ISSN No. 0976-5697

Volume 3, No. 4, July- August 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Algorithms for Preparing Queries in Fuzzy Object-Oriented Database

Maryam kavoosi Balootaki*& Mashaala Abasi Dezfuli
Departemant of computer, Science and research Branch,
Islamic Azad University, Khouzestan, Iran
makavoosi@yahoo.com, abbasi_masha@yahoo.com

Ali Haroonabadi
Islamic Azad University, central Tehran Branch
Tehran, Iran
A.haroonabadi@gmail.com

Hassan Kavoosi Balootaki
Department of Mechanical Engineering,
Izeh Branch, Islamic Azad University, Izeh, Iran
Hakavoosi@yahoo.com

Abstract: In this paper we offer an approach for making queries from a fuzzy object-oriented database in the case that user adds some combined limitations (AND, OR) and assigns weights to them with respect to their importance and expects to obtain ideal answers. Then, using the proposed method the obtained results will be sorted in terms of their degree of dependency on the obtained answer set. Moreover, we suggest an approach enabling us to guarantee that each query will result in an answer even in the case that there is no element meeting all user defined limitations. This would be a beneficial approach in the case that it is impossible to meet all limitations and user expects a guaranteed answer. Implementing this approach will result in flexible queries and causes queries as well as databases to get closer to the real world. In this way, users will have easy interactions with database and the database will have better efficiency in response to fuzzy complex and combined queries.Implementation of this case study reveals that compared with crisp method the proposed approach results in more answers which are closer to real cases.

Keywords: object oriented database, fuzzy queries, AND, OR.

I. INTRODUCTION

Emerging object-oriented database resulted in introducing the concept of *behavior in database*. Before this type of databases, only relational databases were logically modeling data regardless the acts of system on data. The concept of *Existence* in the relational database was replaced by the concept of *Class* in the object-oriented database and attributes and acts were considered at the same time. On the other hand, since in the real world data are uncertain, this propelled researchers and authorities to the concept of *Fuzzy Database*. Applying fuzzy concept to database made it possible to utilize natural linguistic concepts in queries.
Combining fuzzy approach (in order to handle verbal uncertainty and variability) with object-oriented approach resulted in emerging a new strong database so called fuzzy object-oriented database. In this paper, we describe those queries which are presented as combined queries using verbal variables. The objects which are appeared in answers are sorted in terms of their degree of dependency.

II. RELATED STUDIES

Fuzzy set concept has been employed in databases as well as data recovery areas in recent 30 years [1]. Based on fuzzy sets and fuzzy alpha cut theories, [2] introduces general fuzzy queries against classic relational database and extends fuzzy queries interpretations. A research carried out in [3] presents an approach for database query in the case that the priorities of user are bipolar and the saved data in the database is inaccurate. In [4] based on fuzzy logic a comparative model has been presented for data recovery enabling users to recover accurate and special data once they sort them. [5] utilizes two μ_r and I_e complementary attributes for modeling respectively fuzzy concepts and unreliability. Also, it uses them to compute whether or not a *Tuple* is appropriate for a given query. Based on color ground [6] presents a fuzzy approach for image recovery and suggest fuzzy databases for saving and recovering ambiguous data.

III. PROPOSED METHOD

This paper presents an algorithm answering “And” & “Or” queries. It also suggests an approach which guarantees finding an answer. Figures 1, 2 and 3 respectively show the steps of algorithm1, algoritm2 and algorithm 3.
Assume the following query:

Q: $p_1(w_1)$ AND $p_2(w_2)$ AND...AND $p_n(w_n)$ (1)

In which Q is query, P_i is the i^{th} limit and W_i is the weight assigned by user to each limitation.

A. *Algorithm 1:*

a. In the First step consider the limitation with the maximum weight and answer it. Assume that the objects set of $\{O_1, O_2, O_3,\ldots,O_m\}$ is obtained in response to the limitations.
b. Calculate the degrees of dependency of the objects obtained in previous step and make $[\mu_{ij}]$ matrix in which i=1, 2,...m and j=1, 2,…n. μ_{ij} indicates the degree of dependency of the i^{th} object to the j^{th} limitation.
c. Compute the matrix value for each object.

$$O_i\ ,\ i=1,2,\ldots,m\ ,j=1,2,\ldots,n \qquad [\mu_{ij}* w_j] \quad (2)$$

d. T_{norm} calculates the following values. Since we wish to meet all limitations at the same time we use $T_{norm.}$ Here we assume T_{norm}=min

$$O_i\ ,\ i=1,2,\ldots,m\ ,j=1,2,\ldots,n$$
$$T^i{}_{norm}(\mu_{ij}* w_j) \quad (3)$$

e. Eliminate the objects with T_{norm} < = tershold

f. The rest objects are ranked by $T_{norm.}$
g. Objects with the same T_{norm} are ranked descending in terms of their dependency on limitation i.e. the more dependency the more weight.
h. Objects with the same degree of dependency on the first limitation, with respect to its weight, are ranked in terms of other limitations with respect to the order of their weights.

If the query follows (4), we suggest algorithm (2)

$Q:p_1(w_1)$ OR $p_2(w_2)$ OR …OR $p_n(w_n)$ (4)

In which P_i is the i^{th} limit, W_i is the weight assigned to the i^{th} limit and Q is query.

B. Algorithm 2:

a. Start from the limitation with the maximum weight and obtain its answer set. If the obtained answer set is an empty set obtain the answer set of the second limitation (in terms of its weight). Continue this procedure until obtaining a non null answer set.
b. Compute the degree of dependency of objects. Assume that the set of objects {O1,O2,…,Om} exists and $[\mu_{ij}]$ matrix, in which i=1, 2,..m and j=1, 2,…n, determines this dependency. μ_{ij} indicates the degree of dependency of the i^{th} object to the j^{th} limitation.
c. For each object calculate the following values:

$$O_i\ ,\ i=1,2,\ldots,m\ ,j=1,2,\ldots,n \quad [\mu_{ij} * w_j] \quad (5)$$

d. Calculate the S_{norm} of the following parameters. (Since it's not necessary to meet all limitations at the same time you can use S_{norm}). Here we assume that S_{norm}=Max.

$$O_i\ ,\ i=1,2,\ldots,m\ ,j=1,2,\ldots,n$$
$$S^i{}_{norm}\ (\mu_{ij} * w_j) \quad (6)$$

e. Eliminate the objects with S_{norm} <= tershold
f. Sort the rest of objects with respect to the value of $S_{norm.}$
g. Sort the objects with the same S_{norm} and the objects with the same degree of dependency on the first limitation (with respect to its weight) in terms of other limitations regarding the order of the assigned weights. This means that at first sort them with respect to the first limitation (the most important limitation) and then sort the objects with the same degree of dependency on this limitation in terms of the second limitation (with respect to of its importance). Continue this procedure until you sort all objects. Fig. 2 shows the idea of the algorithm 2:

C. Answer Guaranteed:

If in responding to OR queries the T_{norm} of all objects is equal to zero i.e. the answer set is null, we use the following algorithm.

a. Algorithm 3:

a) First of all you see a massage that tell you there is no object meeting the requirements of all limitations, are you wish to continue with the objects which meet some limitations? If user clicks OK the following process will be started:
b) The program starts with the limitation with the maximum weight and calculates its answer set. If the answer set of this limitation is null the answer set of the second limitation (with respect to its weight) is calculated and this will be continued until obtaining a non null answer set or until finishing the process of calculating the answer sets of all limitations (in this case no answer will be derived.)
c) The following value is calculated for all objects.

D_i = (7)

d) The objects with D_i <= tershold are eliminated.
e) The rest of objects are sorted in terms of their degree of dependency on the most important limitation and the objects with the same degree of dependency on this limitation are sorted in terms of the second important limitation. This procedure will be continued until sorting all objects.

D. Case Study:

Example 1:

Assume that a class has been defined according to Fig. 4

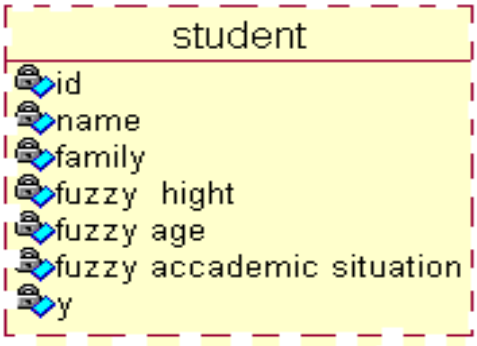

Figure. 2: a fuzzy class defined for students We wish to make this query:

"Find the students who are short(w_1 = 0.8) AND weak in lessons(w_2 = 0.2).",(tereshold =0)

The degrees of dependency of objects are in accordance with table 1. Based on the algorithm 2 the following order (left to right) is obtained. ({O4} is eliminated as its T_{norm}=0):

O_5, O_2, O_3, O_6, O_1

Example 2:

"Find the students who are short (w_1 = 0.8) OR weak in lessons(w_2 = 0.2)." , (tereshold =0)

The values are calculated through the mentioned formula and inserted in table 2. First of all the objects are sorted in terms of S and the order (left to right) would be:

$O_1, O_2, \{O_3, O_4, O_5, O_6\}$

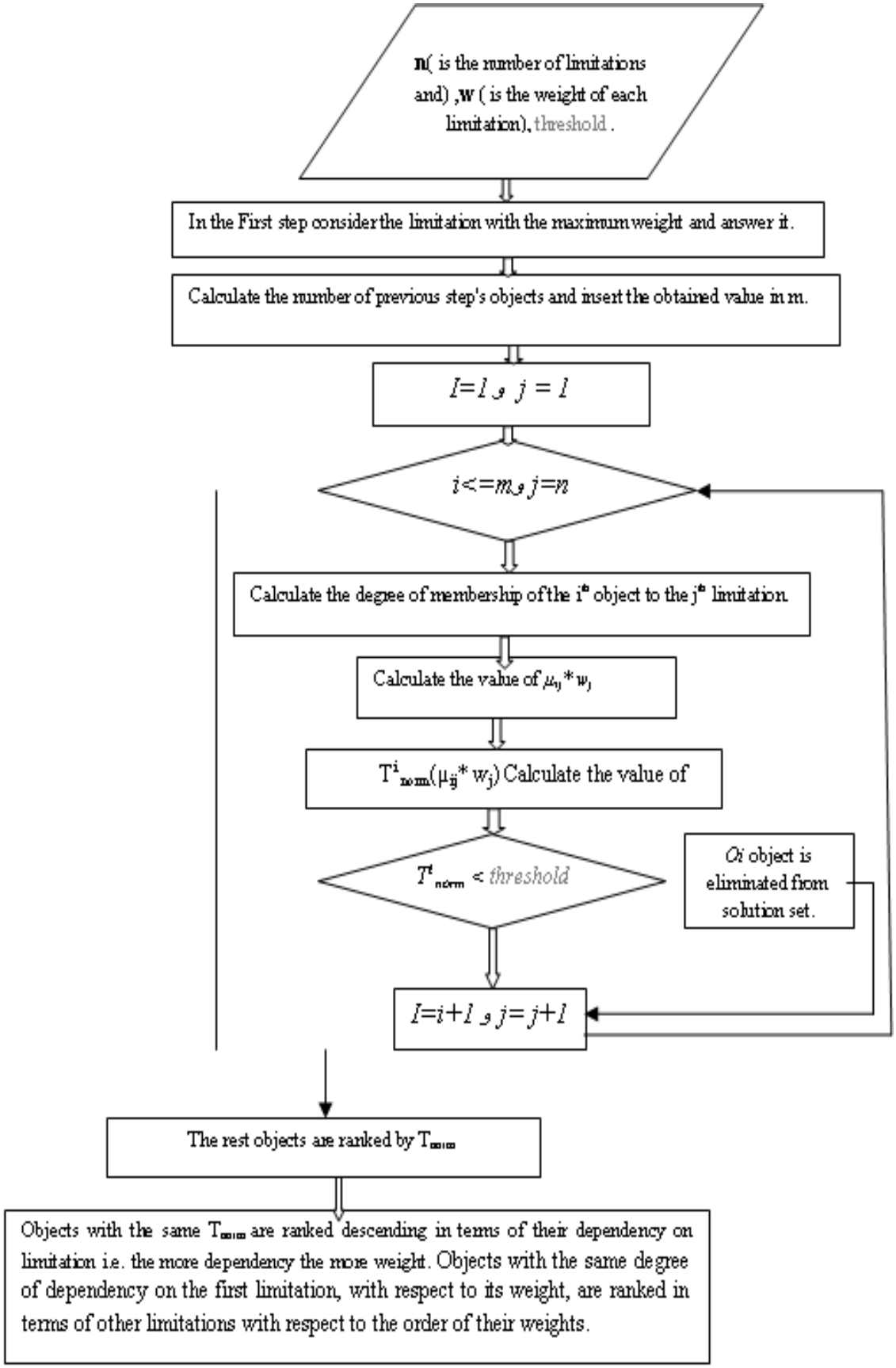

Figure. 1: the idea of the algorithm 1

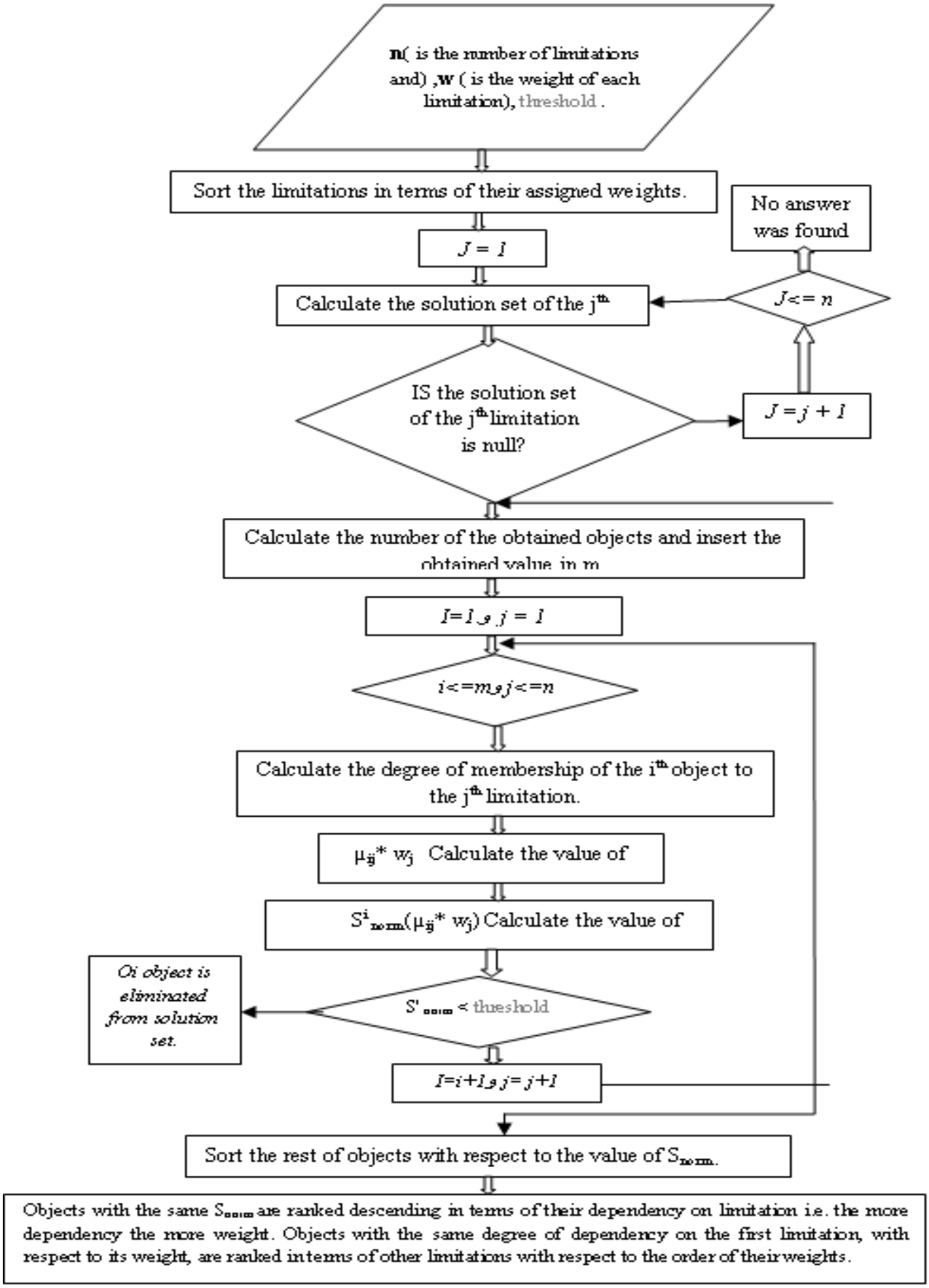

Figure: 3 the idea of the algorithm 2

The objects with the same S_{norm} are sorted in terms of the most important limitation resulting in the following order (left to right):

$O_1, O_2, \{O_3, O_4, O_5, O_6\}$

In next step, the objects with the same degree of dependency on the most important limitation are sorted in terms of their degrees of dependency on the second important limitation (regarding its weight) and the following order is derived:

$O_1, O_2, O_4, O_5, O_3, O_6$

Example 3:

"Find the students who are older(w_1=0.8) AND weak in lessons(w_2 =0.2)". , (tereshold= 0)

The calculated values for this query are in accordance with table 3. As you can see, in all objects we have: T_{norm}=0. So, we use the algorithm 3 and the following order (left to right) is obtained:

O_1, O_3

(In the O_2 object we have D_2=0; so it has been eliminated.)

Table 1: the values computed by algorithm 1 for example 1.

object	$\mu_{i1} * w_1$	$\mu_{i2} * w_2$	T^i_{norm} ($\mu_{ij} * w_j$)
O_1	0.32	0.2	0.2
O_2	0.64	0.12	0.12
O_3	0.8	0.16	0.16
O_4	0.8	0	0
O_5	0.8	0.08	0.08
O_6	0.8	0.18	0.18

Table 2: The Values Computed by Algorithm 2 for Example 2.

object	$\mu_{i1} * w_1$	$\mu_{i2} * w_2$	S^i_{norm} ($\mu_{ij} * w_j$)
O_1	0.32	0.2	0.32
O_2	0.64	0.12	0.64
O_3	0.8	0.16	0.8
O_4	0.8	0	0.8
O_5	0.8	0.08	0.8
O_6	0.8	0.18	0.8

Table 3: The Values Computed by Algorithm3 for Example 3.

object	$\mu_{i1} * w_1$	$\mu_{i2} * w_2$	T^i_{norm}	D_i =
O_1	0.8 *0 = 0	1* 0.2 = 0.2	0	0.2
O_2	0.8* 0 = 0	0 * 0.2 = 0	0	0
O_3	0.8 * 0.7 = 0.56	0 * 0.2 = 0	0	0.56

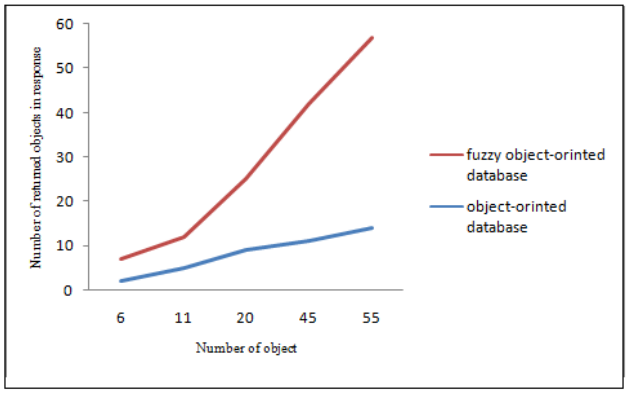

Figure: 4 Comparison between the returned answers of object-oriented database and fuzzy object-oriented database.

IV. CONCLUSION

In Fuzzy object-oriented database it is possible to use linguistic words in the range of users' vocabulary for users' better communication purposes. In this paper a noble approach was proposed for answering (AND, OR) queries in which users assign different weights to the different sections of queries with respect to the importance degrees of the sections. This approach also guarantees the existence of an answer. This approach may result in flexible databases and causes databases to get closer to the real world. Implementation of this case study reveals that compared with crisp method the proposed approach results in more answers which are closer to real cases. Fig. 5 shows this matter.

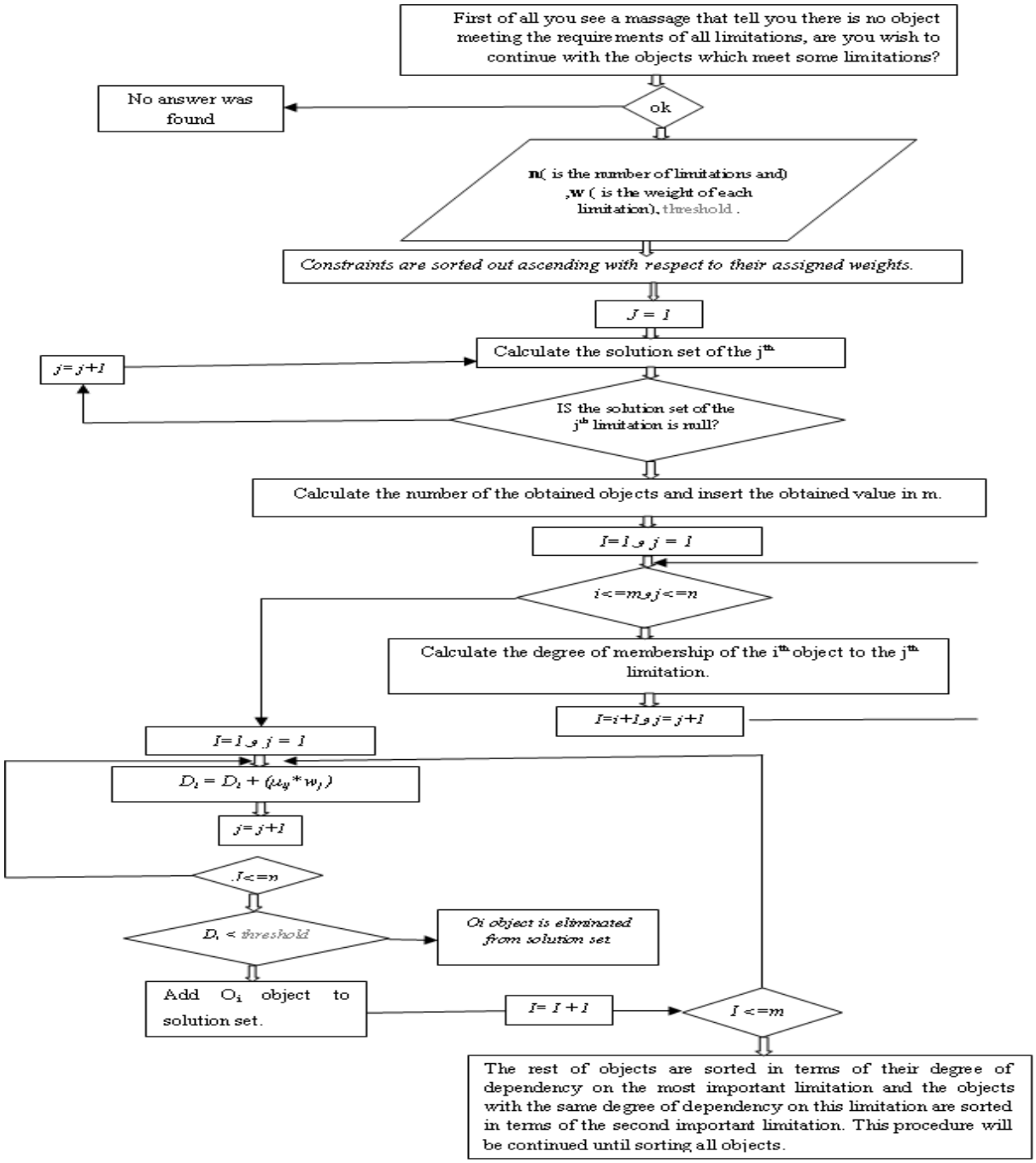

Figure: 5 The Idea Of The Algorithm 3

V. REFERENCES

[1] Bosc, Patrick., Kraft, Donald., Petry, Fred., "Fuzzy sets in database and information systems: Status and opportunities", Fuzzy Sets and Systems, Volume 156, Issue 3, 16 December 2005, Pages 418-426.

[2] Ma, Z.M., Yan, Li., "Generalization of strategies for fuzzy query translation in classical relational databases", Information and Software Technology, Volume 49, Issue 2, February 2007, Pages 172-180.

[3] Patrice Buche, SébastienDestercke, ValérieGuillard, "A flexible bipolar querying approach with imprecise data and guaranteed results", Fuzzy Sets and Systems, Volume 169, Issue 1, 16 April 2011, Pages 51-64.

[4] Aderounmu, G.A., Ajayi, A.O., Soriyan, H.A., " An adaptive fuzzy information retrieval model to improve response time perceived by e-commerce clients", Expert Systems with Applications, Volume 37, Issue 1, January 2010, Pages 82-91.

[5] Chiang, Ding-An., Chow, Louis R., Hsien, Nan-Chen., "Fuzzy information in extended fuzzy relational databases", Fuzzy Sets and Systems, Volume 92, Issue 1, 16 November 1997, Pages 1-20.

[6] Barranco, C.D., Chamorro-Martínez, J., Galán-Perales, E., Medina, J.M., Soto-Hidalgo, J.M.," Retrieving images in

fuzzy object-relational databases using dominant color descriptors", Fuzzy Sets and Systems, Volume 158, Issue 3, 1 February 2007, Pages 312-324.

Short Bio Data for the Author

Maryam Kavoosi Balotaki received the B.S degree in compyter Engineering from the shahid chamran ahvaz university and M.S degree in computer Engineering from ahvaz science and research branch in iran, in 2007.

Ali Haroonabadi received the PH.D degree in computer engineering from the tehran science and research branch in iran, he is faculty member at the computer department of Islamic Azad University, central Tehran Branch. He is a member of the IEEE and springer.

Mashaala Abasi Dezfuli is PHD in computer engineering. He is faculty member at the computer department of Departemant of computer, Science and research branch, Islamic Azad University, Khouzestan, Iran.