of trapezoidal intuitionistic fuzzy numbers and extended fuzzy preference relation for multi-criteria group decision-making based on individual differentiation of decision-makers. Soft Comput 24:397–407. https://doi.org/10.1007/s00500-019-03921-8

Wang YM, Yang JB, Xu DL, Chin KS (2006) On the centroids of fuzzy numbers. Fuzzy Sets Syst 157(7):919–926

Yager R (1980) On choosing between fuzzy subsets. Kybernetes 9(2):151–154. https://doi.org/10.1108/eb005552

Yuan Y (1991) Criteria for evaluating fuzzy ranking methods. Fuzzy Sets Syst 43(2):139–157. https://doi.org/10.1016/0165-0114(91)90073-Y

Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

Zadeh LA (1975) The concept of a linguistic variable and its application to approximate reasoning-I. Inf Sci 8(3):199–249. https://doi.org/10.1016/0020-0255(75)90036-5

Zadeh LA (1975) The concept of a linguistic variable and its application to approximate reasoning-III. Inf Sci 9:43–64. https://doi.org/10.1016/0020-0255(75)90017-1

Ziemba P (2018) NEAT F-PROMETHEE-A new fuzzy multiple criteria decision making method based on the adjustment of mapping trapezoidal fuzzy numbers. Expert Syst Appl 110:363–380. https://doi.org/10.1016/j.eswa.2018.06.008

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.