Dissimilarity Clustering by Hierarchical Multi-Level Refinement

Brieuc Conan-Guez, Fabrice Rossi
brieuc.conan-guez@univ-lorraine.fr, Fabrice.Rossi@univ-paris1.fr

Dissimilarity Clustering

We aim at clustering the object set Ω (with $|\Omega| = N$) into c clusters described by the partition $\mathcal{P} = \{C_1, \ldots, C_c\}$ using only a reflexive and symmetric dissimilarity measure d on Ω.
To achieve this goal, we try to minimize the following generalization of the standard K-means quantization error to this setting, that is [1]:

$$E(\mathcal{P}) = \sum_{C_k \in \mathcal{P}} \frac{1}{|C_k|} \sum_{i,j \in C_k} d(i,j)$$

This can be done greedily using the standard hierarchical clustering analysis (HCA) technique, based on the following linkage criterion:

$$\Delta E(C_p, C_q) = \frac{S_{C_p \cup C_q}}{|C_p| + |C_q|} - \frac{S_{C_p}}{|C_p|} - \frac{S_{C_q}}{|C_q|},$$

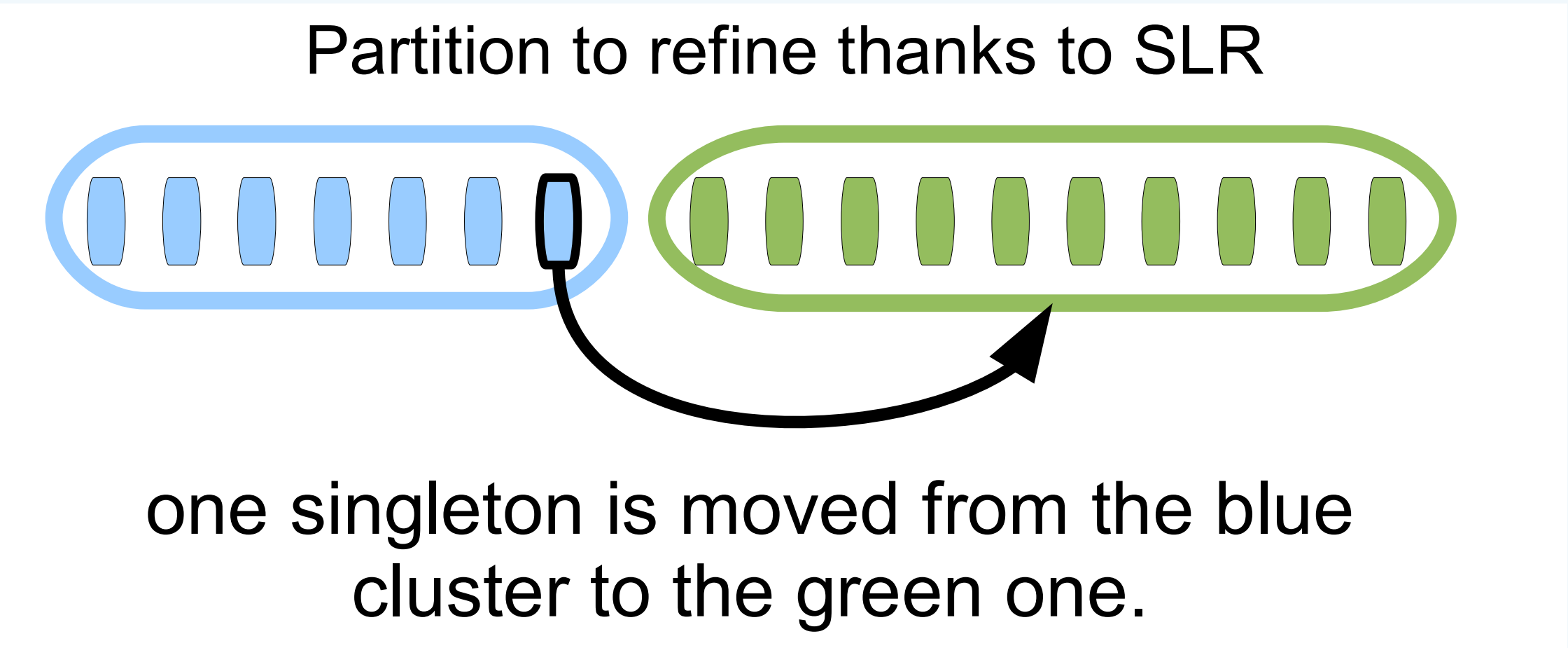
where $S_{C_p} = \sum_{i,j \in C_p} d(i,j)$

Fast Hierarchical Clustering

HCA is implemented using the efficient algorithm proposed by D. Müllner for agglomerative hierarchical clustering [2]. Even though this recent algorithm has a worst case complexity of $O(N^3)$, it is much more efficient than actual commonly used algorithms in practical situations where its observed scaling is in $O(N^2)$. Moreover this algorithm can handle general linkage criteria, such as the one defined above ΔE.

Partition Refinement

A drawback of greedy methods is that bad merging decisions done during the early steps of the HCA cannot be corrected in the later steps, which leads to a suboptimal partition. To limit this problem, a classical approach consists in refining the partition (the chosen level of the hierarchy) thanks to a greedy heuristic. The refinement performs a local search by moving **one object at a time** from one cluster to another.



Partition to refine thanks to SLR

one singleton is moved from the blue cluster to the green one.

In the context of hierarchical clustering, such heuristics are known as Single-Level Refinement (SLR) approaches, as they operate on just one level of the hierarchy: the bottom level.

References

[1] Thomas Hofmann, Joachim M. Buhmann *Pairwise Data Clustering by Deterministic Annealing*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19 (1997)

[2] Daniel Müllner *Modern hierarchical, agglomerative clustering algorithms*, Lecture Notes in Computer Science, Springer (2011)

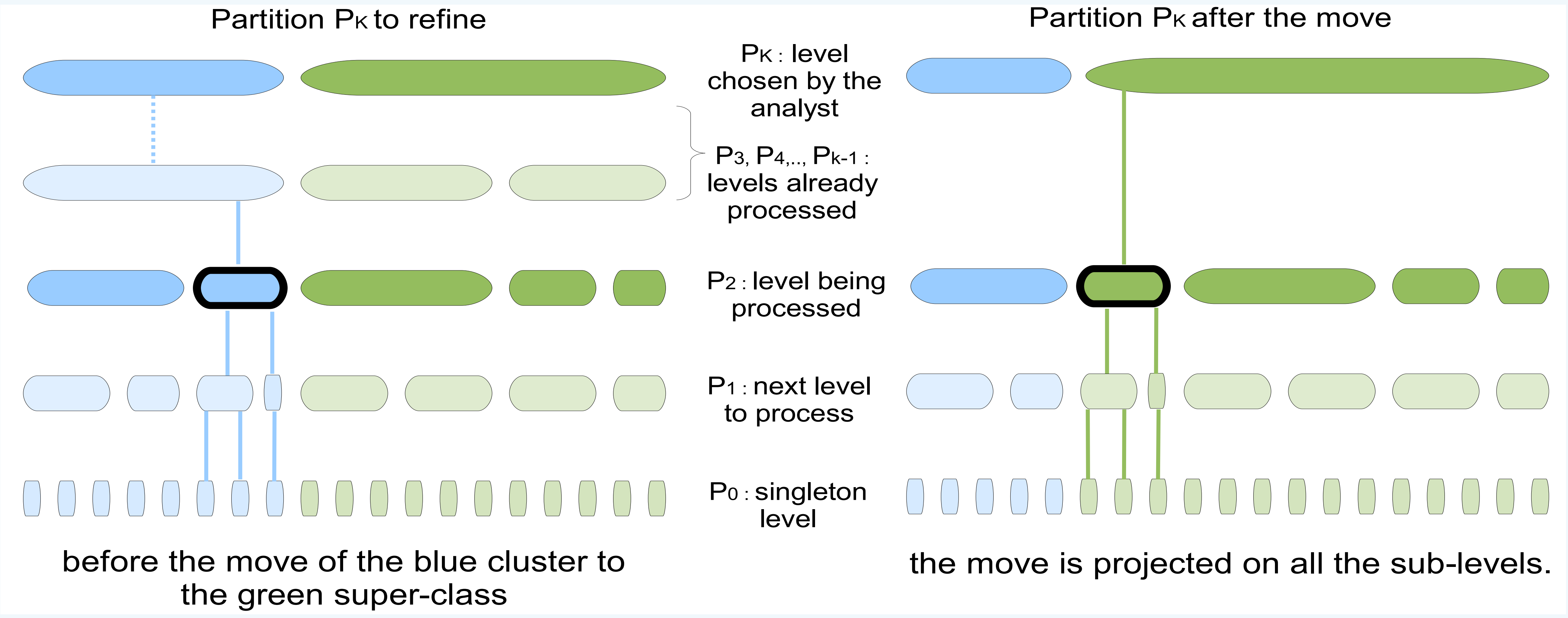
[3] George Kapyris, Eui-hong (Sam) Han, Vipin Kumar: *Multilevel refinement for hierarchical clustering*, tech. report University of Minnesota (1999)

[4] Andreas Noack, Randolf Rotta: *Multi-level Algorithms for Modularity Clustering*, Proceedings of the 8th International Symposium on Experimental Algorithms, Dortmund, Germany (2009)

Multi-Level Refinement

Contrarily to SLR approaches, the Multi-Level Refinement (MLR) approach [3, 4] operates on different levels of the hierarchy. Let $\mathcal{P}_K$ be the level chosen by the analyst (the partition to refine). Given a reduction factor $\alpha < 1$, the MLR considers only a small number of levels $(\mathcal{P}_k)_{0 \leq k \leq K}$ of the hierarchy. The selected levels are those whose sizes are given by a geometric progression $(N, \alpha N, \alpha^2 N, \ldots, \alpha^{K-1} N, |\mathcal{P}_K|)$. This series begins by the partition of singletons $\mathcal{P}_0$ (size N), and ends by the partition to refine $\mathcal{P}_K$ (size $|\mathcal{P}_K|$).

The MLR proceeds in a top-down way. It first applies the greedy refinement heuristic to $\mathcal{P}_K$ considered as a partition of clusters of partition $\mathcal{P}_{K-1}$: this first refinement corresponds to moving entire sub-clusters of objects from one cluster of $\mathcal{P}_K$ to another one. Once this is done, the modifications of $\mathcal{P}_K$ are projected onto $\mathcal{P}_{K-2}$. As a second step, it refines $\mathcal{P}_K$ considered as a partition of clusters of $\mathcal{P}_{K-2}$ and projects once again the modifications. The process repeats identically for all the levels below (i.e. for k varying from $K-3$ to 0). The final step ($k = 0$) corresponds to applying the greedy refinement on the objects as in the SLR.



Partition PK to refine

Partition PK after the move

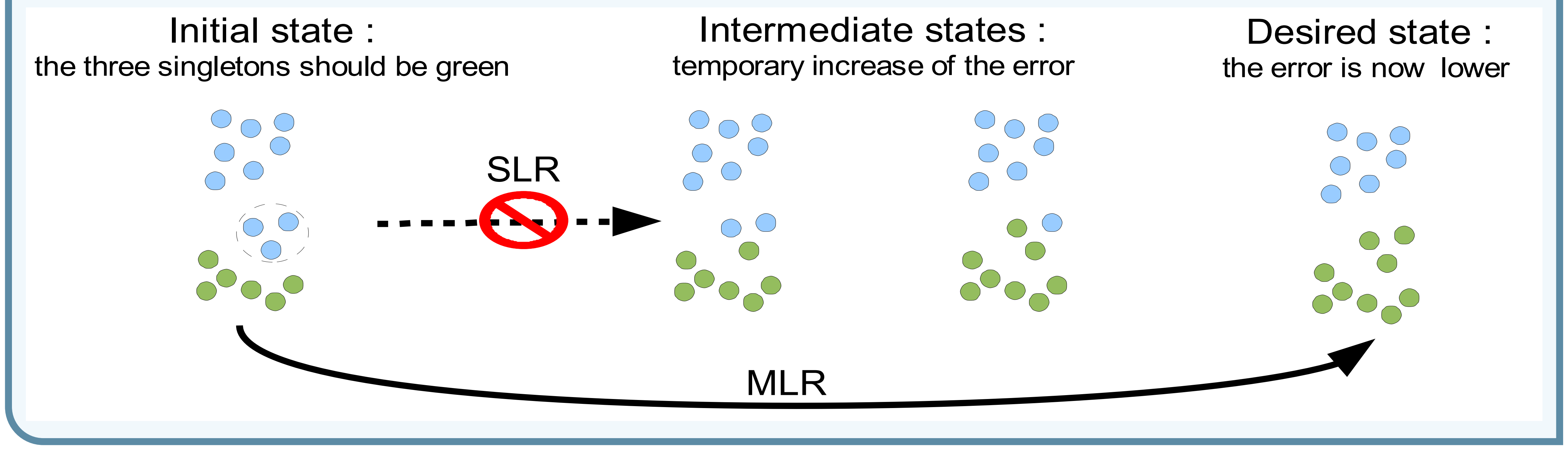
PK : level chosen by the analyst
P3, P4,.., Pk-1 : levels already processed
P2 : level being processed
P1 : next level to process
P0 : singleton level

before the move of the blue cluster to the green super-class

the move is projected on all the sub-levels.

Escaping Local Minima

Moving an entire cluster in one shot allows the MLR to escape more easily from local minima than single-level heuristics. Indeed, for single-level heuristics, the re-assignment of a dense group of objects from one cluster to another is unlikely to happen, as it would imply many individual moves with a temporary large increase of the error measure. For the MLR, on the opposite, the displacement of a group of objects is done in just one operation, which avoids a temporary increase of the error measure.



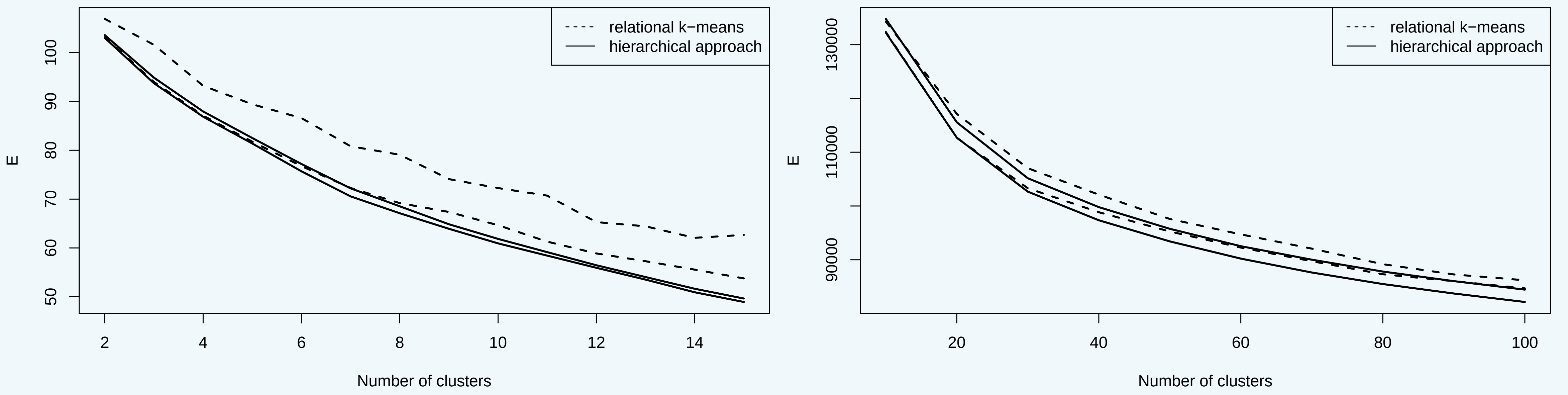
Initial state :
the three singletons should be green

Intermediate states :
temporary increase of the error

Desired state :
the error is now lower

Experiments

The proposed method is tested on two classical dissimilarity data sets: the small size cat cortex database with 65 objects and the large size Copenhagen chromosome database with 4200 objects. Reference performances are provided by the relational k-means (RKM). In all cases, the RKM is started from a number of initial random configurations chosen so that both methods use approximately the same computational ressources.



Quantization error E as a function of the number of clusters for two databases (cat cortex on the left and chromosome on the right) - top solid line: standard HCA, bottom solid line: MLR, dashed lines: best and worst results of the RKM