






Belgian Road Research Centre
Your partner for sustainable roads

PCLUST: An extension of PROMETHEE to interval clustering

Renaud Sarrazin[1,2], Yves De Smet[2], Jean Rosenfeld[3]

[1] MSM division, Belgian Road Research Centre, Brussels, Belgium
[2] CoDE-SMG laboratory, Université libre de Bruxelles, Brussels, Belgium
[3] BEAMS laboratory, Université libre de Bruxelles, Brussels, Belgium

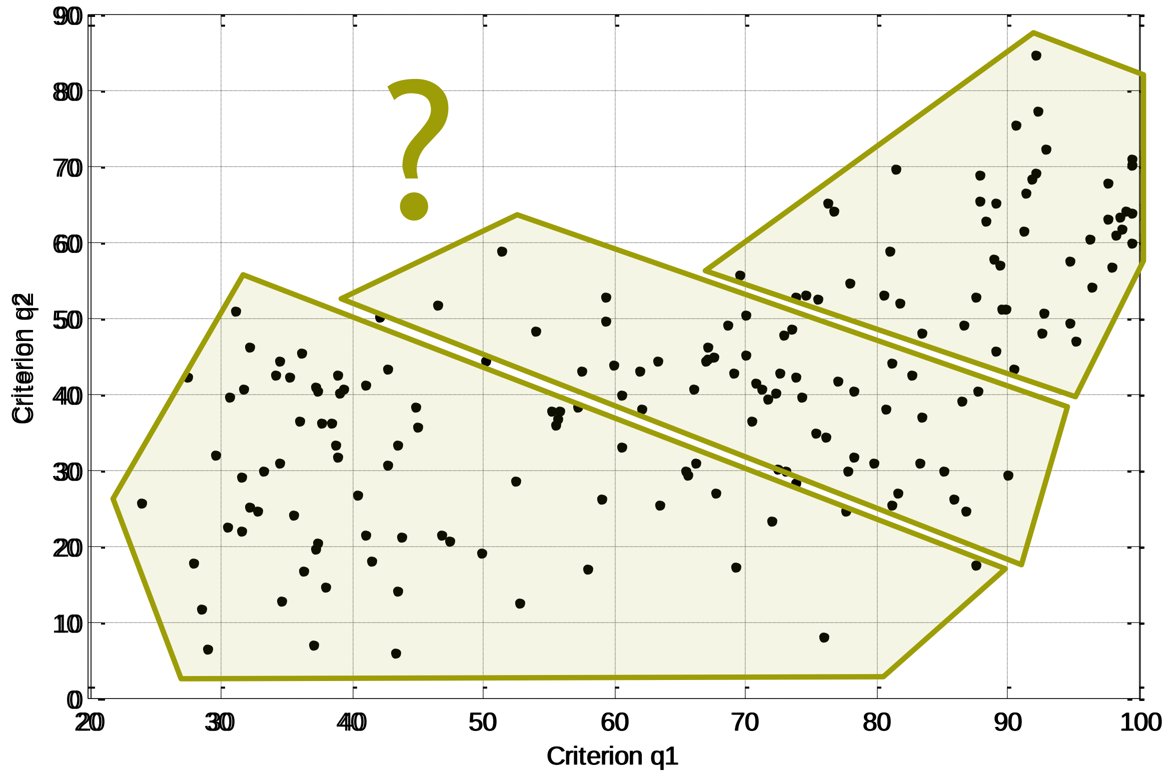
2nd International MCDA Workshop on PROMETHEE: Research and Case Studies – Extensions and Theoretical Developments

Research question

Dealing with a problem of multicriteria clustering

Aim of this research:

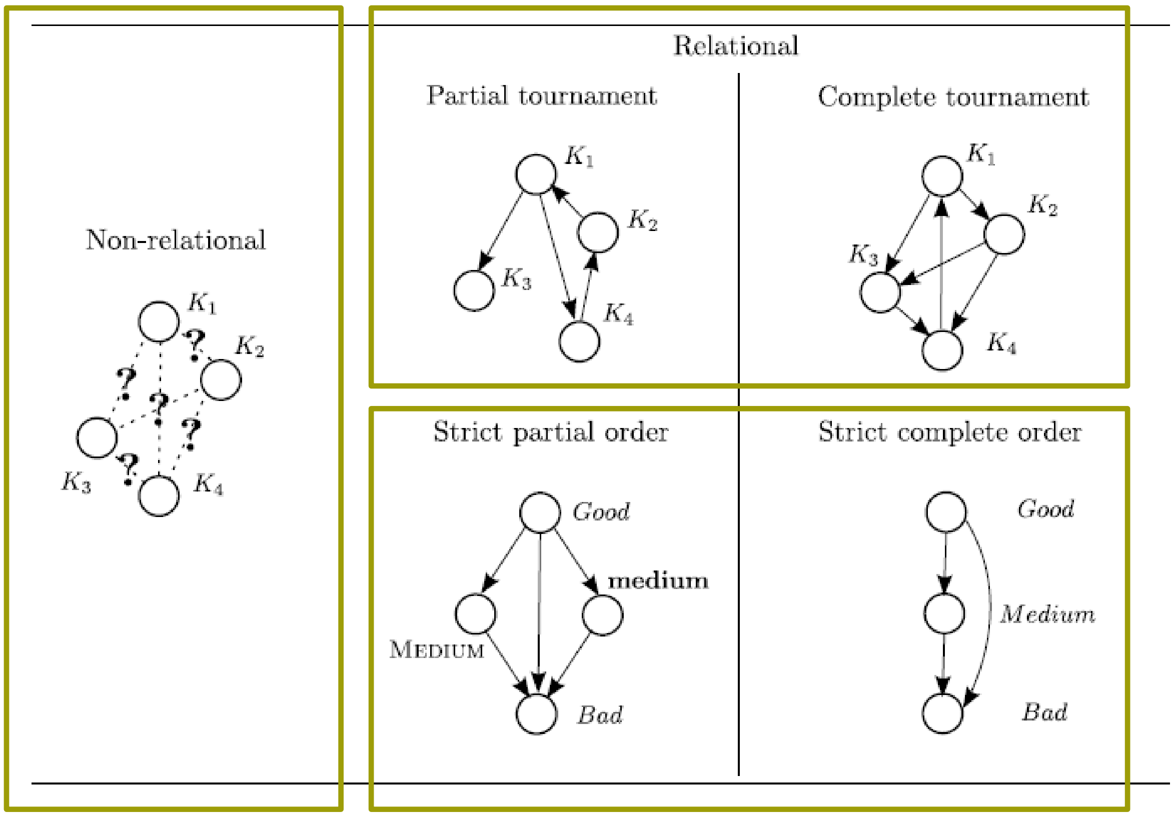
- Consider the multicriteria nature of the problem
- Construct interval clusters (i.e., in partial order)



Context

Problematic of multicriteria clustering

Non-relational, relational and ordered clustering



[A-L Olteanu, 2013]

Context

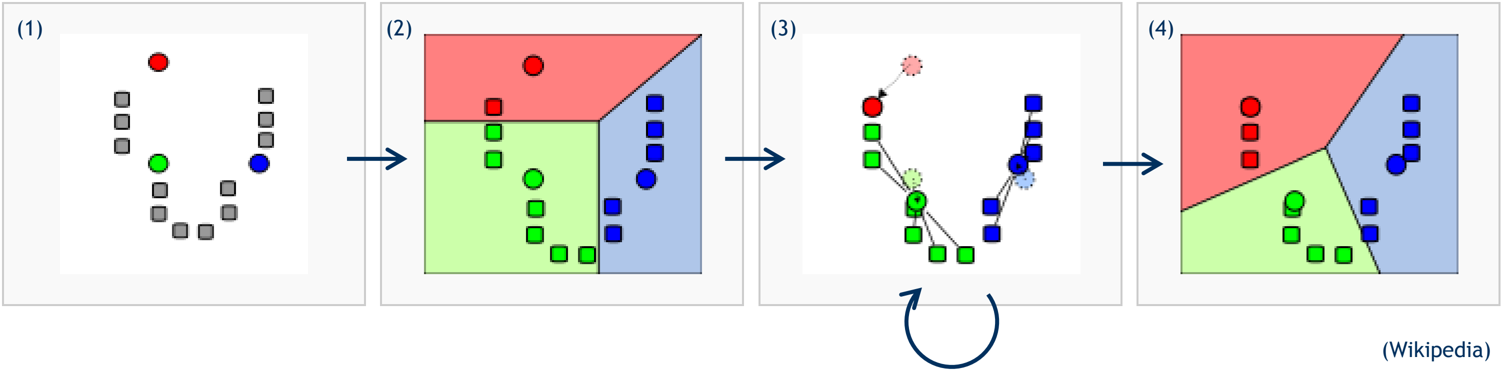
Non-relational clustering: no criteria-dependency

Many methods exists to solve clustering problems (e.g. *k*-means algorithms)
Most of them rely on a **distance measure**!

- Minimize the inner-distance of each group
- Maximize the inter-distance between groups
- Not really appropriate in multicriteria contexts

Use of the preference relations between alternatives!

Illustration of the *k*-means algorithm:



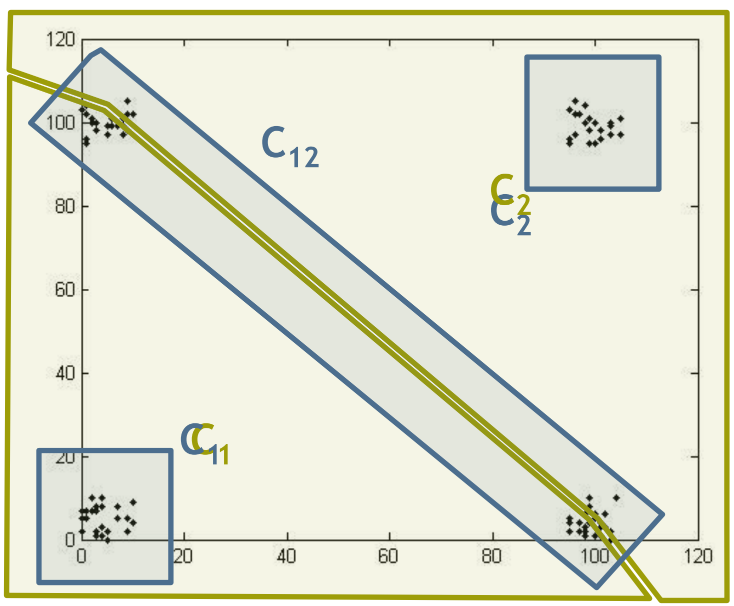
(Wikipedia)

Context

Relational and ordered clustering: criteria-dependency
Consider the additional information given by the criteria

In this contribution, we focus on multicriteria ordered clustering
In particular, we address the problem of interval clustering (i.e. partial order)
Strong interest in using such an approach

Illustrative example:



2 criteria
80 alternatives located in 4 areas of the space
2 principal clusters required

Solution
Completely ordered clustering
Interval clustering

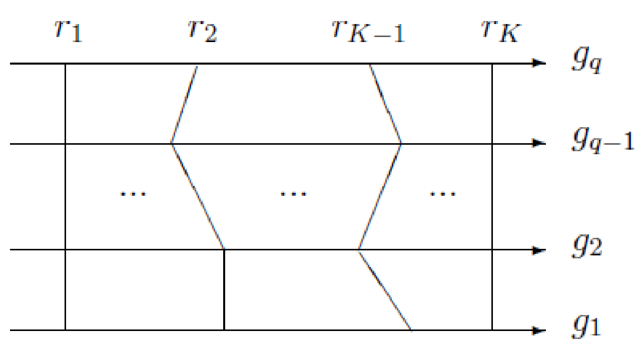
FlowSort method

Developed for solving problems of multicriteria sorting
Let consider:

- $A = \{a_1,\dots,a_n\}$: set of alternatives
- $F = \{g_1,\dots,g_q\}$: set of criteria
- $\kappa = \{C_1,\dots,C_K\}$: set of clusters
- $R = \{r_1,\dots,r_K\}$: set of corresponding central profiles
- $R_i = R \cup \{a_i\}$

Assignment rule

$$C_\phi(a_i) = C_h \text{ if: } \left|\phi_{R_i}(r_h) - \phi_{R_i}(a_i)\right| = \min_{\forall j}\left|\phi_{R_i}(r_j) - \phi_{R_i}(a_i)\right|$$



PCLUST model

Enrichment of the ***k*-means** procedure with the **FlowSort** assignment rule
Adaptation of the assignment rule to interval clustering

Pseudo code

1. Initialization of the central profiles
2. Assignment of each alternatives to the categories
3. Update of the central profiles
4. Repeat until convergence of the model

PCLUST model

1. **Initialization**
   - Randomly
   - Equidistribution of the evaluations

2. **Assignment rule**

$$C_{\phi^+}(a_i) = C_h \text{ if: } \left|\phi^+_{R_i}(r_h) - \phi^+_{R_i}(a_i)\right| = \min_{\forall j}\left|\phi^+_{R_i}(r_j) - \phi^+_{R_i}(a_i)\right|$$

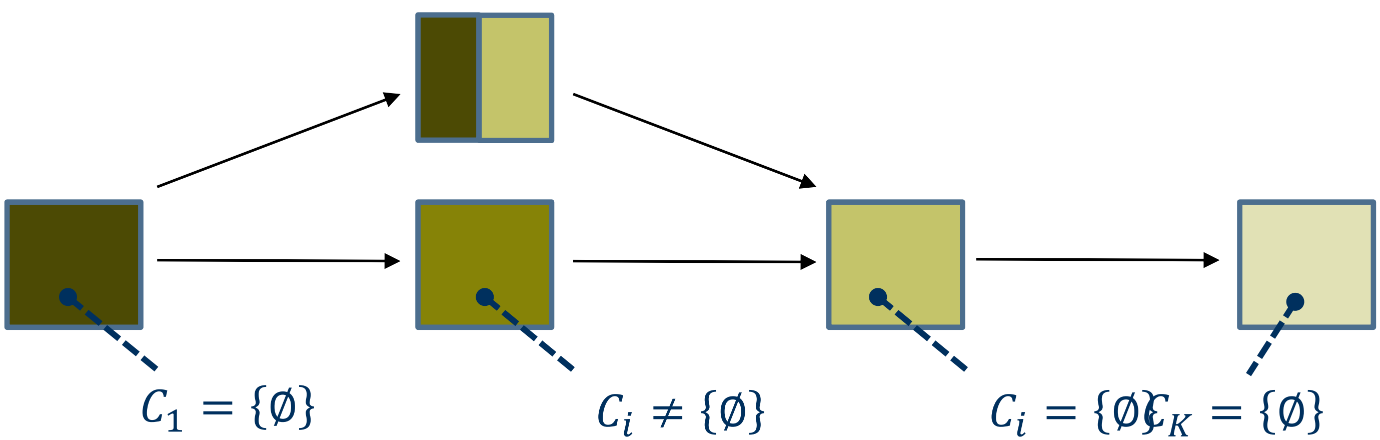
$$C_{\phi^-}(a_i) = C_l \text{ if: } \left|\phi^-_{R_i}(r_h) - \phi^-_{R_i}(a_i)\right| = \min_{\forall j}\left|\phi^-_{R_i}(r_j) - \phi^-_{R_i}(a_i)\right|$$

$$\forall a_i \in A, \forall h, l \in \{1 \dots K\} : \begin{cases} \text{if } C_{\phi^+}(a_i) = C_{\phi^-}(a_i) = C_h & : a_i \in C_h \\ \text{else} & : a_i \in C_{h,l} \end{cases}$$

PCLUST model

3. **Update of the central profiles // First function Upd1**

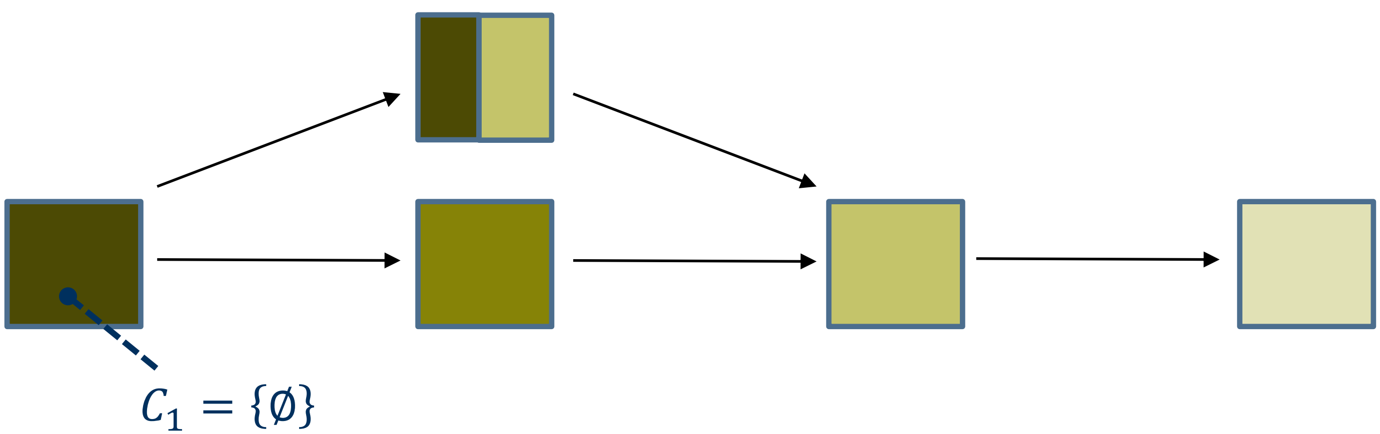
- Non-empty principal categories - median value
- Empty principal categories
  - Extreme categories
    - Interval – median value
    - No interval – bounded random value
  - Non-extreme categories – bounded random value



PCLUST model

3. Update of the central profiles // Second function Upd2

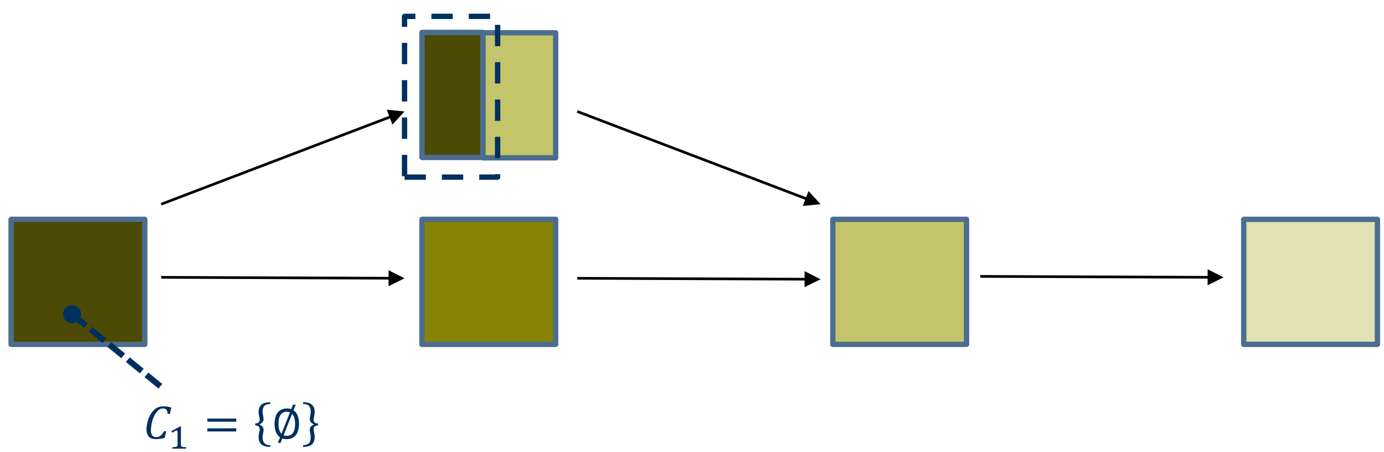
- Non-empty principal categories - median value
- Empty principal categories
    - Extreme categories
        - Interval –
        - No interval – bounded random value
    - Non-extreme categories – bounded random value



PCLUST model

3. Update of the central profiles // Second function Upd2

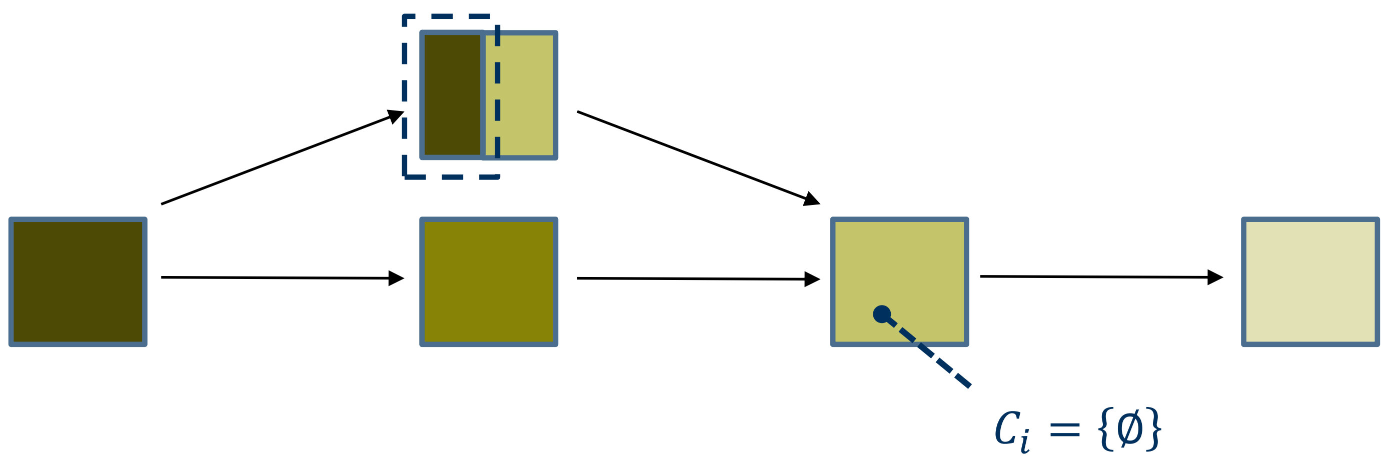
- Non-empty principal categories - median value
- Empty principal categories
  - Extreme categories
    - Interval – median value of the closest alternatives
    - No interval – bounded random value
  - Non-extreme categories – bounded random value



PCLUST model

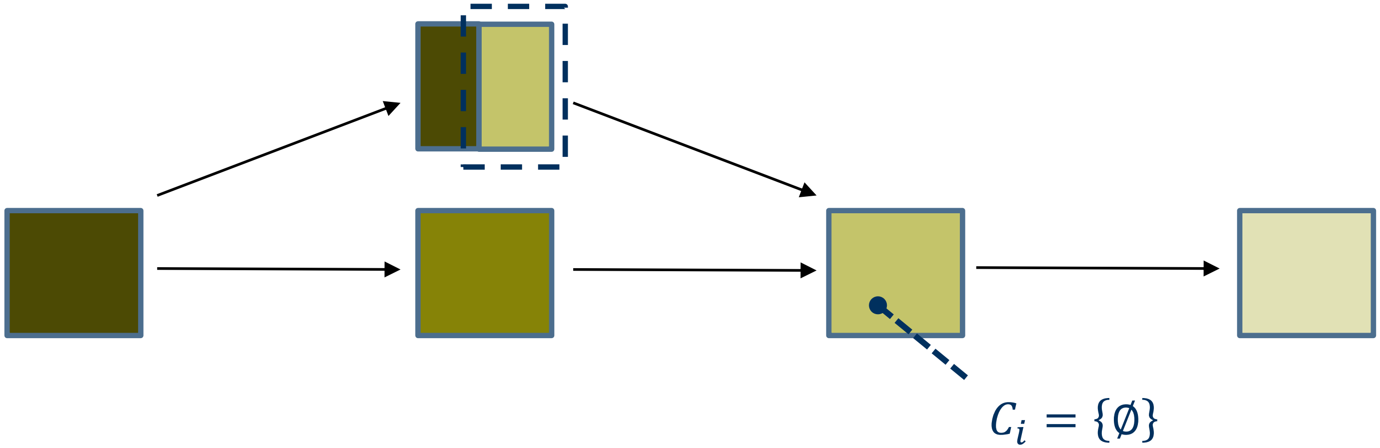
3. Update of the central profiles // Third function Upd3

- Non-empty principal categories - median value
- Empty principal categories
  - Extreme categories
    - Interval – median value of the closest alternatives
    - No interval – bounded random value
  - Non-extreme categories



PCLUST model

3. **Update of the central profiles // Third function Upd3**
   - Non-empty principal categories - median value
   - Empty principal categories
     - Extreme categories
       - Interval – median value of the closest alternatives
       - No interval – bounded random value
     - Non-extreme categories
       - Interval – median value of the closest alternatives
       - No interval – bounded random value



Validation

Evaluation of the update functions and initialization procedures
Comparison with existing procedures (*k*-means and P2CLUST)

Two structured datasets

- Environmental Performance Index (EPI 2014)
- CPU evaluation (UCI repository)

Table 1: Parameters of the EPI dataset

n	178
q	2
w	$\{0.4, 0.6\}$
P_k	$\{q_k = 10, p_k = 50\}$

Table 2: Parameters of the CPU dataset

n	209
q	6
w_k	0.167
P_k	$\{q_k = 0.1, p_k = 0.5\}$

Validation - Quality

Let denote π_{ij} the preference index $\pi(a_i, a_j)$.
Definition of the quality index Qi_{ij} :

$$QI_{ij} = \begin{cases} \pi_{ij} + \pi_{ji} & \text{if } \begin{cases} a_i \in C_h \\ a_j \in C_h \end{cases} \\ \\ 1 - \pi_{ij} + \pi_{ji} & \text{if } \begin{cases} a_i \in C_h \\ a_j \in C_l \\ h > l \end{cases} \\ |0.5 - \pi_{ij}| + |0.5 - \pi_{ji}| & \text{if } \begin{cases} a_i \in C_h \\ a_j \in C_{hx} \\ h \neq x \end{cases} \end{cases}$$

The lower is QI_{ij}, the better is the quality of the final clustering distribution.

Validation - Quality

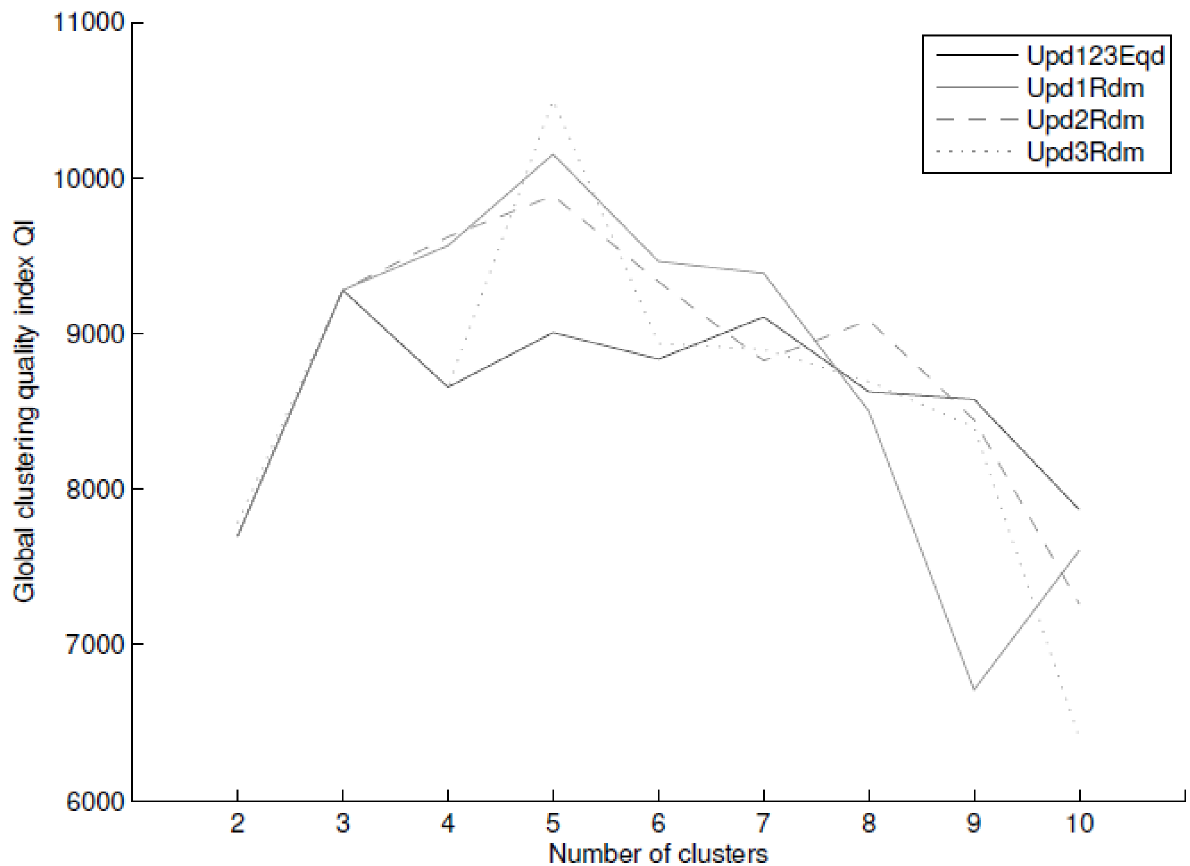


Figure: Evolution of the clustering quality with the number of clusters, 30 tests, EPI dataset

Validation - Quality

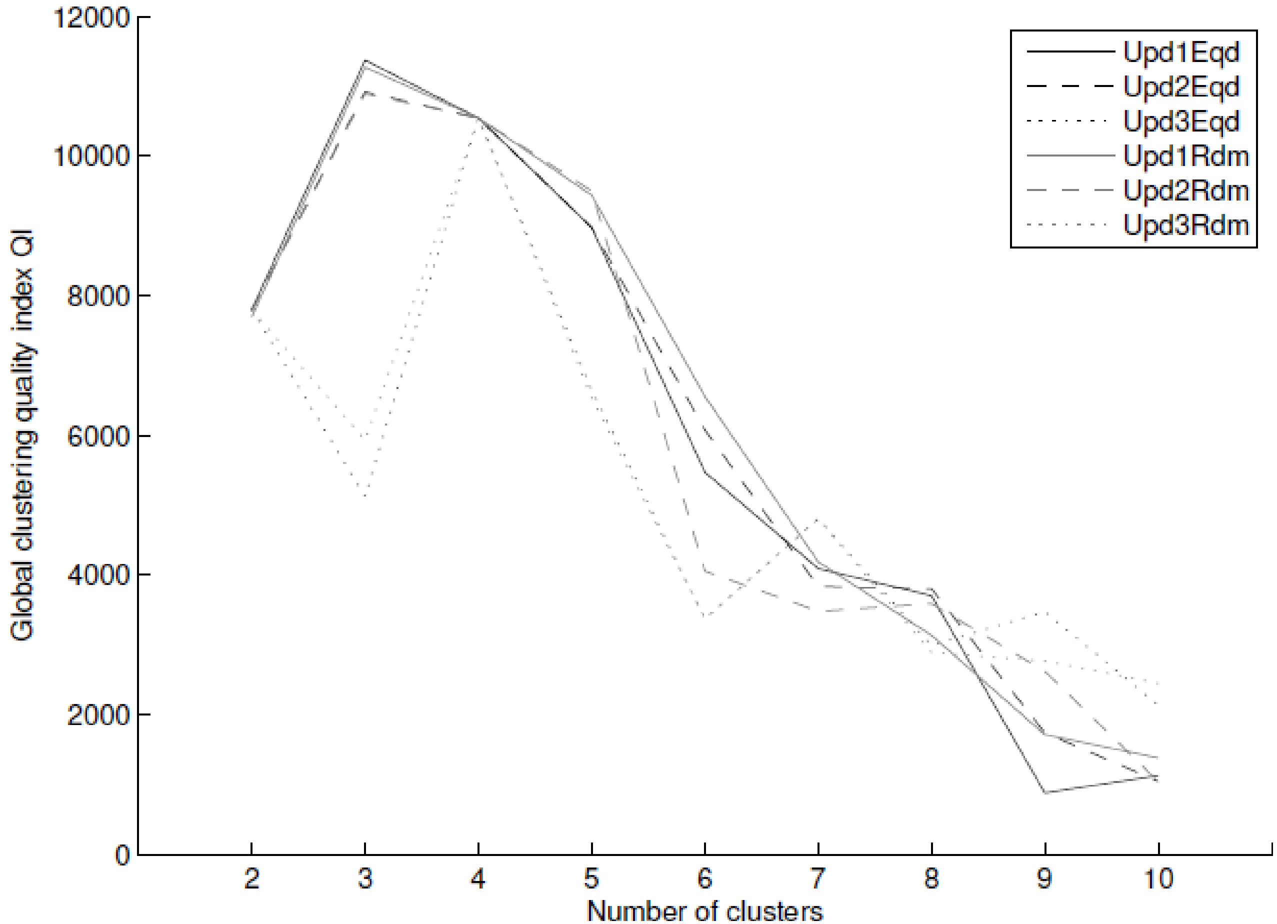


Figure: Evolution of the clustering quality with the number of clusters, 30 tests, CPU dataset

Validation - Quality

Contigency table of two clustering distributions with k = 2 and k = 3 categories (Upd1Rdm, EPI dataset)

	$\|C_1\|$	$\|C_{12}\|$	$\|C_2\|$	$\|C_{23}\|$	$\|C_3\|$	$\sum$
$\|C_1\|$	56	7	7	0	0	70
$\|C_{12}\|$	0	2	50	6	0	58
$\|C_2\|$	0	0	0	13	37	50
$\sum$	56	9	57	19	37	**178**

Limited spread of the alternatives when adding a cluster
Assignment to **principal clusters** in priority
Same results were observed with Upd2 and Upd3

Validation - Convergence

Calculation of the average number of iterations to converge (i_{tot})
Datasets EPI (k=4) and CPU (k=4), 100 runs

	EPI		CPU	
	i_{tot}	std	i_{tot}	std
$Upd1Eqd$	17	0	16.81	4.65
$Upd2Eqd$	17	0	19.91	6.39
$Upd3Eqd$	17	0	25.31	0.49
$Upd1Rdm$	11.09	5.74	14.58	4.79
$Upd2Rdm$	9.50	4.81	15.19	6.03
$Upd3Rdm$	10.11	4.62	17.62	9.94

Influence of the update functions is **not significant**.
Random initialization of the profiles has a stronger influence on the convergence.

Validation - Stability

Calculation of the stability of the clustering S after 100 runs (%), EPI (k=4) and CPU (k=4)
Proportion of distribution $\delta_i(A, \kappa)$ after 100 runs (%), EPI (k=4), Upd1Rdm

	$\mathcal{S}$ (EPI)	$\mathcal{S}$ (CPU)
$Upd1Eqd$	100	93
$Upd2Eqd$	100	99
$Upd3Eqd$	100	100
$Upd1Rdm$	39	88
$Upd2Rdm$	37	92
$Upd3Rdm$	41	95

$\delta_1(A, \kappa)$	3%
$\delta_2(A, \kappa)$	39%
$\delta_3(A, \kappa)$	30%
$\delta_4(A, \kappa)$	2%
$\delta_5(A, \kappa)$	3%
$\delta_6(A, \kappa)$	3%
$\delta_7(A, \kappa)$	15%
$\delta_8(A, \kappa)$	1%
$\delta_9(A, \kappa)$	4%

Stability **globally good**, except for EPI dataset in random initialization.
But the distributions $\delta_2(A, \kappa)$ and $\delta_3(A, \kappa)$ are **very similar** in that case (error < 2%).

Comparison with existing procedures - Quality

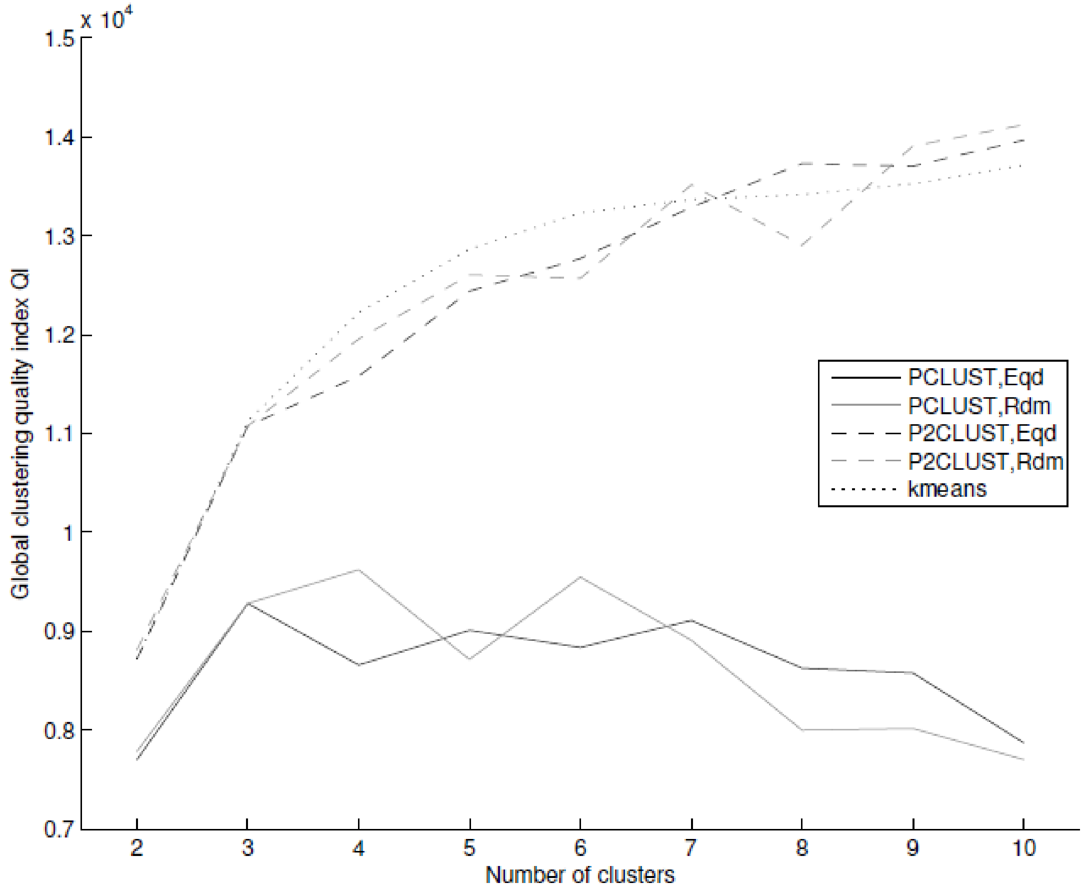


Figure: Evolution of the clustering quality with the number of clusters, 30 tests, EPI dataset. Comparison of the models PCLUST, P2CLUST and *k*-means.

Comparison with existing procedures - Convergence

Calculation of average number of iterations to converge (i_{tot}), standard deviation (*std*) and total calculation time t_{100} (in seconds). Datasets EPI (k=4) and CPU (k=4), 100 runs

	EPI			CPU		
	i_{tot}	std	$t_{100}(s)$	i_{tot}	std	$t_{100}(s)$
PCLUST						
$Upd1Eqd$	17	0	66.88	16.81	4.65	185.91
$Upd2Eqd$	17	0	66.47	19.91	6.39	215.05
$Upd3Eqd$	17	0	70.89	25.31	0.49	276.17
$Upd1Rdm$	11.09	5.74	56.40	14.58	4.79	158.60
$Upd2Rdm$	9.50	4.81	53.73	15.19	6.03	168.11
$Upd3Rdm$	10.11	4.62	57.35	17.62	9.94	226.07
P2CLUST						
Eqd	8	0	44.39	13.52	0.91	178.38
Rdm	5.95	2.49	39.36	9.01	2.80	145.92

P2CLUST converges **slightly faster** when comparing the iterations.
Gain remains moderate even when comparing the calculation times.

Comparison with existing procedures - Stability

Calculation of the stability of the clustering S (%) and the stability allowing 2% of error $S_{2\%}$ (%), EPI (k=4) and CPU (k=4) datasets, 100 runs

	$\mathcal{S}$		$\mathcal{S}_{2\%}$	
	(EPI)	(CPU)	(EPI)	(CPU)
PCLUST				
$Upd1Eqd$	100	93	100	94
$Upd2Eqd$	100	99	100	99
$Upd3Eqd$	100	100	100	100
$Upd1Rdm$	39	88	69	88
$Upd2Rdm$	37	92	70	96
$Upd3Rdm$	41	95	71	96
P2CLUST				
Eqd	100	100	100	100
Rdm	19	61	23	61

Results are **significantly better with the PCLUST model** in random initialization.
Results are similar with the equidistributed initialization strategy.

Conclusions

First extension of PROMETHEE to interval clustering.

Validation of the model on real-world datasets underlines **interesting results**.
Stability and quality of the clustering are particularly good.
Interval clustering allows to generate **higher quality** clustering distributions.
Acceptable convergence of the model.

Limited interest of using preferential information from interval clusters.
Equidistributed initialization leads to more stable clustering.
Random initialization allows the model to converge faster.