Towards better semantics in the multifeature querying

Peter Gurský

UPJŠ, Košice, Slovakia

http://klud.ics.upjs.sk/~gursky

Multifeature querying

The aim is to find top k objects in a potentially huge set of objects using the minimal number of accesses to the sources.

The decision which object is better than the other, is based on the properties of the objects

Usual types of properties:

1. yes/no
   - single, breakfast included, aspect at the sea, Springer proceedings, ...
2. graded to finite number of classes
   - number of stars of hotels, quality of an article, level of education, ...
3. real or integer number
   - salary, price, number of pages, properties in multimedia databases, date, ...
4. text information – derive other properties / reduce searching space
   - first name (→ gender), name of a town (→ distance),... ; ontology object,...

Motivation example

Example 1:

We want to find 5 best accommodations for a conference that are especially cheap, rather near to the center and we prefer new buildings before older ones.

$F(near,cheap,new)=2*near_i+3*cheap_i+new_i$

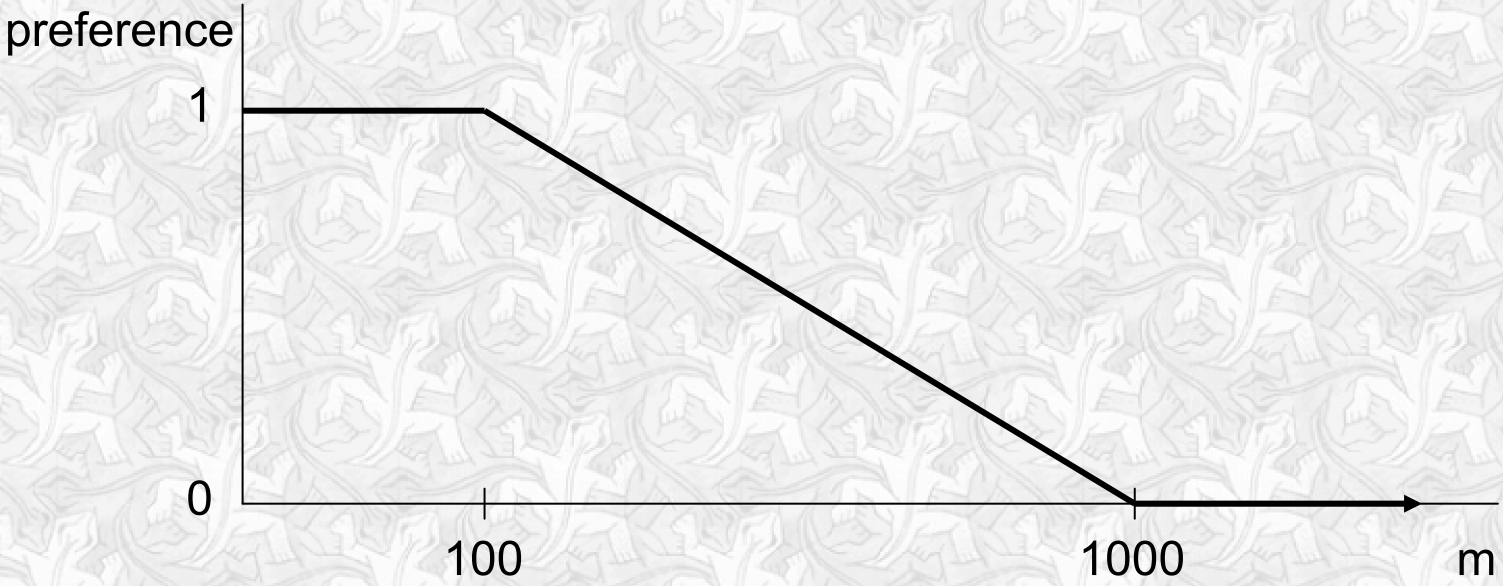
Example 2:

Give me the results of the searching for my key words from portals yahoo, google and seznam.

$F(yahoo,google,seznam)=yahoo_i+2*google_i+2*seznam_i$

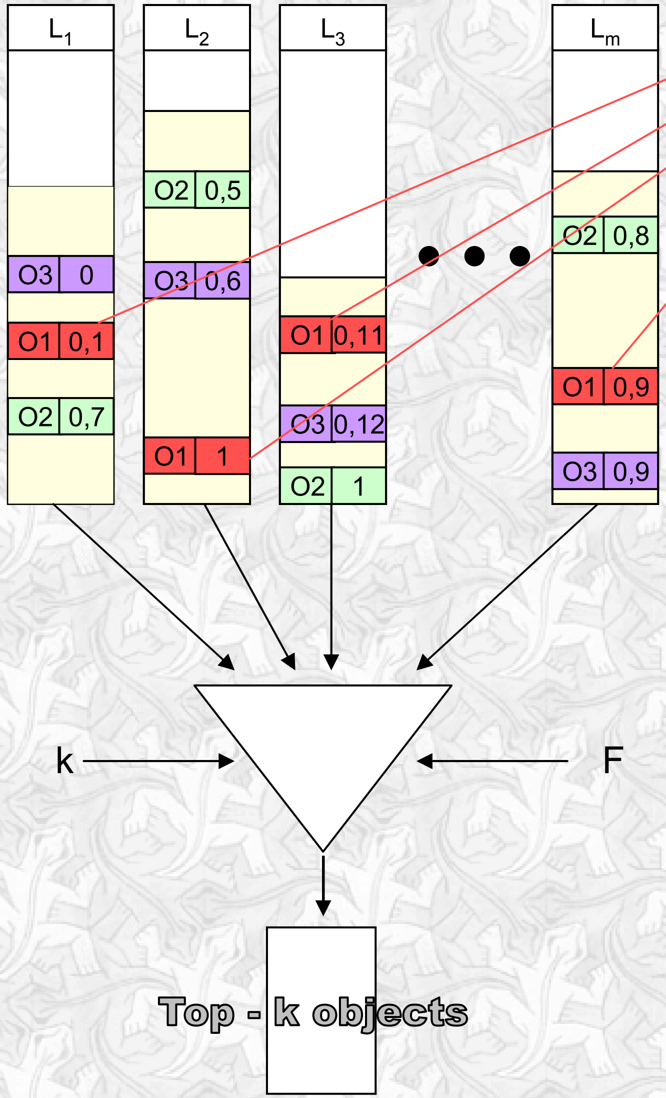
Motivation

What the term “near” means?



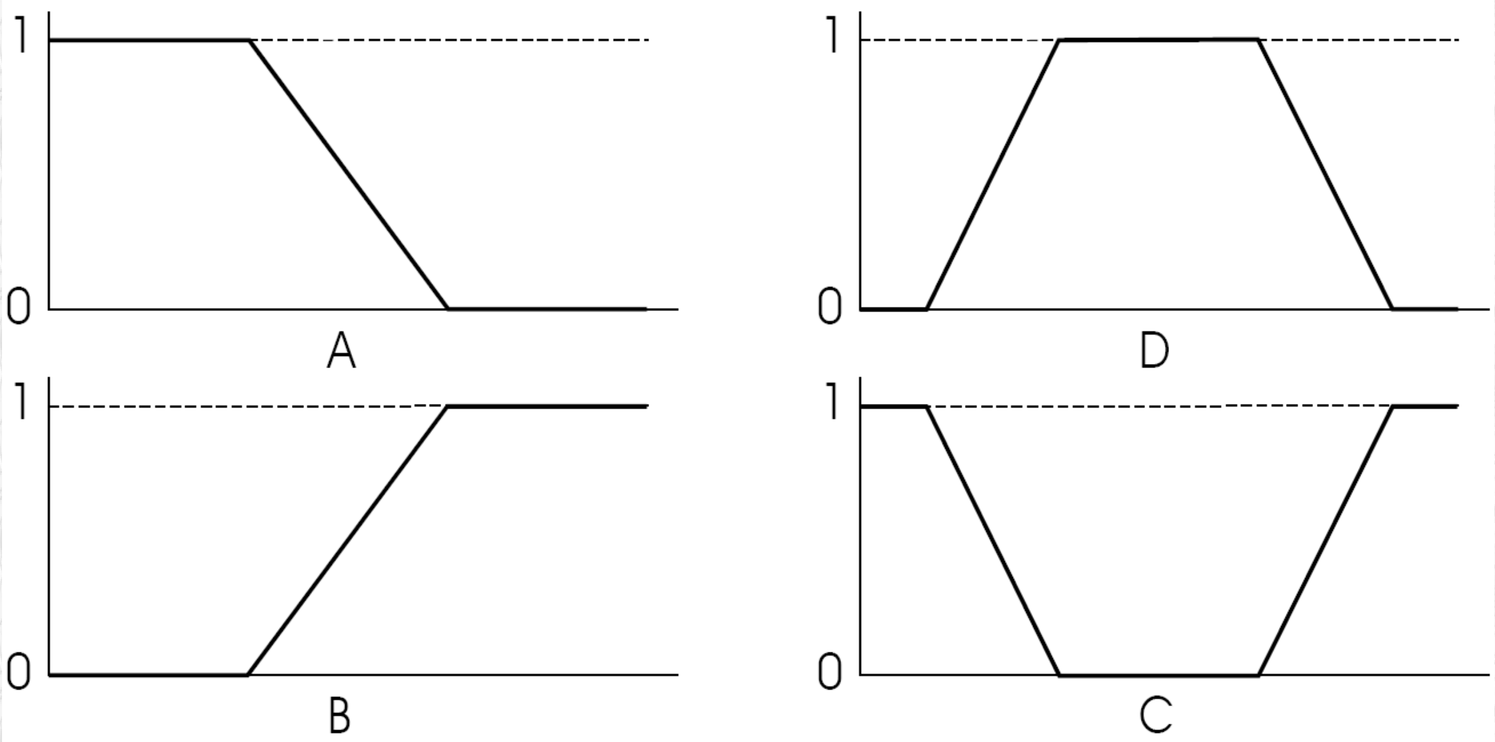
Basic Model

Best-Worst ordering



Fuzzy functions

What the “good distance from the center” means?



Problem

- All current solutions assume, that the sources send their property information ordered from the best value to the worst value.
- None of these solutions allow **the user to specify**, which values of a property are better than the other.
- We discuss about the possibility of preference specification and effective retrieval of top k objects.
- We start with 2 main current approaches and upgrade them

Two types of accesses

- **Sorted access**:
  - return the next greatest value from the i-th list together with a name of an object
- **Random access**:
  - return the value of an object from the i-th list

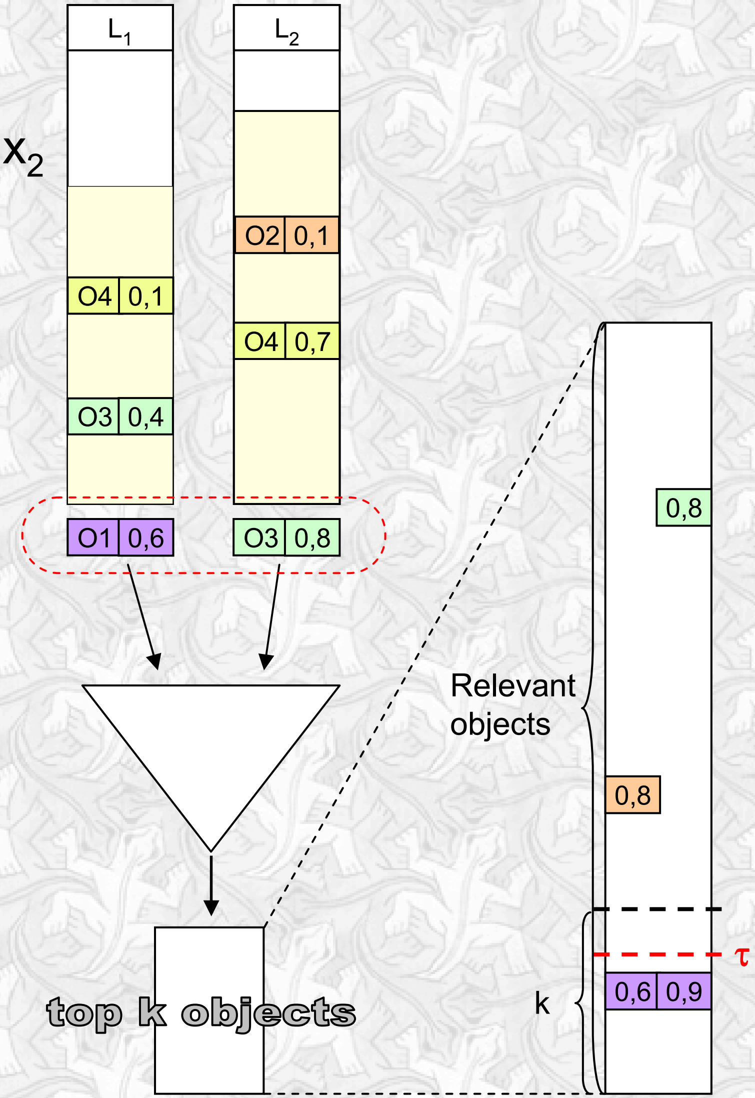
Basic terms

Aggregation function:

$S(X) = F(x_1,x_2) = 2x_1+x_2$

Threshold:

$\tau = F(0.6,0.8)=2.0$



Best function B(X)

Worst function W(X):

S(O1) = F(0.6,0.9) = 1.2+0.9 = 2.1
W(O1) = F(0.6,0.9) = 1.2+0.9 =**2.1**
B(O1) = F(0.6,0.9) = 1.2+0.9 =2.1

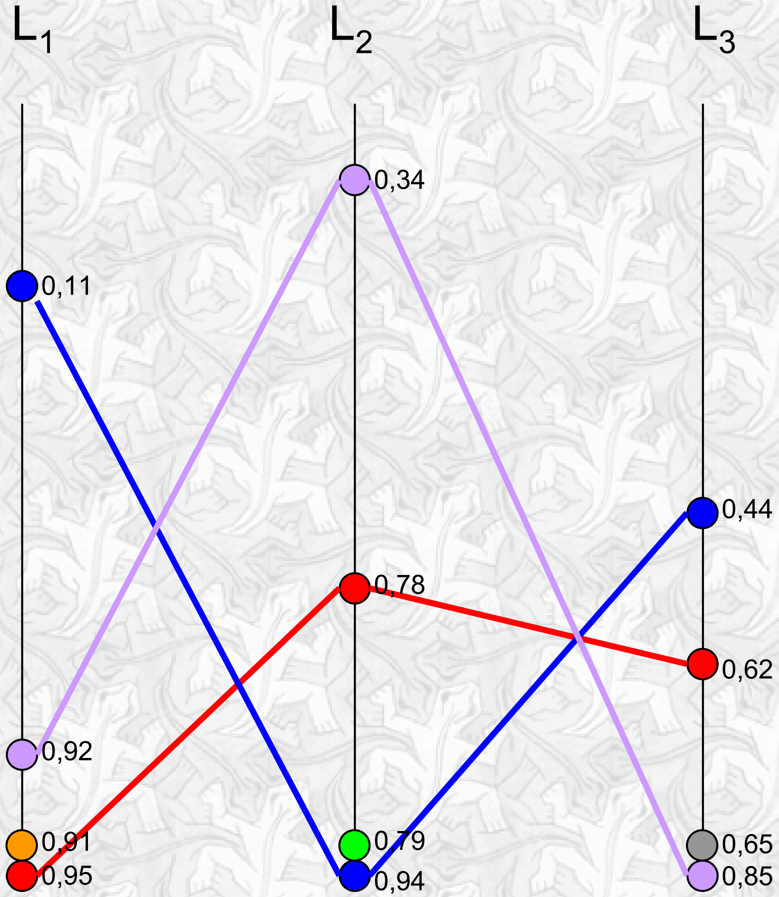
S(O2) = F(0.8,?) = ?
W(O2) = F(0.8,0) = 1.6+0 = **1.6**
B(O2) = F(0.8,0.8) = 1.6+0.8 =2.4

S(O3) = F(?,0.8) = ?
W(O3) = F(0,0.8) = 0+0.8 =**0.8**
B(O3) = F(0.6,0.8) = 1.2+0.8 =2.0

S(O4) = F(?,?) = ?
W(O4) = F(0,0) = **0**
B(O4) = F(0.6,0.8) = 1.2+0.8 =2.0

Threshold algorithm (Fagin 1999)

$F(x_1,x_2,x_3)=2*x_1+3*x_2+x_3$



τ=2*0,91+3*0,79+0,65=**4,84**

Fo_1= 2*0,95+3*0,78+0,62=**4,86**

Fo_2= 2*0,11+3*0,94+0,44=**3,48**

Fo_3= 2*0,92+3*0,34+0,85=**3,71**

3P-NRA (3-phased NRA)

First phase

Goes to all list and constructs the list of relevant objects until the worst value of the kth best object is greater or equal to the threshold.

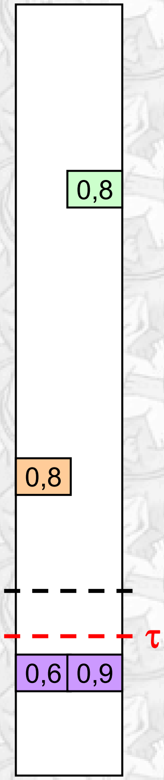
Second phase

Computes best values of all relevant objects. If the value is smaller than the worst value of the kth best object, removes the object from relevant.

Third phase

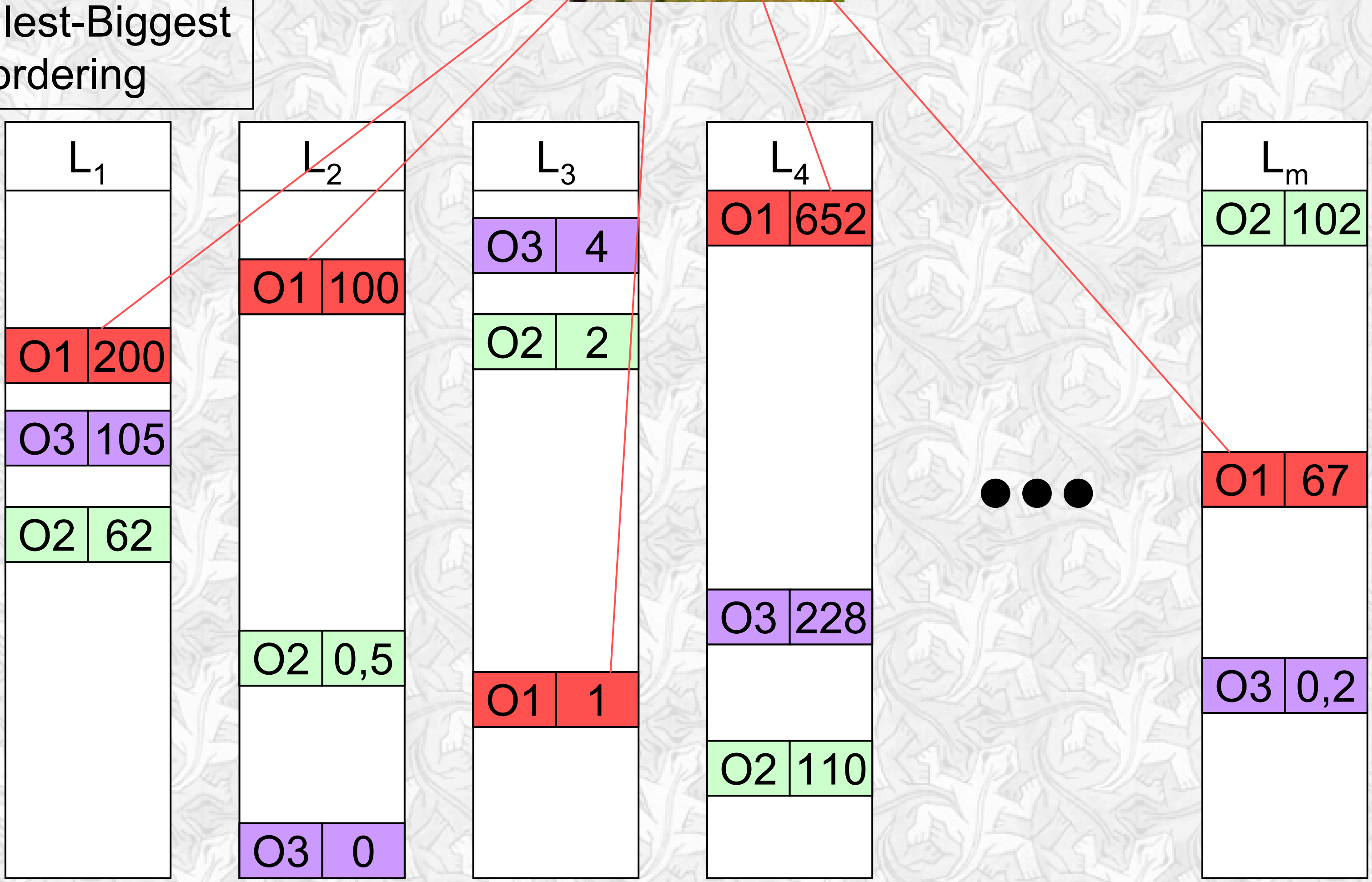
Continues descents to the lists to find other values of the seen objects. Checks the new best values and remove small ones. When threshold falls it goes to the second phase.

Algorithms ends when we have k relevant objects.



Model

Smallest-Biggest ordering



L_1	
O1	200
O3	105
O2	62

L_2	
O1	100
O2	0,5
O3	0

L_3	
O3	4
O2	2
O1	1

L_4	
O1	652
O3	228
O2	110

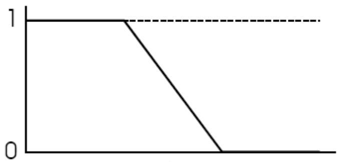
•••

L_m	
O2	102
O1	67
O3	0,2

Fuzzy values

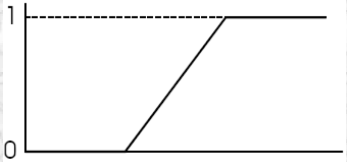
Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Real value	0	1	2	3	4	5	6	7	8	9

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	1	1	0.8	0.6	0.4	0.2	0	0	0



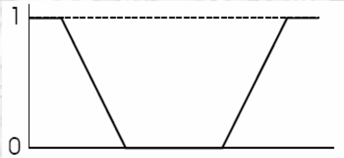
A

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0	0	0.2	0.4	0.6	0.8	1	1	1



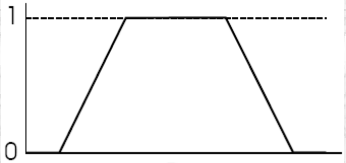
B

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	0.9	0.6	0.3	0	0	0.3	0.6	0.9	1



C

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0.1	0.4	0.7	1	1	0.7	0.4	0.1	0



D

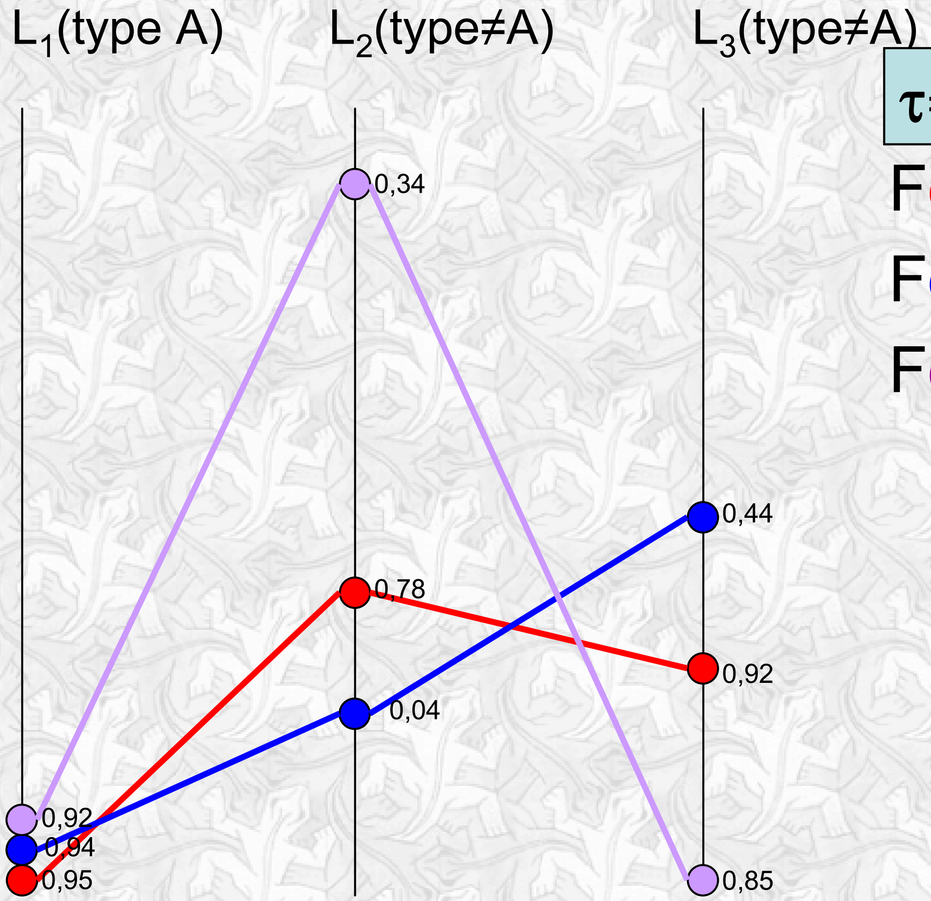
Solution 1

Restricted sorted access

- Bruno, Gravano, and Marian
- For the problem when it is not possible to access some of lists by sorted access
- Can be used to solve our problem
- We need at least one list with the fuzzy function of type A, so we can do the sorted accesses to such lists because the ordering by fuzzy values is the same as by real values
- To the lists with fuzzy function of type B, C or D we do only random accesses
- Threshold falls only with the values of the lists with fuzzy function of type A

Solution 1

$\boxed{F(x_1,x_2,x_3)=2*x_1+3*x_2+x_3}$



$\boxed{\tau=2*0{,}92+3*1+1=\mathbf{7{,}84}}$

$Fo_1= 2*0{,}95+3*0{,}78+0{,}92=\mathbf{5{,}16}$

$Fo_2= 2*0{,}94+3*0{,}04+0{,}44=\mathbf{2{,}44}$

$Fo_3= 2*0{,}92+3*0{,}34+0{,}85=\mathbf{3{,}71}$

Solution 2

Reading whole list or waiting for a maximum

- We will read all the lists that have fuzzy functions of types B or C. Next we will read all the lists that have fuzzy functions of type D until they grow to the maximum fuzzy value.
- We can save accesses mainly to the lists with fuzzy function of type A and partially in the lists with type D.
- This solution can be helpful especially when we extend 3P-NRA (no random access) algorithm.
- We will add a phase zero before the algorithm 3P-NRA

Fuzzy values

$F(x_1,x_2,x_3,x_4)=x_1+x_2+x_3+x_4$

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Real value	0	1	2	3	4	5	6	7	8	9

A

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	1	1	0.8	0.6	0.4	0.2	0	0	0

B

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0	0	0.2	0.4	0.6	0.8	1	1	1

C

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	0.9	0.6	0.3	0	0	0.3	0.6	0.9	1

D

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0.1	0.4	0.7	1	1	0.7	0.4	0.1	0

Fuzzy values

$F(x_1,x_2,x_3,x_4)=x_1+x_2+x_3+x_4$

$\tau=1+0+1+1=\mathbf{3}$

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Real value	0	1	2	3	4	5	6	7	8	9

A

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	1	1	0.8	0.6	0.4	0.2	0	0	0

B

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0	0	0.2	0.4	0.6	0.8	1	1	1

C

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	0.9	0.6	0.3	0	0	0.3	0.6	0.9	1

D

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0.1	0.4	0.7	1	1	0.7	0.4	0.1	0

Object	best	worst
O8	4	1
O9	4	1
O10	4	1
O7	3.8	0.8
O6	3.6	0.6
O5	3.4	0.4
O4	3.2	0.2
O1	3	0
O2	3	0
O3	3	0

Fuzzy values

$F(x_1,x_2,x_3,x_4)=x_1+x_2+x_3+x_4$

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Real value	0	1	2	3	4	5	6	7	8	9

A

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	1	1	0.8	0.6	0.4	0.2	0	0	0

B

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0	0	0.2	0.4	0.6	0.8	1	1	1

C

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	0.9	0.6	0.3	0	0	0.3	0.6	0.9	1

D

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0.1	0.4	0.7	1	1	0.7	0.4	0.1	0

$\tau=1+0+0+1=\mathbf{2}$

Object	best	worst
O10	4	2
O9	3.9	1.9
O8	3.6	1.6
O7	3.1	1.1
O1	3	1
O2	2.9	0.9
O6	2.6	0.6
O3	2.6	0.6
O4	2.5	0.5
O5	2.4	0.4

Fuzzy values

$F(x_1,x_2,x_3,x_4)=x_1+x_2+x_3+x_4$

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Real value	0	1	2	3	4	5	6	7	8	9

A

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	1	1	0.8	0.6	0.4	0.2	0	0	0

B

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0	0	0.2	0.4	0.6	0.8	1	1	1

C

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	0.9	0.6	0.3	0	0	0.3	0.6	0.9	1

D

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0.1	0.4	0.7	1	1	0.7	0.4	0.1	0

$\tau=1+0+0+1=\mathbf{2}$

Object	best	worst
O10	4	2
O9	3.9	1.9
O8	3.6	1.6
O5	2.4	1.4
O4	2.2	2
O7	3.1	1.1
O1	2	1
O2	2	1
O3	2	1
O6	2.6	0.6

Solution 3

Two ways descending

- The same performance in the case of each fuzzy function type as the algorithms TA and 3P-NRA in the original task.
- We need lightly upgrade the functionality of data sources:
  - A source will provide two lists for sorted access - first will send objects with property values ordered from the biggest to the smallest (descending order) and second will send data from the smallest to the biggest (ascending order).
  - Lists can start sending data from the specified value.

Fuzzy values

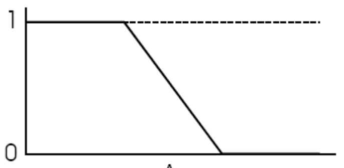
Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Real value	0	1	2	3	4	5	6	7	8	9

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	1	1	0.8	0.6	0.4	0.2	0	0	0

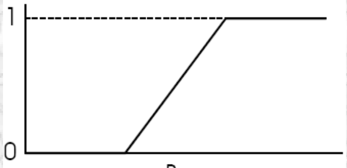
Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0	0	0.2	0.4	0.6	0.8	1	1	1

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	1	0.9	0.6	0.3	0	0	0.3	0.6	0.9	1

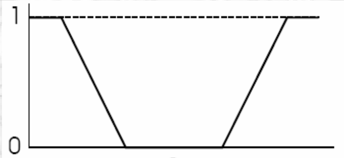
Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0.1	0.4	0.7	1	1	0.7	0.4	0.1	0



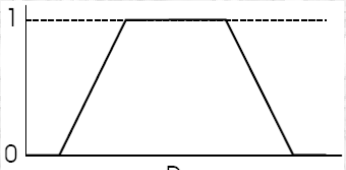
A



B



C



D

Conbination of 2 lists

Object	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Fuzzy value	0	0.1	0.4	0.7	1	1	0.7	0.4	0.1	0

O7
O4
O6
O5

Conclusion

- We extended the model of distributed multifeature querying by adding user specification of preferences to properties values.
- Proposed solutions are needed especially in the cases when we cannot reorder the lists in provided sources.

Thank you for your attention.

http://klud.ics.upjs.sk/~gursky