NSWI166 Introduction to recommender systems and user preferences

Jednu 1/3 Peter Vojtáš, KSI MFF UK

dvě 1/3 Láďa Peška, KSI MFF UK

6/12 **Querying - top-k** Fagin-Lotem-Naor class of models

Outline of this lecture

Information models and ordering – seen from the point of view of RS=recommender systems and UP=user preferences

Querying - top-k – FLN=Fagin-Lotem-Naor class of models

- Web service motivation (?also in RS and UP)
- FLN data model
- FLN data model – viewed as LMPM
- FLN TA=threshold algorithm for top-k
- FLN TA seen geometrically (illustrative 2D in LMPM)
- FLN data model and W3C RDF (web resources)
- FLN TA heuristics

Motto: "The purpose of abstraction is not to be vague, but to create a new semantic level in which one can be absolutely precise."

— Edsger W. Dijkstra, "The Humble Programmer" 1972 ACM Turing Lecture, see [Human-Centered Approach to Static-Analysis-Driven Developer Tools]

Motivation – new aspects

- **so far** from "no match" to "close", multicriterial
  - Ideal values separately for attributes – conflicts
  - Self explainable → visual / geometric

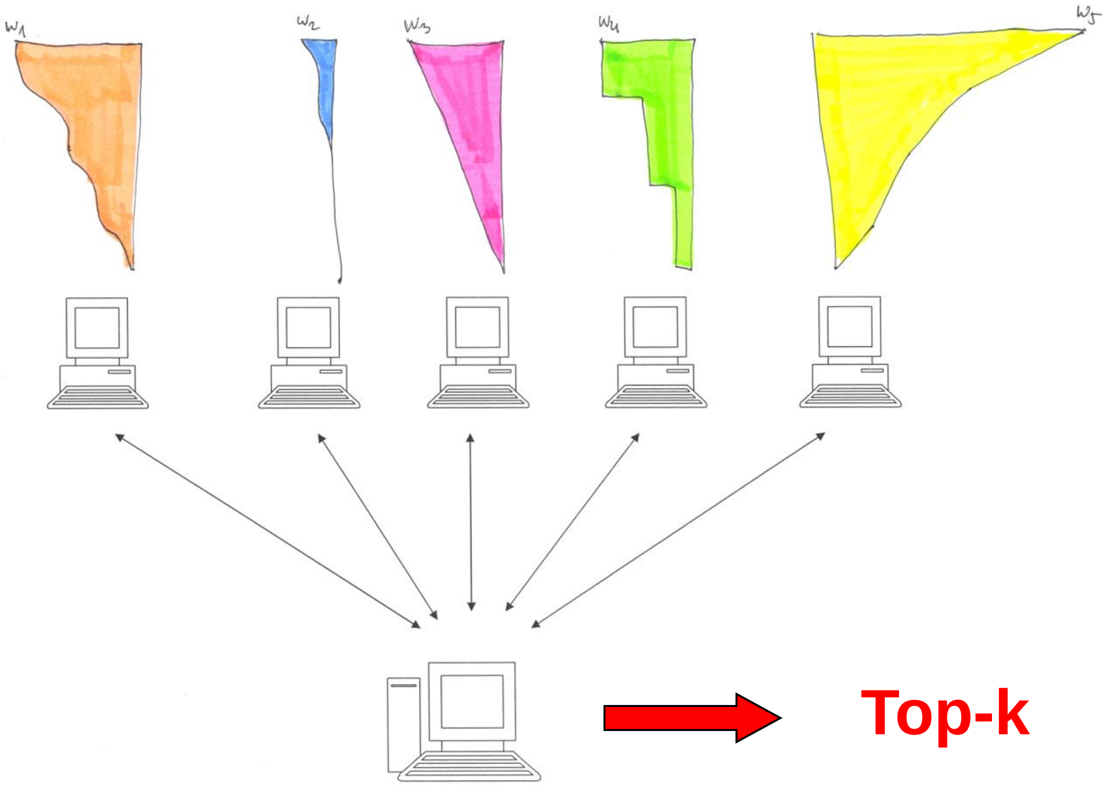
    ↓ ↓

    tableaux → Lean Startup (your idea)
- **new aspect** – attributes distributed/external sources
  - [BGM02] N. Bruno, L. Gravano, A. Marian. Evaluating Top-k Queries over Web-Accessible Databases, *ICDE 2002 - International **C**onference on **D**ata **E**ngineering, San Jose,*
    - Goal: Find best restaurants for a user:
    - Close to address: "2290 Broadway"
    - Price around $25
    - Good rating

Web services – access mode – data types

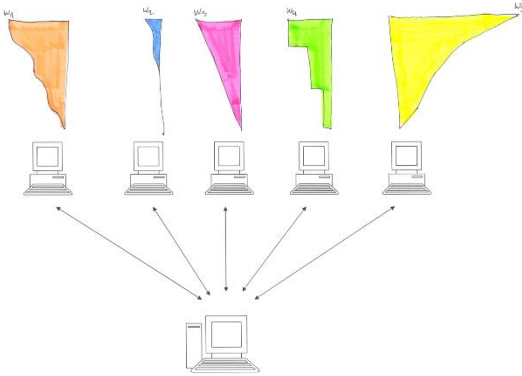
- *MapQuest* returns the distance between two addresses.
- *NYTimes Review* gives the price range of a restaurant.
- *Zagat* gives a food rating to the restaurant.
- We follow paper [FLN] R. Fagin, A. Lotem, M. Naor, Optimal aggregation algorithms for middleware. Journal of Computer and System Sciences 66 (2003) 614–656 JCSS2003
  - Access mode – sorted, direct (random), stateless, …
  - From multimedia middleware (IBM Almaden Garlic project) top-k optimal querying to our multiuser LMPM

Data model Fagin – Lotem - Naor



We need method/prototype/algorithm for top-k queries
Possibly without scanning whole data ? “order-by” ?

Data model Fagin-Lotem-Naor-FLN

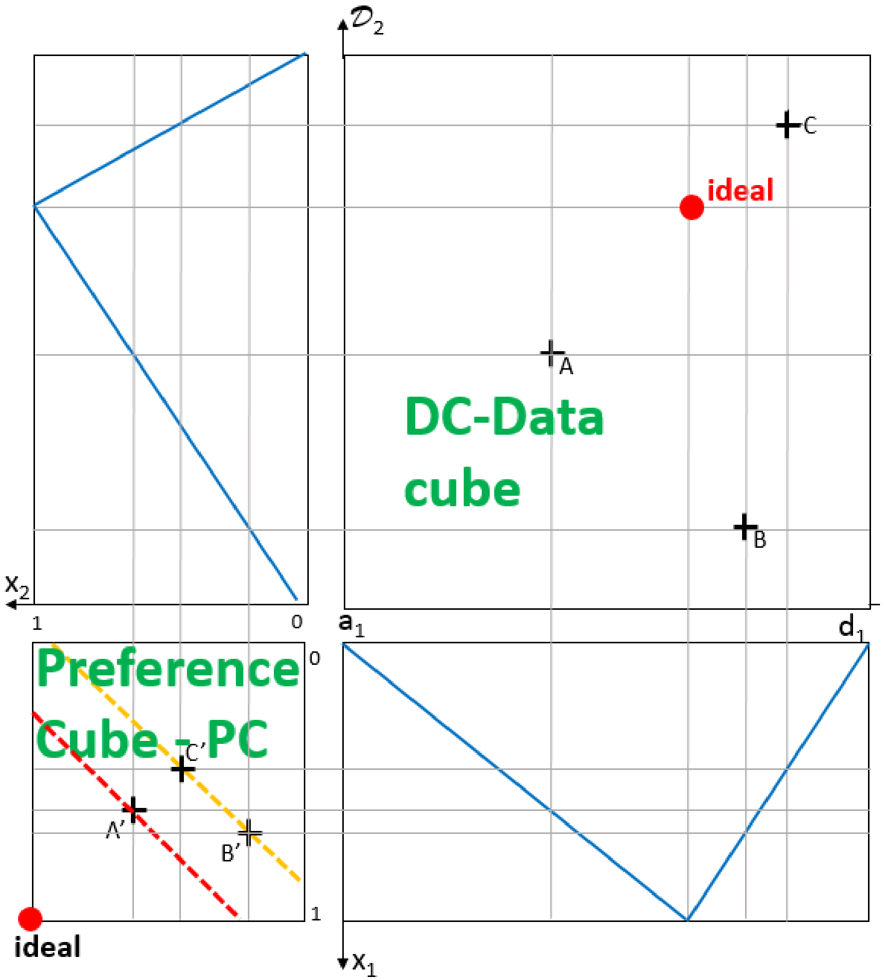


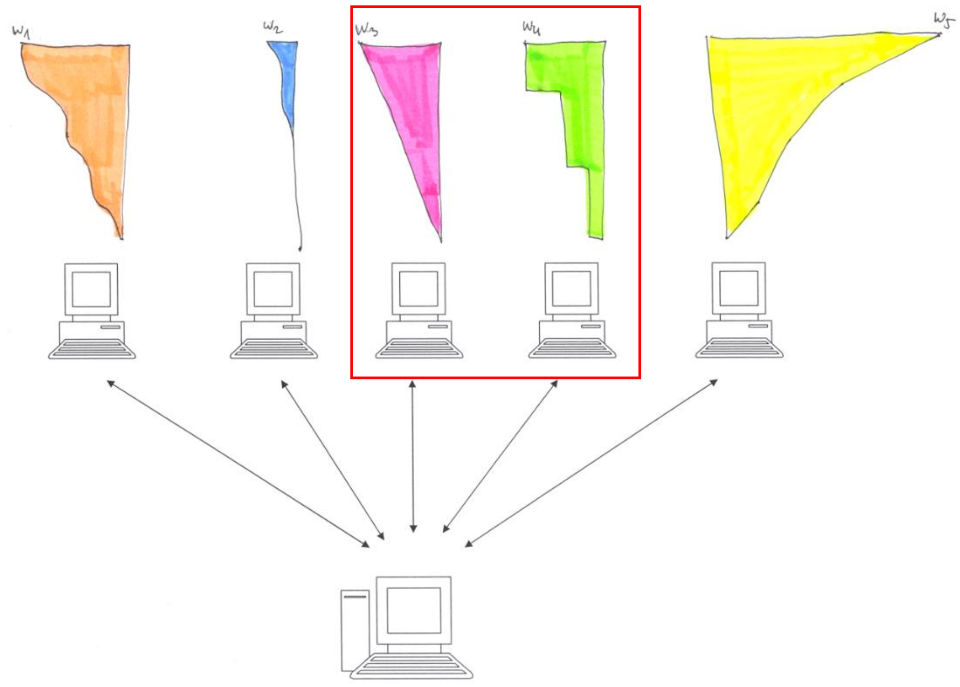
Objects $\{R_i : i \leq N\}$, m attributes
R has scores $x_1^R, ..., x_m^R \in [0, 1]$
Data in m ordered lists $L_1, ..., L_m$
record in L_i looks like (R, x_i^R), is sorted in descending order by the x_i^R value
Data access:
- sequential – price c_S
- Direct (random) , knowing id. R) – price c_R
  overall price $s*c_S + r*c_R$

Combination function $t:[0,1]^m \rightarrow [0,1]$, monotone, i.e.
$x_i \leq y_i$ implies $t(x_1, ..., x_m) \leq t(y_1, ..., y_m)$

We follow paper [FLN] R. Fagin, A. Lotem, M. Naor, Optimal aggregation algorithms for middleware. Journal of Computer and System Sciences 66 (2003) 614–656

FLN data model – viewed as LMPM





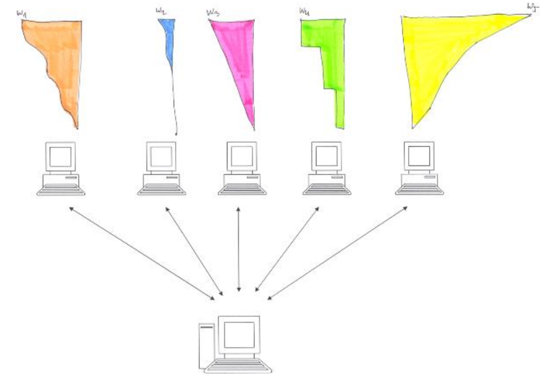
FLN data model – viewed as LMPM

In LMPM object globally ordered by
$r_{\mathbf{f},t}(oid) = t(f_1(oid.A_1),\ldots, f_j(oid.A_j),\ldots, f_m(oid.A_m))$
In FLN Objects $\{R_i : i \leq N\}$, m attributes - R_i iff oid=i
R has scores $x_1^R, \ldots, x_m^R \in [0, 1]$ - $x_j^R = f_j(oid.A_j)$
Ordered by $t(R) = t(x_1^R, \ldots, x_m^R) = r_{\mathbf{f},t}(oid)$ – **so far same** …
Differences:

- FLN do not restrict to linear f_j and t
- FLN assumes data in m ordered lists $L_1, \ldots, L_m$ (indexes) or mode of access on the server side
- Data access – sequential, direct (random),
- price c_S, price c_R, overall price $s*c_S + r*c_R$

We will add more - our model is multiuser – formally is FLN single user, Garlic Almaden has a graphical query interface

FLN threshold algorithm TA

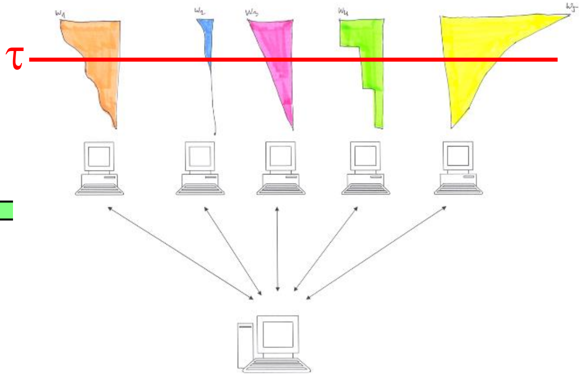


1. **Do** sorted access in parallel to each of the m sorted lists L_i. **As** an object R is seen under sorted access in some list, **do** random access to the other lists to find the grade x_j^R of object R in every list L_j . **Then** compute the grade $t(R) = t(x_1^R, ..., x_m^R)$ of object R.

If this grade is one of the k highest we have seen, **then** remember object R and its grade t(R) (ties are broken arbitrarily, so that only k objects and their grades need to be remembered at any time).

(It may seem wasteful to do random access to find a grade that was already determined earlier. As we discuss later, this is done in order to avoid unbounded buffers)

Threshold algorithm TA



2. **For** each list L_i, let $\underline{x}_i$ be the grade of the last object seen under sorted access. **Define** the threshold value τ to be

$$\tau = t(\underline{x}_1, \ldots, \underline{x}_m)$$

As soon as at least k objects have been seen whose grade is at least equal to τ; **then** halt. **Else** go to 1.

3. **Let** Y be a set containing the k objects that have been seen with the highest grades. The **output** is then the graded set $\{(R, t(R)) | R \in Y\}$ (ordered by t(R)).

TA algorithm - illustration

I am looking for a hotel close to beach, cheap, good

close		
H1		0,9
H2		0,8
H3		0,5
H4		0,4

cheap		
H3		0,9
H2		0,8
H4		0,5
H1		0,3

quality		
H2		0,9
H3		0,8
H1		0,5
H4		0,3

stack c_1		
H2		0,8
H3		0,68
H1		0,63

stack c_2		
H2		0,8
H3		0,68
H1		0,63

…

Threshold τ_1 $(3*0{,}9+2*0{,}9+0{,}9)/6=0{,}9$ So far I do not know the best

Threshold τ_2 $(3*0{,}8+2*0{,}8+0{,}8)/6=0{,}8$ 0,8≥0,8 … H2 is the best

Threshold τ_3 = 0.5, hotel H4 has overall preference degree 0.416…

Threshold algorithm is incremental in top-k

P	100m	P	Long	P	Shot	P	High	P	400m	P	110mh	P	Discus	P	Pole	P	Javelin	P	1500r	threshold
1	942	1	1089	4	847	1	915	2	964	1	985	4	840	2	1004	1	892	5	799	9277
4	938	2	1010	3	841	5	915	8	924	3	976	1	827	4	972	6	861	1	798	9062
2	922	3	982	9	831	7	859	1	919	4	946	3	803	11	941	2	843	12	770	8816
17	915	8	932	1	810	12	831	9	909	7	936	5	803	10	910	3	839	10	760	8645
3	897	6	908	8	800	4	803	14	877	10	936	7	800	12	910	14	797	11	734	8462
10	890	11	898	16	796	13	803	3	873	9	929	9	796	9	880	15	763	3	721	8349
14	885	4	891	10	780	2	776	17	873	2	916	11	748	1	849	5	746	4	706	8170
8	883	5	859	7	776	3	776	10	872	8	913	2	732	6	849	16	737	6	703	8100
6	876	12	854	6	772	14	776	5	870	6	903	8	698	8	849	7	735	2	686	8019
9	863	7	853	17	769	15	776	4	866	12	897	12	696	3	819	10	715	16	679	7933
13	863	9	840	5	765	6	749	7	858	14	886	15	691	7	819	17	711	8	665	7847
5	858	13	840	2	751	8	749	13	849	15	870	14	688	15	790	11	709	9	664	7768
16	854	10	799	11	739	16	749	6	846	17	853	10	672	5	760	8	672	13	640	7584
7	843	15	797	13	715	9	723	16	819	11	842	13	668	13	760	4	656	15	636	7459
12	841	14	788	14	708	11	696	12	808	13	841	6	655	17	731	13	653	17	628	7349
11	793	17	774	12	667	10	670	15	803	16	817	16	653	16	673	12	617	7	621	7088
15	784	16	769	15	666	17	644	11	791	5	798	17	608	14	645	9	593	14	563	6861

P	Athlete	Points	
1	Šebrle CZE	9026	3
2	Nool EST	8604	5
3	Dvorak CZE	8527	5
4	Lobodin RUS	8465	5
5	Zsivoczky HUN	8173	7
6	Ambrosch AUT	8122	8
7	Kürtösi HUN	8099	9
8	Warners NED	8085	9
9	Hämäläinen FIN	8028	9
10	Jensen NOR	8004	10
11	Schönbeck GEF	7891	11
12	Niklaus GER	7891	11
13	Tebbich AUT	7632	13
14	Llanos PUR	7613	13
15	SchnallingerAU	7576	14
16	Walser AUT	7546	14
17	Walser AUT	7506	14

Found versus confirmed = In which step was the object above threshold, e.g., Sebrle in the step 3

Objects seen in first ● second ● third ● step

TA algorithm is correct

FLN – theorem 4.1. If the aggregation function t is monotone, then TA correctly finds the top k answers (ties are ordered arbitrarily).

Proof. Let Y be as in Step 3 of TA. We need only show that every member of Y has at least as high a grade as every object z not in Y.

By definition of Y; $z \notin Y$, either it's grade was not one of the k highest or z has not been seen in running TA. So assume that z was not seen. Assume that the fields of z are $x_1^z, \ldots, x_m^z$.

Therefore, $x_i^z \leq \underline{x}_i$ for every i: Hence, $t(x_1^z, \ldots, x_m^z) \leq \tau = t(\underline{x}_1, \ldots, \underline{x}_m)$; where the inequality follows by monotonicity of t.

But by definition of Y; for every y in Y, $t(y) \geq \tau$. Therefore, for every y in Y we have $t(y) \geq \tau \geq t(z)$ as desired.

Threshold algorithm TA - wording as in FLN and small changes, additional notation (cycle counter)

Given k > 0 (can work incrementally then put k=0), data in ordered lists L_j, sequential/random access, cycle counter c:=1

1. Do sorted access in parallel to each of the m sorted lists L_i, either the service is stateless (get c's element) or not (server "knows c" → next)

- As an object R is seen under sorted access in some list,
  DO random access to the other lists to find the grade x_j^R of object R in every list L_j
  THEN compute the grade of R, $t(R) = t(x_1^R, ..., x_m^R)$
- IF this grade is one of the k highest we have seen,
  THEN remember in Y^c object R and its grade t(R) (ties are broken arbitrarily, can work incrementally)
- GO TO 2

Threshold algorithm TA - wording as in FLN and small changes, additional notation (cycle counter) - cont'd

2. For each list L_i, let $\underline{x}_i^c$ be the grade of the last object seen under c's sorted access. Put $T^c = (\underline{x}_1^c, ..., \underline{x}_m^c)$

- Define the threshold value τ^c to be

$$\tau^c = t(\underline{x}_1^c, ..., \underline{x}_m^c)$$

If k>0 and AS SOON AS at least k objects in Y^c have been seen whose grade is at least equal to τ^c (incremental variant: elements of Y^c above τ^c are confirmed)
THEN GOTO 3 ELSE IF N=c THEN GOTO 3 ELSE c:= c+1 and GO TO 1

3. Let Y^c be a set containing the k objects that have been seen with the highest grades. The output is then the graded set $\{(R, t(R)) | R \in Y\}$ (ordered by t(R)).

TA algorithm geometrically

User's screen?

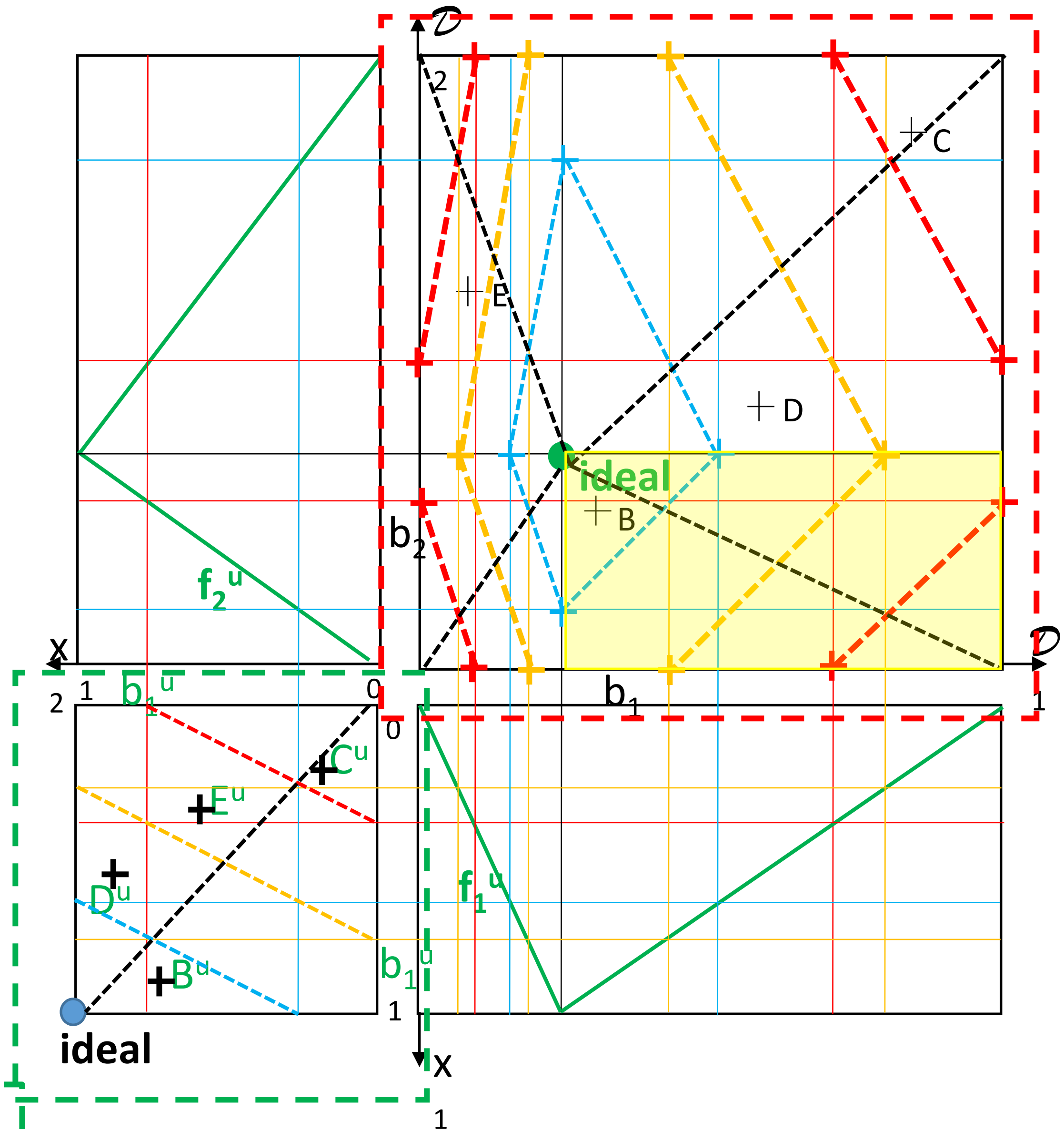
We would like to keep our model
- intuitive
- self explanatory

We would like to visualize also methods / prototypes / algorithms
- both

Top-k querying
- and

Learning user's preferences

We start at preference cube



FLN-TA-runs here

FLN-TA graphically (2D)

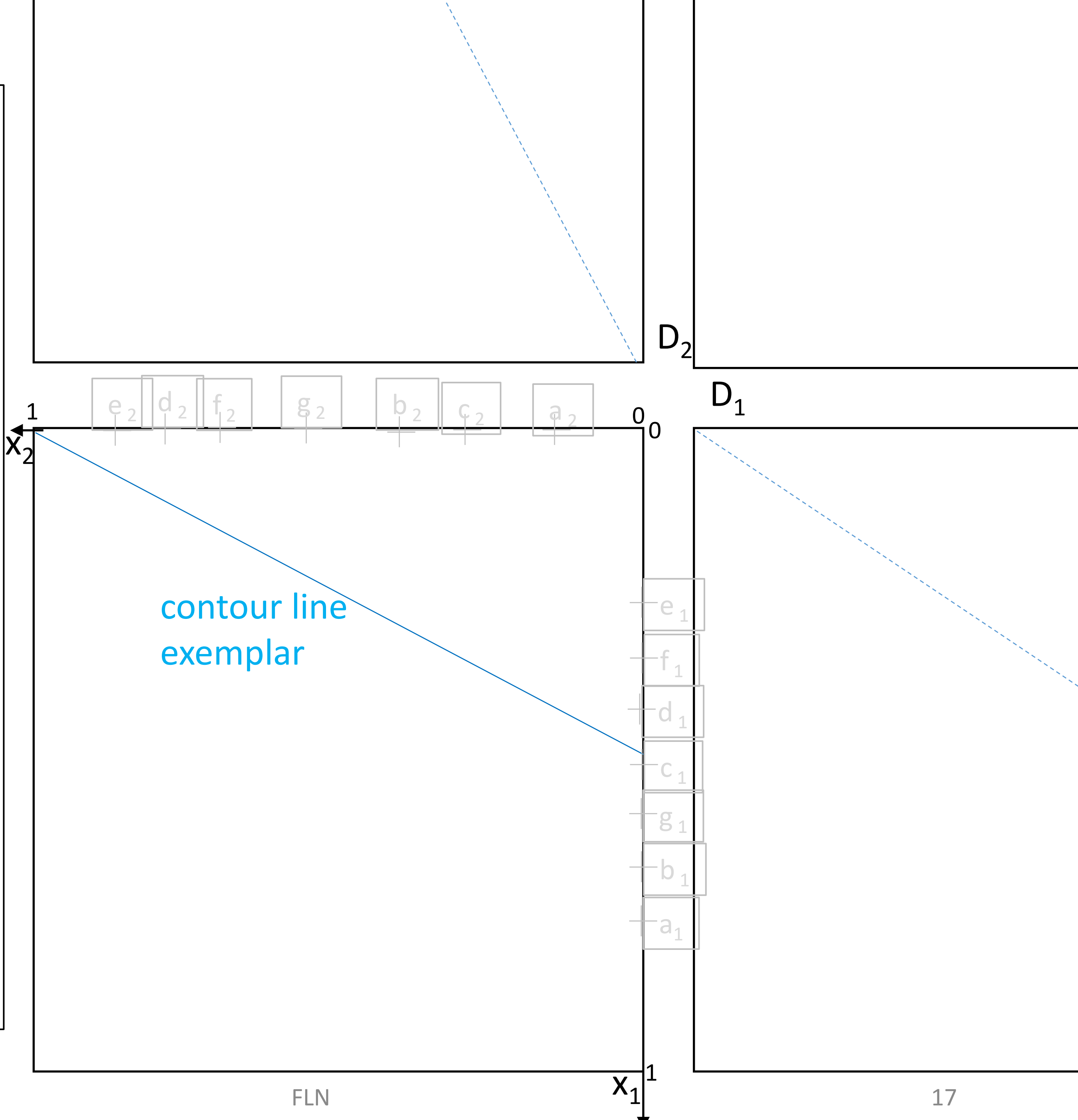
Objects $\{R_i : i \leq N\}$, R have m-many score $x_1^R, \ldots, x_m^R$

Data in m ordered lists $L_1, \ldots, L_m$, record in L_i looks like (R, x_i^R), is sorted in descending order by the x_i^R value

Combination function $t:[0,1]^m \rightarrow [0,1]$, monotone wrt Pareto ord

User u is fixed, task is to compute top-k as efficiently as possible, f_i^u gave x_i^R

Preference cube with images of data pts A,…,G on x_1, x_2 axes (lists L_1, L_2) at middleware is not seen … with aggregation given by one contour line.



FLN-TA graphically (2D)

cycle counter c:=1

Step 1

Part "Do sorted access "

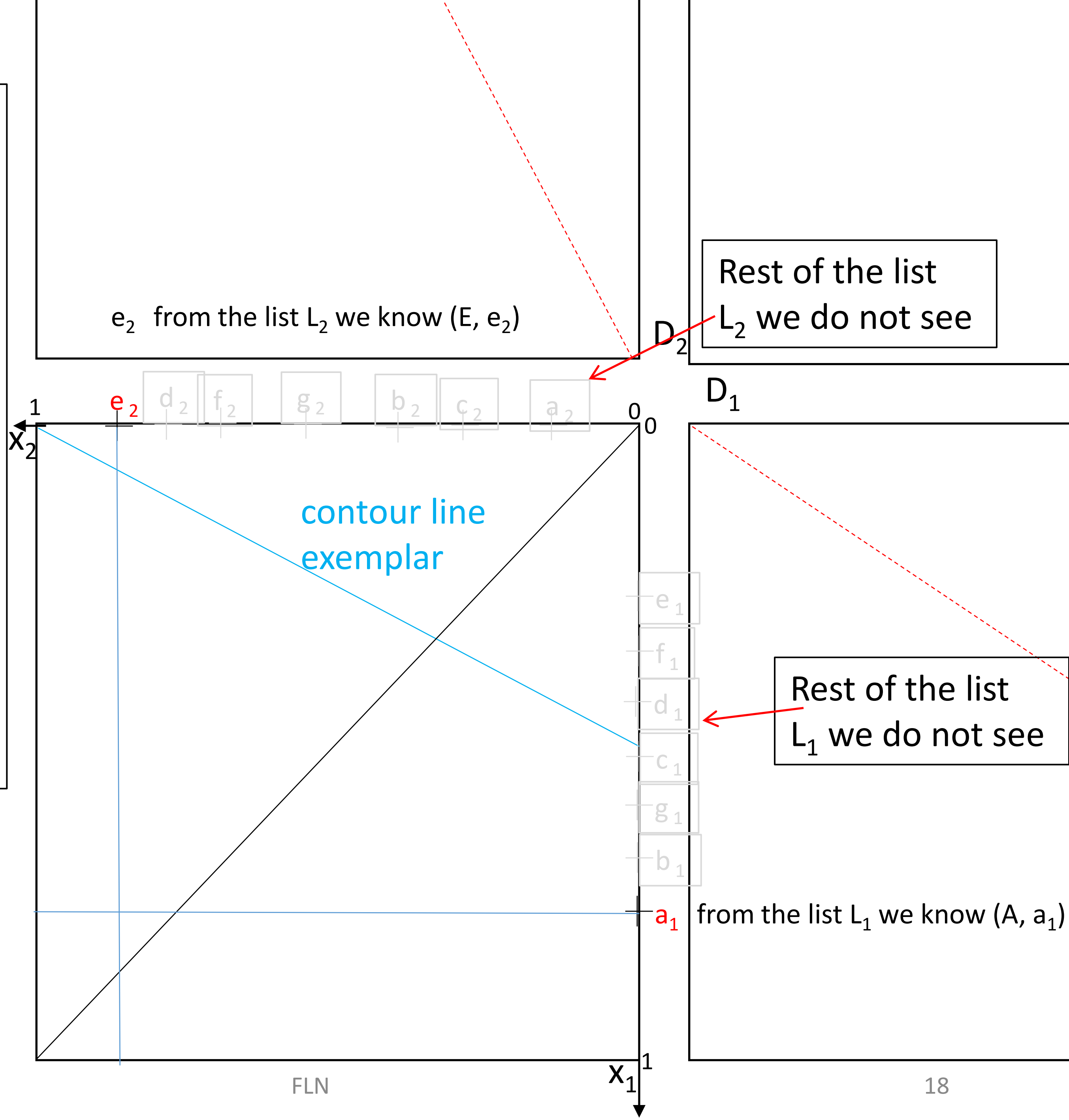
aggregation given by ex. contour line.

Axes x_1, x_2 can be viewed as lists, in this case we see only (by sequential access)

$L_1 = \{(A,a_1)\}$

$L_2 = \{(E,e_2)\}$

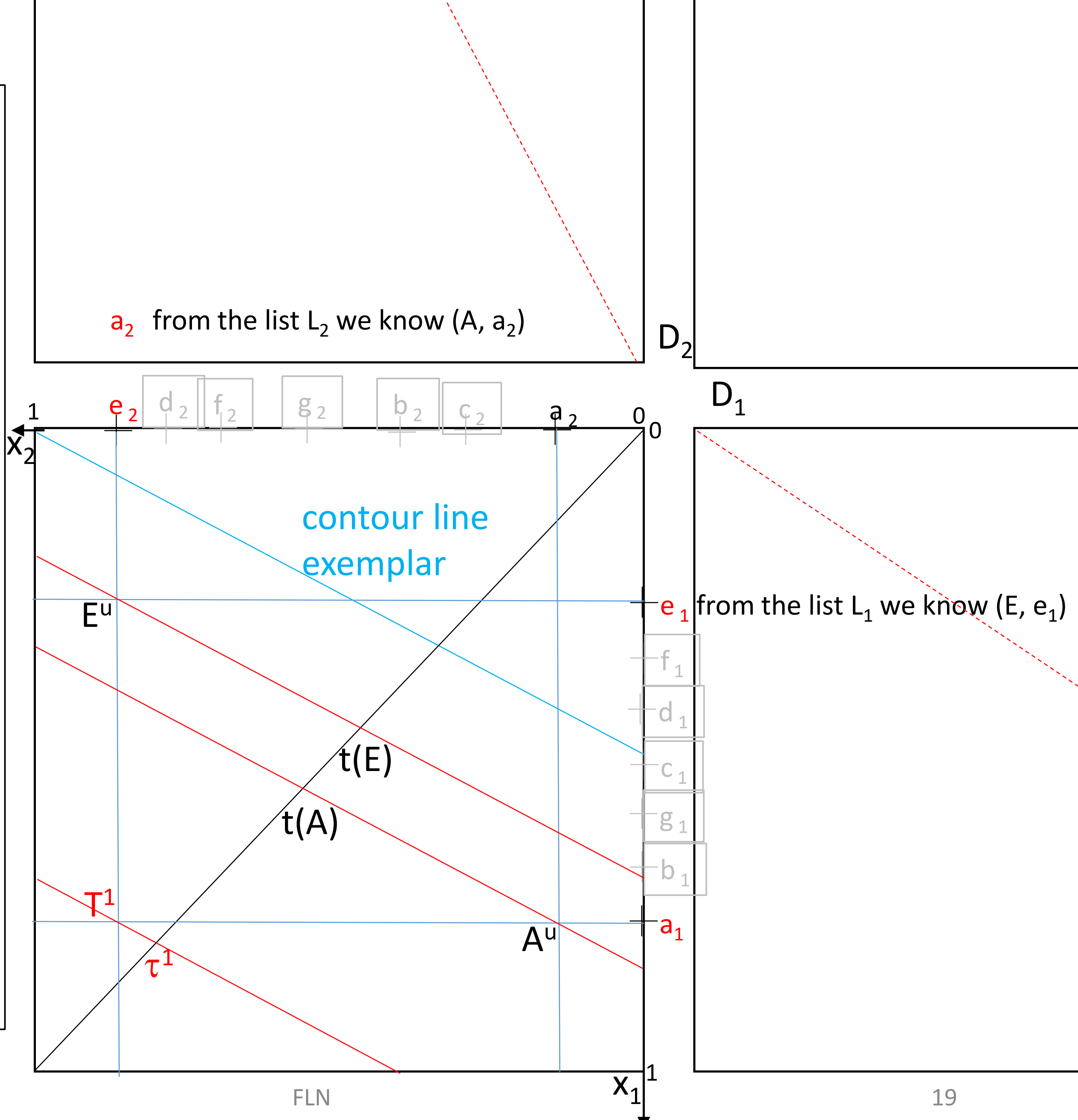
Nevertheless we know object ID's and so we can go to part 2



FLN-TA graphically (2D)

cycle counter c:=1
Step 1
Part “DO random access ”

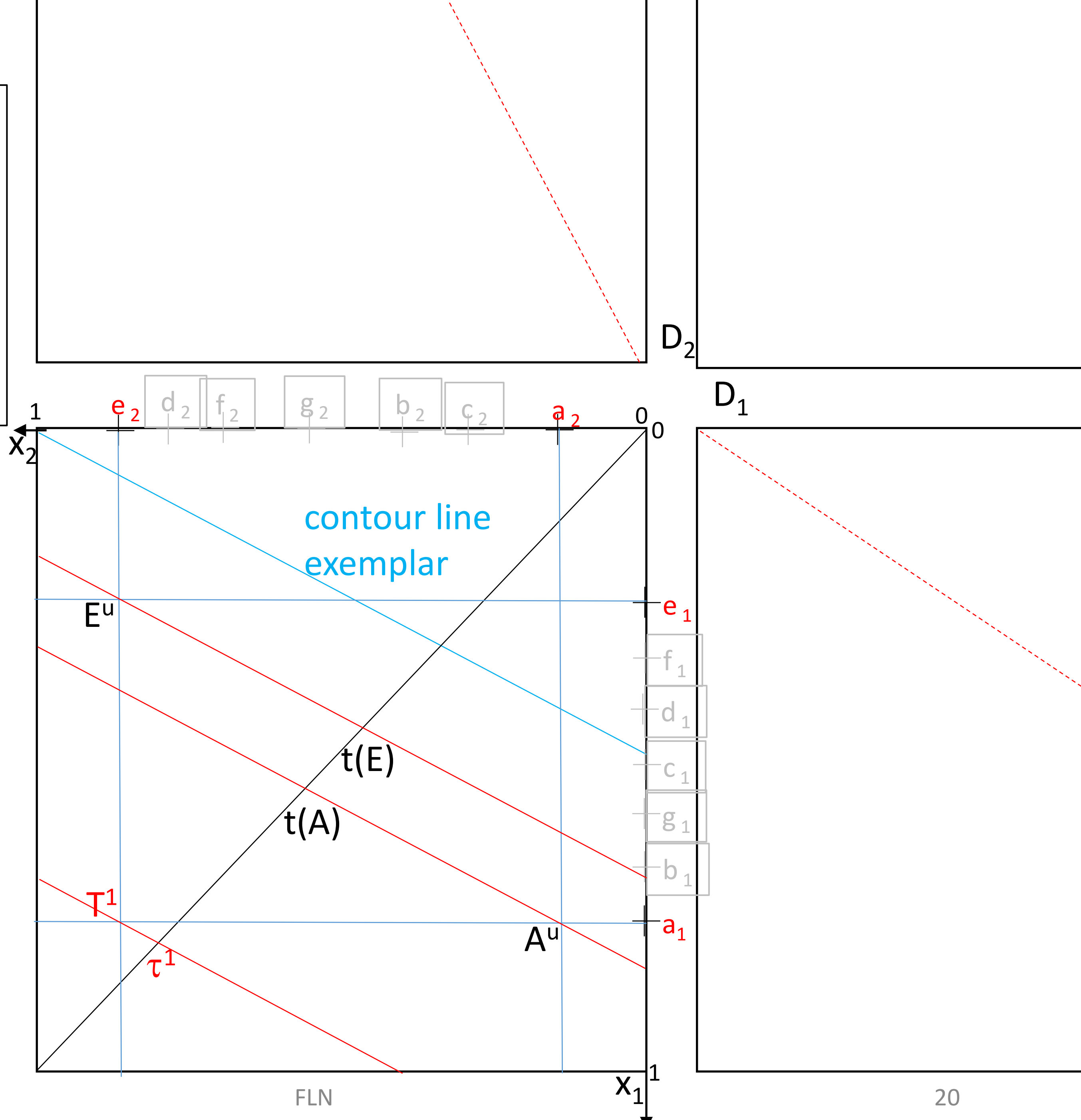
this gives a_1, e_2,
Axes x_1, x_2 viewed as lists, in this case
$L_1 = \{(A,a_1), (E,e_1)\}$
$L_2 = \{(E,e_2), (A,a_2)\}$
We have PC images $A^u = (a_1,a_2)$, $E^u = (e_1,e_2)$
Part “compute t(R) ”
gives t(A), t(E)
Part “in stack Y^1 we have”
$Y^1 = \{(A, t(A)), (E, t(E))\}$
Step 2 gives
$\underline{x}_1^1 = a_1$, $\underline{x}_2^1 = e_2$,
$T^1 = (a_1 , e_2)$, and it’s contour line intersects diagonal at τ^1



FLN-TA graphically (2D)

cycle counter c:=1
Step 2
Part “compare Y^1 and τ^1”

ELSE c:= c+1 = 2
and GO TO 1
applies

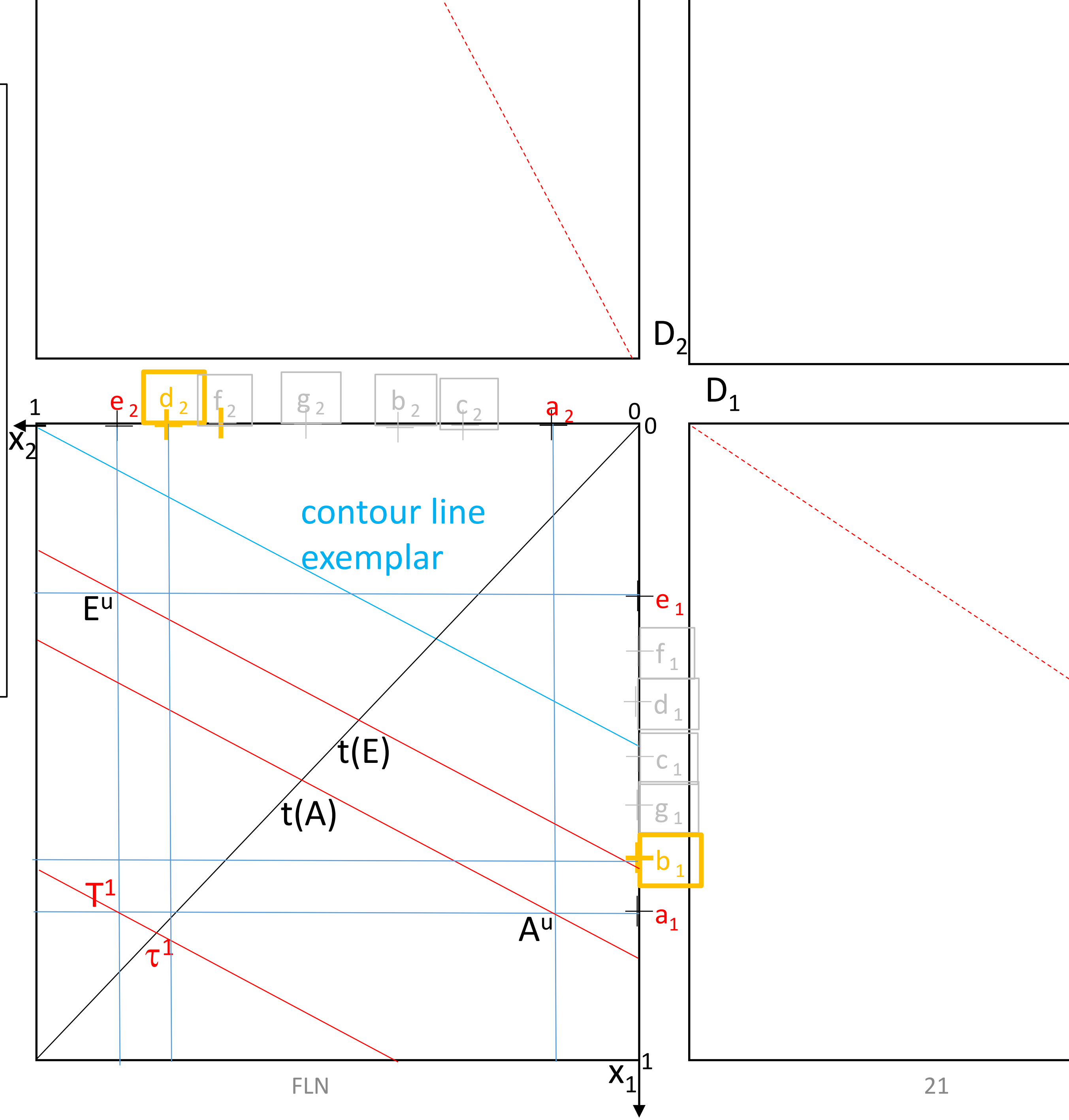


FLN-TA graphically (2D)

cycle counter c:=2
Step 1
Part "Do sorted access "

Axes x_1, x_2 and/or lists we see (by sequential access)
L_1 ={(A,a_1),(B,b_1),…(E, e_1)}
L_2 ={(E,e_2),(D,d_2),…(A,a_2)}

Nevertheless we know object ID's and so we can go to part 2



FLN-TA graphically (2D)

cycle counter c:=2

Step 1

Part “DO random access ”

this gives d_1,b_2,

We have PC images B^u, D^u

Part “compute t(R) ”

gives t(B), t(D)

Part “in stack Y^2 we have”

an ordered stack

Y^2 = {(B, t(B)), (A, t(A)), (D, t(D)), (E, t(E))}

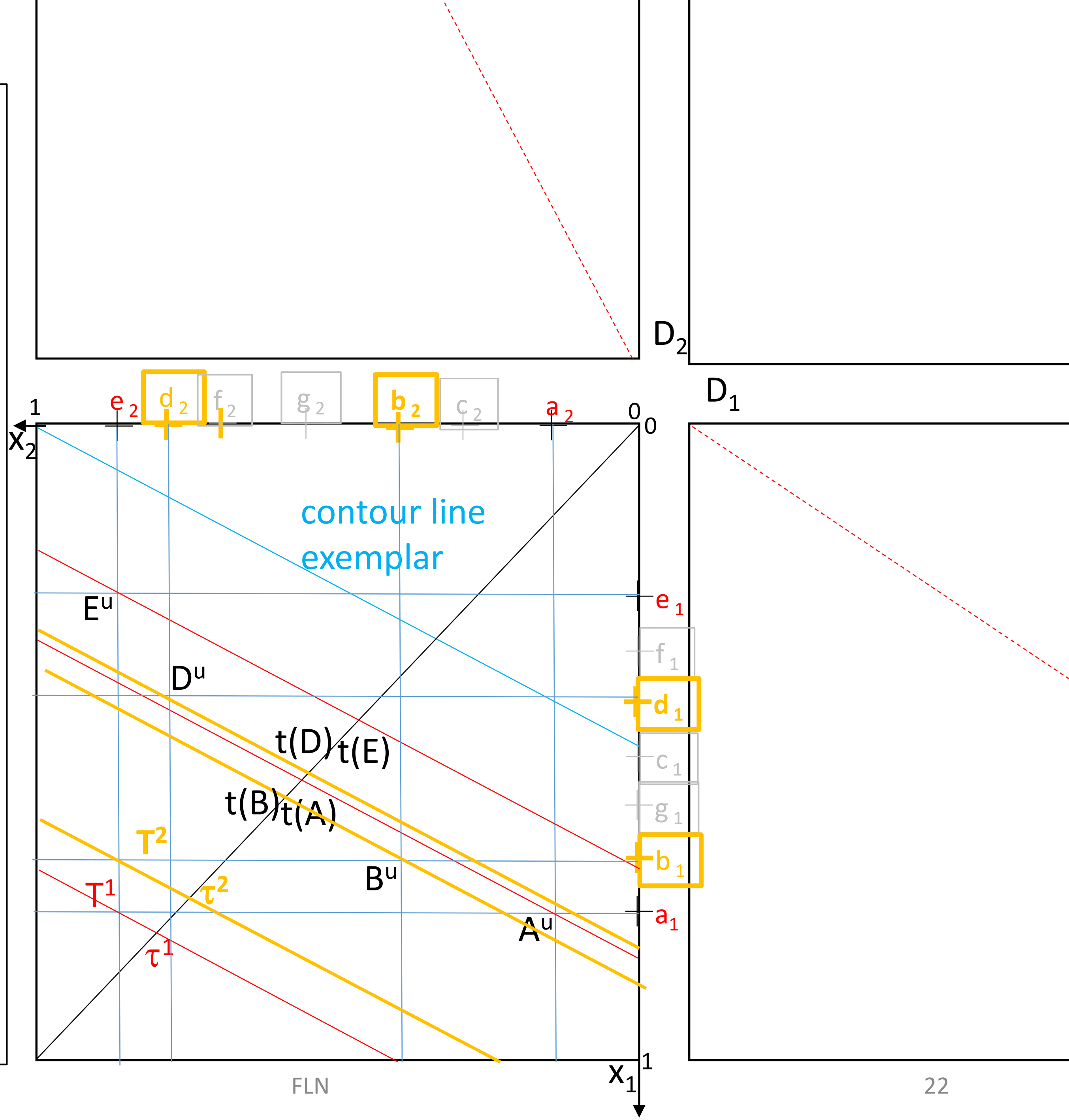
Step 2 gives

$\underline{x}_1{}^2 = b_1$, $\underline{x}_2{}^2 = d_2$,

T^2= (b_1 , d_2), and it’s contour line intersects diagonal at τ^2

Part “compare Y^2 and τ^2”

gives c:= 3 and GO TO 1



FLN-TA graphically (2D)

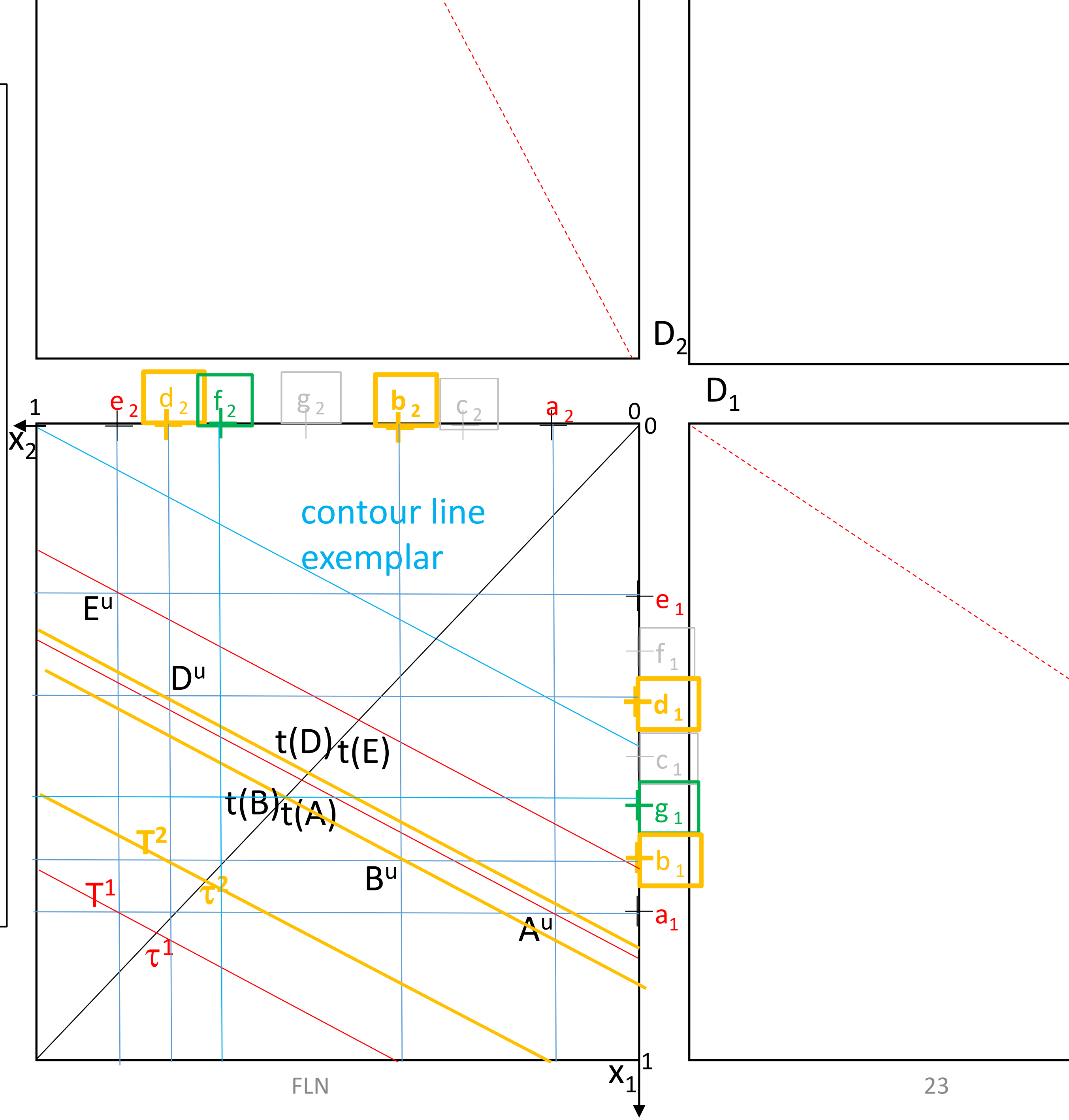
cycle counter c:=3
Step 1, part … 2, part …

Colors can help us to distinguish cycles …

Anyway, it is a **mess** in this small dimension

In future **we omit** depicting
- data cube part
- attribute preferences

We enlarge preference cube and solutions will be coded by colors and notation as here



FLN-TA graphically (2D)

Big picture

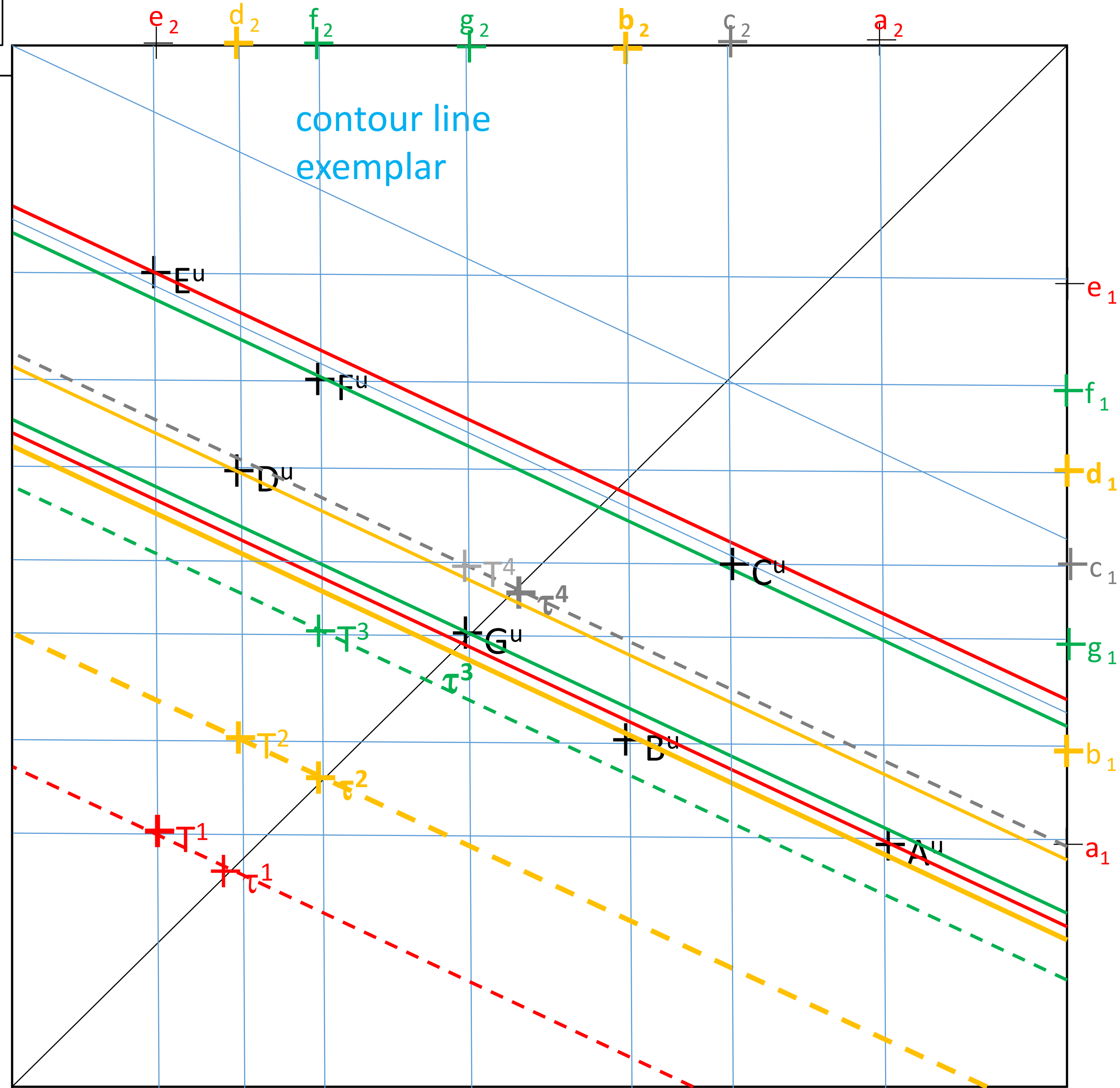
In this case we need 4 steps to decide the winner

Certified order of items so far is B, A, G, D (for F, C, E we have to wait)

It seems that number of steps needed to decide the winner is at most n/2 + 1 where n is the # of items ($\lfloor n/2 \rfloor + 1$)

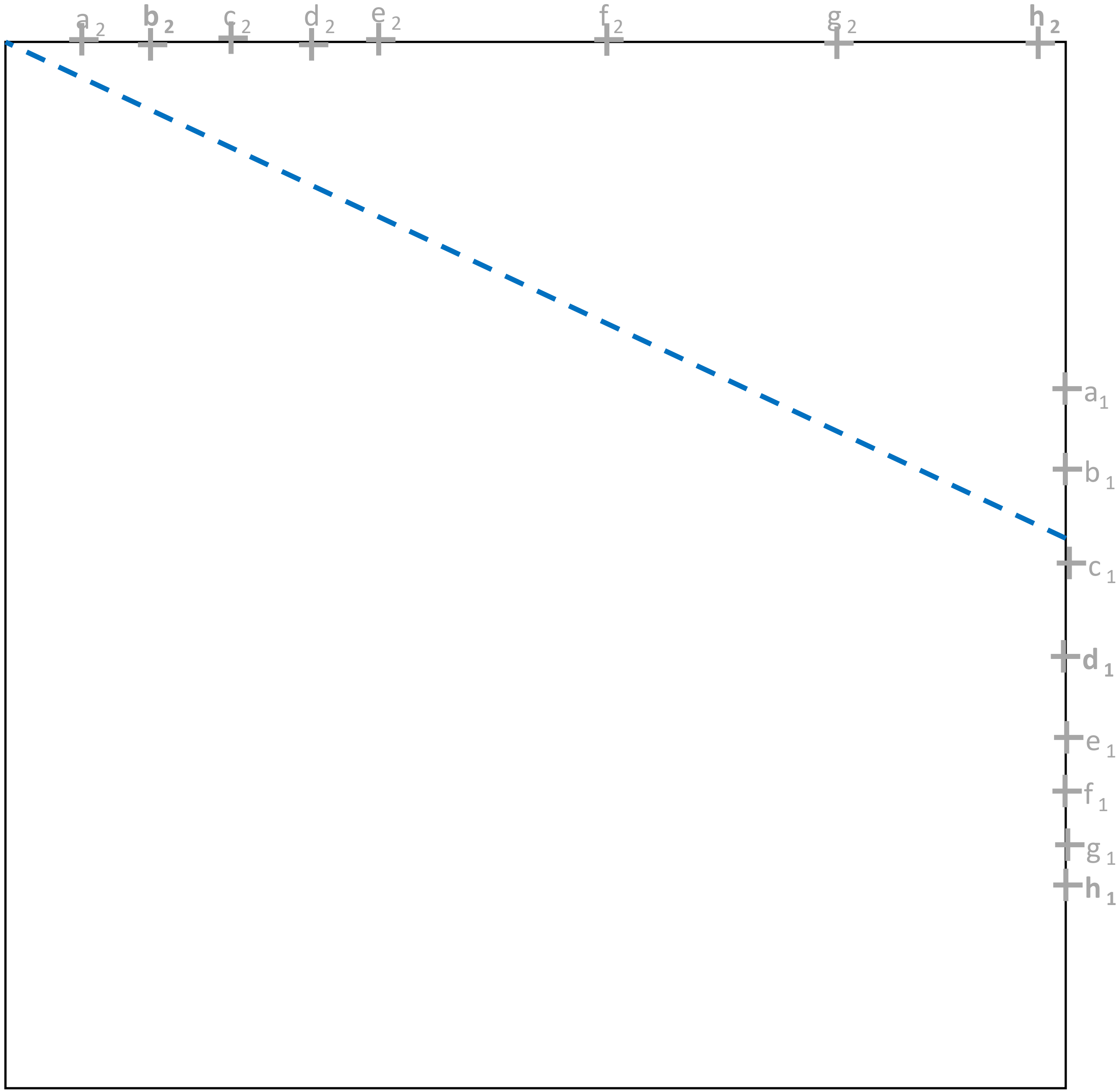
?can we find data distribution such that TA ends (decides winner / decides all) in arbitrary number of steps $s \leq (\lfloor n/2 \rfloor + 1)$?

In 3-D, 4-D, …?



FLN-TA graphically (2D)
Lab/homework 1

Run threshold algorithm
Use notation as in previous slide
Use colors to denote steps/cycles of TA

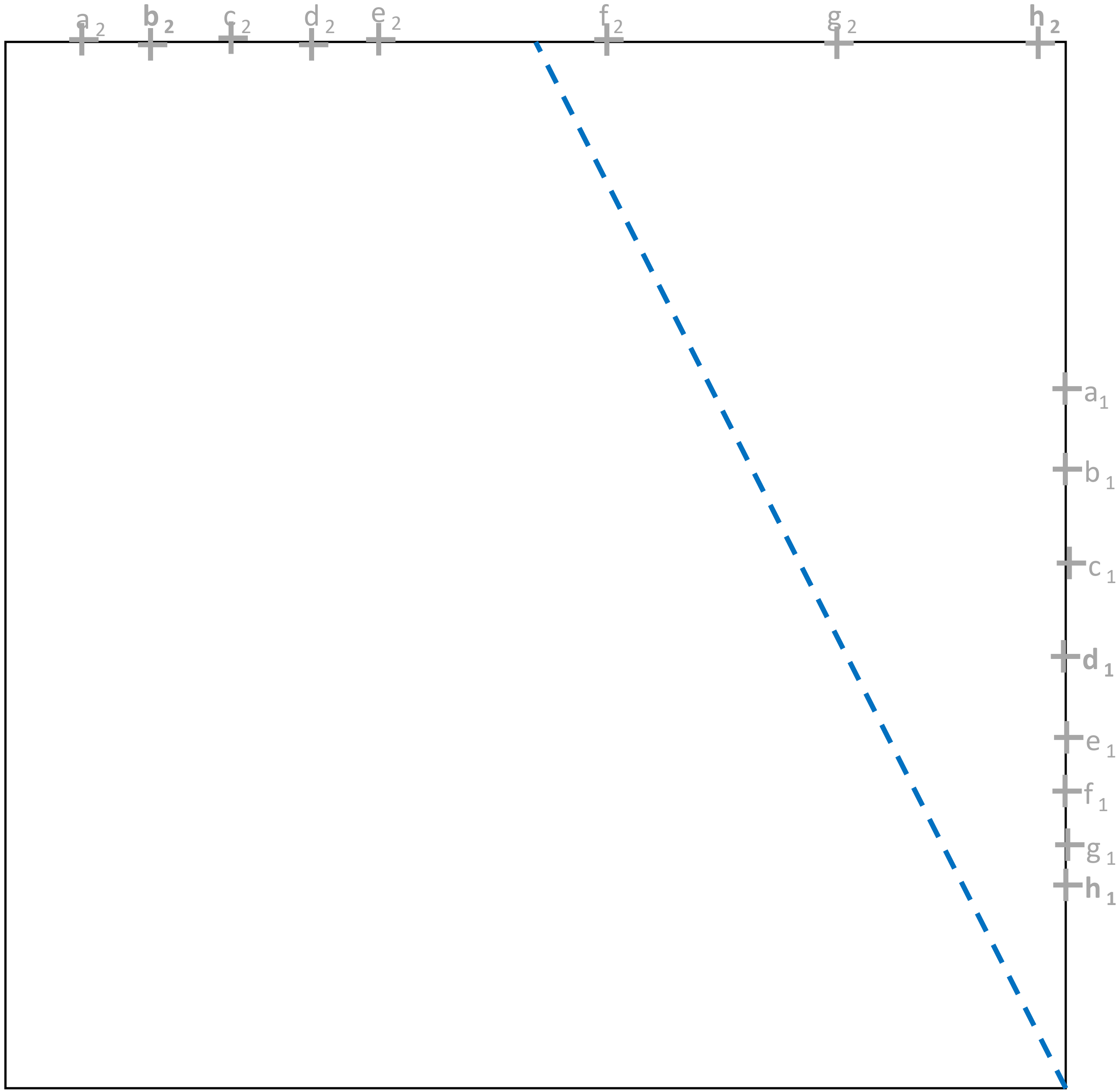


FLN-TA graphically (2D)
Lab/homework 2

Run threshold algorithm

Use notation as in previous slide

Use colors to denote steps/cycles of TA

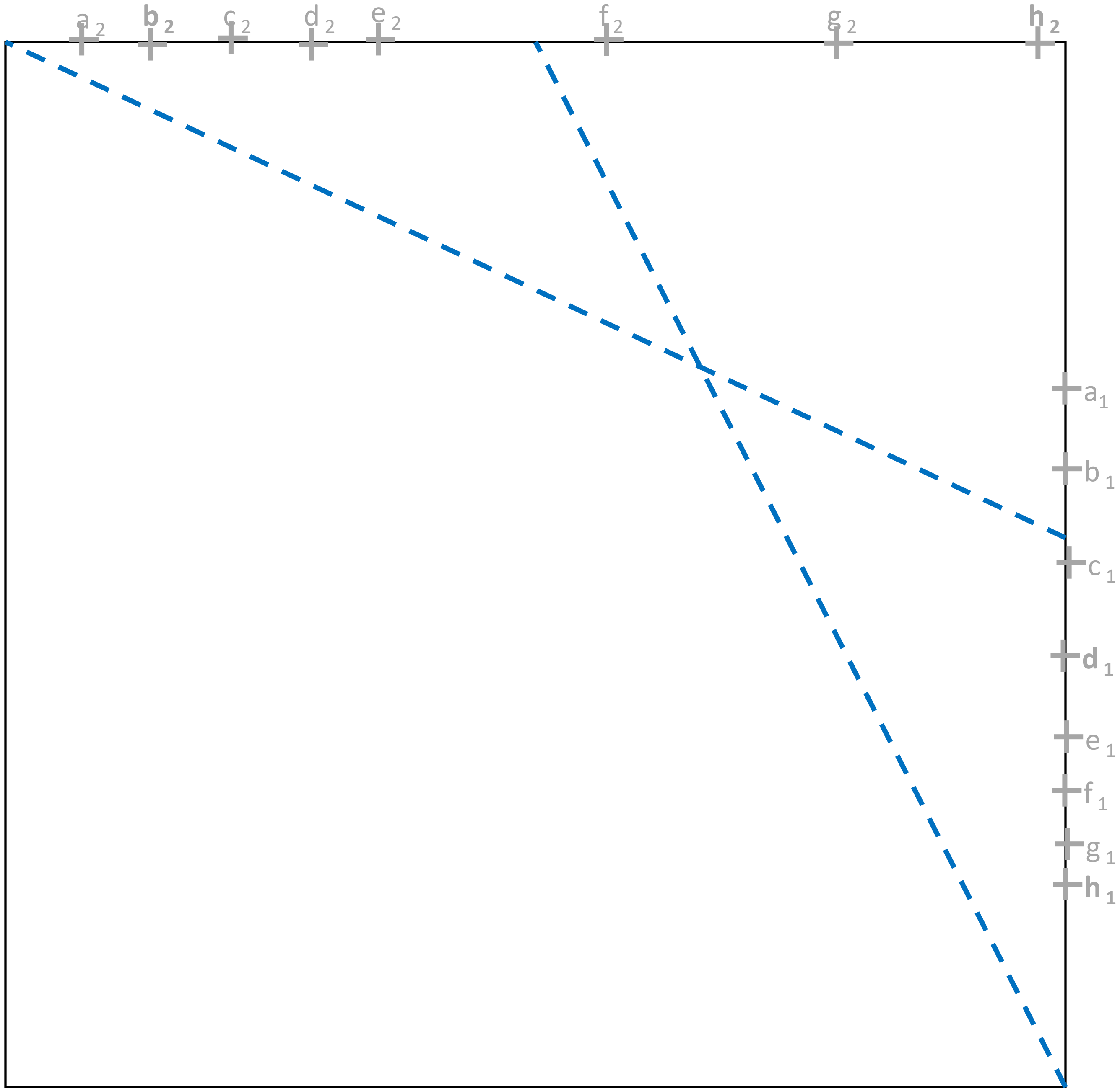


FLN-TA graphically (2D)
Lab/homework 3

Run threshold algorithm
Use notation as in previous slide
Use colors to denote steps/cycles of TA

Lists are same, **how does aggregation influence run of TA**?

Is there a **neighborhood of contour line** with same TA run and outcome?



FLN-TA graphically (2D)
Lab/homework 4

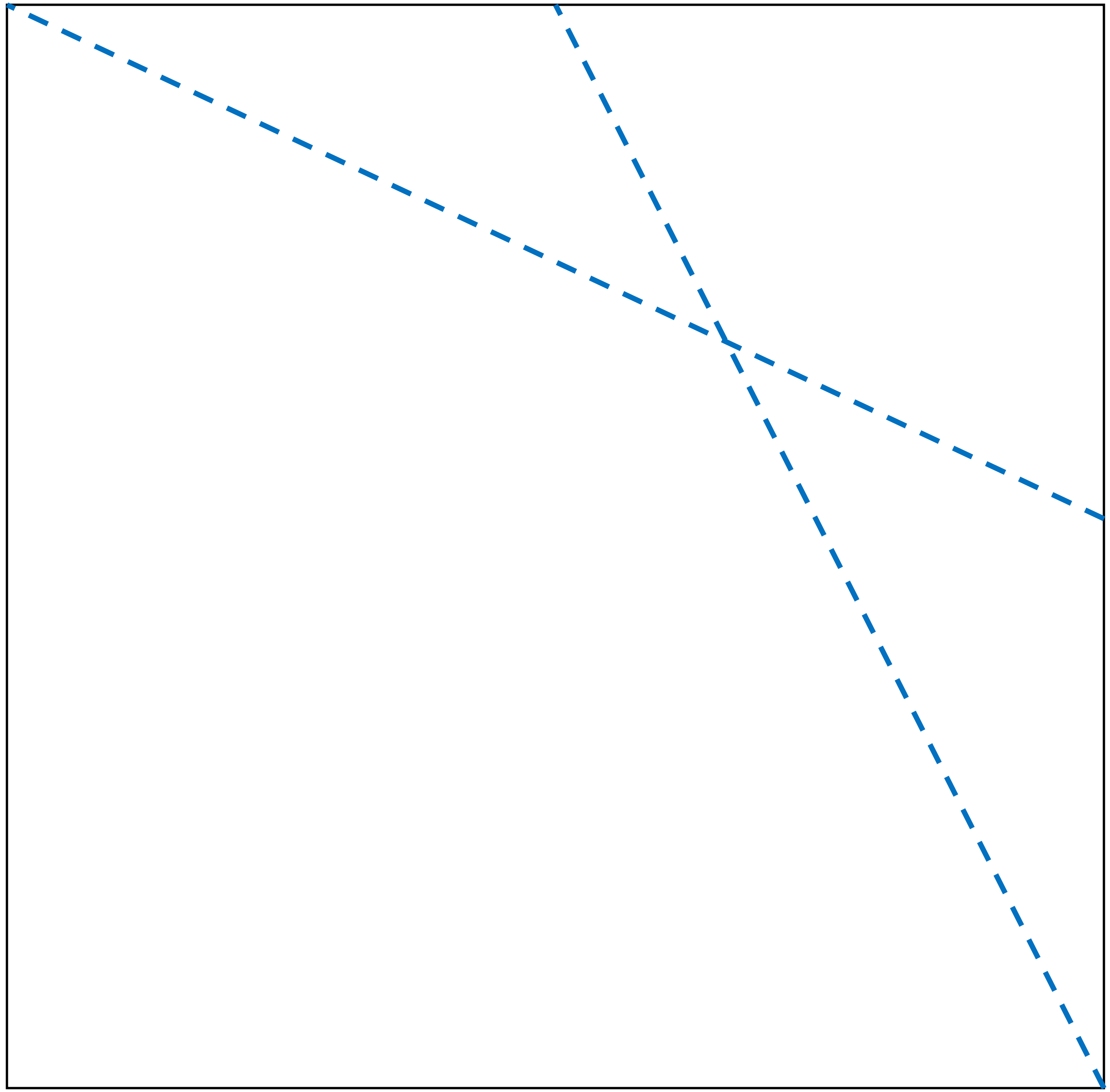
Experiment with mutual influence of data and aggregation distribution.

Is the hypothesis ?can we find data distribution such that TA ends (decides winner / decides all) in arbitrary number of steps $s \leq (\lfloor n/2 \rfloor + 1)$? **true** ?

Check it in **higher dimensions** in PC, 4-D, 5-D, …

Try **randomly generated** data and threshold

Run threshold algorithm
With notation, colors, etc. same as in previous



σημαντικος (ΣΗΜΑΝΤΙΚΟΣ) – sémantika, význam

Problém webu-lidé rozumí, stroje ne

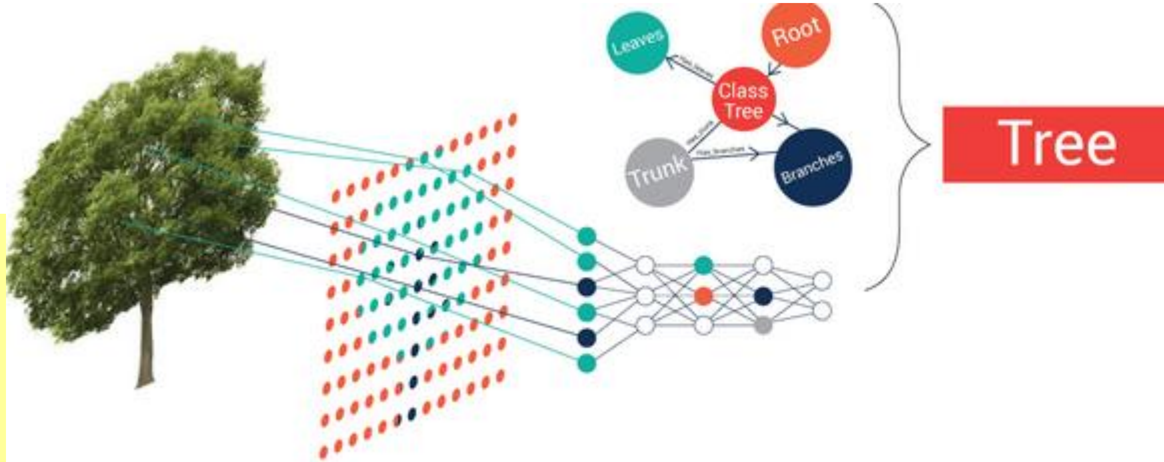
林克昌 根留台灣 可能增高

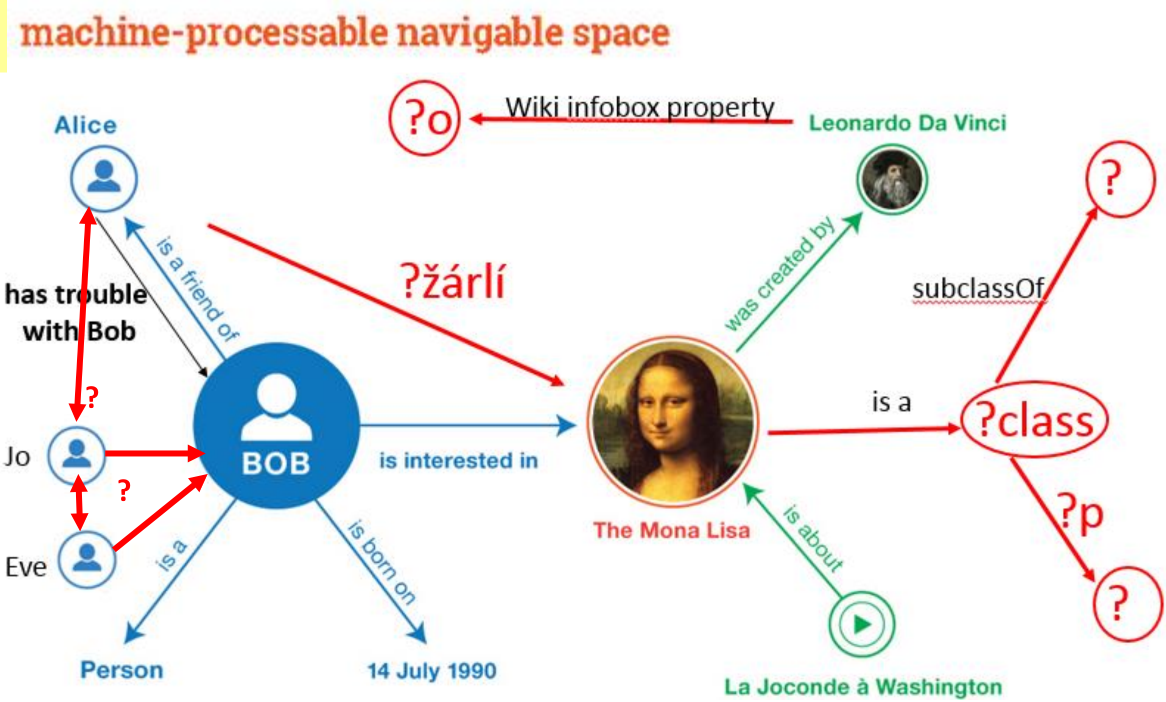
在愛樂者熱心奔走之下，華裔名指揮家林克昌根留台灣的可行性又提升了幾分。兩廳院主任李炎、國家音樂廳樂團副團長黃奕明日前親赴林克昌、石聖芳寓所拜會，並提出多場客席邀約。此外，台灣省立交響樂團團長陳澄雄也早早「下訂」，邀請林克昌赴台中霧峰，從八月十日起訓練省交，為期長達一個月。

在台灣諸多公家樂團中，陳澄雄是以實際行動表達對林克昌肯定的樂界人士之一，曾多次公開表示對林克昌指揮才華的欽佩，而且幾乎每個樂季都邀請林克昌客席演出。

此外，林克昌上個月赴俄羅斯與頂尖的「俄羅斯國家管絃樂團」灌錄了柴可夫斯基晚期三大交響曲以及「羅密歐與茱麗葉」、「斯拉夫進行曲」、「義大利隨想曲」，最後的DAT母帶也在前兩天寄回台灣。製作人楊忠衡與林克昌試聽之後，都對錄音效果—尤其音質表現感到相當滿意，楊忠衡估計呈現了七分林克昌指揮神韻。

俄羅斯國家管絃樂團首席布魯尼日前也讚譽林克昌的指揮藝術有三大特點：一是控制自如的彈性速度；二是強烈的動態對比；三是宛如呼吸歌唱的旋律處理。這些對錄音師而言都構成很大挑戰。俄國錄音師雖然採用多軌混音，但定位、場面都有可觀之處。。

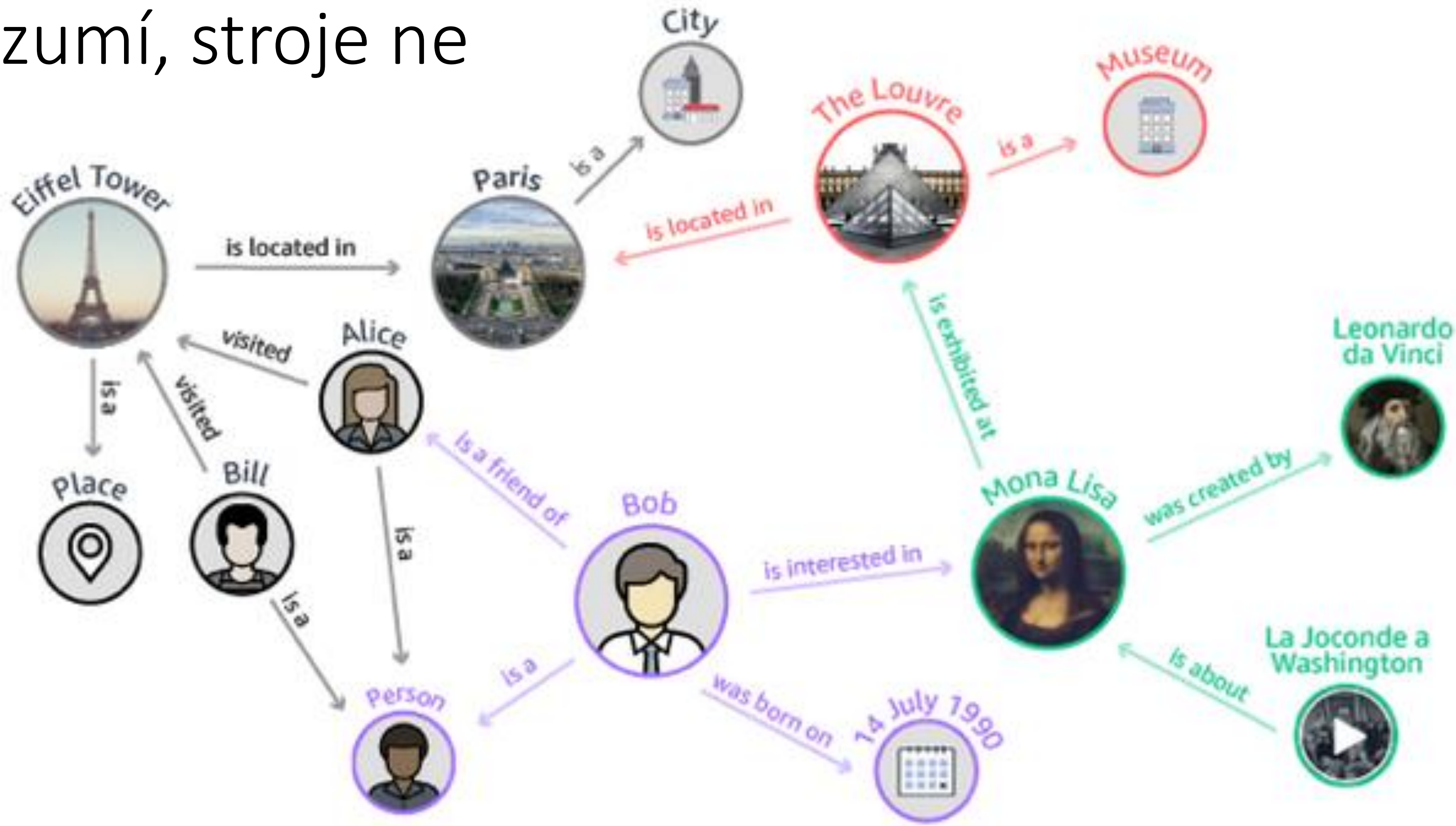




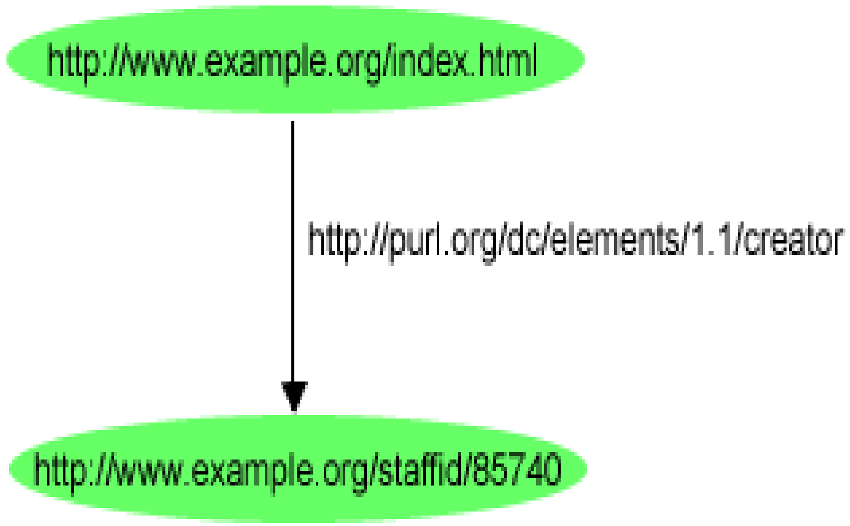
Více bývalá NSWI108 a nově také NDBI021, také multimodální data

σημαντικος (ΣΗΜΑΝΤΙΚΟΣ) – sémantika, význam

Problém webu-lidé rozumí, stroje ne



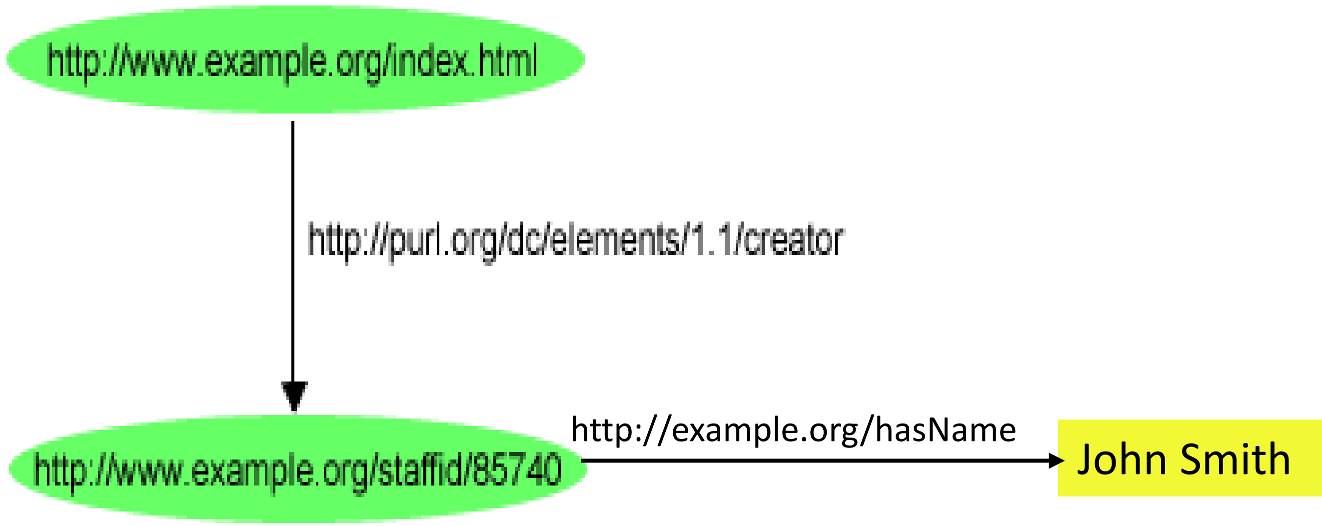
RDF – named oriented graph



- RDF - "Resource Description Framework"
- W3C recommendation (http://www.w3.org/RDF)
- RDF is a data model

Example from W3C Resource Description Framework (RDF) Primer http://www.w3.org/TR/rdf-primer/

RDF – oriented graph

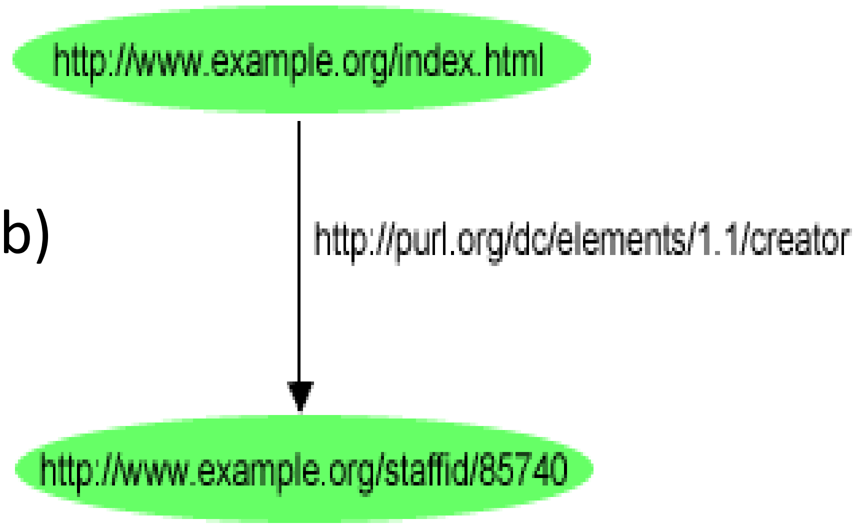


- uses URI (IRI) for unique resource identification, taken from XML
- graph has named vertices and edges
- Literals are data values, which are not resources, string of symbols, with possible data type

Example from W3C Resource Description Framework (RDF) Primer http://www.w3.org/TR/rdf-primer/

RDF terminology of sentence analysis

Terminology W3C		Sentence analytical level in Czech
Subject		podmět
Predicate (verb)		přísudek
Object		předmět



This sentence in natural language reads as:
http://www.example.org/index.html has a **creator** whose staffid value is **85740**

→ Collision of „linguistic“ and OOP terminology

Example from W3C Resource Description Framework (RDF) Primer http://www.w3.org/TR/rdf-primer/

RDF and terminologies of ER, OOP, …

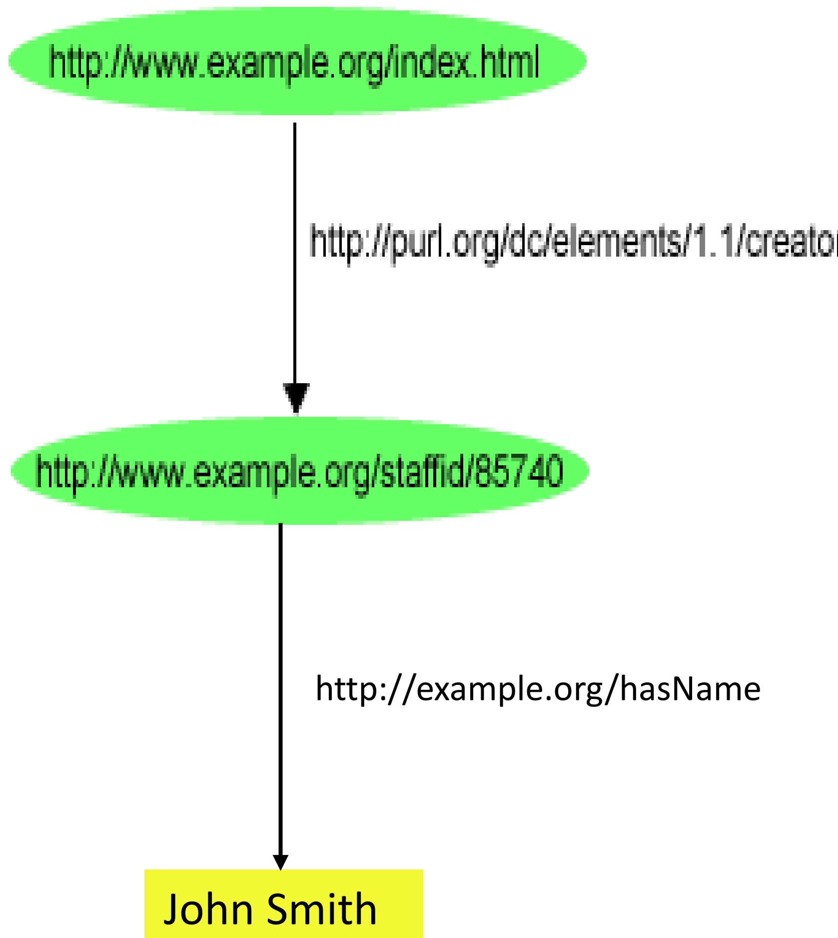
Entity, object

relation

Entity, object

Attribute

value



subject

verb

object/subject

verb

object

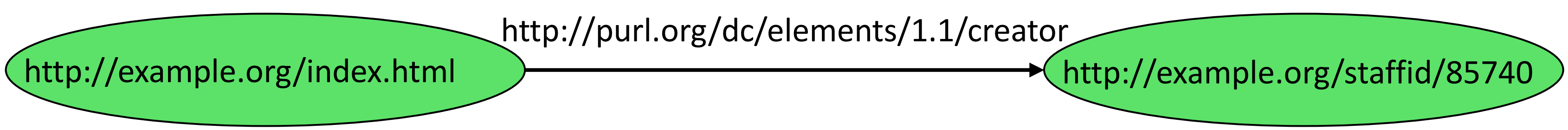
Example from W3C Resource Description Framework (RDF) Primer http://www.w3.org/TR/rdf-primer/

Languages of web services use RDF – XML syntax

- as in XML, we can use name spaces
- proper RDF elements, with name space rdf:

```
<?xml version=“1.0" encoding="utf-8"?>
 <rdf:RDF xmlns:rdf=''http://www.w3.org/1999/02/22-rdf-syntax-ns#“
             xmlns:ex=''http://example.org/“
             xmlns:dc=„http://purl.org/dc/elements/1.1/“ >

  <rdf:Description rdf:about="http://www.example.org/index.html">
      <ex:creator>
  <rdf:Description rdf:about="http://www.example.org/staffid/85740">
     </ ex:creator >
  </rdf:Description>
 </rdf:RDF>
```



Example from W3C Resource Description Framework (RDF) Primer http://www.w3.org/TR/rdf-primer/

RDF – XML syntax tripple

- element rdf:Description is coding „subject“, it’s URI is attribute value of rdf:about
- each subelement of rdf:Description is „predicate“, it’s URI is the name, this contains „object“ of the triple as further rdf:Description

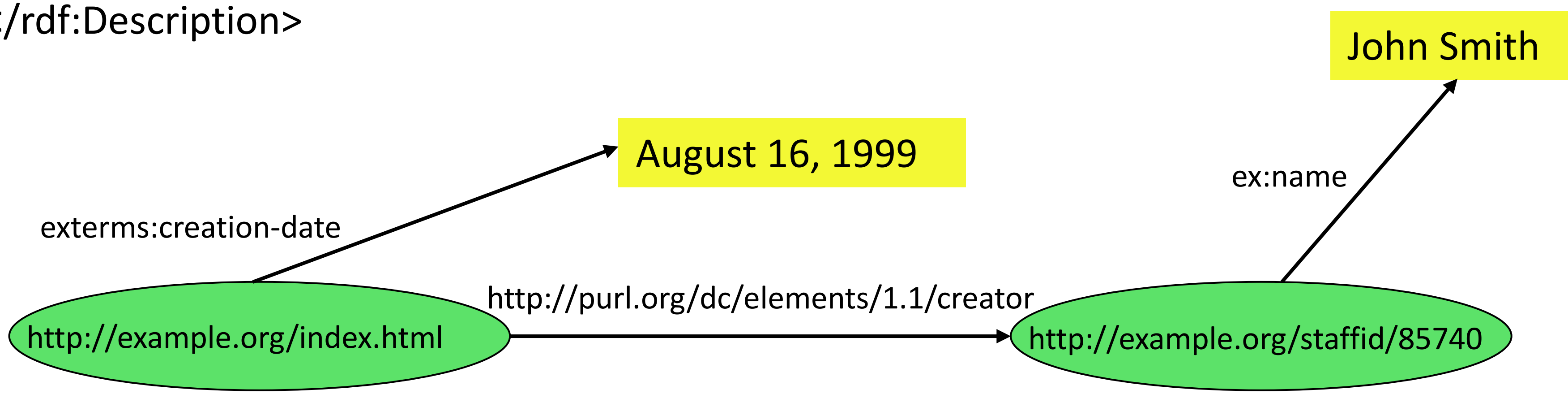
```
<rdf:Description rdf:about="http://www.example.org/index.html">
     <ex:creator>
<rdf:Description rdf:about="http://www.example.org/staffid/85740">
    </ ex:creator >
</rdf:Description>
</rdf:RDF>
```

Example from W3C Resource Description Framework (RDF) Primer http://www.w3.org/TR/rdf-primer/

RDF – XML syntax

- Untyped literals can be specified as text in content of element „predicate“
- single element „subject“ can contain more „predicate“ subelements
- „object“ rdf:Description can serve as „subject“ for next triple

```
<rdf:Description rdf:about="http://www.example.org/index.html">
  <ex:creator>
     <rdf:Description rdf:about="http://www.example.org/staffid/85740">
          < ex:name > John Smith </ ex:name >
  </ ex:creator >
  <exterms:creation-date > August 16, 1999 </ exterms:creation-date >
</rdf:Description>
```



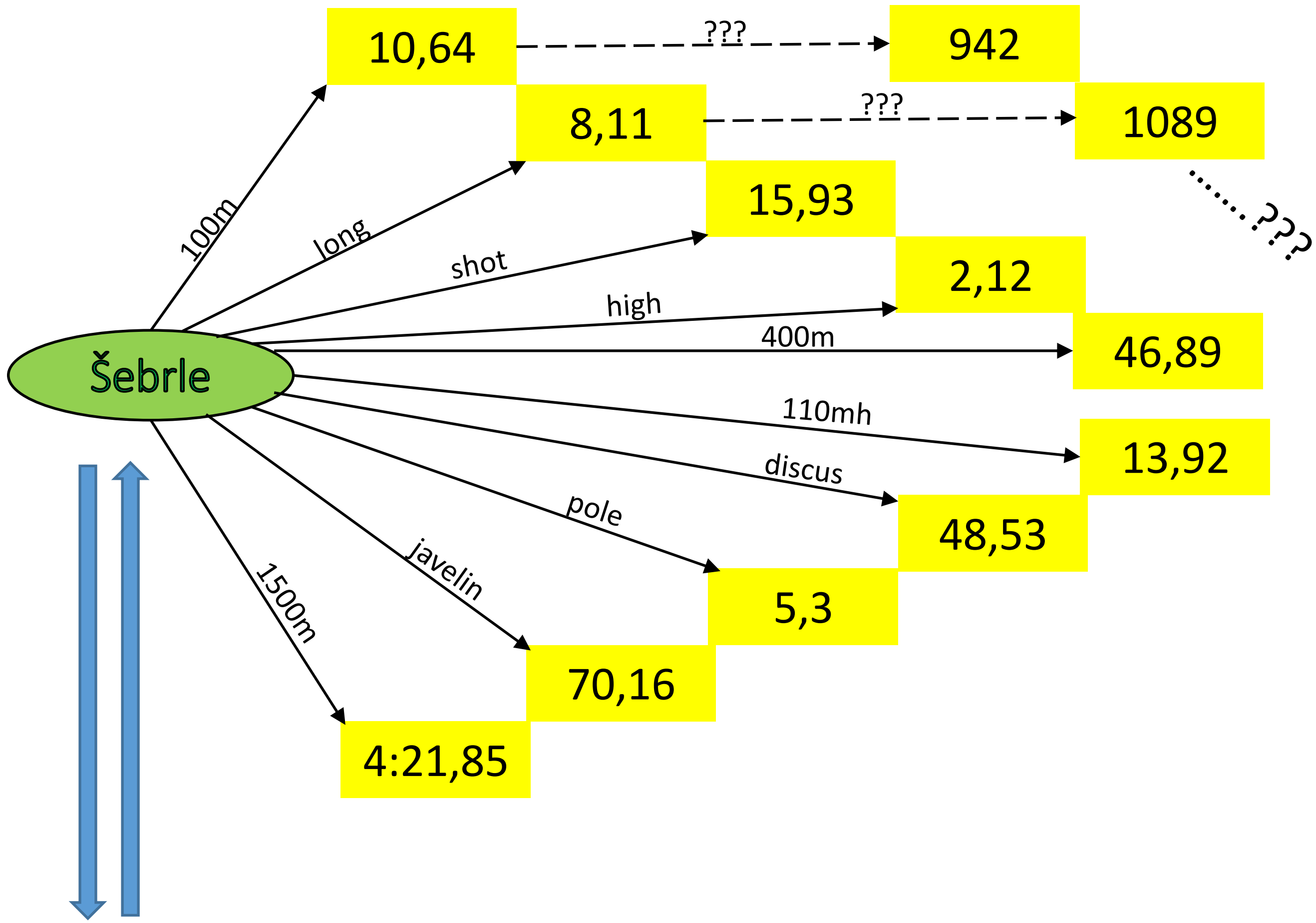
Example from W3C Resource Description Framework (RDF) Primer http://www.w3.org/TR/rdf-primer/

Transformations without loss of information

Relational to RDF – decomposition – when does it make sense?

RDF to relational – lot of joins – integration of web resources (like FLN-TA only needed for top-k)

Graph databases, store as graphs? optimize in the time of query? … W3C rules too strict!?



P	Athlete	Points		P	100m	P	Long	P	Shot	P	High	P	400m	P	110mh	P	Discus	P	Pole	P	Javeli n	P	1500m
1	Šebrle CZE	9026		1	10,64	1	8.11	4	15.93	1	2.12	2	46,89	1	13,92	4	48.53	2	5.3	1	70.16	5	4.21,85
2	Nool EST	8604		4	10,66	2	7.8	3	15.83	5	2.12	8	47,70	3	13,99	1	47.92	4	5.2	6	68.15	1	4.21,98
3	Dvorak CZE	8527		2	10,73	3	7.69	9	15.67	7	2.06	1	47,79	4	14,22	3	46.74	11	5.1	2	66.94	12	4.26,13
4	Lobodin RUS	8465		17	10,76	8	7.49	1	15.33	12	2.03	9	48,01	7	14,30	5	46.73	10	5	3	66.66	10	4.27,65

Athlete	Points		P	100m	P	Long	P	Shot	P	High	P	400m	P	110mh	P	Disc us	P	Pole	P	Javelin	P	1500m
Šebrle CZE	9026		1	942	1	1089	4	847	1	915	2	964	1	985	4	840	2	1004	1	892	5	799
Nool EST	8604		4	938	2	1010	3	841	5	915	8	924	3	976	1	827	4	972	6	861	1	798

FLN Model– viewed as LMPM-RDF data, for web services + multiuser, …

- In LMPM object globally ordered by $r_{\mathbf{f},t}(oid) = t(f_1(oid.A_1),\ldots)$
- In FLN Objects $\{R_i : i \leq N\}$, (attributes are hidden, $R_i = oid_i$)
- R has scores $x_1^R, ..., x_m^R \in [0, 1]$ - $x_j^R = f_j(oid.A_j)$
- Ordered by $t(R) = t(x_1^R, ..., x_m^R) = r_{\mathbf{f},t}(oid)$ – **so far same** …
- FLN do not restrict to linear f_j and t
- FLN assumes data in m ordered lists $L_1, ..., L_m$ (indexes) or mode of access on the server side
- **Lists are in fact RDF data**, as (R, x_i^R) can be seen as a triple $R \xrightarrow{A_i \text{ preference}} x_i^R$
- Data access – sequential, direct (random), price c_S, price c_R, overall price $s*c_S + r*c_R$
- **We will add more** - our model is multiuser – FLN is single user + we consider learning user models, measure quality of models, …

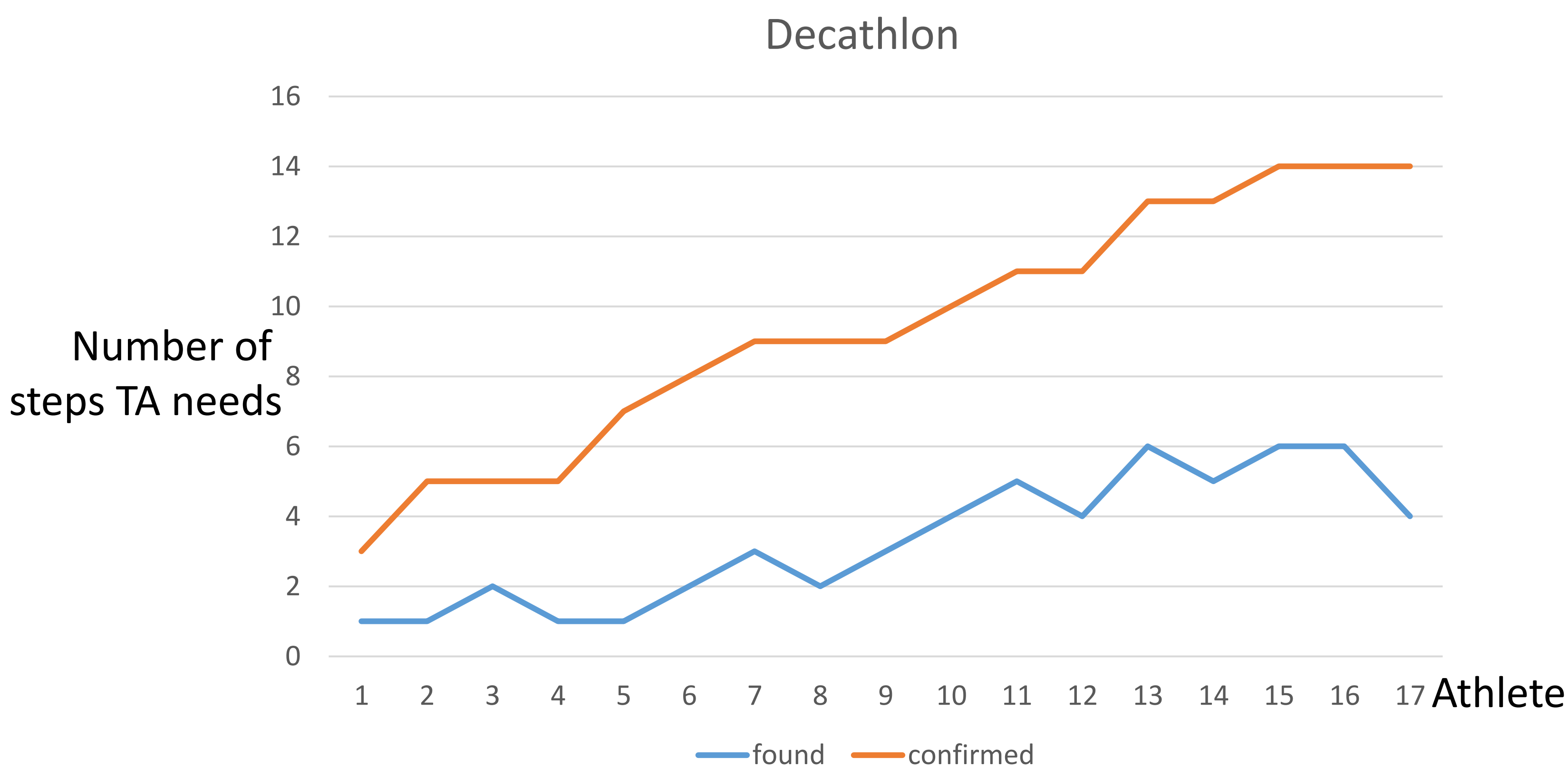
Alternatives

- Various server responses
  - No random access (some servers have both, some only one type of access, ...)
  - Answer is an ordered (step-by-step) list od object ID's (preference degree computation hidden – probably depending on user's behavior)
  - Answer is an ordered (step-by-step) list od object ID's and attribute values (preference degree computation hidden – probably depending on user's behavior)
- It is about data integration
  - preference degree computation on server side hidden – probably depending on user's behavior
  - preference degree computation on middleware side – our task?
- In house data?
  - Is TA useful at all?

Found versus confirmed in TA

Assume – data structure is given – computed offline $(N \log N)^m$

We can not influence found; we can try to influence confirmed



Found versus confirmed - heuristics

Confirm TA as early as possible, if t(R) bigger, τ smaller

$$\uparrow \quad t(R) \geq \tau = t(\underline{x}_1, \ldots, \underline{x}_m) \quad \downarrow$$

Heuristics can use

$$\partial t / \underline{x}_i$$

– if it is our first access to services in L_i , …

If we know from previous access some estimation of distribution of grades (attribute values when f_i is known) then

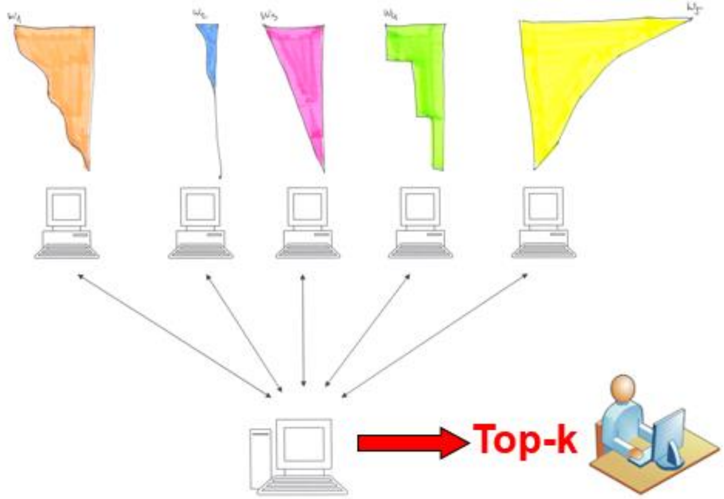
Heuristics can use

$$\text{estimationOf}(j^{th} \text{ element of } L_i) * \partial t / \underline{x}_i$$

Test heuristics on data (maybe it is domain dependent – e.g., Reuters collection has exponential distribution of lists)

Can we predict found? What is the probability that at step c(k) the list is already fixed?

Design experiments!



- Server
  - Create lists for u_i
  - Which data accessed

  by different users?
- middleware
  - Change aggregation
  - Measure influence
- client
  - Incremental run, when found, when confirmed
  - Step-by-step execution, measure top-k, 1-hit,

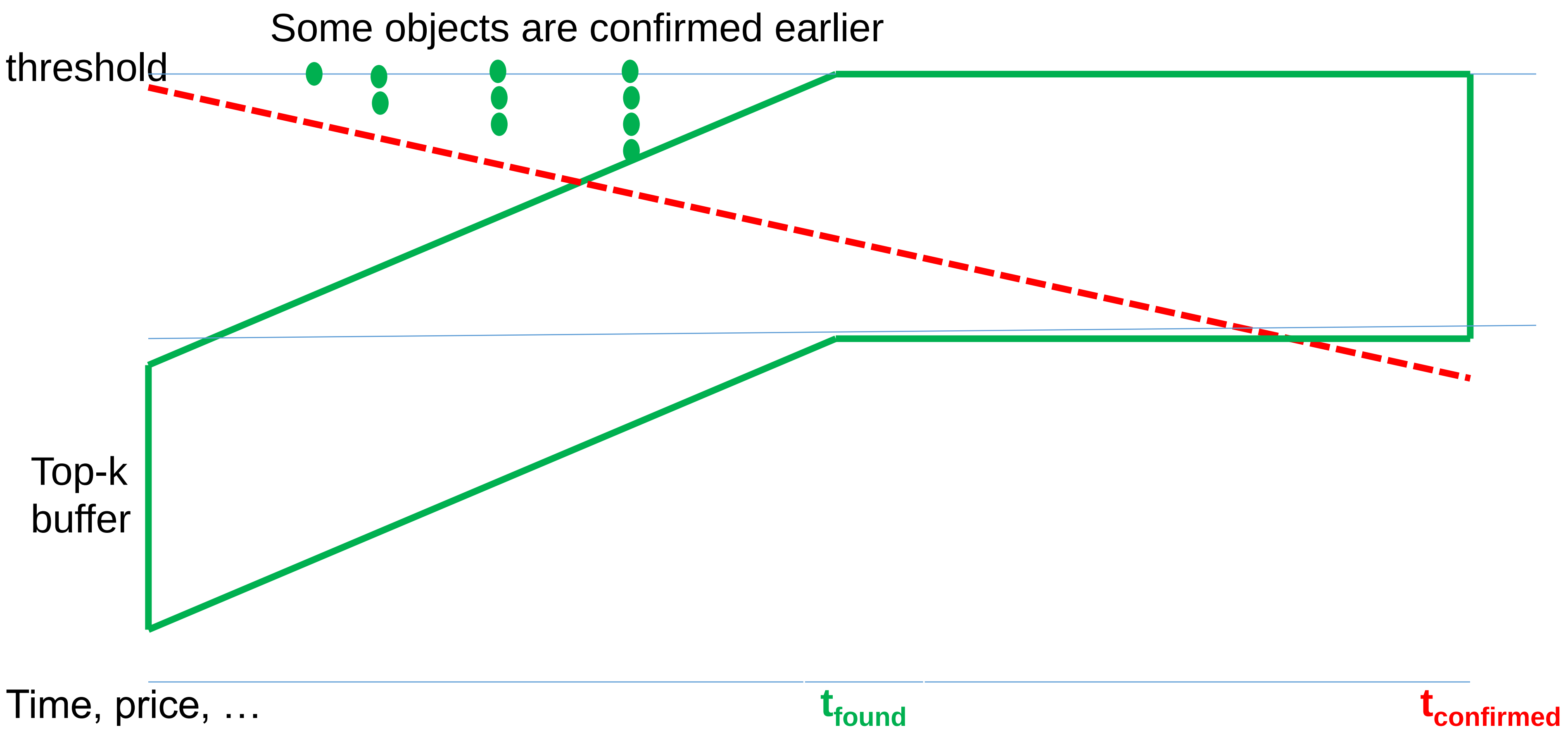
- Separate middleware server

Like IBM, Google, …

- Maybe an e-shop
- Maybe a “smart” app client

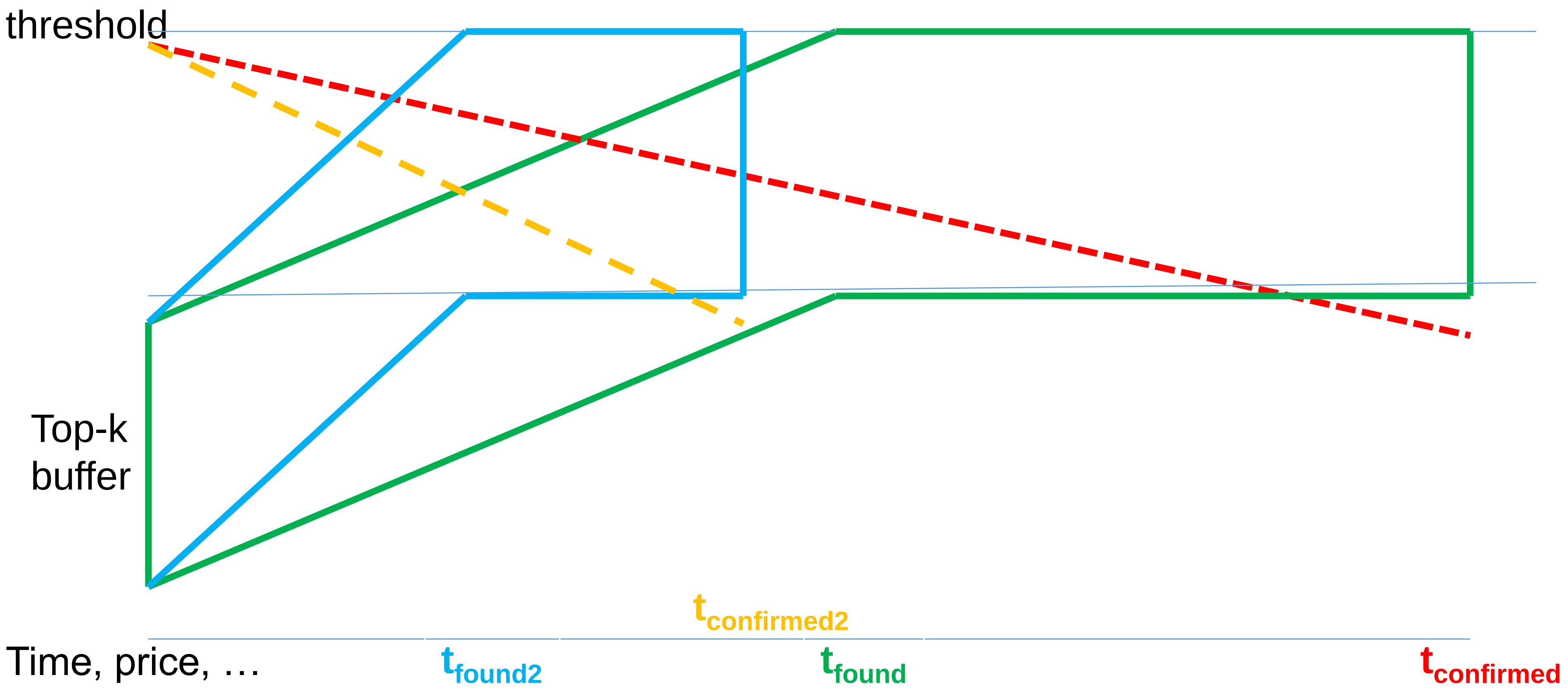
middleware

Visualizing TA – price ≈ time



Speed up ideas …

- Guntzer, Balke, Kießling [VLDB 2000]
- With [P. Gurský]



Heuristics GBK, PG, …

Increase frequency and depth of sequential access

$t(R) \geq \tau$

Where $\underline{x}_i * \partial t(\underline{x}_1, \ldots, \underline{x}_i, \ldots, \underline{x}_m) / \partial \underline{x}_i$ decrease most

Where $\underline{x}_i * \partial t(\underline{x}_1, \ldots, \underline{x}_i, \ldots, \underline{x}_m) / \partial \underline{x}_i$ decrease least

Discrete $\partial t(\underline{x}_1^{H(j)}, \ldots, \underline{x}_m^{H(j)}) / \partial \underline{x}_i * (\underline{x}_i^{H(j)} - \underline{x}_i^{H(j)+p})$

See PG page 18-20

How about TA in data cube?

Consider data cube with items B, C, D, E and preference cube with their images . During construction on x_1, x_2 axes are lists

$L_1 = \{(B,b_1), (D,d_1), (E,e_1),..$

$L_2 = \{(D,d_2), (B,b_2), (E,e_2)...$

Random access givesd$_1$,b$_2$

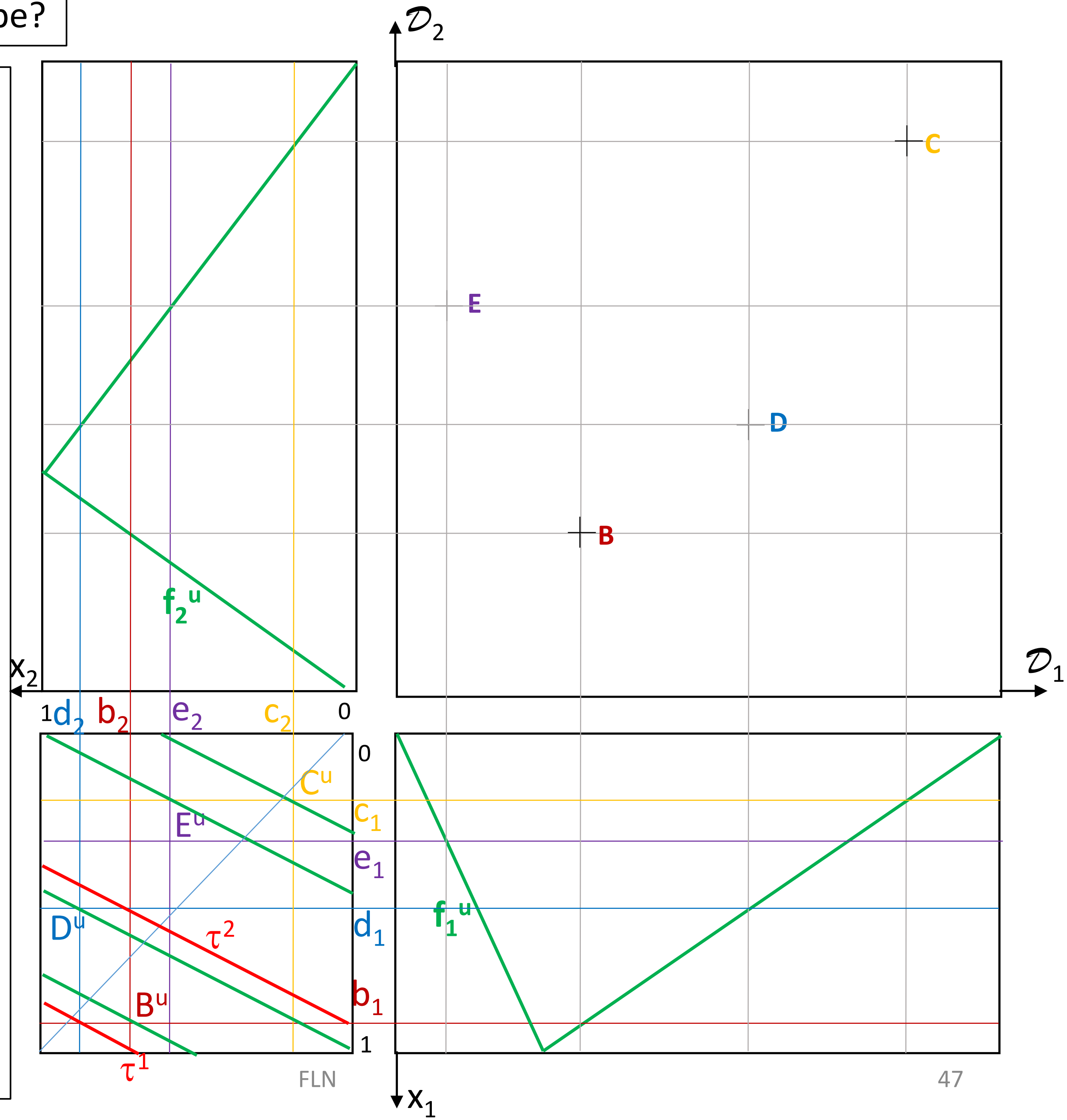
$B^u = (b_1,b_2)$, t(B) – see diagonal, c=1, $\underline{x}_1^1 = b_1$, $\underline{x}_2^1 = d_2$, $T^1 = (b_1, d_2)$

$Y^1 = \{(B, t(B)), (D, t(D))\}$,

$\tau^1 > t(B)$,

For c=2, $\underline{x}_1^2 = d_1$, $\underline{x}_2^2 = b_2$, $Y^2 = \{(B, t(B)), (D, t(D))\}$, $\tau^2 < t(B) < t(D)$, so we have top-2

But, we do not know which data points to take first, …



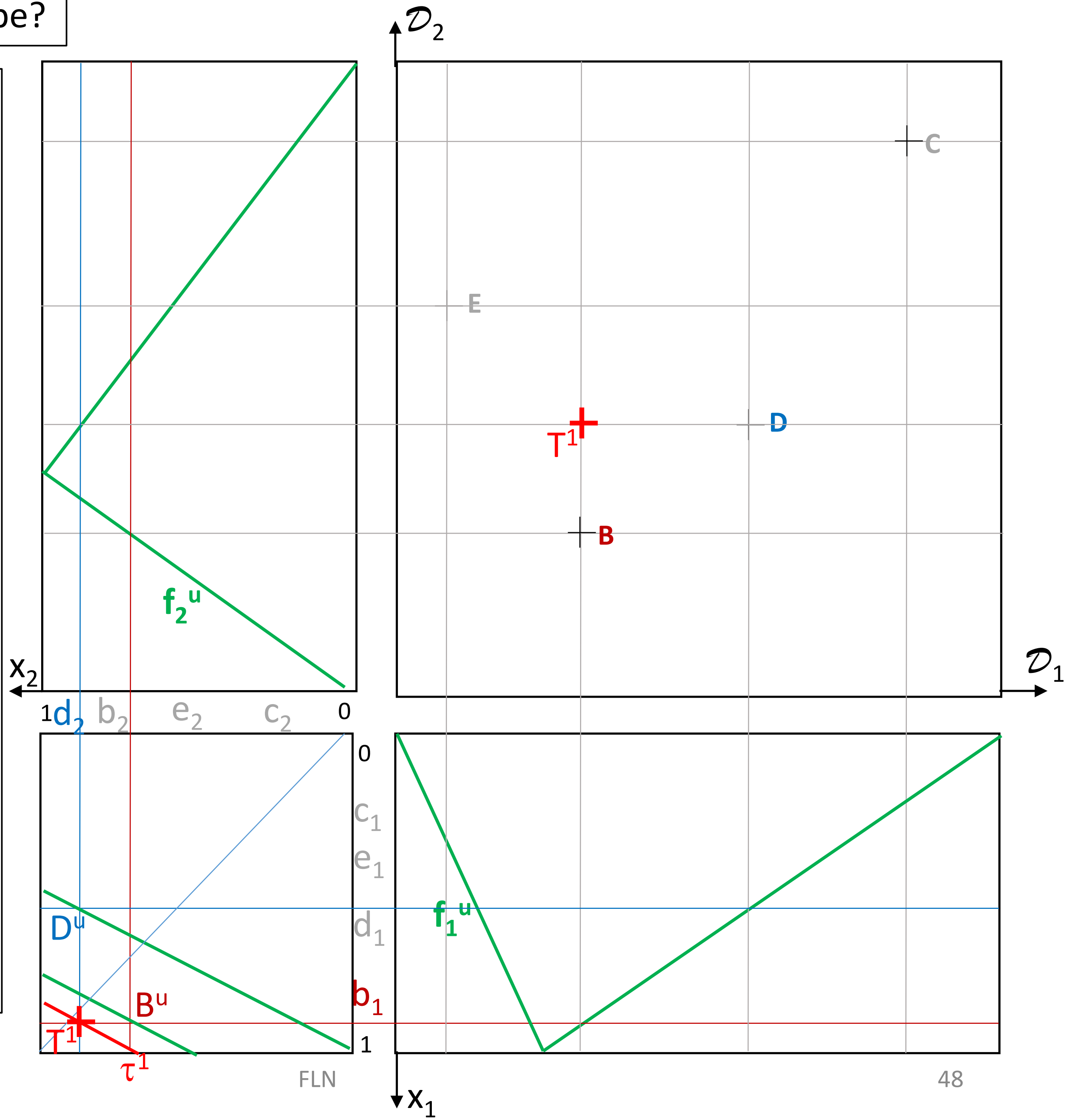
How about TA in data cube?

But, we do not know which data points to take first, …

So it is wrong to assume we have data cube with items B, C, D, E and preference cube …

It is more realistic to assume that server S_1 knows $B.\mathcal{A}_1$, $C.\mathcal{A}_1$, $D.\mathcal{A}_1$, $E.\mathcal{A}_1$, … and also f_1^u . Then server S_1 is able to create $L_1 = \{(B,b_1), (D,d_1), (E,e_1)$,.. Similarly server S_2 with $L_2 = \{(D,d_2), (B,b_2), (E,e_2)$… In step 1 we (at middleware we know t and are able to compute τ^1) do not get best item.

…



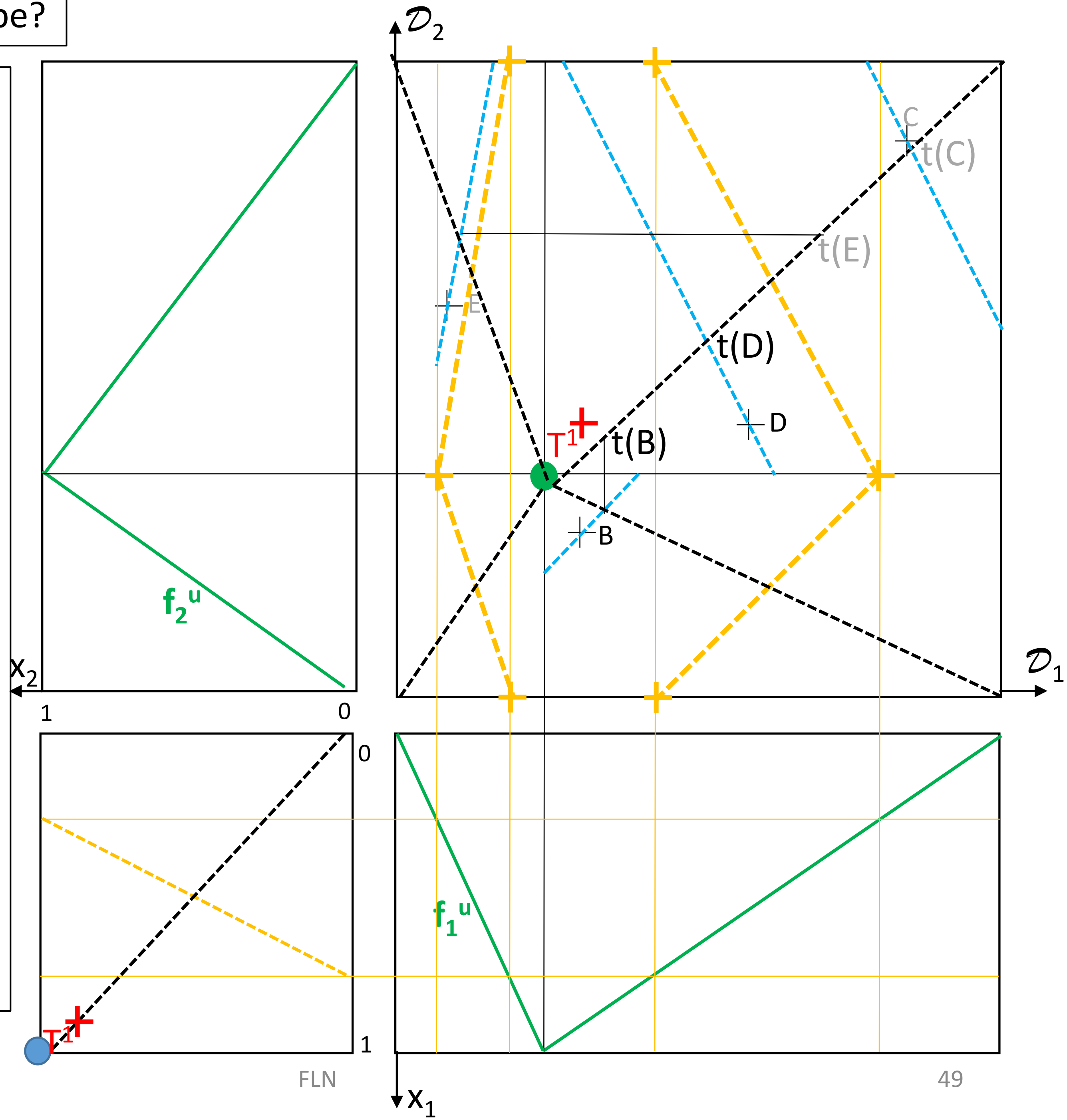
How about TA in data cube?

We can illustrate in DC what happens during TA first step computation, or

... in case of **in-house** data we have data cube with items B, C, D, E and the aggregation function t in preference cube, user preferences f_i^u . ...

It is more realistic to compute one contour lines in DC, then take parallel CL of the point in the quadrant (here the blue ones), intersect with the quadrant diagonal, proportional value of these in North-East quadrant gives ordering,

this gives t(B) < t(D) ...



End of lecture

Questions?
Comments?