Combining Information from Distributed Evolutionary *k*-means

Murilo Coelho Naldi
Department of Exact and Technological Sciences
Federal University of Viçosa - UFV
Rio Paranaíba, Brazil
murilocn@ufv.br

Ricardo José Gabrielli Barreto Campello
Institute of Mathematics and Computer Sciences
University of São Paulo - USP
São Carlos, Brazil
campello@icmc.usp.br

***Abstract*—One of the challenges for clustering resides in dealing with huge amounts of data, which causes the need for distribution of large data sets in separate repositories. However, most clustering techniques require the data to be centralized. One of them, the *k*-means, has been elected one of the most influential data mining algorithms. Although exact distributed versions of the k-means algorithm have been proposed, the algorithm is still sensitive to the selection of the initial cluster prototypes and requires that the number of clusters be specified in advance. This work tackles the problem of generating an approximated model for distributed clustering, based on *k*-means, for scenarios where the number of clusters of the distributed data is unknown. We propose a collection of algorithms that generate and select *k*-means clustering for each distributed subset of the data and combine them afterwards. The variants of the algorithm are compared from two perspectives: the theoretical one, through asymptotic complexity analyses; and the experimental one, through a comparative evaluation of results obtained from a collection of experiments and statistical tests.**

***Keywords*-clustering; *k*-means; distributed data sets;**

I. Introduction

Data clustering is a fundamental conceptual problem in data mining, in which one aims at determining a finite set of categories to describe a data set according to similarities among its objects. The solution to this problem often constitutes the final goal of the mining procedure — having broad applicability in areas that range from image and market segmentation to document categorization and bioinformatics (e.g. see [1], [2]).

Many clustering algorithms have been proposed in the literature [1], [3]. Among them, the *k*-means methods has been investigated for more than half a century [4]. Recently, *k*-means has been elected one of the top ten most influential data mining algorithms for being simple, scalable and easy to adapt for different application domains [3]. However, *k*-means is sensitive to the selection of the initial cluster prototypes, as it may converge to suboptimal solutions if the initial prototypes are not properly chosen [1]. In addition, it requires that the number of clusters, k, be specified in advance. This can be quite restrictive in practice, since the number of clusters in a data set is generally unknown, especially in real-world applications involving high dimensional and/or distributed data.

In order to circumvent *k*-means limitations, approximation algorithms include the hybridization of *k*-means with some sort of general purpose meta-heuristic adapted to the clustering problem. Evolutionary algorithms are meta-heuristics widely believed to be able to provide satisfactory suboptimal solutions to NP-hard problems at acceptable time. From a combinatorial optimization perspective, clustering problems can be formally classified as NP-hard [5]. Probably for this reason, several evolutionary approaches for clustering problems have been proposed in the literature [5], [6].

A challenge for clustering resides in the substantial growth of data generated in many fields over the years. This growth causes a need for the distribution of large data sets in separate repositories, also called *data sites*. In many scenarios, the data are naturally distributed, i.e., have been generated and stored in different data sites. Large distributed data sets demand computational techniques that are able to extract relevant information with good computational performance and scalability. However, most clustering techniques require the data to be centralized, which may not be feasible in many cases due to computational limitations. Moreover, there is another complicating factor, which is the need to preserve the privacy of data among data sites, a legal obligation in some countries in Europe, the United States and other countries [7].

The present work tackles the problem of generating a model for distributed clustering, based on *k*-means, for scenarios where the number of clusters of the distributed data is unknown. For such, we propose the generation and selection of *k*-means clustering locally in each data site and, after that, the combination of the obtained clusters into a single global solution that represents the whole data set. The variants of the algorithms are compared from two perspectives: the theoretical one, through asymptotic complexity analyses; and the experimental one, through a comparative evaluation of results obtained from a collection of experiments and statistical tests.

The remainder of this article is organized as follows: in Section II, a brief review of the related work is made. Then, a collection of *k*-means based algorithms to cluster distributed data are proposed in Section III. This collection of algorithms is experimentally compared in Section IV. The

1522-4899/12 $26.00 © 2012 IEEE
DOI 10.1109/SBRN.2012.11

main conclusions are then summarized in Section V.

II. Related Work

The adaptation of a (centralized) clustering technique for distributed data sets can be divided into two different approaches. The first approach consists of generating an exact distributed model of the clustering algorithm, i.e., distributing the algorithm in such way that its result will be identical to the centralized (original) version of the algorithm. Alternatively, it is possible to obtain a clustering result that approximates those obtained by the centralized version of an algorithm. This second approach is known as approximated models [8] and they are developed to have better computational performance or reduced data transmission in comparison to the correspondent exact model.

An exact distributed version of the *k*-means algorithm was proposed in [9], which consists of multiple transmissions and updates of centroids (clusters' means) among data sites. Although this exact version guarantees the same results of the centralized *k*-means, the multiple transmissions can make the algorithm excessively slow, especially if *k*-means is executed multiple times in order to estimate the initial cluster prototypes or the number of clusters in the data.

Approximated algorithms can significantly reduce the number of transmissions and the amount of data transmitted, as the algorithm is based on data representatives or approximations instead of the data itself [8]. The main idea consists of applying parts of the algorithm on different subsets of the data set and, after that, combing the results into a single solution. The method used to combine the results depends on the type of data distribution. The most typical scenario consists of different objects of the data set distributed among data sites. In this scenario the objects share the same feature space. Markov chains and the Monte Carlo method are used in [10] to combine clustering information and obtain a final clustering model of the data set. Another approach unifies locally clustering solutions (obtained in the data sites) into a global solution based on the distance among their representatives (e.g. centroids) [11]. A similar idea is employed in [12] for collaboration problems.

III. Combinations of Distributed Clustering (CDC)

The Combinations of Distributed Clustering (CDC) is a collection of algorithms aimed to combine distributed information about the data set into a final clustering solution. They are composed by *data nodes* and a *master node*. A data node is a fraction of the CDC algorithm responsible for all processing which involves direct access to a subset of the data. Each data node is responsible for one subset of the data and no subset can be processed by more than one data node. In general, each data site has, at least, one data node. Data nodes are capable of transmitting information to other data nodes or to the master node. Differently from the data nodes, the master node does not have direct access to the data set (or part of it). Instead, it receives information from the data nodes and combines this information into a global clustering solution.

Generally speaking, the CDC algorithms have two main steps: the first is the generation of clustering models and the second combines the models obtained in the first step. These steps will be presented in Sections III-A and III-B.

A. Generation of Clustering Models

The first step of the CDC algorithms consists of clustering locally the subsets of the distributed data set, i.e., to generate data clusters for each of the data nodes separately. In the present work, the *k*-means algorithm will be used in this step in order to generate hard partitional clusters (here called partitions for the sake of simplicity) for the subsets in each data site. A partition of a data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, composed of a-dimensional feature or attribute vectors $\mathbf{x}_j$, is a collection $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_k\}$ of k non-overlapping data subsets $\mathbf{C}_i$ (clusters) such that $\mathbf{C}_1 \cup \mathbf{C}_2 \cup ... \cup \mathbf{C}_k = \mathbf{X}$, $\mathbf{C}_i \neq \emptyset$ and $\mathbf{C}_i \cap \mathbf{C}_l = \emptyset$ for $i \neq l$. In order to circumvent the *k*-means limitations presented in Section I, we used the F-EAC [13] algorithm to estimate the number of clusters and the correspondent centroids of *k*-means with linear computational cost.

The F-EAC was developed to evolve partitions generated by the use of k-means and evolutionary operators. These partitions are represented by individuals, that in turn are codified by genotypes. The set of genotypes is called population. Algorithm 1 presents the main F-EAC steps, in which g is the current generation, P_g is the current population, $|P|$ is the population size and $\mathsf{S_C}$ is the stopping criterion.

The Simplified Silhouette validation criteria [14] is used here as the fitness function (Step 6). An elitist strategy keeps the best solution (Step 9) [15]. The other solutions are chosen by a selection operator (e.g., proportional selection such as the roulette selection [15] $-$ Step 10).

One parameter of F-EAC is the population size $|P|$. Empirical evidence suggests that this type of algorithm is robust for distinct value choices for this parameter [13], [16], [17] and values such as $|P| = 10$ enable the algorithm to obtain good partitions in reasonable computational time. This value will be adopted in the present work.

The F-EAC uses two cluster oriented mutation operators. The first eliminates one or more clusters, adding its objects into the clusters with the closest centroids. The second splits one or more clusters into two new clusters each. The proportion of application between the two operators is adjusted dynamically based on the performance obtained by each one in the previous generation. This work considers the performance of the operators individually for each genotype, which is a simple way of doing this. If the usage of one operator generated a child with fitness higher than its father, this operator will be used in the mutation of this child

Algorithm 1 F-EAC.

1: $g \leftarrow 1$;
2: initialize randomly a population P_g;
3: **repeat**
4: **for** $i = 1, ..., |P|$ **do**
5: apply the k-means algorithm to each genotype;
6: evaluate each genotype according to the fitness function;
7: **end for**
8: **if** $\mathsf{S_C}$ is not satisfied **then**
9: apply elitist strategy;
10: select genotypes from P_g;
11: **for all** selected genotypes **do**
12: select which clusters will be mutated by proportional selection;
13: apply the mutation operators in the selected clusters to create new genotypes;
14: **end for**
15: copy the new genotype to the next population P_{g+1} and increments g;
16: **end if**
17: **until** $\mathsf{S_C}$ is satisfied
18: **return** P_g;

afterwards. Otherwise, the other operator will be used. If the genotype belongs to the initial population or was selected by elitism, it has 50% of chance of being mutated by each operator.

Some possible stopping criteria $\mathsf{S_C}$ for F-EAC (applied in Step 17) are: the definition of a maximum number of generations, a threshold for population diversity, and others [5]. The stopping criteria adopted in the experiments presented in this work are discussed in Section IV. Once the F-EAC stops during the first step of the CDC algorithms, the centroids of the fittest (best evaluated) partition of the data subset and the number of objects in each cluster of this partition are transmitted to the master node.

For a more detailed description of the F-EAC algorithm, please refer to [13], [17].

B. Combinations of Clustering Models

After receiving the centroids and the number of objects in each cluster from every data node, the master node generates a combined partition of the data set. Because the centroids share the same feature space (as the objects of the data set), it is possible to cluster these centroids with clustering algorithms in order to obtain a meta-partition.

Be $MX = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_c\}$ the set of centroids from all data nodes, resulted from the the first step of CDC algorithms, where c is the total number of centroids. A meta-partition is a collection $M\pi = \{MC_1, MC_2, ..., MC_{mk}\}$ of meta-clusters MC_i for which $MC_1 \cup MC_2 \cup ... \cup MC_{mk} = MX$, $MC_i \neq \emptyset$ and $MC_i \cap MC_l = \emptyset$ for $i \neq l$. The meta-partition can be converted into a global partition of the original data set, replacing each centroid by the objects it represents.

During the creation of the meta-partition, it is possible to adjust the position of the meta-centroid — i.e. the centroid of the meta-cluster — to become the mean of the objects represented in this meta-cluster. For such, the meta-centroid is calculated as the average of the centroids weighted by the number of objects they represent. Thus, the j-th meta-centroid is calculated as presented in Equation (1):

$$\mathbf{mc}_j = \frac{\sum\limits_{\mathbf{c}_i \in MC_j} \mathbf{c}_i |C_i|}{\sum\limits_{\mathbf{c}_i \in MC_j} |C_i|} \tag{1}$$

where $|C_i|$ is the cardinality of the cluster for which $\mathbf{c}_i$ is the centroid.

In this work, three algorithms were chosen to combine the set of meta-centroids MX: the algorithm F-EAC (Algorithm 1) and the hierarchical algorithms single-link and complete-link [1]. As the hierarchical clustering is a nested sequence of hard partitional clusterings, each level of the hierarchy is evaluated with the Simplified Silhouette index (the same used as the fitness function for F-EAC) and the best partition is chosen as the final solution. In CDC algorithms, the calculation of the Simplified Silhouette index is exact, i.e., it obtains the same result as the centralized version of the index. This is done by transmitting the meta-centroids from the master node to the data nodes and retrieving the Simplified Silhouette value of each object from the data nodes.

C. Complexity of the CDC Algorithms

Assuming that the i-th data node has $n[i]$ objects, the first step of the CDC algorithms has a computational complexity equivalent to the F-EAC [17], i.e., $O(g_t \cdot |P| \cdot t \cdot \hat{k}_{max} \cdot (n[1] + n[2] + ... + n[dn])$ or $O(g_t \cdot |P| \cdot t \cdot \hat{k}_{max} \cdot n)$, where g_t is the F-EAC number of generations, $|P|$ is the population size, t is the number of k-means iterations, $\hat{k}_{max}$ is the maximum number of clusters codified by a genotype during the evolutionary search, dn is the number of data nodes, and n is the number of objects in the distributed data set. If executed in parallel, this complexity is reduced to $O(g_t \cdot |P| \cdot t \cdot \hat{k}_{max} \cdot n_{max})$, where n_{max} is the maximum number of objects in a data node. Moreover, if $n_{max} \approx n/dn$, then the complexity of the CDC first step is $O(g_t \cdot |P| \cdot t \cdot \hat{k}_{max} \cdot n/dn)$. The second step of the CDC algorithms has the computational complexity of the clustering algorithm adopted in this step. However, the number of clustered objects is the total number of centroids resulted from the first step (c). Hierarchical algorithms have computational complexity $O(c^2 \cdot \log c)$ [1]. If the F-EAC was adopted, the computational cost of the second step is $O(g_t \cdot |P| \cdot t \cdot \hat{k}_{max} \cdot c)$ [17].

The CDC algorithms memory allocation cost in the first step is $O(\hat{k}_{max} \cdot |P| \cdot n_{max})$. In the second step, this cost is $O(\hat{k}_{max} \cdot |P| \cdot c)$ if the F-EAC algorithm is adopted. Otherwise, if the hierarchical algorithms are used, the dissimilarities between centroids cause the memory allocation cost of $O(c^2)$.

The amount of data transferred among the data nodes and the master node by the CDC algorithms is also related to the clustering algorithm used in their second step. To apply the Simplified Silhouette index, the meta-centroids must be transmitted to the data nodes, thus increasing the amount of data transmitted by the algorithms. If the hierarchical algorithms are adopted, the total number of meta-centroids is $2 + 3 + \cdots + k_{max} \approx k_{max}(k_{max} + 1)/2$, which requires transmissions with sizes of order $O(k_{max}^2 \cdot a \cdot dn)$. If the F-EAC is adopted, then the data transmission is estimated as $O(g_t \cdot |P| \cdot \hat{k}_{max} \cdot a \cdot dn)$ in the worst case. It is important to note that the transmission cost of the algorithm does not depend on the number of objects in the data set, which makes the algorithm scalable in relation to this aspect.

IV. Experiments

Experiments with a collection of data sets were carried out to evaluate the CDC algorithms. This evaluation compares different CDC variants, concerning three aspects: the quality of the resulted partitions, the execution time, and the total amount of data transmitted between data nodes. The quality of the resulted partitions is measured with the Jaccard external index [18] in relation to the known clusters or "golden truth". The execution time and amount of data transmitted was measured using Matlab software in computers with quad core 3.0 Ghz processors and 12 GB of RAM memory. These calculations consider that the data nodes executed in parallel and that the data was transmitted concurrently. It is important to note that the experiments presented in this section aims at comparing each of those aspects independently, as the execution and transmission time are machine dependent.

A collection of 80 artificial data sets created by a ellipsoidal cluster generator in [19] was chosen for the experiments presented here. Each data set of the collection was distributed among data nodes in two ways: balanced and unbalanced. The balanced distribution consists of distributing the data set among data partitions maintaining the same proportion of objects each known cluster has in relation to the original data set. When this proportion is not maintained, the distribution is considered unbalanced. For a detailed description of the methods for data distribution adopted here, please refer to [20].

A. Parameters and Variants

In the experiments presented here, two stopping criteria $\mathsf{S_C}$ will be adopted for the F-EAC (Step 17 of Algorithm 1). The first consists of finding a partition with fitness value equal or higher than a reference value v_r. To calculate the v_r value for a data set or subset, the following procedure was executed:

1) Initialize $i \leftarrow 1$.
2) Execute *k*-means 100 times with random initial prototypes and the k value as the known number of clusters.
3) Each partition resulted from Step 2 is evaluated with the same validation criteria used by the F-EAC.
4) The best validation value is stored in $\mathbf{v}[i]$.
5) If $i < 31$ then $i \leftarrow i + 1$ and return to Step 2.
6) Calculate the reference value as $v_r = \sum_{i=1}^{30} \frac{\mathbf{v}[i]}{30}$.

Additionally, a maximum number of generations $g_{max} = 100$ is adopted as a second stopping criterion. The F-EAC initial population is composed of partitions with number of clusters k randomly chosen in the interval $[2, n^{1/2}]$, a commonly used rule of thumb for *k*-means based evolutionary algorithms [21], [16], [17]. Comparing the results for the different population sizes adopted in [17], it is possible to observe that larger populations tend to make F-EAC converge to good solutions in fewer generations, which, by their turn, are more computationally costly than those related to smaller populations. These results suggest that the algorithm is reasonably robust to the choice of the population size $|P|$, especially if one considers that its effectiveness in solving the clustering problem is barely affected by this choice. Experiments in [17] show that $|P| = 10$ obtains good results and will be adopted here. During the F-EAC, the *k*-means convergence is attained when no significant difference is observed between the values of the centroids in two consecutive iterations, for which a threshold of 10^{-3} is adopted in this work, and a maximum number of iterations t is also imposed to the algorithm. Empirical evidence suggests that $t = 5$ or less repetitions will suffice [17]. So, this value is also adopted here.

When hierarchical algorithms are applied on the combination of clusters, the CDC algorithms result on a hierarchy composed of nested partitions with number of clusters from 2 to $n^{1/2}$ and there is no other stopping criterion. Here, every level of the hierarchy is evaluated with the Simplified Silhouette index and the correspondent partition with the best result is chosen.

In the following experiments, four variants of the CDC algorithms were compared for 2 types of data set distributions:

1) CDC-*sl*: hierarchical single-link applied on a set of F-EAC clusters, balanced data set distribution.
2) CDC-*sl* (U): the same as the previous variant, for unbalanced data set distribution.
3) CDC-*al*: hierarchical average-link applied on a set of F-EAC clusters, balanced data set distribution.
4) CDC-*al* (U): the same as the previous variant, for unbalanced data set distribution.
5) CDC-*FEAC*: F-EAC applied on a set of F-EAC clusters, balanced data set distribution.

6) CDC-*FEAC* (U): the same as the previous variant, for unbalanced data set distribution.
7) CDC-*FEAC* (10g): F-EAC applied on a set of F-EAC clusters, balanced data set distribution. This variant has a third stopping criterion for the F-EAC at the combination of clusters. The algorithm stops after 10 consecutive generations without improvement on the fitness value of the best partition.
8) CAD-*FEAC* (10g)(U):the same as the previous variant, for unbalanced data set distribution.

B. Results

The CDC variants were executed 30 times for each data set and the mean and standard deviation error values of the Jaccard index, the execution time and amount of data transferred were compared. The (non-parametric) Friedman test [22] was applied o verify the statistical significance of the differences between the mean values, since the ANOVA test assumes that the compared samples are drawn from populations with normal distributions and similar variances [23], which could not be done here. When the null hypothesis of the test was rejected, which indicates that there is statistical evidence to support that the compared means are different, a post-hoc multiple comparison procedure [24] was applied using Matlab® to find which differences did exhibit statistical significance. To maintain the actual level of statistical confidence in 95%, a Bonferroni adjustment [25] was applied to the critical values from the $t-$distribution to compensate for multiple comparisons. The best results and results which are not statistically different from the best one are presented in bold in Tables I, II and III for distributions in 5, 20, and 80 data nodes, respectively.

CDC Variant	Jaccard	Time (*s*)	Transmission (*KB*)
1 CDC-*sl*	0.7816 (0.1503)	**3.14 (2.07)**	255.44 (183.42)
2 CDC-*sl* (U)	0.7499 (0.1843)	4.91 (7.69)	**212.05 (183.89)**
3 CDC-*al*	**0.7974 (0.1582)**	3.19 (2.08)	255.44 (183.42)
4 CDC-*al* (U)	0.7725 (0.1865)	4.94 (7.69)	**212.05 (183.89)**
5 CDC-*FEAC*	0.7634 (0.1425)	15.28 (15.60)	2055.53 (3235.86)
6 CDC-*FEAC* (U)	0.7469 (0.1684)	26.28 (22.06)	2882.62 (3628.17)
7 CDC-*FEAC* (10g)	0.7469 (0.1482)	8.05 (5.05)	865.79 (939.65)
8 CDC-*FEAC* (10g) (U)	0.7242 (0.1729)	11.03 (9.34)	948.26 (974.23)

Table I
MEAN AND STANDARD DEVIATION VALUES OBTAINED BY THE COMPARED ALGORITHMS WHEN APPLIED TO THE COLLECTION OF DATA SETS DISTRIBUTED AMONG 5 DATA NODES.

The results in Table I indicate that the variants of the CDC algorithms that adopt hierarchical clustering algorithms at the second main step presented the best results when the data sets were distributed among 5 data nodes. The variant with the best Jaccard mean values is the CDC-*al*, followed closely by the CDC-*sl*. When computational time is considered, the results between these variants are inverted. They also presented the lower amount of data transmitted.

Comparing Tables I and II, the quality of the CDC-*sl* and CDC-*al* partitions were reduced when the number of data nodes increased. This reduction may be a result from the overlap between clusters, evidenced by the increased (almost three times) number of clusters in the CDC first step (please refer to [20] for a more detailed explanation). The same cannot be said about the CDC-*FEAC* variants, which resulted in the best partitions, as presented in Table II. Although the CDC-*FEAC* (10g) variants resulted in Jaccard mean values which are not as good as those obtained with the original CDC-*FEAC*, they are close. Additionally, the stopping criterion of 10 generations without improvement allowed computational and transmission savings.

CDC Variant	Jaccard	Time (*s*)	Transmission (*KB*)
1 CDC-*sl*	0.5711 (0.2542)	**1.66 (0.92)**	1290.41 (840.20)
2 CDC-*sl* (U)	0.5878 (0.2525)	**2.07 (2.16)**	**1283.34 (836.64)**
3 CDC-*al*	0.6940 (0.2520)	2.26 (1.49)	1290.41 (840.20)
4 CDC-*al* (U)	0.6874 (0.2552)	2.18 (2.22)	**1283.34 (836.64)**
5 CDC-*FEAC*	0.7650 (0.1531)	9.76 (10.44)	7857.03 (11479.87)
6 CDC-*FEAC* (U)	**0.7776 (0.1672)**	14.89 (12.97)	15154.84 (19411.24)
7 CDC-*FEAC* (10g)	0.7541 (0.1584)	5.27 (3.55)	3686.41 (3802.90)
8 CDC-*FEAC* (10g) (U)	0.7596 (0.1719)	5.57 (3.81)	4351.37 (4639.09)

Table II
MEAN AND STANDARD DEVIATION VALUES OBTAINED BY THE COMPARED ALGORITHMS WHEN APPLIED TO THE COLLECTION OF DATA SETS DISTRIBUTED AMONG 20 DATA NODES.

CDC Variant	Jaccard	Time (*s*)	Transmission (*KB*)
1 CDC-*sl*	0.2664 (0.2410)	**0.97 (0.65)**	**5127.95 (3342.77)**
2 CDC-*sl* (U)	0.2789 (0.2478)	**0.91 (0.61)**	**5124.81 (3338.31)**
3 CDC-*al*	0.5327 (0.2694)	3.76 (3.58)	**5127.95 (3342.77)**
4 CDC-*al* (U)	0.5725 (0.2571)	2.50 (2.05)	**5124.81 (3338.31)**
5 CDC-*FEAC*	0.7052 (0.2241)	18.41 (15.49)	61525.95 (74502.38)
6 CDC-*FEAC* (U)	**0.7481 (0.1906)**	18.07 (14.31)	68201.59 (80820.47)
7 CDC-*FEAC* (10g)	0.6921 (0.2283)	5.53 (3.60)	16158.76 (16939.77)
8 CDC-*FEAC* (10g) (U)	0.7348 (0.1910)	5.30 (3.28)	17386.12 (18348.24)

Table III
MEAN AND STANDARD DEVIATION VALUES OBTAINED BY THE COMPARED ALGORITHMS WHEN APPLIED TO THE COLLECTION OF DATA SETS DISTRIBUTED AMONG 80 DATA NODES.

Once more, the increasing of the data nodes (20 to 80) reduced the mean quality of the CDC-*sl* and CDC-*al*, as indicated by the Jaccard mean values in Tables II and III. Nevertheless, the CDC-*FEAC* variants had lower quality variation, which indicates robustness in relation to the number of data nodes and centroids resulted from the first step of the algorithms. However, their computational time and the amount of data transmitted were higher than those resulted from the variants based on hierarchical algorithms. It is possible to reduce these costs with the use of the 10 generation limit, with the drawback of some quality loss.

In relation to the type of distribution, the unbalanced distribution only reduced the quality of the partitions obtained with 5 data nodes. For 20 and 80 data nodes, the quality of the partitions was better or very close to the quality resulted from the variants applied on balanced data. This indicates that the CDC algorithms seems to be able to obtain good results even if the data set known clusters are distributed

with unbalanced known clusters.

V. Conclusion

This work proposed the generation and selection of *k*-means clustering locally in each data site and the combination of the obtained clusters into a single global solution that represents the whole data set. The experiment results obtained from the 8 CDC variants compared showed that, when the hierarchical algorithms are adopted, this combination has lower computational time and data transmission. However, the adoption of the F-EAC at the second step of the algorithms showed to be robust to variations in the data distribution. The use of the limit of 10 generations without improvement on the quality of partitions reduced the cost of the algorithm, causing minor quality loss. This version should be recommended for scenarios where the performances of the CDC algorithms are unknown.

Acknowledgment

The authors acknowledge the Brazilian agencies CNPq, FAPEMIG and FAPESP for the financial support.

References

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999. [Online]. Available: citeseer.ist.psu.edu/jain99data.html

[2] R. Xu and I. Wunsch, D., "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.

[3] X. Wu, *Top ten algorithms in data mining.* Taylor & Francis, 2009.

[4] D. Steinley, "K-means clustering: A half-century synthesis," *British Journal of Mathematical and Statistical Psychology*, vol. 59, pp. 1–34(34), May 2006.

[5] E. Falkenauer, *Genetic Algorithms and Grouping Problems.* John Wiley & Sons, 1998.

[6] E. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. Ponce Leon F. de Carvalho, "A survey of evolutionary algorithms for clustering," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 2, pp. 133–155, March 2009.

[7] H. Hijmans, "Recent developments in data protection at european union level," *ERA-Forum, Online First*, vol. 12, pp. 1–13, 2010.

[8] K. Hammouda and M. Kamel, "Hierarchically distributed peer-to-peer document clustering and cluster summarization," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 5, pp. 681–698, May 2009.

[9] I. S. Dhillon and D. S. Modha, "A data-clustering algorithm on distributed memory multiprocessors," in *Revised Papers from Large-Scale Parallel Data Mining, Workshop on Large-Scale Parallel KDD Systems, SIGKDD*. London, UK: Springer-Verlag, 2000, pp. 245–260.

[10] S. Merugu and J. Ghosh, "A privacy-sensitive approach to distributed clustering," *Pattern Recognition Letters*, vol. 26, no. 4, pp. 399–410, 2005.

[11] Y. Dong, S. Cao, K. Chen, M. He, and X. Tai, "Pfhc: A clustering algorithm based on data partitioning for unevenly distributed datasets," *Fuzzy Sets and Systems*, vol. 160, no. 13, pp. 1886 – 1901, 2009, theme: Information Processing and Applications. [Online]. Available: http://www.sciencedirect.com/science/article/B6V05-4V17CJH-9/2/b11420094ccdb7cb14556f98923efb63

[12] W. Pedrycz and P. Rai, "Collaborative clustering with the use of fuzzy c-means and its quantification," *Fuzzy Sets Syst.*, vol. 159, no. 18, pp. 2399–2427, 2008.

[13] V. Alves, R. Campello, and E. Hruschka, "Towards a fast evolutionary algorithm for clustering," in *IEEE Congress on Evolutionary Computation, 2006, Vancouver, Canada*, 2006, pp. 1776–1783.

[14] E. R. Hruschka, R. J. G. B. Campello, and L. N. de Castro, "Evolving clusters in gene-expression data," *Information Sciences*, vol. 176, no. 13, pp. 1898–1927, 2006.

[15] M. Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, 1998.

[16] R. J. G. B. Campello, E. R. Hruschka, and V. S. Alves, "On the efficiency of evolutionary fuzzy clustering," *Journal of Heuristics*, vol. 15, no. 1, pp. 43–75, 2009.

[17] M. C. Naldi, R. J. G. B. Campello, E. R. Hruschka, and A. C. P. L. F. Carvalho, "Efficiency issues of evolutionary *k*-means." *Applied Soft Computing.*, vol. 11, no. 2, pp. 1938–1952, 2011.

[18] P. Jaccard, "Nouvelles recherches sur la distribution florale," *Bull. Soc. Vandoise des Sci. Nat.*, vol. 44, pp. 223–270, 1908.

[19] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Trans. on Evolutionary Computation*, vol. 34, pp. 56–76, 2007.

[20] M. Naldi, "Técnicas de combinação para o agrupamento centralizado e distribuído de dados," Ph.D. dissertation, Instituto de Ciências Matemáticas e Computação , ICMC-USP, 2011.

[21] M. Pakhira, S. Bandyopadhyay, and U. Maulik, "A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification." *Fuzzy Sets Systems*, vol. 155, no. 2, pp. 191–214, 2005.

[22] M. Hollander and D. A. Wolfe, *Nonparametric Statistical Methods*. Wiley-Interscience, 1999.

[23] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[24] Y. Hochberg and A. C. Tamhane., *Multiple Comparison Procedures*. John Wiley & Sons, 1987.

[25] O. J. Dunn, "Multiple comparisons among means," *Journal of the American Statistical Association*, vol. 56, no. 293, pp. 52–64, 1961. [Online]. Available: http://www.jstor.org/stable/2282330