Building rankings from hierachical systems of ordinal indicators

Marco Fattore, Alberto Arcagni
University of Milano – Bicocca

Neuchâtel, October 2018

Hierarchical Multi Indicator Systems

- Indicators are organized by **categories** representing different **topics** of the analysis
- The **goal** is to get a synthesis and a ranking for the whole system
- The **usual solution** is to get global ranking from the poset obtained by the **intersection** of the intermediate rankings

Inefficiency

- Ranking extraction is a “complexity” reduction process which necessarily **loses information** on the input posets
- **Repeated** ranking extraction as in the usual solution (firstly to extract intermediate ranking then for the global one) increases the information loss
- There is usually **no control** on such information loss

Our approach

- We want to **preserve as much complexity as possible**, so we want to find out a way to "put together" the input posets into a "synthetic" one
- The construction of the "synthetic" poset must be performed as an **"order structures" preserving** process
- We want to **control** the information loss

The algorithm

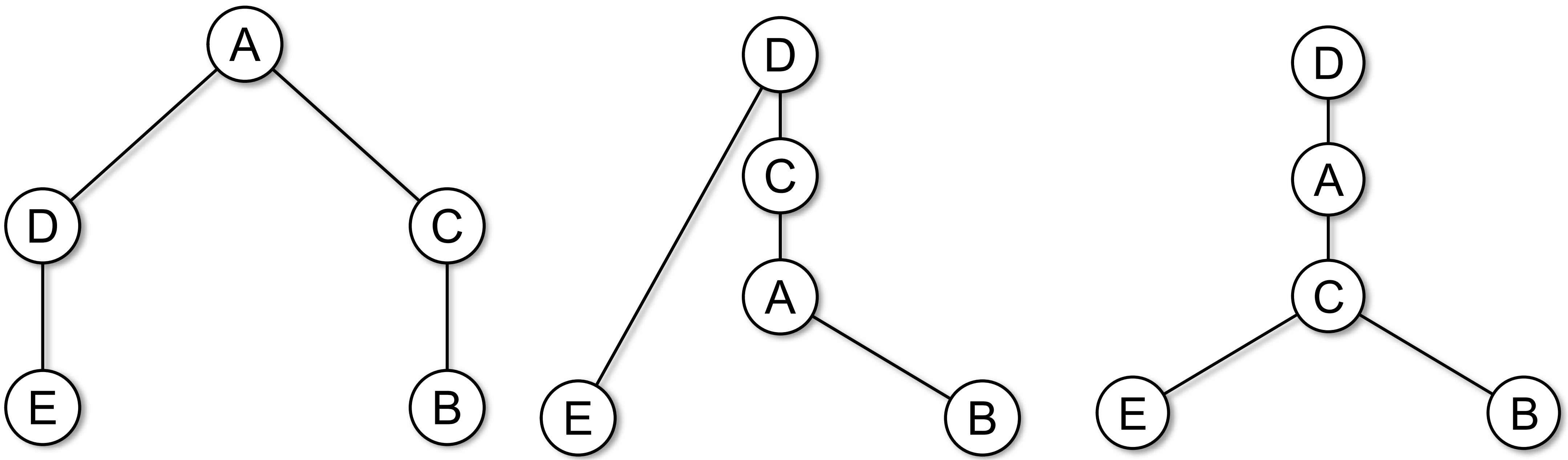
- The **object of each iteration** is a poset that is a synthesis of the input posets
- The **initial poset** is the one composed by all the comparabilities shared by the input posets but also the one with the largest "complexity": the intersection poset
- For **each iteration** a comparability is introduced to reduce the "complexty". For each comparability is evaluated a loss function
- The comparability introduced is the one that minimize it
- The algorithm has a finite **number of iterations**, lower or equal to the number of incomparabilities of the intersection poset
- The value of the loss function associated to the inserted comparability is associated to the iteration

Loss function

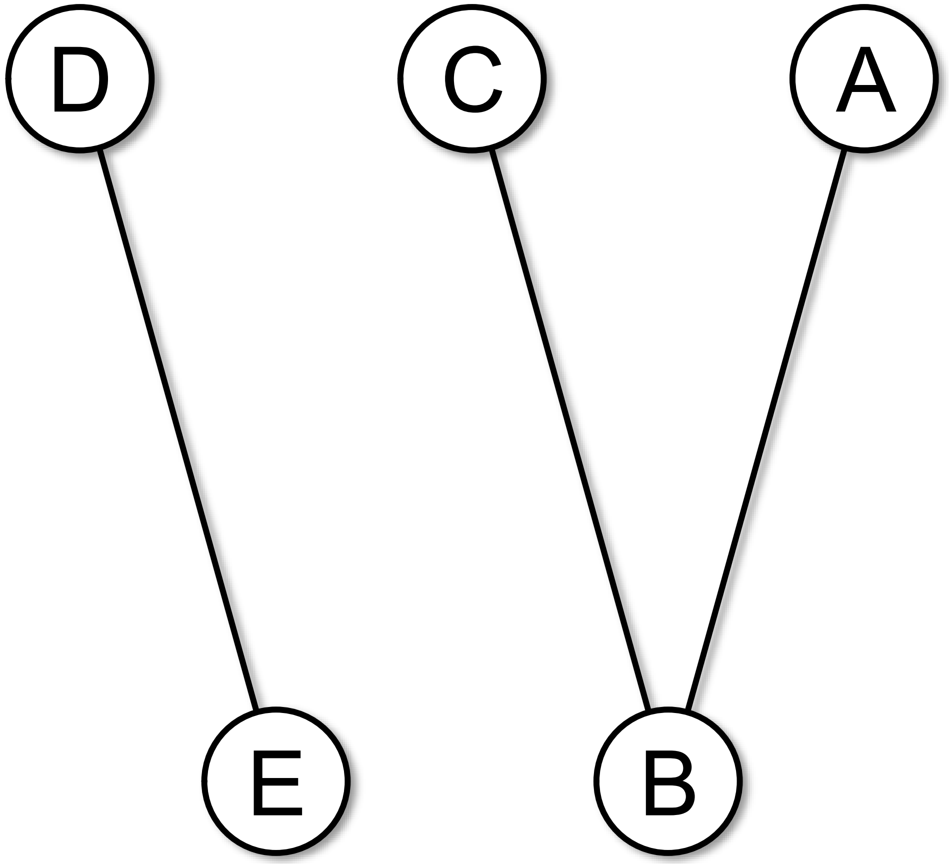
$$d(l) = \frac{1}{k}\sum_{j=1}^{k}\frac{\left\|\mathrm{MRP}(l) - \mathrm{MRP}(\Pi_j)\right\|_1}{\left\|\mathrm{MRP}(\Pi_j)\right\|_1}$$

- The loss function is the **average of the distances** of the MRP matrix of the iteration poset from the MRP matrices of the input posets

Example: input posets

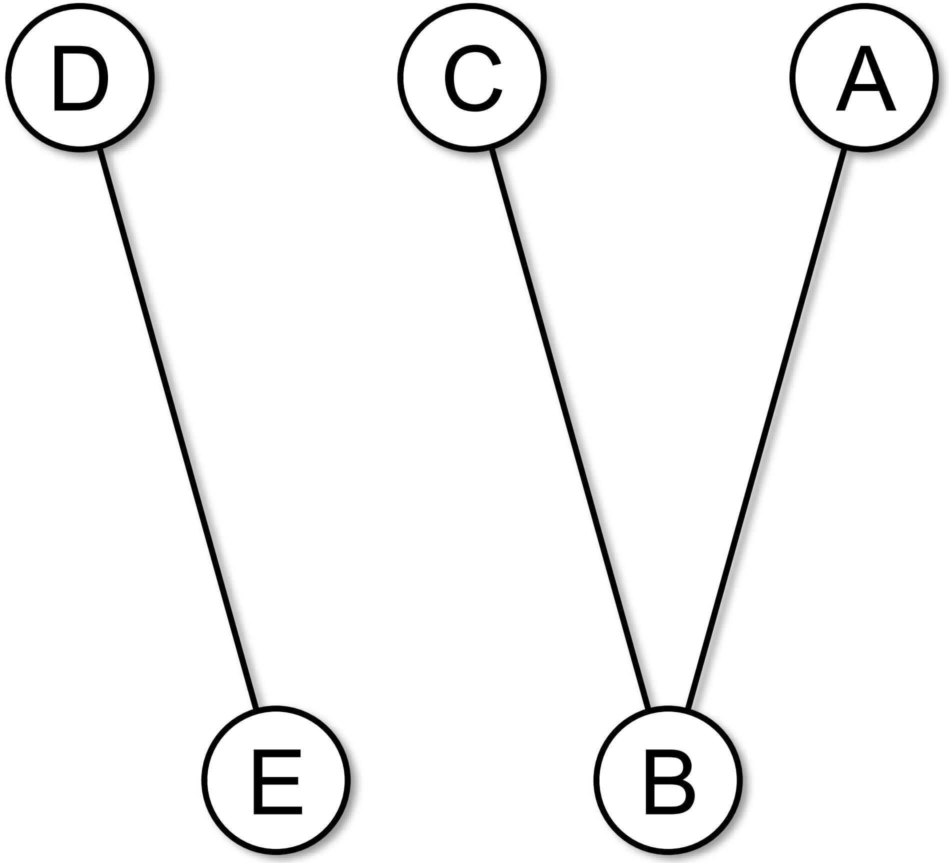


Example: Intersection



$$d(0) = \frac{0.3 + 0.4 + 0.4}{3} = 0.3667$$

Example: First iteration

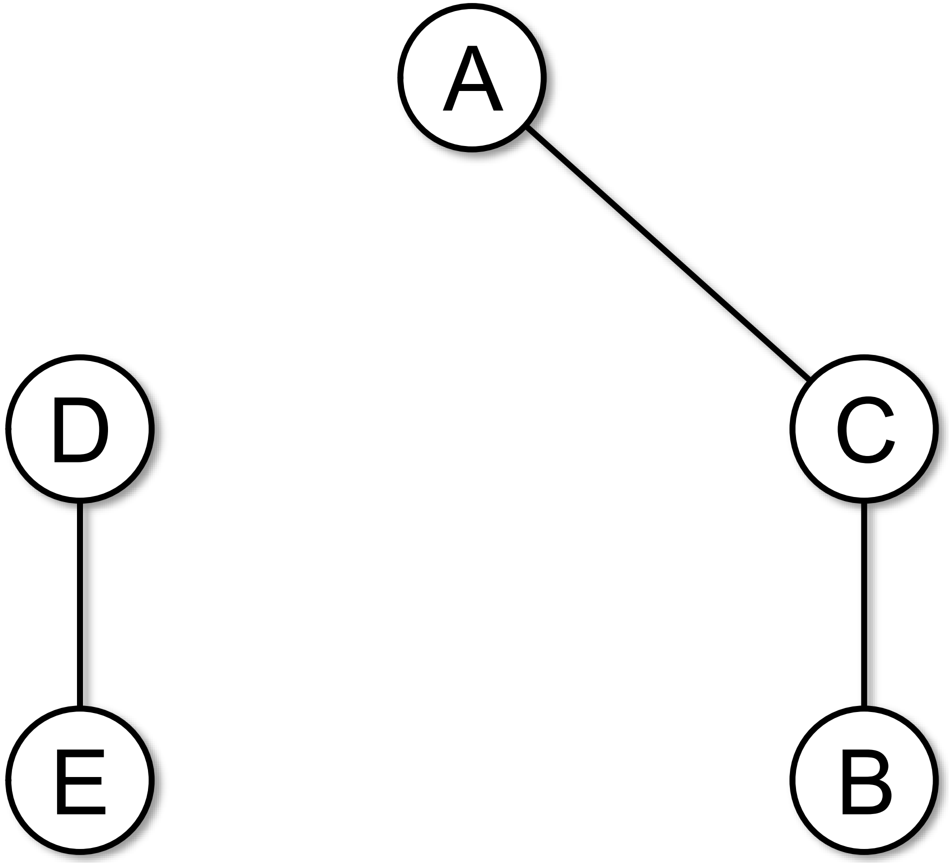


$d(1)$

	A	B	C	D	E
A			0.4222	0.3545	0.5667
B				0.3593	0.4333
C	0.3400			0.3485	0.5333
D	0.4370	0.7667	0.5185		
E	0.3556	0.5167	0.3889		

7 incomparabilities

Example: 2nd iteration

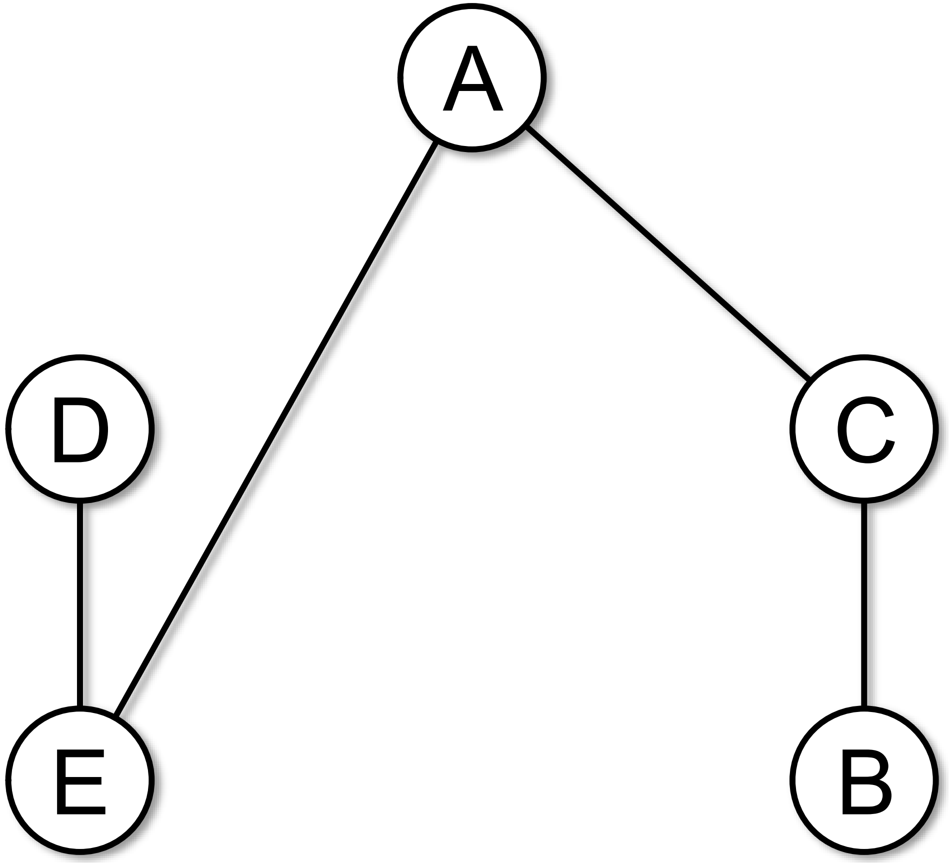


$d(2)$

	A	B	C	D	E
A				0.3333	0.6000
B				0.3370	0.4222
C				0.3381	0.5111
D	0.3556	0.7333	0.5333		
E	0.3259	0.4667	0.3556		

6 incomparabilities

Example: 3rd iteration

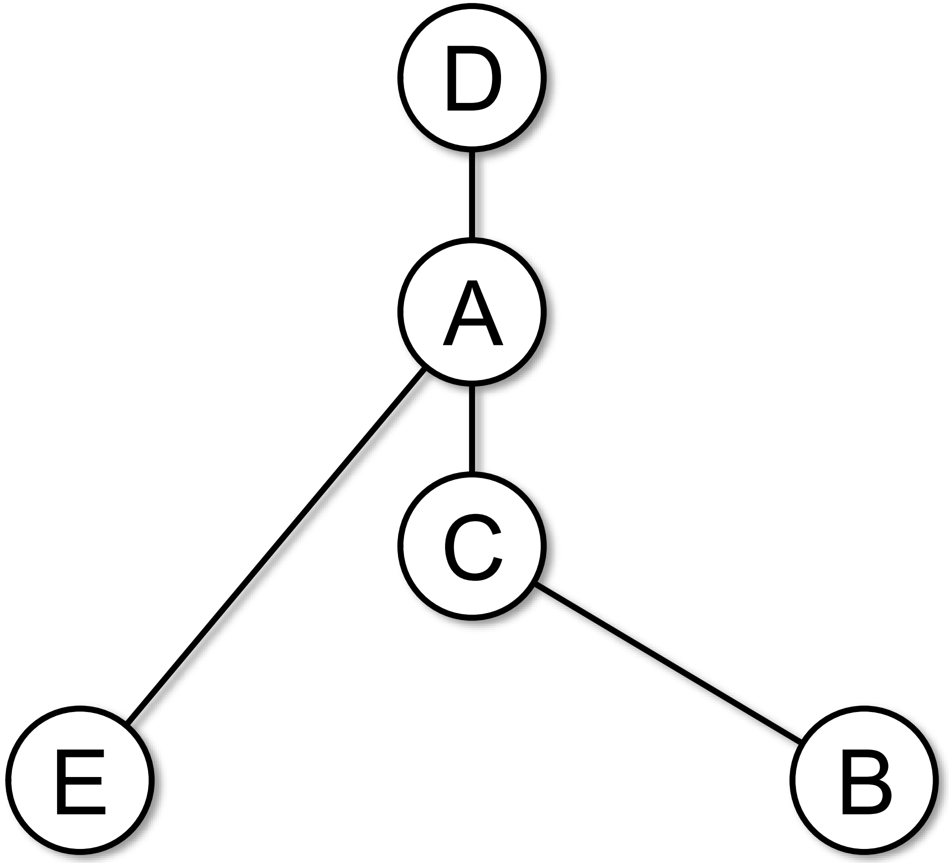


$d(3)$

	A	B	C	D	E
A				0.2778	
B				0.3167	0.4000
C				0.3111	0.5000
D	0.3556	0.7333	0.5333		
E		0.4667	0.3556		

5 incomparabilities

Example: 4th iteration

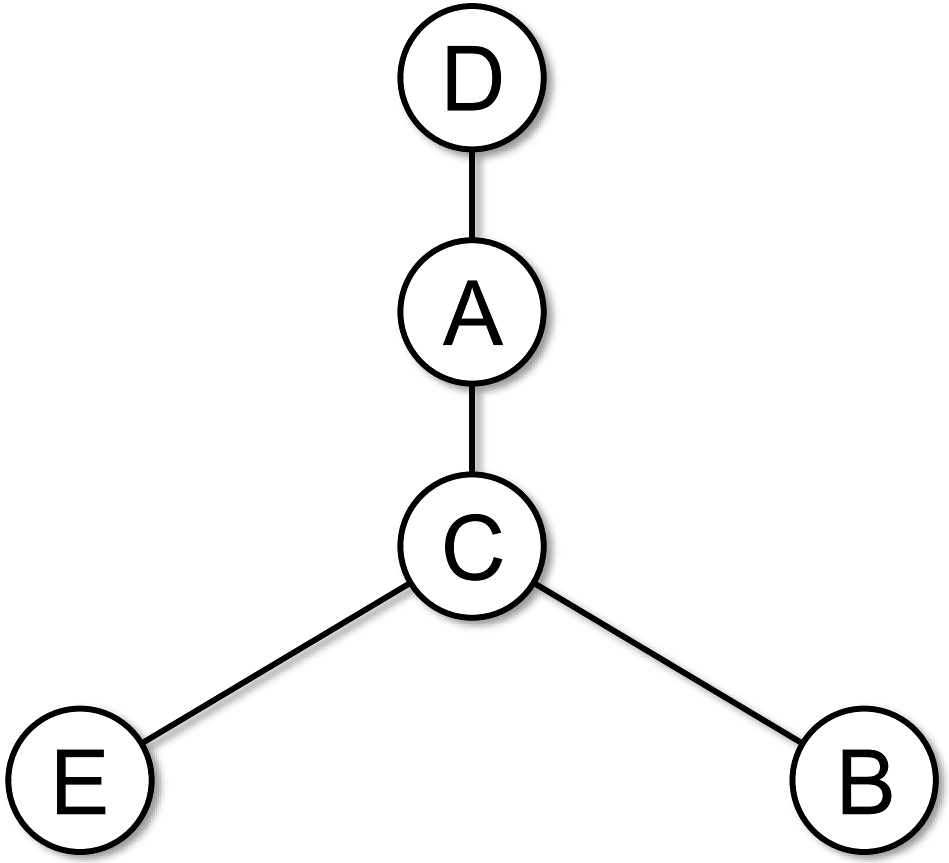


$d(4)$

	A	B	C	D	E
A					
B					0.3667
C					0.4667
D					
E		0.3556	0.2556		

2 incomparabilities

Example: 5th iteration

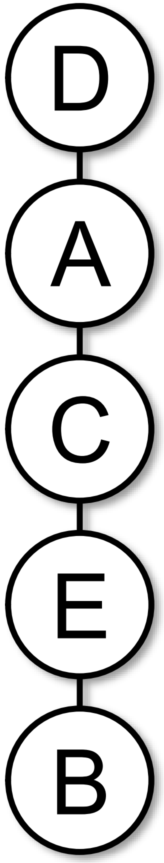


$d(5)$

	A	B	C	D	E
A					
B					0.3222
C					
D					
E		0.3556			

1 incomparability

Example: Complete order

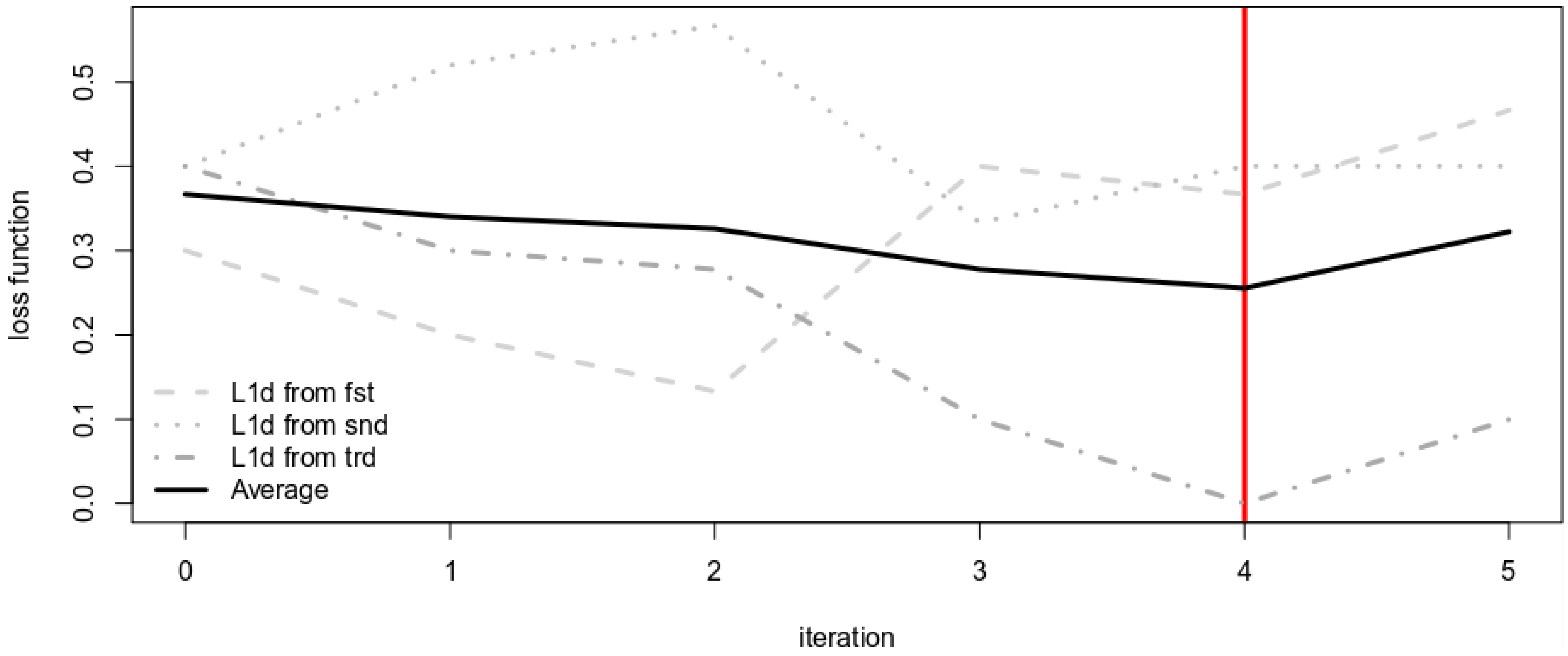


$d(5)$

	A	B	C	D	E
A					
B					
C					
D					
E					

0 incomparabilities

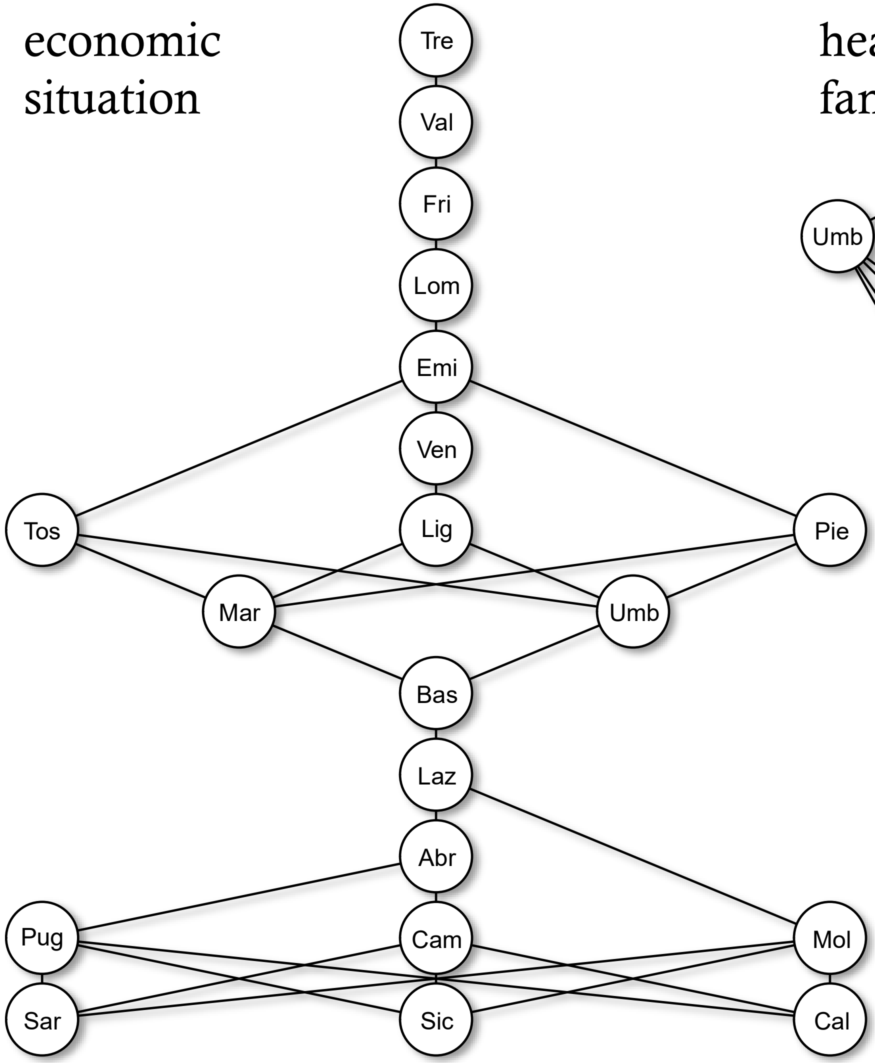
Example: Loss function



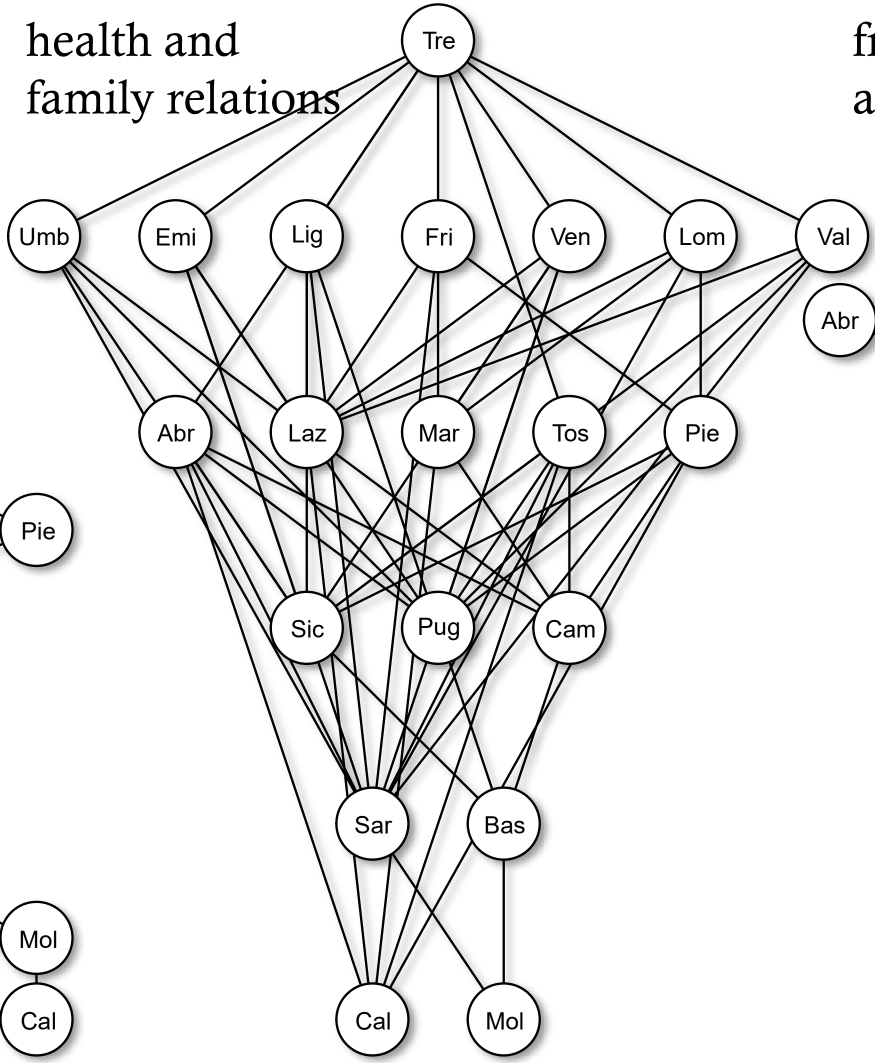
5 iterations

Multipurpose survey: aspects of daily life

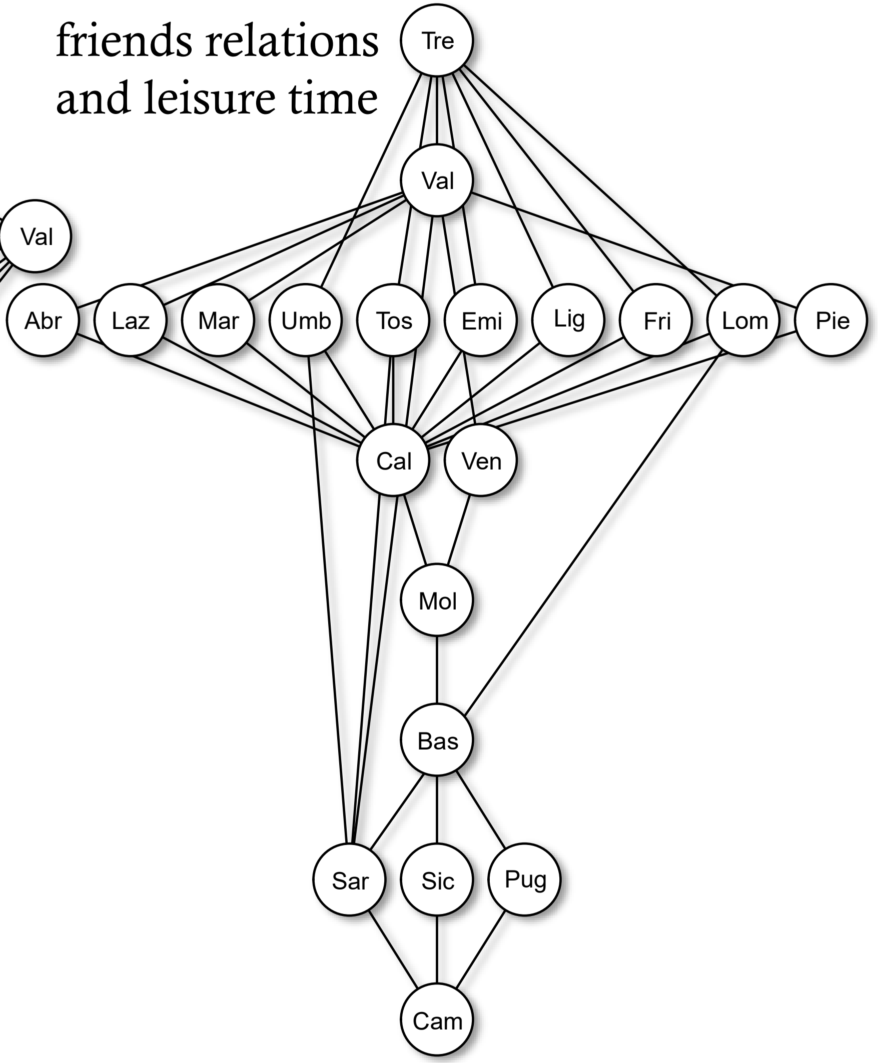
Satisfaction with economic situation



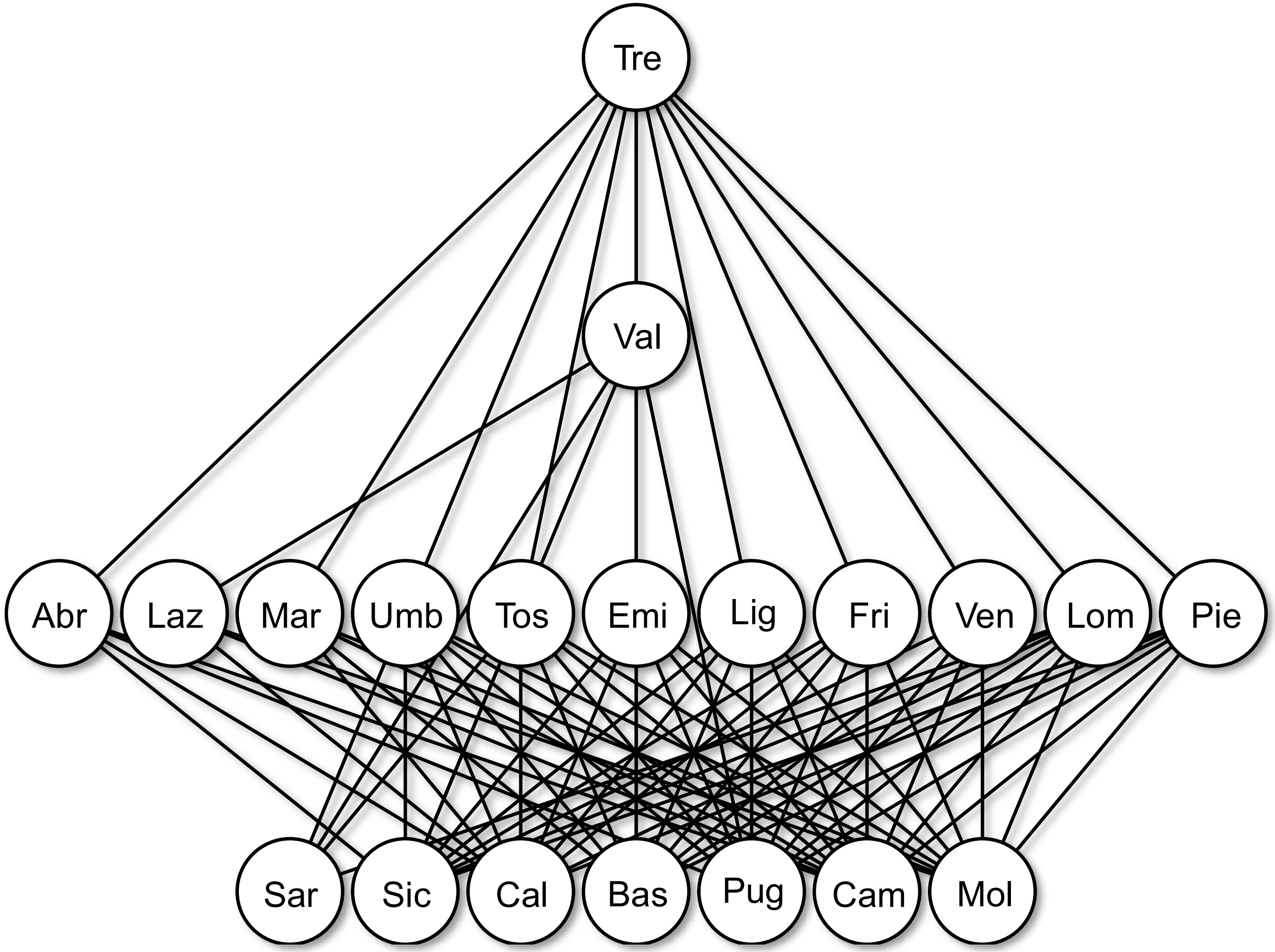
Satisfaction with health and family relations



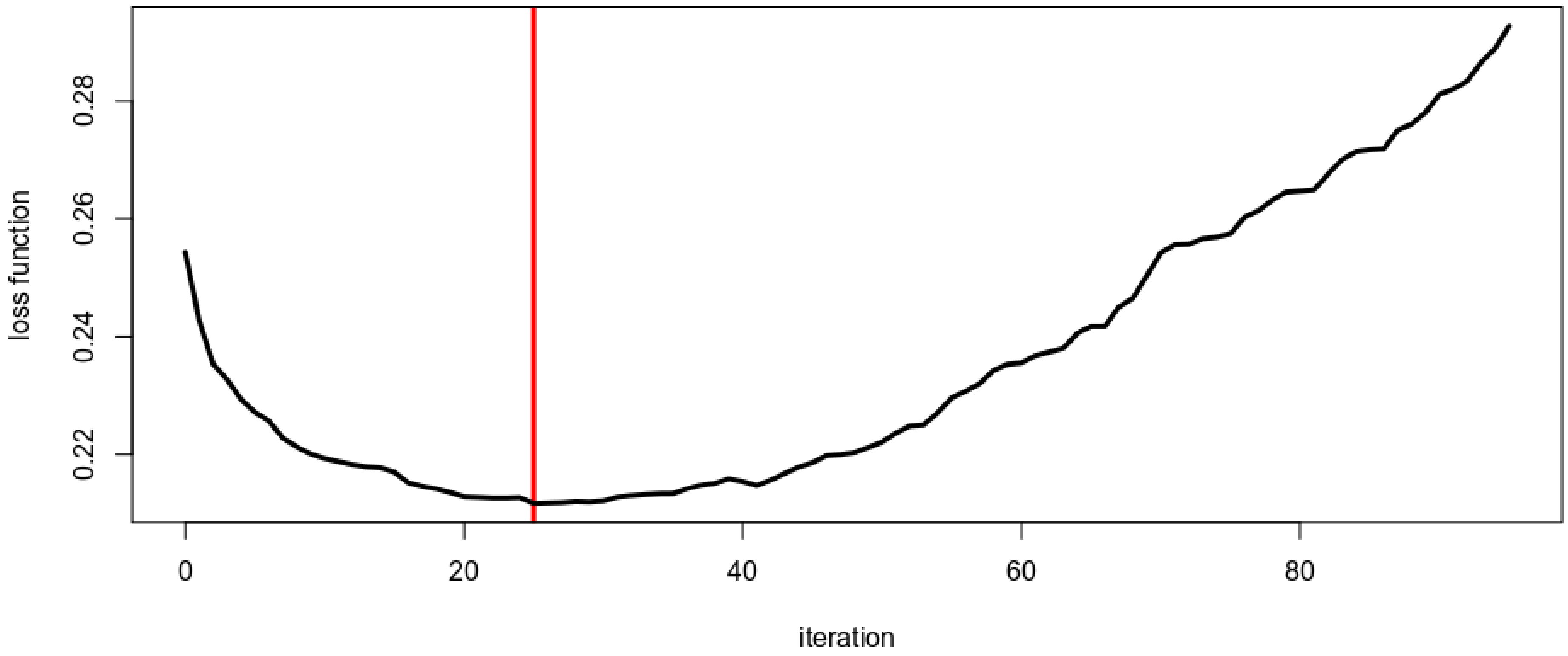
Satisfaction with friends relations and leisure time



Intersection

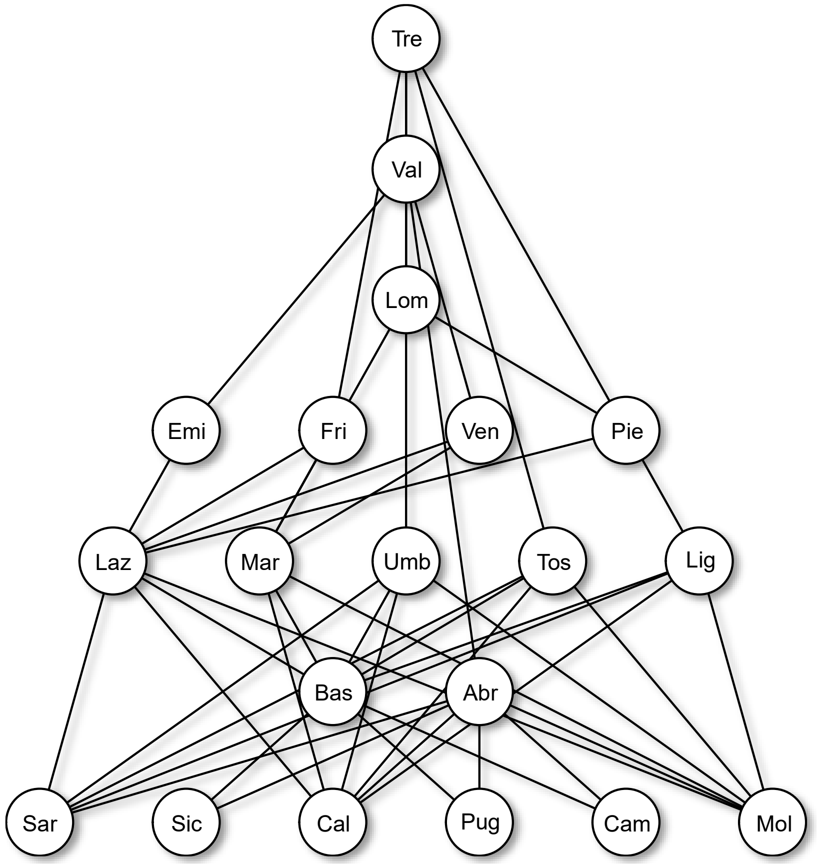


Loss function



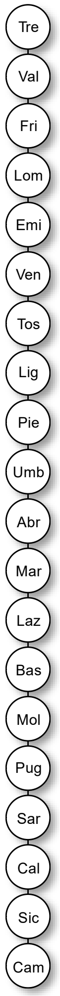
94 iterations

Optimum



- Optimum at 25-th iteration
- $d(25) = 0.2117$

Complete order



1	Trentino – Alto Adige	11	Abruzzo
2	Aosta Valley	12	Marche
3	Friuli Venezia Giulia	13	Lazio
4	Lombardy	14	Basilicata
5	Emilia – Romagna	15	Molise
6	Veneto	16	Apulia
7	Tuscany	17	Sardinia
8	Liguria	18	Calabria
9	Piedmont	19	Sicily
10	Umbria	20	Campania

- $d(94) = 0.2889$
- $\frac{d(94)}{d(25)} = 1.3694$

Conclusions

- The researcher can **choose** the step poset to adopt: the one that minimize the loss function or the final one that minimize the "dimensionality", a complete order
- If the researcher choose the complete order, he has the corresponding absolute value of the loss function and can compare it with its minimum
- The algorithm is **heuristic** but it returns reasonable results
- Differently by the "usual solution" the "dimensionality" reduction is **applied only once** and comparabilities are introduced taking into account all the input posets and not a synthesis of them
- Different **loss functions** can be proposed in order to reduce the **computational intensity** and to search **different optima**
- The algorithm is going to be released in the **R package PARSEC**