Polygon Intersection Based Algorithm for Fuzzy Set Compatibility Calculations

Sukgamon Sukpisit, Supaporn Kansomkeat, Pannipa Sae Ueng, Apirada Thadadech, and Srđan Škrbić

Abstract—**PFSQL is an extension of the SQL language that allows usage of fuzzy logic in SQL queries. In query statements, variables can take both fuzzy and non-fuzzy values. Normally, different types of values cannot be compared directly. Therefore, it is necessary to implement fuzzy compatibility calculation to solve this problem. This paper proposes a method of fuzzy compatibility calculation implementation that determines compatibility degree of two fuzzy sets. The compatibility value is calculated using polygon intersection algorithm. To prove the correctness of the proposed method, the application has been developed and tested with 360 compatibility cases of different randomly generated fuzzy values. The experimental results show that our algorithms can handle various types of intersections between any two fuzzy sets.**

Index Terms—**Compatibility, fuzzy database, fuzzy query, PFSQL.**

I. Introduction

In the real world applications, some information might be vague, ambiguous, uncertain, imprecise or incomplete. Fuzzy logic has become a successful approach to handle this kind of information [1]. At the same time, methods of incorporating fuzziness into relational databases, such as fuzzy data models that are introduced by Ma *et al*. [2] and Vucetic *et al*. [3], are studied. In 2012, Škrbić and Racković introduced PFSQL (Prioritized Fuzzy Structured Query Language) that represents a set of extensions to SQL using priority fuzzy logic, together with a new fuzzy relational data model based on fuzzy extensions of the relational model [1].

PFSQL allows fuzzy logic concepts to be used in queries. Variables in query statements can be assigned both fuzzy and crisp values [4]. For example, a.wealth = triangle (13, 18, 20). Normally, a non-fuzzy value (a.wealth) and a fuzzy value (triangle (13, 18, 20)) cannot be compared directly because they are of different type. To solve this problem, the fuzzy compatibility calculations must be used. In this paper, we propose a method of implementation of fuzzy compatibility calculations between fuzzy sets. Our algorithm is capable of calculating intersection of every pair of the following types: triangular fuzzy number, trapezoidal fuzzy number, intervals, fuzzy shoulders and crisp values. This algorithm may then be used for wide spectrum of problems, but our interest is to use it for the implementation of different types of fuzzy queries. For example, it can be applied to the implementation of an interpreter for the fuzzy XQuery language proposed by Ueng and Škrbić in [5].

This paper is organized as follows. In the next section, we introduce the algorithms that we propose for compatibility calculations. Our implementation and testing results are presented in Sections III and IV, respectively. Section V is the conclusion.

Manuscript received May 20, 2015; revised January 18, 2016. Authors are partially supported by Ministry of Education and Science of the Republic of Serbia, through project no. ON 174023: Intelligent techniques and their integration into wide-spectrum decision support.

S. Sukpisit, S. Kansomkeat, and A. Thadadech are with the Department of Computer Science, Faculty of Science, Prince of Songkla University, Songkhla, Thailand (e-mail: sukgamon.s@psu.ac.th, supaporn.k@psu.ac.th, apirada.t@psu.ac.th).

P. S. Ueng and S. Škrbić are with the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Novi Sad, Serbia (e-mail: pannipa@dmi.uns.ac.rs, srdjan.skrbic@dmi.uns.ac.rs).

II. Compatibility Calculation

Our research focuses on five fuzzy types: triangular fuzzy numbers, trapezoidal fuzzy numbers, fuzzy shoulders, intervals and crisp values. In this section we describe the algorithm capable of determining the compatibility degree of two fuzzy sets of those types.

The compatibility calculation process is separated into three steps. First, the intersection area of two fuzzy sets is determined. Second, the size of the shape of the intersection area is calculated. Finally, a compatibility value is obtained using the compatibility equation.

A. Determining Intersection Area

An intersection area of two fuzzy sets is determined in 2-dimensions: vertical (x) and horizontal (y). We can assume that the shape of any characteristic function is a polygon. Each edge of a polygon can be transformed into linear equation ($y = mx + c$) and used for calculations in that form. For example, a fuzzy triangle shape has 3 coordinates: (*LeftOffset*, 0), (*Maximum*, 1) and (*RightOfsset*, 0). The bottom edge ($y = 0$) is not used for compatibility computation, so a fuzzy triangle have two edge-equations, *LeftEdge* and *RightEdge*. Table I shows coordinates and edge-equations of all characteristic functions used in this paper. A fuzzy trapezoidal shape has two edge-equations same as triangle and one additional edge-equation called *CenterEdge* which is simple – $y = 1$.

There are two types of fuzzy shoulder shapes – ascending or right shoulder and descending or left shoulder. There are two edge-equations in both types: *LeftEdge* and *RightEdge*. One edge-equation of them is constant depending on its type. A fuzzy interval is a line graph that starts from (*LeftMax*, 1) and ends at (*RightMax*, 1). The area below that line graph gives a rectangular shape that can be calculated easily but it is more complex to determine a common area with other shapes.

The main activity in determining intersection area step is the coordinate and edge-equations specification of the intersection area. The coordinates are transformed into objects. Each object provides *x* and *y* coordination called *Coordinate*. All objects are clockwise or counter clockwise added to a coordination table. The sequence of adding is important because non-consecutive adding may lead to incorrect results in the polygonal area calculation step.

This paper considers five types of fuzzy sets. For the sake of brevity, only the simplest case, triangle and triangle, is demonstrated in this paper. In this case, we only use left and right edges to find coordinates of the intersection area. Listing 1 shows the algorithm that calculates coordinates of the intersection area.

TABLE I: COORDINATES AND EDGE-EQUATION OF ALL CHARACTERISTIC FUNCTIONS

Characteristic functions	**Coordinates**	**Edge equations**
Triangle		
	(LeftOffset, 0)	LeftEdge, RightEdge
	(Maximum, 1)	
	(RightOffset, 0)	
Trapezoidal		
	(LeftOffset, 0)	LeftEdge, CenterEdge, RightEdge
	(LeftMaximum, 1)	
	(RightMaximum, 1)	
	(RightOffset, 0)	
Right Shoulder		
	(ZeroPoint, 0)	LeftEdge, RightEdge
	(Maximum, 1)	
	$(\infty, 1)$	
	$(\infty, 0)$	
Left Shoulder		
	(0, 0)	LeftEdge, RightEdge
	(0, 1)	
	(Maximum, 1)	
	(ZeroPoint, 0)	
Interval		
	(LeftMaximum, 0)	
	(LeftMaximum, 1)	
	(RightMaximum, 1)	
	(RightMaximum, 0)	
Crisp value		
	(*X*, *Y*)	

LISTING. 1: ALGORITHM FOR DETERMINING COORDINATES OF THE INTERSECTION AREA BETWEEN TWO FUZZY TRIANGLES

Algorithm GetCoordinates (FuzzyTriangle A, FuzzyTriangle B).

01. Compare *LeftOffset* and *RightOffset* of 2 fuzzy sets.
02. If there is intersection area
03. Store a coordinate ($LeftOffset_{max}$, 0).
04. Find a coordinate of interception of $LeftEdge_A$ and $LeftEdge_B$.
05. Find a coordinate of interception of $LeftEdge_A$ and $RightEdge_B$.
06. Find a coordinate of interception of $RightEdge_A$ and $LeftEdge_B$.
07. Find a coordinate of interception of $RightEdge_A$ and $RightEdge_B$.
08. Store coordinate ($RightOffset_{min}$, 0).
09. End if
10. End

The algorithm starts from checking intersection area of two fuzzy sets by comparing *LeftOffset* and *RightOffset* for the two. If there is no intersection area, the algorithm ends. If there is an intersection area, the first coordinate of it is ($LeftOffset_{max}$, 0). The $LeftOffset_{max}$ can be obtained by calculating the maximum value of $LeftOffset_A$ and $LeftOffset_B$. This coordinate is stored in the coordination table. In order to find and store coordinates in clockwise direction, we start by comparing $LeftEdge_A$ with $LeftEdge_B$. If these two edges overlap, the coordinates of the overlapping point is stored in the second row of the coordination table. To find and store the next three coordinates, method proceeds in the same manner. The last coordinate is obtained by calculating the minimum value of $RightOffset_A$ and $RightOffset_B$.

B. Calculating Intersection Area

In our research, the cyclic polygon calculation proposed in [6] is used to calculate the intersection area. This method uses coordinates of a polygon for the area calculations. The area is calculated by the following equation:

$$\text{Area} = \left|\frac{(x_1y_2 - x_2y_1) + (x_2y_3 - x_3y_2) + \cdots + (x_ny_1 - x_1y_n)}{2}\right| \tag{1}$$

To demonstrate the process, we describe compatibility calculation for two fuzzy sets: *triangleA* (12, 15, 18) and *triangleB* (14, 16, 17). The three attributes of a triangular fuzzy number are *LeftOffset*, *Maximum* and *RightOffset* respectively. Fig. 1 shows the fuzzy sets, *triangleA* and *triangleB*. There are four coordinates of the intersection of these two fuzzy sets. Fig. 2 shows coordinates of *triangleA* and *triangleB*, and their coordination table.

When you submit your final version, after your paper has been accepted, prepare it in two-column format, including figures and tables.

$$\left|\frac{((14\times 0.8) - (15.6\times 0)) + ((15.6\times 0.5) - (16.5\times 0.8)) + ((16.5\times 0) - (17\times 0.5)) + ((17\times 0) - (14\times 0))}{2}\right|$$

$$= 1.35$$

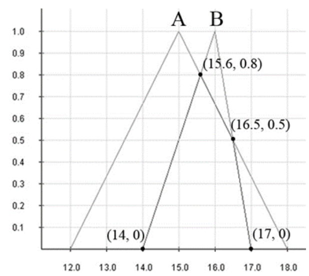


Fig. 1. *TriangleA* and *triangleB.*

Attr.	**X**	**Y**
LeftOffset	12	0
Maximum	15	1
RightOffset	18	0

triangleA

Attr.	**X**	**Y**
LeftOffset	14	0
Maximum	16	1
RightOffset	17	0

triangleB

X	**Y**
14	0
15.6	0.8
16.5	0.5
17	0

Coordination Table

Fig. 2. Coordinates of *triangleA* and *triangleB*, and their coordination table.

C. Calculating Compatibility Value

To obtain compatibility value, the compatibility equation [4] will be applied. The equation of compatibility of the fuzzy set A to the fuzzy set B is given below.

$$C = \frac{P(A \cap B)}{P(A)} \tag{2}$$

$P(A \cap B)$ is the area of intersection between the two fuzzy sets and $P(A)$ is the area of the source fuzzy set *A*.

Compatibility value is a number that varies from 0 to 1. Zero means incompatible, and one means fully compatible.

As stated before, we focus on five types of fuzzy sets. Each fuzzy set has a different shape of the characteristic function. An area of each shape can be obtained by mathematical methods. For example, the area of triangular fuzzy set can be calculated by $\left|\frac{\text{width} \times \text{height}}{2}\right|$.

In Fig. 1, the area of the intersection is 1.35. The area of the source fuzzy set (*triangleA*) is 3. Therefore, the compatibility value for these two fuzzy sets equals 0.45.

III. Implementation

To support our ideas, we developed the application that provides graphical user interface for calculation of the intersection area of two fuzzy sets and determining coordinates of the intersection area. Compatibility value is then obtained using this data. The application has two functions: manual testing and random testing. The manual testing function is used for a single test. User can identify two fuzzy sets and then this function will return the compatibility value. Fig. 3 shows the user interface for the manual testing function. The random testing function generates cases randomly. In this case, user can indicate type of fuzzy sets, boundary values and the number of generated cases. The boundary values include minimum value and maximum value. Fig. 4 shows the user interface of the random testing function.

Fig. 3. User interface of the manual compatibility testing application.

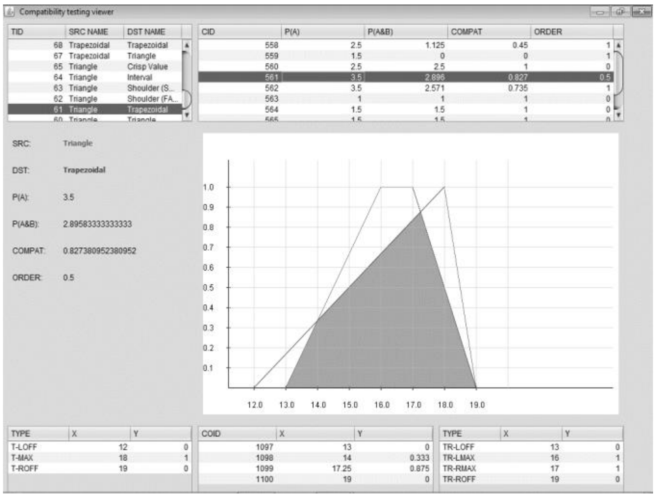


Fig. 4. User interface of the random compatibility testing application.

In our application, when the compatibility calculation processes are finished, the image of fuzzy sets and their intersection area will appear on the screen.

This application was developed on Java platform with the use of PostgreSQL to store fuzzy set attributes and cases of the random testing function.

IV. Testing Results

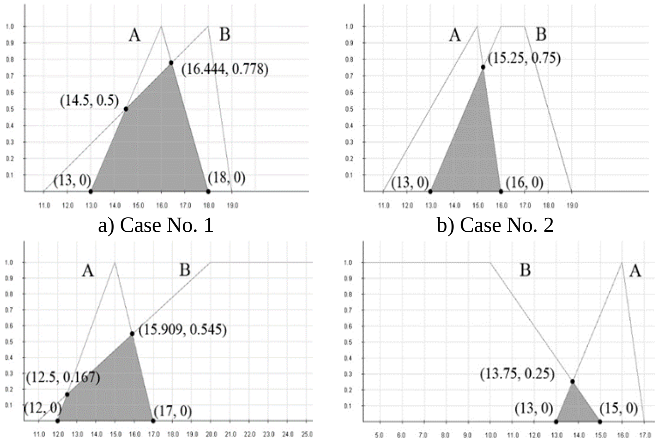
To prove the correctness of our algorithms, some compatibility cases are generated. To reduce the collecting bias, fuzzy sets are generated randomly using the described random testing function of our application. For thorough testing, each fuzzy type was compared with other types. For example, fuzzy triangle is compared with other four types including itself. There are two types of fuzzy shoulders, therefore there are six comparison pairs. Each pair has ten cases of compatibility testing. Totally, there are 360 compatibility cases in our experiment.

TABLE II: A Comparisons between Fuzzy Triangle and Other Types

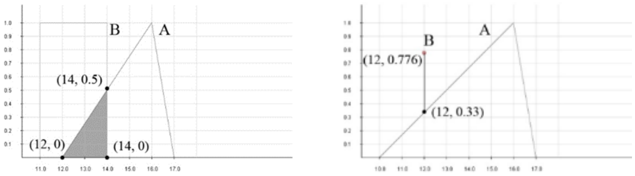
No.	Type of fuzzy A		P(A)	Type of fuzzy B		$P(A \cap B)$	C
	Attributes	Val.		Attributes	Val.		
1	Fuzzy triangle		2.5	Fuzzy triangle		2.2222	0.8889
	$LeftOffset_A$	13		$LeftOffset_B$	11		
	$Maximum_A$	16		$Maximum_B$	18		
	$RightOffset_A$	18		$RightOffset_B$	19		
2	Fuzzy triangle		2.5	Fuzzy trapezoidal		1.125	0.45
	$LeftOffset_A$	11		$LeftOffset_B$	13		
	$Maximum_A$	15		$LeftMax_B$	16		
	$RightOffset_A$	16		$RightMax_B$	17		
				$RightOffset_B$	19		
3	Fuzzy triangle		2.5	Fuzzy shoulder (FC)		1.553	0.6212
	$LeftOffset_A$	12		$ZeroPoint_B$	11		
	$Maximum_A$	15		$Maximum_B$	20		
	$RightOffset_A$	17					
4	Fuzzy triangle		2	Fuzzy shoulder (SB)		0.25	0.125
	$LeftOffset_A$	13		$Maximum_B$	10		
	$Maximum_A$	16		$ZeroPoint_B$	15		
	$RightOffset_A$	17					
5	Fuzzy triangle		2.5	Fuzzy interval		0.5	0.2
	$LeftOffset_A$	12		$LeftMax_B$	11		
	$Maximum_A$	16		$RightMax_B$	14		
	$RightOffset_A$	17					
6	Fuzzy triangle		3.5	Crisp value			0.3333
	$LeftOffset_A$	10		X_{crisp}	12		
	$Maximum_A$	16		Y_{crisp}	0.78		
	$RightOffset_A$	17					
7	Fuzz triangle		1.5	Fuzzy trapezoidal		0	0
	$LeftOffset_A$	10		$LeftOffset_B$	14		
	$Maximum_A$	11		$LeftMax_B$	17		
	$RightOffset_A$	13		$RightMax_B$	18		
				$RightOffset_B$	19		

If the range in the random testing function is too long, fuzzy sets can be positioned too far from each other and have no compatibility. To overcome this problem, the minimum and maximum values of boundaries are set to be 10 and 20, respectively.

Table II shows some cases of comparisons between fuzzy triangle and other types. The intersection areas of the cases 1 to 5 have 4, 3, 4, 3 and 3 coordinates, respectively. The last two cases have no intersection area. The shapes and coordinates of intersection area of each case are shown in Fig. 5.



a) Case No. 1 b) Case No. 2 c) Case No. 3 d) Case No. 4



e) Case No. 5 f) Case No. 6

Fig. 5. User interface of the random compatibility testing application.

V. Conclusion

This paper proposes the algorithm for fuzzy compatibility calculation of two fuzzy sets. The compatibility measure is used to compare two fuzzy sets. First we introduced the algorithms able to determine the intersection area between two fuzzy sets. After that, the compatibility calculation processes is explained and illustrated. Within the paper, the application that provides GUI for compatibility calculations is developed. The testing results are generated randomly by this application. In these results, various shapes of intersection areas are recognized correctly by our implementation. In this way we illustrated the power of the proposed algorithms to handle various types of intersections between any two fuzzy sets of the five fuzzy membership function types that we described.

It is our intent to use the proposed algorithms for compatibility calculations inside the interpreter for the fuzzy logic enriched XQuery language.

In the future, we plan to develop and implement the algorithms capable of calculating fuzzy ordering. Fuzzy ordering is important operation for queries that contain relational operators, as well as for those that contain aggregate functions like MIN, MAX and SUM.

References

[1] S. Škrbić, M. Racković, and A. Takači, "Prioritized fuzzy logic based information processing in relational databases," *Knowledge-Based Systems*, vol. 38, pp. 62-73, 2013.

[2] Z. M. Ma, F. Zhang, and L. Yan, "Fuzzy information modeling in UML class diagram and relational database models," *Applied Soft Computing*, vol. 11, pp. 4236-4245, 2011.

[3] M. Vucetic, M. Hudec, and M. Vujoševic, "A new method for computing fuzzy functional dependencies in relational database systems," *Expert Systems with Applications*, vol. 40, pp. 2738-2745 ,2013.

[4] S. Škrbić and M. Racković, *Fuzzy Databases*, Faculty of Sciences, University of Novi Sad, Novi Sad, 2013.

[5] P. S. Ueng and S. Škrbić, "Implementing XQuery fuzzy extensions using a native XML database," in *Proc. 13th IEEE International Symposium on Computational Intelligence and Informatics*, 2012, pp. 305-309.

[6] I. Pak, "The area of cyclic polygons: Recent progress on Robbins' conjectures," *Advances in Applied Mathematics*, vol. 34, pp. 690-696, 2005.

Sukgamon Sukpist is a master student at the Department of Computer Science, Faculty of Science, Prince of Songkla University, Songkhla, Thailand. He received his B.Sc. degree in information and communication technology from Prince of Songkla University, Songkhla, Thailand in 2010. His research interest is fuzzy XQuery. He is a software developer at the Computer Center, Prince of Songkla University since 2010.

Supaporn Kansomkeat is an assistant professor at the Department of Computer Science, Faculty of Science, Prince of Songkla University, Songkhla, Thailand. Her current research interests include software testing, test process improvement and fuzzy XQuery. She received a PhD in computer engineering from Chulalongkorn University in 2007. She was a publicity co-chair for the International Conference on Asia-Pacific Software Engineering Conference. She is a general secretariat for the 12th International Joint Conference on Computer Science and Software Engineering.

Pannipa Sae Ueng is a Ph.D. student at the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Novi Sad, Serbia. She received her B.Sc. degree in computer science from the Department of Computer Science, Faculty of Science, Prince of Songkla University, Songkhla, Thailand and M.Sc. degree in computer science from the Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand. She has worked as a lecturer at the Department of Computer Science, Faculty of Science, Prince of Songkla University since 2007. Her research interests are fuzzy database and fuzzy XQuery.

Apirada Thadadech is an assistant professor at the Department of Computer Science, Faculty of Science, Prince of Songkla University, Songkhla, Thailand. She received her M.Sc. degree in computer science from University of Philippines Losbanos, Philippines in 1990. In 2013, she was elected as a head of the Department of Computer Science, Faculty of Science, Prince of Songkla University, Songkhla, Thailand. Her research interest is software engineering.

Srđan Škrbić is an associate professor at the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Novi Sad, Serbia. He received his B.Sc. degree in computer science in 2001. M.Sc. in computer science in 2004 and Ph.D. in computer science in 2009 at the Faculty of Sciences, University of Novi Sad. In 2006, he received the "Mileva Maric – Einstein" prize for his M.Sc. thesis. He was elected as a head of the Chair for information technologies and systems in 2011. Recently he has focused on reseach topics in high performance and parallel scientific computing. He coauthored more than 50 research papers.