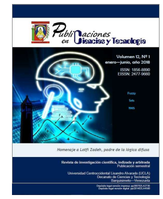
Publicaciones en Ciencias y Tecnología

Publicaciones en Ciencias y Tecnología. Vol.12, N^{o}2, Julio-Diciembre (2018) 105–116



Artículo de investigación

Ordenamiento de números difusos usando centroides: Aspectos prácticos

Ordering fuzzy numbers using the centroid: Practical aspects

Soraya Carrasquel[a], David Coronado[a], Ricardo Monascal[a], Rosseline Rodríguez[a]

[a] *Universidad Simón Bolívar. Caracas, Venezuela.*

Recibido: 31-03-2018 Aceptado: 02-10-2018

Resumen

Este artículo corresponde a la segunda parte de un estudio donde se explora el ordenamiento de números difusos con miras a su uso en consultas a bases de datos. En la primera parte: se presentó una nueva propuesta que compara dos números difusos usando la abscisa del centroide; y se demostró que es una relación orden total. En esta segunda parte: se abordan casos de estudio que comprenden una vasta diversidad de posibles situaciones de comparación de números difusos con distintas representaciones; se presenta una implementación en Haskell del método propuesto; y se pone en evidencia su adecuación a la intuición humana.

Palabras clave: números difusos, ordenamiento, centroide, bases de datos, Haskell.
Código UNESCO: 120304- Inteligencia Artificial.

Abstract

This paper corresponds to the second part of the study exploring the ordering of fuzzy numbers with the purpose of supporting the database queries configuration. In the first part: a new proposal was presented comparing two fuzzy numbers using the centroid abscissa; and it was shown to be a total order relation. In this second part: study cases are addressed that include a vast diversity of possible situations for comparison fuzzy numbers with different representations; a Haskell implementation of the proposed method is presented; and thus, we demonstrate its adaptation to human intuition.

Key words: fuzzy numbers, ordering, centroid, databases, Haskell.
UNESCO Code: 120304- Artificial intelligence.

1. Introducción

El concepto de número difuso fue introducido por Zadeh [1] con el propósito de manipular valores aproximados. Hay dominios de aplicación donde los números difusos se han incorporado para representar la imprecisión e incertidumbre en los datos. En el área de bases de datos se han realizado diversos esfuerzos para dar soporte a información imperfecta y requerimientos vagos. Así surge el modelo entidad relación extendido difuso FuzzyEER [2] y el lenguaje para bases de datos relacionales extendido difuso FSQL [2], los cuales admiten atributos cuyo valor sea un número difuso.

Impresa: ISSN 1856-8890. Dep. Legal pp200702LA2730. Digital: ISSN 2477-9660 Dep. Legal ppi201402LA4590. Licencia CC BY-NC-SA

Email addresses: `scarrasquel@usb.ve` (Soraya Carrasquel), `dcoronado@usb.ve` (David Coronado), `rmonascal@usb.ve` (Ricardo Monascal), `crodrig@usb.ve` (Rosseline Rodríguez)

En consultas a bases de datos es frecuente requerir que los resultados sean presentados según el orden para algunos de los atributos. Sin embargo, FSQL no permite que en el ordenamiento de una consulta se usen atributos definidos como números difusos. Esta limitación puede subsanarse hallando un ordenamiento de números difusos que sea adecuado.

En las últimas décadas se han propuesto numerosos métodos para el ordenamiento de números difusos [3]. Según Buckley y Eslami [4] hay más de cien publicaciones sobre ordenamientos de números difusos. Yuan [5] sostiene que no existe un método único de ordenamiento que satisfaga todas las necesidades: más aún, estos métodos no siempre concuerdan entre sí y algunos de ellos incluso parecen contradecir la propia intuición en algunos casos. Nasseri et al [6] sostienen que la mayoría de los métodos presentados no pueden comparar cualquier par de números difusos, y algunos métodos no concuerdan con la intuición humana. Por otro lado, Chu y Charnsethikul [7] dicen que, a pesar de los méritos, algunos de los métodos de ordenamiento son computacionalmente complejos y en otros es difícil implementar la conexión entre el procedimiento de puntuación y los valores de evaluación difusos finales, lo que limita su aplicabilidad.

En este trabajo se toma el ordenamiento propuesto en [8] que calcula la abscisa del centroide del número difuso y luego se comparan los números difusos en base al resultado de este cálculo. Este método permite comparar números difusos dados trapezoidalmente con números difusos sin importar cuál sea su forma de representación. Compara números difusos dados por extensión, dados por función trapezoidal, de un solo punto (precisos); y cualquier combinación de éstos. Además, en este trabajo se hace un estudio de la aplicación de este método sobre diversos casos de estudio, a fin de observar su adecuación. Ésta se comprueba en base al resultado esperado según la intuición humana, lo cual sería una ventaja respecto a muchos otros métodos, según lo expresado tanto por Yuan [5] como por Nasseri et al [6].

Rao y Shankar [9] también presentaron un procedimiento para ordenar números difusos con definición trapezoidal, que usa el cálculo del centroide. Para ello, dividen el trapecio en dos triángulos y un rectángulo, y calculan el centroide de cada figura geométrica. Entonces, encuentran el circuncentro del triángulo formado por los tres centroides, para finalmente, ordenar de acuerdo a la abscisa de este punto. Aunque este método resulta de fácil implementación y se puede usar incluso, con números precisos, no puede ser usado para números difusos definidos por extensión, lo cual hace que no sea un método completo y limitaría mucho su aplicabilidad. Por tal razón se decide usar el método planteado en [8].

El resto del artículo se estructura así: la sección 2 expone los conceptos teóricos que sustentan este trabajo. En la sección 3 se detalla el proceso de implementación del método en el lenguaje de programación funcional *Haskell*. Posteriormente, se estudian los diferentes casos posibles de comparación: en la sección 4 cuando ambos números difusos son dados por extensión, en la sección 5 cuando ambos números difusos se definen trapezoidalmente y en la sección 6 cuando un número difuso viene dado por extensión y el otro trapezoidalmente. En estas secciones, se presenta además un análisis de los resultados obtenidos luego de la implementación en *Haskell*. Finalmente, se dan las conclusiones y trabajos futuros.

2. Marco teórico

Los *conjuntos difusos* [10] se caracterizan por una función de membresía μ, cuyo rango está en el intervalo real $[0, 1]$, cuánto más se acerca a 1 el grado de membresía de un elemento, éste está más posiblemente (o certeramente) incluido en el conjunto. Así, 0 es la medida de completa exclusión y 1 la medida de completa inclusión.

Se conoce como *núcleo* al conjunto clásico conformado por los elementos completamente incluidos en un conjunto difuso. Los elementos parcialmente incluidos, es decir, aquéllos cuyo grado de membresía es estrictamente mayor que cero y menor que uno, conforman el *borde* del conjunto difuso. El conjunto clásico formado por la unión del núcleo y el borde se denomina *soporte* del conjunto difuso. Es decir el *soporte* es el conjunto de los elementos que no están completamente excluidos del conjunto difuso, aquéllos cuyo grado de membresía es distinto de cero.

Un *número difuso* se define como un conjunto difuso con soporte en los números reales que satisface que su función de membresía es normalizada, el núcleo es un conjunto pleno, además es convexa, semicontinua superior y su soporte está acotado [2]. Los números difusos pueden ser representados de diferentes formas. En este trabajo se usarán las dos formas más comunes: definición por extensión y definición trapezoidal. En el caso en que un número difuso aproxima un valor entero, se puede usar una definición por extensión. Un *número difuso A definido por extensión* se representa como un conjunto de pares de la forma $A = \{\mu_A(x_1)/x_1, \mu_A(x_2)/x_2, \ldots, \mu_A(x_n)/x_n\}$.

Una representación de número difuso muy usada es de función de membresía trapezoidal. En esta caso, se especifica con una cuaterna (x_1, x_2, x_3, x_4) de elementos ordenados del dominio, con $x_1 \leq x_2 \leq x_3 \leq x_4$ que definen la función

$$\mu(x) = \begin{cases} 0 & \text{si } x < x_1 \\ \dfrac{x_2 - x}{x_2 - x_1} & \text{si } x_1 \leq x < x_2 \\ 1 & \text{si } x_2 \leq x \leq x_3 \\ \dfrac{x_4 - x}{x_4 - x_3} & \text{si } x_3 < x \leq x_4 \\ 0 & \text{si } x > x_4 \end{cases}$$

Un número preciso x puede representarse como el trapecio degenerado (x, x, x, x).

2.1. Ordenamiento basado en el centroide

Muchas estructuras y sistemas mecánicos, entre otros, actúan como si sus masas estuvieran concentradas en un solo punto, llamado *centro de masa*. La localización de este punto depende de dos factores principales, la forma de la estructura y el material con el cual está construida, a través de su densidad de masa. En el caso en que la estructura esté elaborada de un único material (densidad constante), el centro de masa dependerá sólo de la forma de la estructura, en este caso se llama *centroide*.

Con base en lo anterior, el centroide es una medida que puede usarse como representante de un número difuso, y así ordenarlos de acuerdo con la cercanía de este valor al origen. En [11], se plantea un ordenamiento de números difusos a partir del cálculo de la distancia euclídea entre el centroide de cada número difuso y un punto fijo, ocasionalmente el origen. Posteriormente, en [12], los autores corrigen un error en la fórmula para el cálculo del centroide usada en [11] y muestran un ejemplo de la variación de los resultados con ambas fórmulas.

Por otra parte, en [13] se propone un método de ordenamiento basado en el área entre el centroide del número difuso y el origen. En [14] se define un método de ordenamiento que consiste en: primero, cortar el trapecio en dos triangulos y un rectángulo proyectando al eje X los vértices superiores del mismo; segundo, encontrar el centroide de cada figura recién construída; tercero, calcular el centroide del triangulo formado por los tres centroides anteriores; y cuarto, ordenar de acuerdo a las abscisas de este centroide. Este método es parcialmente modificado en [15] usando líneas de Euler.

Una de las desventajas de los métodos mencionados previamente es que sólo es aplicable a números difusos trapezoidales. Es así como en [8] se propone calcular el centroide de cada número difuso a comparar y ordenar de acuerdo a la abscisa de éstos. Además dan fórmulas para el cálculo del centroide tanto de números difusos trapezoidales como números difusos definidos por extensión. En este trabajo se implementa el método propuesto en [8], y además, se realiza una comparación con el método presentado en [14].

La siguiente relación $\leq_C$ entre números difusos fue propuesta en [8]. La misma se basa en el cálculo de la abscisa del centroide del número difuso. Se demostró además que dicha relación es una relación de orden total y que permite comparar números difusos dados en forma trapezoidal, por extensión y números precisos.

Sean dos números difusos A y B. Sean $\overline{A}$ y $\overline{B}$ las respectivas abscisas de sus centroides, se dice que $A \leq_C B$ si y solo si $\overline{A} \leq \overline{B}$. Además, si $\overline{A} = \overline{B}$ se dice que $A =_C B$.

Para calcular la abscisa del centroide se obtuvieron las siguientes fórmulas:

Caso 1 Si A es número preciso, $A = x$, se puede representar por extensión $A = \{ 1/x \}$ o trapezoidalmente $A = (x, x, x, x)$. En cualquier caso, la abscisa de su centroide es el propio valor x.

Caso 2 Si A es un número difuso dado por extensión como $A = \{ \mu_1/x_1, \mu_2/x_2, \ldots, \mu_n/x_n \}$ su centroide viene dado por la media ponderada de los x_i con pesos μ_i

$$\overline{A} = \frac{\sum_{k=1}^{n} x_k \mu_k}{\sum_{k=1}^{n} \mu_k}. \tag{2.1}$$

Caso 3 Si A es un número difuso, no preciso, y está dado trapezoidalmente como $A = (x_1, x_2, x_3, x_4)$, entonces la abscisa de su centroide $\overline{A}$ es:

$$\overline{A} = \frac{1}{3}\left(x_1 + x_2 + x_3 + x_4 + \frac{x_1 x_2 - x_3 x_4}{x_4 - x_1 + x_3 - x_2}\right) \tag{2.2}$$

3. Experiencia práctica de implementación

El método de ordenamiento basado en la abscisa del centroide [8] fue implementado en una primera versión usando el lenguaje de programación *Haskell*. Este lenguaje es de muy fácil uso, siendo muy apropiado para definiciones matemáticas por su paradigma funcional.

Esta implementación se realizó con el objetivo de probar el método sobre un conjunto de casos de prueba. A continuación se darán algunos detalles de esta implementación.

En un primer lugar se definieron los tipos asociados a los números difusos: `Trapecio`, `Extension` y `NumDifuso`. El primero representa la definición trapezoidal de un número difuso a través de una tupla con los cuatro puntos de inflexión del trapecio. Su definición en *Haskell* es

```
type Trapecio = (Double, Double, Double, Double).
```

El segundo permite especificar el conjunto de pares μ_i/x_i que corresponde el valor de la función de membresía μ_i para el valor x_i. Los pares (membresía, valor) cuya función de membresía es mayor que cero se almacenan en una lista. Su definición en *Haskell* es

```
type Extension = [(Double, @Double)]
```

El tercer tipo es una generalización que representa un número difuso cualquiera, a través de una unión (`Either`) en *Haskell*, cuya definición es

```
type NumDifuso = Either Trapecio Extension
```

Por ejemplo, si se quiere definir el trapecio $(2, 4, 6, 8)$ se usaría la sintaxis:

```
Left (2.0, 4.0, 6.0, 8.0)
```

Mientras que para definir el número difuso por extensión { 0,5/1, 1/2, 0,8/3, 0,5/4 } se usaría la sintaxis:

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

Las palabras reservadas `Right` y `Left` indican el tipo de definición que se eligió dentro de la unión con base en la posición que ésta ocupa: a la izquierda para los trapecios y a la derecha para las definiciones por extensión.

Luego, se implementó el operador de comparación `menorFuzzy` para números difusos que usa el cálculo de la abscisa del centroide.

```
menorFuzzy :: NumDifuso -> NumDifuso -> Bool
menorFuzzy nd1 nd2 = (absCentroide nd1) < (absCentroide nd2)
```

Como se observa este operador llama a la función `absCentroide` que calcula la abscisa del centroide dependiendo del tipo de número difuso que corresponda (trapezoidal o por extensión).

```
absCentroide ::  NumDifuso -> Double
absCentroide (Left (x1,x2,x3,x4))=
                          (x1+x2+x3+x4+((x1*x2-x3*x4)/(x4-x1+x3-x2)))/3
absCentroide (Right nde) = (sum $ map mulAbsOrd nde)/(sumMembresia nde)
```

En el caso de un número difuso trapezoidal, en la función `absCentroide` se realiza directamente el cálculo según la definición. Para los números difusos definidos por extensión es necesario calcular la sumatoria de los productos $\mu_i * x_i$ para cada i que aparezca en el conjunto difuso `nde` y dividirlo entre la sumatoria de todas los valores de membresía de dichos pares. Para ello se usan las siguientes funciones auxiliares:

```
mulAbsOrd :: (Double,Double) -> Double
mulAbsOrd (x,y) = x*y

sumMembresia :: Extension -> Double
sumMembresia [] = 0
sumMembresia ((a,b):t) = a + sumMembresia(t)
```

Para realizar las pruebas con los casos de estudio se construyeron tablas donde se comparan dos números difusos. Esto se hizo con la implementación de la función `construirTabla` en *Haskell*, la cual toma una lista de pares de números difusos `[[NumDifuso]]` y devuelve una tabla de tuplas

```
(NumDifuso, NumDifuso, Double, Double, Bool),
```

donde los dos primeros elementos son los números a comparar, el tercer elemento es un valor real con el valor de la abscisa del centroide del primer número difuso, el cuarto elemento es un valor real con el valor de la abscisa del centroide del segundo número difuso, y el quinto elemento es un valor booleano con el resultado de la comparación $A \leq_C B$, siendo A el primer número difuso y B el segundo número difuso de la tupla. A continuación la especificación completa de la función `construirTabla`.

```
construirTabla :: [[NumDifuso]] -> [(NumDifuso, NumDifuso, Double, Double, Bool)]
construirTabla [] = []
construirTabla tabla =
    let
       nd1 = head (head tabla)
       nd2 = head (tail (head tabla))
    in ((nd1, nd2,(absCentroide nd1),(absCentroide nd2),
        (menorFuzzy nd1 nd2)):(construirTabla (tail tabla)))
```

Esta tabla es necesario transformarla a un *string* para su visualización por pantalla a través de la siguiente función en *Haskell*.

```
tablaToString :: [(NumDifuso, NumDifuso,Double,Double, Bool)] -> String
tablaToString tabla = unlines $ map show tabla
```

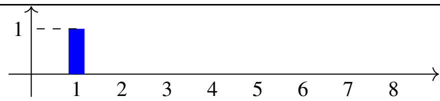
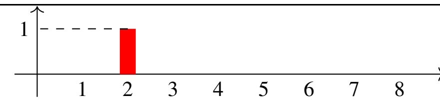
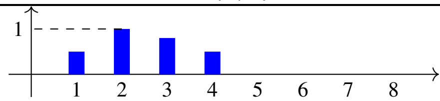
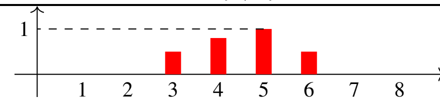
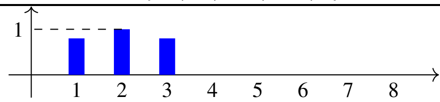
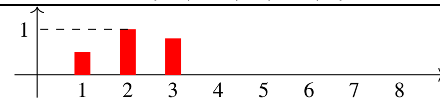
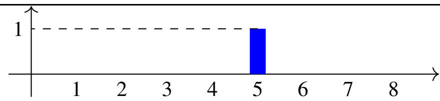
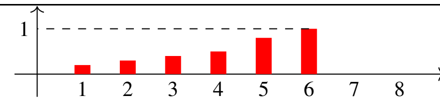
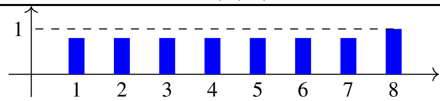
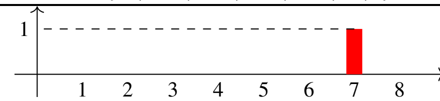
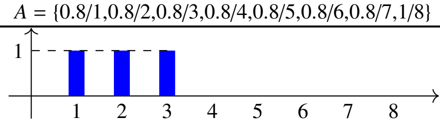
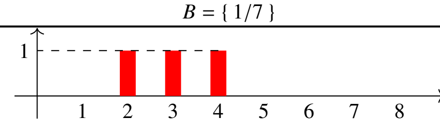
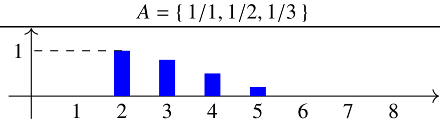
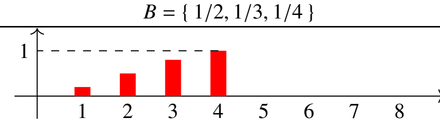
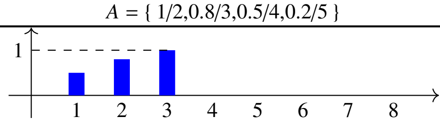
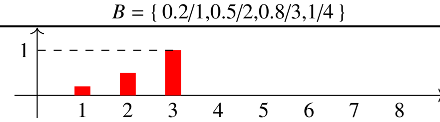
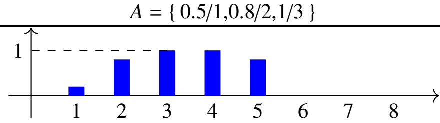
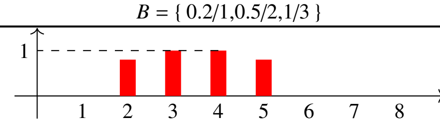
Finalmente, se escribió el programa principal para construir las tablas correspondientes a todos los casos de prueba realizados

```
main :: IO()
main = do
   hSetBuffering stdout NoBuffering
   putStrLn "\n\nTabla 1: números difusos por extensión\n\n"
   putStr $ tablaToString (construirTabla tabla1)
   putStrLn "\n\nTabla 2: números difusos trapezoidales\n\n"
   putStr $ tablaToString (construirTabla tabla2)
   putStrLn "\n\nTabla 3: números difusos combinados\n\n"
   putStr $ tablaToString (construirTabla tabla3)
```

4. Estudio con representación por extensión

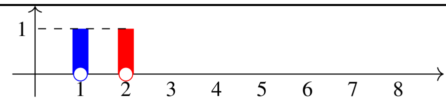
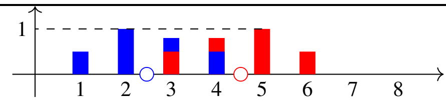
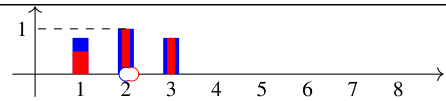
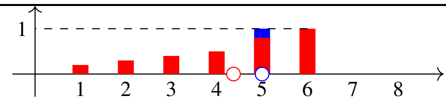
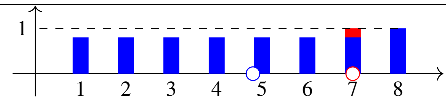
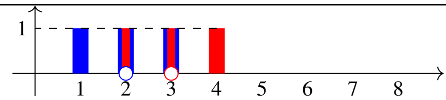
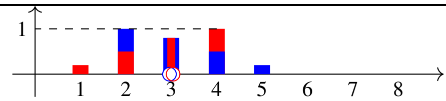
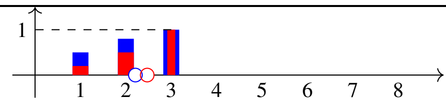
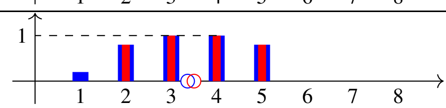
En el Cuadro 1 se presentan los números difusos correspondientes a los casos de estudio preparados. En cada fila del cuadro se tiene un par de números difusos A y B. El diseño del caso de estudio se hizo de forma tal que, según la intuición de los usuarios, se esperaría que A sea menor o igual que B.

Cuadro 1: Pares de números difusos dados por extensión a ser comparados

No	A	B
1	 $A = \{ 1/1 \}$	 $B = \{ 1/2 \}$
2	 $A = \{ 0.5/1, 1/2, 0.8/3, 0.5/4 \}$	 $B = \{ 0.5/3, 0.8/4, 1/5, 0.5/6 \}$
3	 $A = \{ 0.8/1, 1/2, 0.8/3 \}$	 $B = \{ 0.5/1, 1/2, 0.8/3 \}$
4	 $A = \{ 1/5 \}$	 $B = \{0.2/1, 0.3/2, 0.4/3, 0.5/4, 0.8/5, 1/6\}$
5	 $A = \{0.8/1, 0.8/2, 0.8/3, 0.8/4, 0.8/5, 0.8/6, 0.8/7, 1/8\}$	 $B = \{ 1/7 \}$
6	 $A = \{ 1/1, 1/2, 1/3 \}$	 $B = \{ 1/2, 1/3, 1/4 \}$
7	 $A = \{ 1/2, 0.8/3, 0.5/4, 0.2/5 \}$	 $B = \{ 0.2/1, 0.5/2, 0.8/3, 1/4 \}$
8	 $A = \{ 0.5/1, 0.8/2, 1/3 \}$	 $B = \{ 0.2/1, 0.5/2, 1/3 \}$
9	 $A = \{0.2/1, 0.8/2, 1/3, 1/4, 0.8/5\}$	 $B = \{ 0.8/2, 1/3, 1/4, 0.8/5 \}$

Los resultados de la implementación en *Haskell* se muestra en el Cuadro 2. El resultado esperado, de acuerdo a la intuición de los usuarios consultados, es $A \leq_C B$.

Cuadro 2: Resultado de la comparación de pares de números difusos definidos por extensión

No	Gráfica de la comparación de A y B	$\overline{A}$	$\overline{B}$	Conclusión
1		1	2	$A \leq_C B$
2		2.46	4.53	$A \leq_C B$
3		2	2.13	$A \leq_C B$
4		5	4.37	$B \leq_C A$
5		4.61	7	$A \leq_C B$
6		2	3	$A \leq_C B$
7		2.96	3.04	$A \leq_C B$
8		2.21	2.47	$A \leq_C B$
9		3.36	3.5	$A \leq_C B$

Al observar el Cuadro 2, se evidencia que en ocho de los nueve casos de estudio, la comparación resultó como se esperaba. Sin embargo, se destaca la fila 4, donde no se obtuvo el resultado esperado. La cual corresponde a un número preciso vs. un número difuso.

En el mismo Cuadro 2, se observa que la fila 1 compara dos números precisos dando el resultado esperado. Así mismo, en la fila 5 se compara un número difuso vs. un número preciso, dando también el resultado esperado. Por tal razón se pensó que quizás lo que estaba fallando para el caso de la fila 4 era la intuición de los usuarios, y no el método.

Si se analizan los detalles para el caso inesperado, como la definición

$$A = \{\,1/5\,\}$$
$$B = \{0{,}2/1,\ 0{,}3/2,\ 0{,}4/3,\ 0{,}5/4,\ 0{,}8/5,\ 1/6\}$$

y el gráfico, visto más de cerca en la Figura 1, se puede notar que el número difuso A (en azul) tiene un solo valor del dominio con grado de membresía distinto de cero. Ese valor es el 5 cuyo grado de membresía es 1. Mientras que el número difuso B (en rojo) tiene cinco valores del dominio con grado de membresía distinto de cero, uno de los cuales, el valor 6 tiene grado de membresía igual a 1.

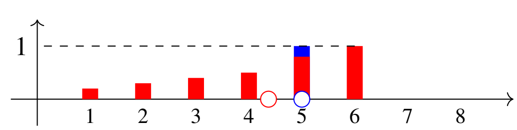


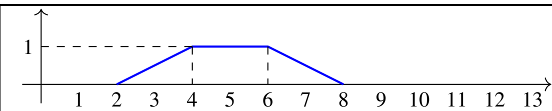
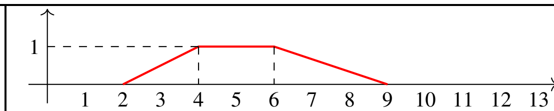
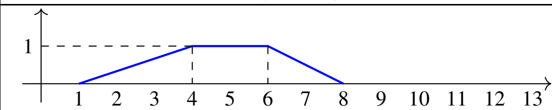
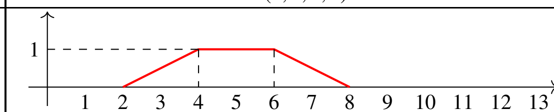
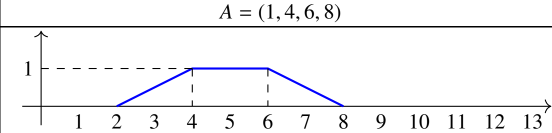
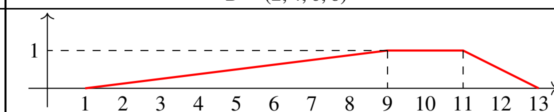
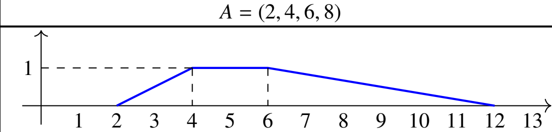
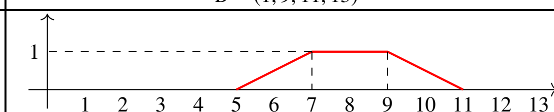
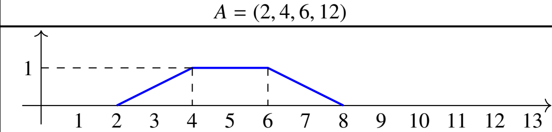
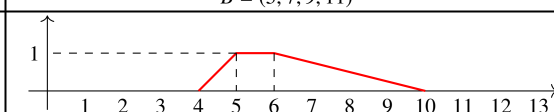
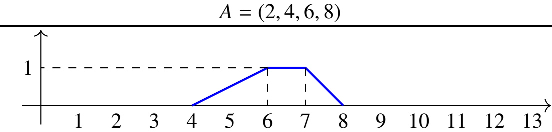
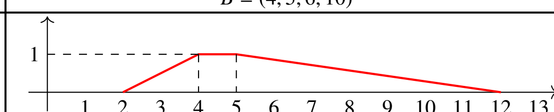
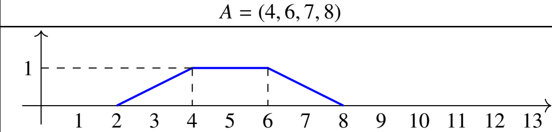
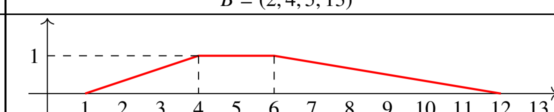
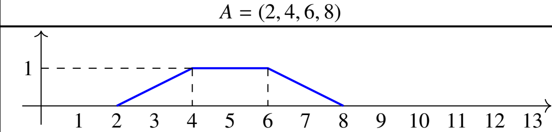
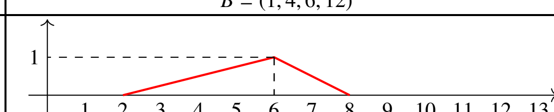
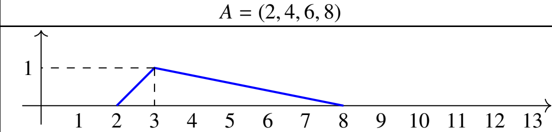
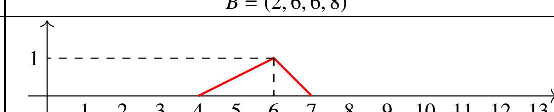
Figura 1: Gráfica ampliada del ejemplo con resultado no esperado

Por esta razón al calcular la abscisa del centroide, o punto de equilibrio de la barra metálica con masas correspondientes al número difuso B aparece antes del valor 5. En cambio la barra asociada al número difuso A sólo tiene la masa en el punto 5. Por lo que la abscisa del centroide cae en el valor 5. Esta es la razón por la que para este caso, la comparación a través del valor de la abscisa del centroide es contraria a la intuición.

5. Estudio con representación trapezoidal

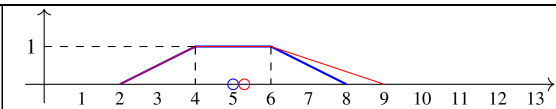
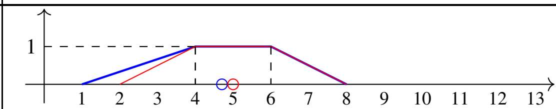
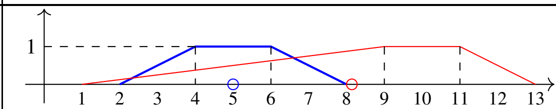
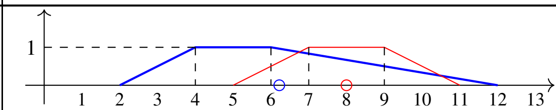
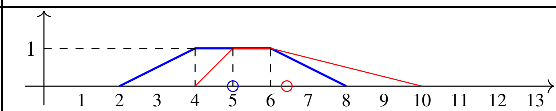
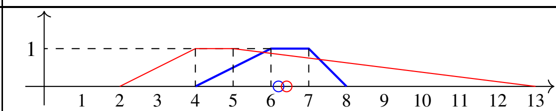
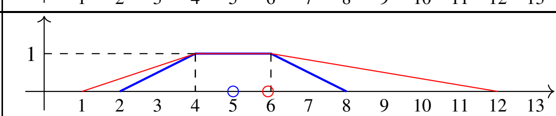
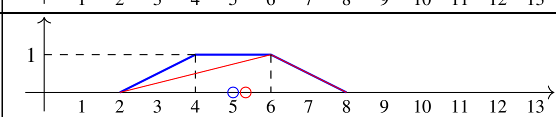
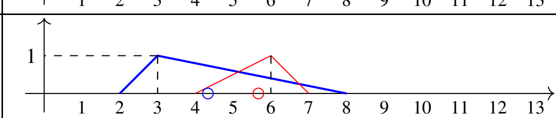
En el Cuadro 3, se muestran los nueve pares de números difusos dados trapezoidalmente a ser comparados. La intuición indica que el número difuso A (en azul) será menor que el número difuso B (en rojo).

Cuadro 3: Pares de números difusos dados trapezoidalmente a ser comparados

No	A	B
1	 $A = (2, 4, 6, 8)$	 $B = (2, 4, 6, 9)$
2	 $A = (1, 4, 6, 8)$	 $B = (2, 4, 6, 8)$
3	 $A = (2, 4, 6, 8)$	 $B = (1, 9, 11, 13)$
4	 $A = (2, 4, 6, 12)$	 $B = (5, 7, 9, 11)$
5	 $A = (2, 4, 6, 8)$	 $B = (4, 5, 6, 10)$
6	 $A = (4, 6, 7, 8)$	 $B = (2, 4, 5, 13)$
7	 $A = (2, 4, 6, 8)$	 $B = (1, 4, 6, 12)$
8	 $A = (2, 4, 6, 8)$	 $B = (2, 6, 6, 8)$
9	 $A = (2, 3, 3, 8)$	 $B = (4, 6, 6, 7)$

Se obtuvieron los resultados mostrados en el Cuadro 4. Destaca que en *todos* los casos, se obtuvo el resultado esperado. Esto indica que la intuición sugerida por los usuarios consultados es correcta en el caso de los números difusos dados trapezoidalmente, y que el método cumple con esta intuición.

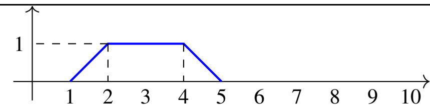
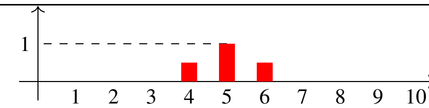
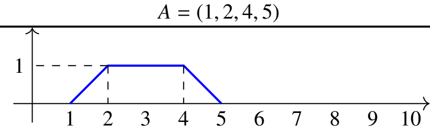
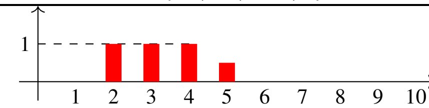
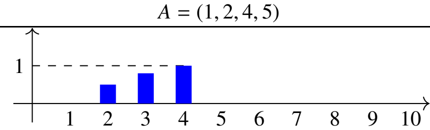
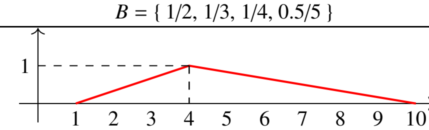
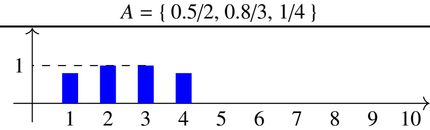
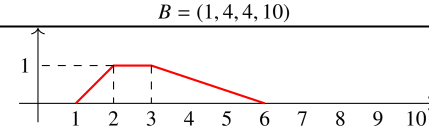
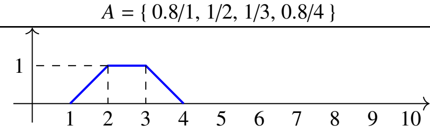
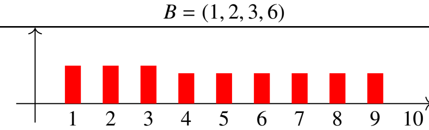
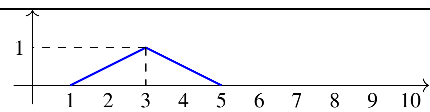
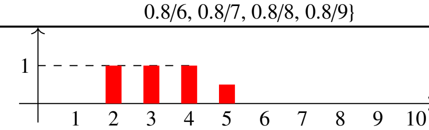
Cuadro 4: Resultados de comparar pares de números difusos de forma trapezoidal

Nro	Gráfica de la comparación de A y B	$\overline{A}$	$\overline{B}$	Conclusión
1		5	5.29	$A \leq_C B$
2		4.7	5	$A \leq_C B$
3		5	8.14	$A \leq_C B$
4		6.22	8	$A \leq_C B$
5		5	6.42	$A \leq_C B$
6		6.2	6.42	$A \leq_C B$
7		5	5.92	$A \leq_C B$
8		5	5.33	$A \leq_C B$
9		4.33	5.66	$A \leq_C B$

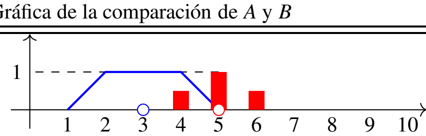
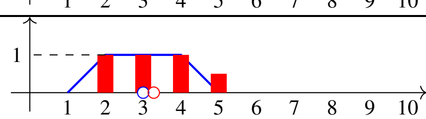
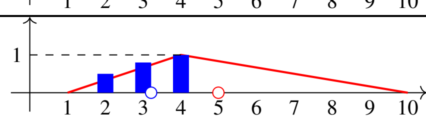
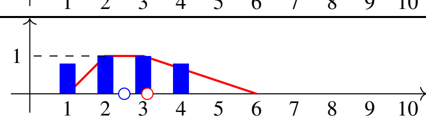
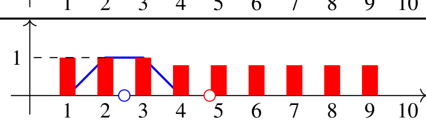
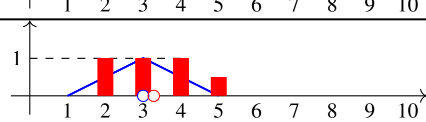
6. Estudio con representación mixta

Se muestra en esta sección, seis pares de números difusos a ser comparados, donde uno de los números difusos está dado por extensión el otro trapezoidalmente. En el Cuadro 5, se muestran los seis pares de números difusos a ser comparados, donde uno está dado por extensión y el otro trapezoidalmente. Se espera intuitivamente, el resultado $A \leq_C B$. En el Cuadro 6 se muestra el resultado obtenido mediante la implementación en *Haskell*. Nuevamente, se obtuvo el resultado esperado intuitivamente en todos los casos. Lo cual muestra la buena adecuación del método propuesto.

Cuadro 5: Pares de números difusos en formatos mixtos

No	A	B
1	 $A = (1, 2, 4, 5)$	 B = { 0.5/4, 1/5, 0.5/6 }
2	 $A = (1, 2, 4, 5)$	 B = { 1/2, 1/3, 1/4, 0.5/5 }
3	 A = { 0.5/2, 0.8/3, 1/4 }	 $B = (1, 4, 4, 10)$
4	 A = { 0.8/1, 1/2, 1/3, 0.8/4 }	 $B = (1, 2, 3, 6)$
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}
6	 $A = (1, 3, 3, 5)$	 B = { 1/2, 1/3, 1/4, 0.5/5 }

Cuadro 6: Resultados de comparar pares de números difusos en formato mixto

Nro	Gráfica de la comparación de A y B	$\overline{A}$	$\overline{B}$	Conclusión
1		3	5	$A \leq_C B$
2		3	3.28	$A \leq_C B$
3		3.21	5	$A \leq_C B$
4		2.5	3.11	$A \leq_C B$
5		2.5	4.76	$A \leq_C B$
6		3	3.28	$A \leq_C B$

7. Conclusiones - trabajos futuros

Luego de un estudio teórico de diversos métodos de ordenamiento para números difusos se concluyó que la mejor opción es tomar aquel método que se adapte al dominio de aplicación. En el caso particular de las bases de datos donde el proceso de ordenar los resultados de las consultas suele ser una actividad recurrente, el camino preferido es el marcado por la intuición de los usuarios. En este sentido, se diseñaron un conjunto de casos de estudio que podrían presentarse con atributos cuyos valores sean números difusos. En el diseño se consideraron diferentes definiciones y combinaciones de estos datos. Para cada par de datos se escogió el orden entre ellos con base en la intuición de los usuarios.

Se puede concluir que el ordenamiento *basado en la abscisa del centroide* es un *buen* ordenamiento ya que el resultado obtenido al aplicarlo es el esperado intuitivamente por los usuarios consultados. El método basado en la abscisa del centroide tiene la ventaja que permite realizar comparaciones sobre números difusos definidos trapezoidalmente, y por extensión, así como, con números precisos, lo cual no es posible realizar por otros métodos.

En este trabajo se presentó una experiencia de implementación del método de ordenamiento basado en la abscisa del centroide, en el lenguaje de programación *Haskell*. Se observó que este lenguaje es sencillo, de fácil uso y muy adecuado para definiciones matemáticas por su paradigma funcional. Por lo cual se recomienda su uso en la implementación de otros métodos de ordenamiento a fin de hacer estudios similares al realizado en el presente trabajo. Es de resaltar que los casos estudios presentados en este artículo fueron definidos en forma abstracta. Sería recomendable en un trabajo futuro buscar un dominio de aplicación real y datos de ese dominio de aplicación, para hacer nuevas comparaciones contextualizadas en la que quede aún más clara la semántica de los datos y la intuición de la comparación.

El ordenamiento que se propuso teóricamente en la primera parte de este trabajo, y que se estudió en forma práctica en esta segunda parte, fue concebido para su uso en consultas a bases de datos, permitiendo ordenar los resultados por atributos que contienen números difusos. En este sentido, en trabajos futuros debe implementarse este ordenamiento como parte de la extensión de un gestor de bases de datos para permitir esta funcionalidad.

8. Agradecimientos

Este trabajo es resultado del Proyecto de Grupo "*Desafíos del Modelo Relacional Difuso*" que cuenta con el apoyo de la Universidad Nacional Experimental Politécnica "Antonio José de Sucre" (UNEXPO), Vicerrectorado de Barquisimeto. Agradecemos el apoyo brindado por el profesor Carlos Lameda, coordinador adjunto de este proyecto, y al profesor Leonid Tineo, también coordinador adjunto, por sus valiosos aportes para concretar este trabajo. "*Conozco, oh Señor, que el hombre no es señor de su propio camino, ni del hombre que camina es el ordenar sus pasos*" (Jeremías 10:23)

Referencias

[1] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning—I. *Information sciences*, 8(3):199–249, 1975. OnLine.

[2] J. Galindo; A. Urrutia; M. Piattini. Fuzzy databases: Modeling, design and implementation. *Idea Group Publishing Hershey, USA*, 2006. OnLine.

[3] Pushpinder Singh. A novel method for ranking generalized fuzzy numbers. *Journal of Information Science and Engineering*, 31:1373–1385, 2015. OnLine.

[4] J.J Buckley; E. Eslami. Fuzzy ordering of fuzzy numbers. *International Journal of Uncertainty, Fuzziness and Knowledge–Based Systems*, 12(01):105–114, 2004. OnLine.

[5] Y. Yuan. Criteria for evaluating fuzzy ranking methods. *Fuzzy Sets and Systems*, 43(2):139–157, 1991. OnLine.

[6] S. H. Nasseri; F. Taleshian; Z. Alizadeh; J. Vahidi. A new method for ordering *LR* fuzzy number. *The Journal of Mathematics and Computer Science*, 4(3):283–294, 2012. OnLine.

[7] T.C. Chu; P. Charnsethikul. Ordering alternatives under fuzzy multiple criteria decision making via a fuzzy number dominance based ranking approach. *International Journal of Fuzzy Systems*, 15(3):263–273, 2013.

[8] S. Carrasquel; D. Coronado; R. Monascal; R. Rodriguez. Ordenamiento de números difusos usando centroides. aspectos teóricos. *Publicaciones en Ciencia y Tecnología*, 12(2):57–67, 2018. OnLine.

[9] P. Bushan Rao; N. R.Shankar. Ranking fuzzy numbers with a distance method using circumcenter of centroids and an index of modality. *Advances in Fuzzy Systems*, pages 1–7, 2011. OnLine.

[10] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965. OnLine.

[11] Ching-Hsue Cheng. A new approach for ranking fuzzy numbers by distance method. *Fuzzy Sets and Systems*, 95(3):307–317, 1998. OnLine.

[12] Y.M Wang; J.B. Yang; D.L. Xu; K.S. Chin. On the centroids of fuzzy numbers. *Fuzzy sets and systems*, 157(7):919–926, 2006. OnLine.

[13] Ta-Chung Chu; Chung-Tsen Tsao. Ranking fuzzy numbers with an area between the centroid point and original point. *Computers & Mathematics with Applications*, 43(1-2):111–117, 2002, OnLine.

[14] H. B. Mitchell; P. A. Schaefer. On ordering fuzzy numbers. *International Journal of Intelligent Systems*, 15(11):981–993, 2000. OnLine.

[15] A. H. Ganesh; M. Suresh. Ordering of generalised trapezoidal fuzzy numbers based on area method using euler line of centroids. *Advances in Fuzzy Mathematics*, 12(4):783–791, 2017.

Sobre los autores

Soraya Carrasquel
Licenciada en Matemáticas, Magister Scientiarum en Matemáticas. Profesora e Investigadora en la Universidad Simón Bolívar, Caracas, Venezuela. Adscrita al Departamento de Computación y Tecnología de la Información.
Correo: scarrasquel@usb.ve - ORCID

David Coronado
Licenciado en Matemáticas, Magister Scientiarum en Matemáticas. PhD. en Matemáticas. Profesor e Investigador en la Universidad Simón Bolívar, Caracas, Venezuela. Adscrito al Departamento de Computación y Tecnología de la Información. Correo: dcoronado@usb.ve - ORCID

Ricardo Monascal
Ingeniero en Computación. Magister Scientiarum en Computación. Profesor e investigador en la Universidad Simón Bolívar, Caracas, Venezuela. Adscrito al Departamento de Computación y Tecnología de la Información.
Correo: rmonascal@usb.ve - ORCID

Rosseline Rodríguez
Ingeniero en Computación. Magister Scientiarum en Computación. Profesora e investigadora en la Universidad Simón Bolívar, Caracas, Venezuela. Adscrita al Departamento de Computación y Tecnología de la Información.
Correo: crodrig@usb.ve - ORCID