Parallel Text Document Clustering Based on Genetic Algorithm

Madina Mansurova[1], Vladimir Barakhnin[2,3], Sanzhar Aubakirov[1], Yerzhan Khibatkhanuly[1], and Aigerim Mussina[1]

[1] Al-Farabi Kazakh National University, Almaty, Kazakhstan
[2] Institute of Computational Technologies SB RAS, Novosibirsk, Russia
[3] Novosibirsk State University, Novosibirsk, Russia
mansurova01@mail.ru,bar@ict.nsc.ru,{c0rp.aubakirov,x.erzhan, aygerimmusina}@gmail.com

Abstract. This work describes parallel implementation of the text document clustering algorithm. The algorithm is based on evaluation of the similarity between objects in a competitive situation, which leads to the notion of the function of rival similarity. Attributes of bibliographic description of scientific articles were chosen as the scales for determining similarity measure. To find the weighting coefficients which are used in the formula of similarity measure a genetic algorithm is developed. To speed up the performance of the algorithm, parallel computing technologies are used. Parallelization is executed in two stages: in the stage of the genetic algorithm, as well as directly in clustering. The parallel genetic algorithm is implemented with the help of MPJ Express library and the parallel clustering algorithm using the Java 8 Streams library. The results of computational experiments showing benefits of the parallel implementation of the algorithm are presented.

Keywords: clustering algorithm, genetic algorithm, parallel computing.

1 Introduction

The volume of the digital content increases every day. This impedes the process of selection of the most appropriate material, when searching for the necessary information. Clustering is one of the instruments that allows to perceive large volumes of information. Clustering is a process of dividing a set of text documents of the electronic database into classes when the elements united into one class (called a cluster) have a greater similarity than the elements referring to different classes. The process of text document clustering is resource intensive; the problem gets more complicated with the increase in the volume of the data being processed. To solve this problem, researches apply the different technologies of parallel computing.

The aim of this work is development of parallel FRiS-Tax algorithm for clustering of scientific articles. For clustering, the measure of rival similarity was taken as the proximity measure. To automate the search for the most appropriate

weighting coefficients in the formula of similarity measure, a genetic algorithm was developed.

The paper is organized as follows. The relevance of research is substantiated in Section 1. Section 2 describes the problem of text document clustering and the accepted proximity measure. Also, Section 2 presents FRiS-Tax clustering algorithm. Section 3 describes a genetic algorithm to find the most appropriate weighting coefficients in the formula of similarity measure. Section 4 presents parallel versions of genetic and clustering algorithms. Then, the results of computational experiments and the obtained data analysis are given. In conclusion, the results of the performed work are summarized.

2 Clustering algorithm with competitive similarity function

This work deals with the problem of clustering publications from bibliographic databases which allows automating the process of choosing publications for a concrete researcher or a group of working together researchers. In the clustering problem, each cluster is described with the help of one or several identifiers called centroids. These are centers of gravity or central objects of clusters. FRiS-Tax algorithm ([1, 2, 3]) is chosen as a clustering algorithm. Comparison of FRiS-Tax algorithm with the existing analogs is presented in [1] and the results of FRiS-Tax exceed the results of competitors. The experiments with FRiS-Tax algorithm showed its high efficiency when solving the clustering problem, and demonstrated the usefulness of rival similarity functions in different problems of data analysis. To measure similarity, we propose to take the attributes of the bibliographic descriptions of documents as scales.

Let D be a set of documents. Similarity measure m on the D set is defined as follows:

$$m : D \times D \to [0, 1], \tag{1}$$

and in the case of complete similarity, function m has the value 1, in case of complete difference - 0. Calculation of the similarity measure is performed by the formula of the type:

$$m(d_1, d_2) = \sum\nolimits_{i=1}^{n} a_i m_i(d_1, d_2), \tag{2}$$

where i is the index of the element (attribute) of the bibliographic description, a_i are weighting coefficients, $m_i(d_1, d_2)$ is the measure of similarity by the i-th element (in other words, by i-th scale), and n is the number of considered attributes.

The measure of rival similarity is introduced as follows. In the case of the given absolute value of similarity $m(x, y)$ between two objects, the rival similarity of object a with object b on competition with c is calculated by the following formula:

$$F_{b/c}(a) = \frac{m(a,b) - m(a,c)}{m(a,b) + m(a,c)}, \tag{3}$$

where F is called a function of rival similarity or FRiS-function. The values of F change within the range from $+1$ to -1. This is what we call the function of rival similarity or FRiS-function. Function F agrees well with the mechanism of perception of similarity and difference which are used by a person when he compares a certain object with two other objects.

Similarity between the object and cluster is assigned by the same principle. In order to evaluate the rival similarity of object z with the first cluster, the absolute similarity $m(z,1)$ of z with this cluster and similarity $m(z,2)$ with the cluster-competitor are taken into account. We use the value of similarity of object z with the nearest or typical representative of the given cluster as the value of similarity of object z with the cluster. In this case, the value of the rival similarity is calculated by the formula:

$$F_{1/2}(z) = \frac{m(z,1) - m(z,2)}{m(z,1) + m(z,2)}.$$

The clustering method can be described as follows. Let a set of objects of sampling A be given. The similarity of objects united into clusters is taken as a rival similarity with the central object of the cluster. Such objects were called pillars of clusters [1]. The peculiarity of the problem of dividing a set into clusters is that at the initial stage the reference of the objects of sampling to this or other cluster in unknown. All the objects of set A are likely to refer to one cluster. If we fix a set of centroids of this cluster $S = \{s_1, s_2, \ldots, s_k\}$, then for each object $a \in A$ it is possible to find the distance $m(a, s_{a1})$ (from the object to the nearest centroid from set S). But the absence of a cluster-competitor does not allow to determine the distance of the object to the nearest pillar of the cluster-competitor. In this regard, in the first stage, a virtual cluster-competitor is introduced the pillar of which is placed from each object of sampling at a fixed distance equal to m^*. Then, the value of rival similarity of object a with the nearest to it pillar s_{a1} from S in comparison with the virtual competitor is written as:

$$F^*_{s_{a1}}(a) = \frac{m(a, s_{a1}) - m^*}{m(a, s_{a1}) + m^*}.$$

The number of pillars in the clustering problem will be chosen in such a way so that the value of competitive similarity of each object of sampling A with the nearest to it pillar from S is maximum:

$$\overline{F}(S) = \sum\nolimits_{a \in A} F^*_{s_{a1}}(a) \to \max_S. \tag{4}$$

The proposed algorithm chooses the number of clusters automatically. The user only assigns the limit number of clusters K, among which he would like to have the best variant of clustering. The algorithm subsequently seeks for solution

of the problem of clustering for all values $k = 1, 2, \ldots, K$, so that to choose the best of them (Fig. 1).

The advantages of using FRiS-Tax algorithm are shown in [4]. Firstly, the use of FRiS-compactness as a criterion of information capability of features at random distributions of images showed a significant advantage in comparison with a widely used criterion of minimum of errors when recognizing the test sampling by *Cross Validation* or *One Leave Out* methods. Secondly, at normal distributions, FRiS-algorithm first chooses the pillars located in the area of mathematical expectation, and, if distributions are polymodal and images are linearly inseparable, the pillars will be in the centres of modes. In the process of recognition, the decision is made in the favor of that image the pillar of which is similar to the control object most of all and the value of the function of similarity of the object with the chosen image allows to judge about the reliability of the taken decision. And finally, the use of FRiS-function for solution (at international contest Data Mining Cup 2009) of the problem of predicting the values of variables measured in absolute scale allowed it creator to hold the 4th place among 321 teams. Thus, the efficiency of using FRiS-function in algorithms for solution of problems of predicting quantitative variables is demonstrated.

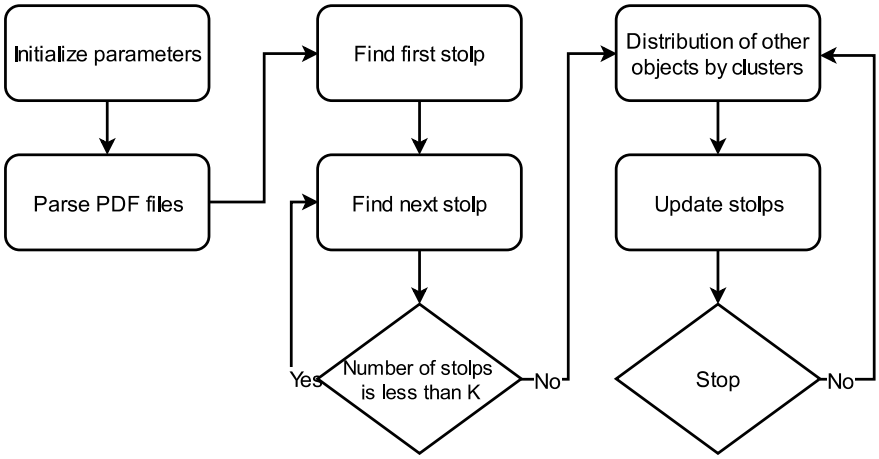

Fig. 1. Algorithm of clustering FRiS-Tax.

3 A genetic algorithm for adjustment of coefficients in the formula of similarity measure

In this work, we have chosen:

– the year of issue;
– code UDC;

– key words;
– authors;
– series;
– annotation;
– title

as attributes of division of articles from bibliographic databases into clusters. To choose weighting coefficients which are used in the formula of similarity measure (2), a genetic algorithm was developed. The genetic algorithm refers to heuristic algorithms of search which is used for solving the problems of optimization and modeling by random selection, combination and variation of the sought-for parameters using the mechanisms similar to natural selection in nature [5]. It should be noted that earlier the weighting coefficients were adjusted manually by an experimental way (see [3]) and the change of the problem domain of documents required a new series of experiments. The use of genetic algorithm allows automating the search for the most acceptable weighting coefficients in the formula of similarity measure.

The genetic algorithm is executed in the following stages ([5]):

1) Creation of initial population.
2) Selection.
3) Choice of parents.
4) Crossover.
5) Mutations.

The description of realization of genetic algorithm stages as applied to the problem of clustering is presented below (Fig. 2).

3.1 Creation of initial population

To create the initial population and its further evolution, it is necessary to have an ordered chain of genes or a genotype. According to [5], in some, usually random, way a set of genotypes of initial population is created. These genotypes are estimated using a "fitness-function" as a result of which each genotype is associated with a definite value ("fitness") that determines how well the genotype described by it solves the set-up task. For this task, a chain of genes has a fixed length equal to 13 and presents a set of parameters made up on the basis of attributes of bibliographic description of documents.

3.2 The structure of a chromosome

In genetic algorithms, the individuals entering the population are presented by ordered subsequent genes or chromosomes with coded in them sets of the problem parameters. Figure 3 presents the structure of a chromosome consisting of 13 genes. Abbreviations in Figure 3 present the first letters of the gene's name, for example, $UseAbstract = UAb$.

The values which can be taken by genes are presented in the right column of Table 1. Genes from the given genotype are used as follows. Let us consider

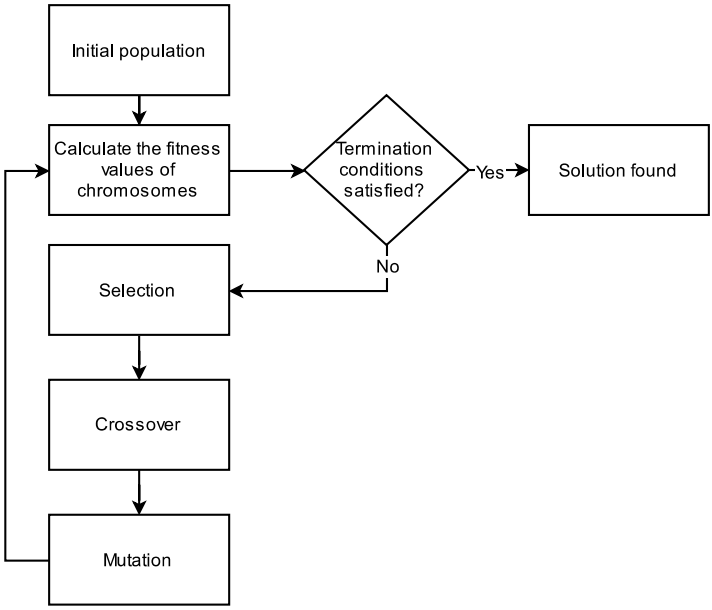

Fig. 2. Genetic algorithm.

the genes the values of which vary within the range from 0 to 3. If the value of gene is equal to 0, it is not used in creation of population. If the value is more than 0, this value presents the corresponding weight of gene: *authorsWeight*, *keywordsWeight*, *titleWeight*, *abstractTextWeight*. These weights are used further, when calculating proximity measure m according to formula (2).

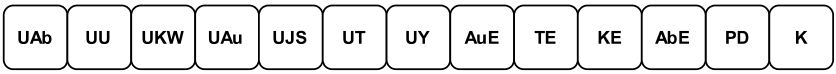

Fig. 3. The structure of a chromosome.

Genes from Table 1 which end in the word *Equality*: *AuthorEquality*, *TitleEquality*, *KeywordsEquality*, *AbstractEquality* define the way of comparing the attributes of documents and are used in creation of population only if the corresponding values of genes *UseAuthors*, *UseTitle*, *UseKeyWords*, *UseAbstract* are positive. If the values *AuthorEquality*, *TitleEquality*, *KeywordsEquality*, *AbstractEquality* are equal to 0, we use usual comparison by the method *Equals* to compare lists of authors, names of articles and keywords. If the values of *AuthorEquality*, *TitleEquality*, *KeywordsEquality* are equal to 1, we use Levenstein distance for evaluation of proximity measure of attributes [6]. If the value of gene

Table 1. A set of genes

N	Genes	Possible values
1	POSSIBLE-DIFFERENCES	0-3
2	UseAbstract	0-3
3	UseUdk	0-1
4	UseKeyWords	0-3
5	UseAuthors	0-3
6	UseJournaSeria	0-1
7	UseTitle	0-3
8	UseYear	0-1
9	AuthorEquality	0-1
10	TitleEquality	0-1
11	KeywordsEquality	0-1
12	AbstractEquality	0-1
13	K(number of clusters)	2-12

AbstractEquality is equal to 1, the algorithm of shingles [7] is used for evaluation of proximity measure of annotations.

The values of genes *UseUdk*, *UseJournaSeria*, *UseYear* are binary, i.e. depending on the values the genes are either used or not used, in case of being used, +1 is added to the measure m. The gene *POSSIBLE_DIFFERENCES* is a threshold value, when evaluating proximity by Levenstein distance. The value of this gene varies from 0 to 3. If *POSSIBLE_DIFFERENCES* = 0, the compared names, authors, keywords must completely coincide. If, when comparing, the calculated Levenstein distance is less than the threshold value, the corresponding weight: *AuthorsWeight*, *titleWeight* or *KeywordsWeight* is added to the proximity measure m. If Levenstein distance exceeds the threshold value, it is concluded that the attributes are different.

3.3 Selection

In genetic algorithm, a set of individuals, each with its own genotype, is a certain solution of the clustering problem. Let us suppose that we have generated an individual, that is a set of weighting coefficients is given to determine the measure of similarity.

At the stage of selection, the parents of the future individual are determined with the help of *Roulette Selection* [8], *Tournament Selection* [9], and *Elitism Selection* [9] methods. Selection by *Roulette Selection* proceeds as follows. The values of fitness of all individuals are summed and we obtain a certain value sum, then choose a random number between 0 and the sum. A cycle is started according to the number of individuals, their fitnesses are summed and as soon as the sum exceeds the random number, we return the index of the individual which was the last to take part in summing. When using *Tournament Selection*, n tournaments are realized to choose n individuals. When using *Elitism Selection*,

the individuals with the greatest fitness securely pass on to a new population. The use of elitarism usually allows to accelerate convergence of the genetic algorithm. The disadvantage of the strategy of elitarism is that the probability of getting into the local minimum increases.

3.4 Crossover

The survived individuals take part in reproduction. The crossover operator combines two chromosomes (parents) to produce a new chromosome. The new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. For this, the following methods are used: *One point crossover*, *Two point crossover*, *Uniform crossover*, and *Variable to Variable crossover*. Figure 4 presents the stage of crossover of the genetic algorithm.

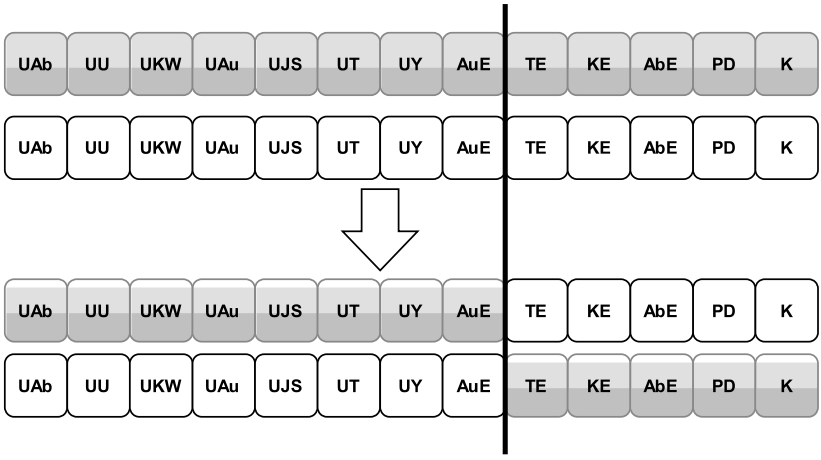

Fig. 4. Stage of one point crossover.

3.5 Mutation

The stage of mutation is necessary not to let the solution of the problem get into a local extremum. It is supposed that, after the crossover is completed, part of the new individuals undergo mutations. The essence of mutation operator is as follows. In the chromosome under study, a random number of genes is picked out randomly. The coefficient of mutation determines the intensity of mutations. It determines the fraction of genes subjected to mutation on the current iteration taking into consideration their total amount. In our case, 25% of all individuals are selected which are subjected to mutation (Fig. 5).

Thus, the genetic algorithm for the clustering problem is executed in two stages: a stage of initialization and a stage of iterations.

The stage of initialization:

The first generation is formed.

The stage of iterations:

1) Clustering is performed by FRiS-Tax algorithm.

2) The value of the fitness-function is calculated.

3)The value of the fitness-function is compared with the threshold value of quality. For this problem, the threshold value is equal to 0.8. If the predetermined value of clustering quality is reached, the algorithm stops.

4)If not, a new generation is formed: selection of individuals, reproduction, and mutations are performed.

5)Transition to step 1 is carried out.

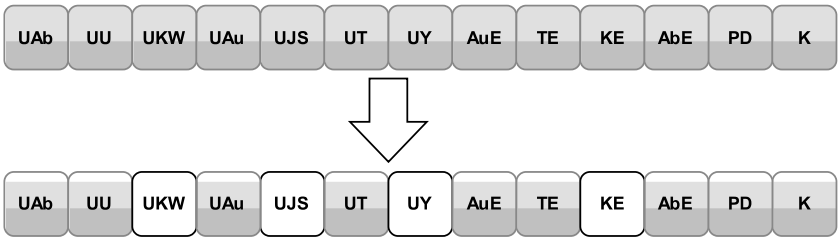

Fig. 5. Stage of mutation.

3.6 Evaluation of the quality of clustering

In the algorithm, a fitness-function is given which allows to determine how well the clustering problem is solved. In this work, the quality of the obtained clusters is evaluated using the measures of estimation - *Purity* [10] and *Root mean square deviation* [11].

The measure *Purity* is an external criterion of the quality of clustering which is calculated as follows:

$$purity = \frac{1}{N} \sum_{k} max_j |w_k \bigcap c_j|,$$

where $W = w_1, w_2, \ldots, w_k$ is the result of clustering performed by an expert, $C = c_1, c_2, \ldots, c_j$ is the result of clustering performed by the program. Then, in order to determine which of the individuals was not selected and is dying and which of them survived and will take part in reproduction, we take the threshold for the values of fitness-function. The individual dies if the function returns the value which is less than the taken threshold. *Root mean square deviation* is also used for evaluation of the quality of clustering. The lower the value of this function, it is the better the quality of clustering.

4 Development of the parallel clustering algorithm

With the increase in the amount of documents to be processed, the time of the clustering process realization increases exponentially, therefore, the aim of developing a parallel algorithm of clustering is justified. Parallelization is carried out in two stages of the algorithm of clustering. Firstly, during selection of individuals in the genetic algorithm when clustering is performed with different sets of weighting coefficients. The program is written in *Java*, and this stage of the parallel algorithm is performed using *MPJ Express*, an implementation of an *MPI*-like API which can execute on a variety of parallel platforms ranging from multicore processors to compute clusters [12]. Secondly, it is directly in the course of performing the clustering algorithm. In FRiS-Tax algorithm, the most complex computing process is traversal of all objects of selection and testing each of them for the role of a pillar. Parallelization of this stage is implemented with the help of technology *Java 8 Streams* [13].

4.1 Parallelization of the genetic algorithm

The steps of a parallel version of the genetic algorithm are presented below (Fig. 6).

1) N processes are started using MPJ. The number N depends on the number of individuals in the first generation, i.e. if we increase the number of individuals up to 64, then 64 processes are started. Each process is started on a separate computing node.

2) Each process reads-out a file with articles which are to be divided into clusters.

3) The master-process generates N random chromosomes and sends them to the rest processes.

4) Each process takes one chromosome and creates an individual, computes the value of fitness-function and sends it to the master-process.

5) Master-process checks whether there is an individual with the value of fitness-function greater or equal to the given threshold value (0.8). If such individual is found, the master-process informs all the rest processes that the individual is found and can stop the work.

6) If not, the master-process stars selection. Crossover takes place within selection.

7) After the parents are determined, a new individual is born. The old generation does not take part in the further work.

8) After that, mutation is performed, random values are assigned to random genes. The mutation coefficient is 25%. When the master-process performs selection, crossover and mutation, the rest processes wait.

4.2 Parallelization of the clustering algorithm

The load test revealed the two slowest stages in the FRiS-Tax clustering algorithm. They are the methods of finding the first pillar and finding the next pillar,

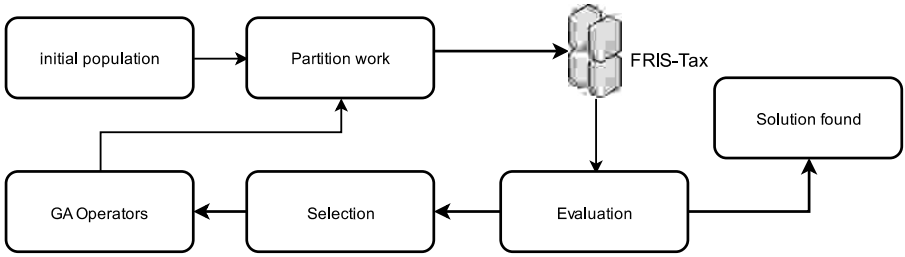

Fig. 6. Parallel clustering algorithm FRiS-Tax.

which are doing $N * (N-1)$ and $N * (N-1) * M$ operations, where N is the number of articles and M is the number of already found pillars. To accelerate these methods, the technology *Java 8 Streams* was used. Since repeated $(N-1)$ and $(N-1) * M$ times operations in methods finding first and finding next pillar respectively are simple and their result need to be summarized at the end, it is reasonable to implement here *parallel()* method of *Java* 8. The *Java* runtime partitions the stream into multiple substreams.

Finding first pillar:

1) Get article i from articles list, i is an iterator over list of articles.

2) Calculate its $\overline{F}(S)$ by formula (4) with all articles, except itself. Since $\overline{F}(S)$ is equal to the sum of $F_(s_{a1})^*(a)$ we can implement *parallel*() method on stream of articles. Each core receive substream and does calculation.

3) When all substreams executed, their result summarize by method *sum*(). Here at the second step we divide $N-1$ operations between cores in a processor.

Finding next pillar:

1) Get article i from articles list, i is an iterator over list of articles.

2) Calculate its $F_(s_{a1})^*(a)$ with all articles, except itself, and all found pillar as showed in formula (3). We implement *parallel*() method on stream of articles list and on stream of defined pillars.

3) When all substreams executed, their results summarized by method *sum*(). Here at the second step we divide $(N-1) * M$ operations between cores in a processor.

5 The results of the computing experiment

To study the efficiency of the developed parallel algorithm, we carried out computing experiments in the Laboratory of computer sciences of RIMM at al-Farabi KazNU on the cluster including 16 computing nodes.

For performing analysis and clustering, the journal "Vestnik KazNU" of 2008-2015 was used as initial data. Sampling includes 95 pdf documents. The total number of articles is 2837. The choice of the initial data is conditioned by the fact that all documents were divided into series (mathematics, biology, philosophy, etc.) and further divisions do not cause difficulties, when using measures of similarity based on only bibliographic descriptions or titles of the articles. In order to

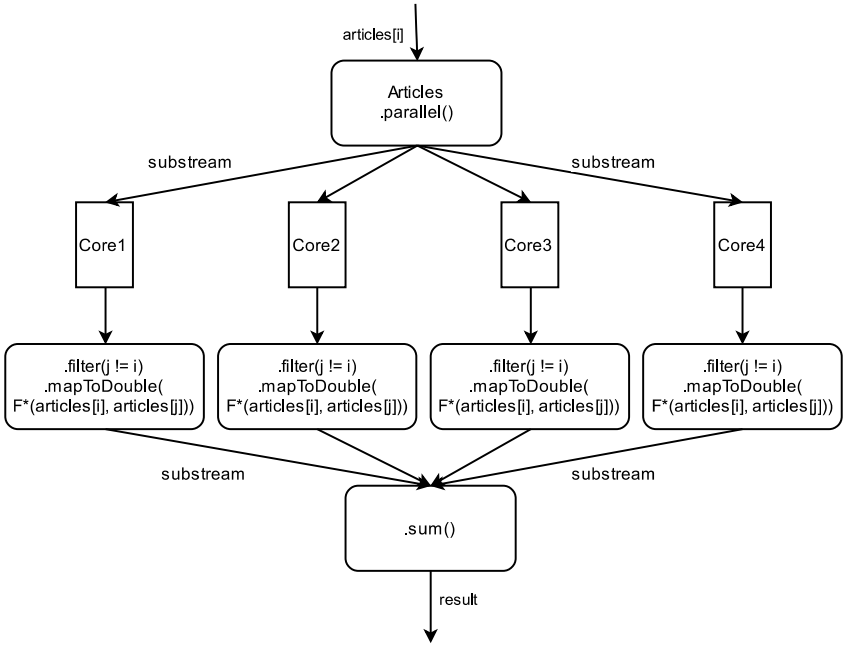

Fig. 7. Stream Parallelization on 4-core processor.

evaluate the quality of division of sampling, this body was divided into clusters with the help of an expert into the problem domain. The time of execution was determined as follows. We made measurements of the time of clustering processes for the clusters being formed on one computer node and several computer nodes for parallel realization. Figure 7 presents the dependency of time for realization of the clustering algorithm on the number of processes. Figure 8-9 present acceleration and efficiency of parallel realization. As is seen in the constructed diagrams, with the increase in the number of processes, acceleration increases to a certain value which is related to the expenditure of communication. The optimum number of processes proved to be 8 at which the maximum value of acceleration was observed but the highest value of efficiency was achieved with 4 processes. Figures 10-11 presents distribution of the documents to the resulting clusters.

The second initial sampling consists of 522 scientific articles of the journal "Siberian mathematical journal". Each article has a code of classifier MSC2010 which was taken as a reference. This allowed to objectively evaluate the quality of clustering when using fitness Purity, the initial sampling was divided in advance by the code of classifier into 8 large clusters. The computing experiment showed the following best gene - [8, 2, 4, 4, 2, 2, 1, 0, 1, 2, 0] at which the value of fitness-function was equal to 80%.

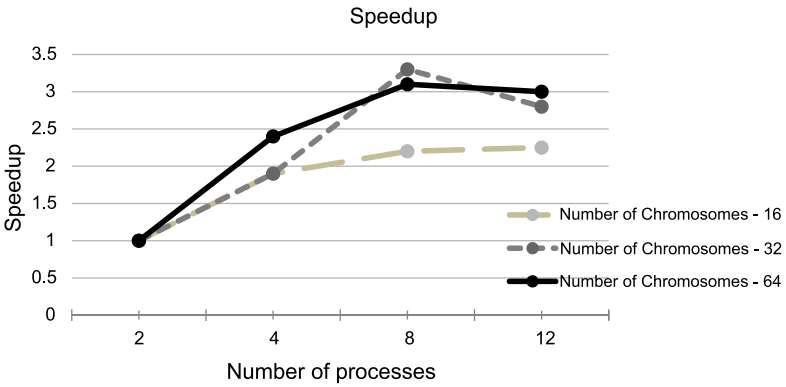

Fig. 8. Speedup of parallel clustering algorithm FRiS-Tax.

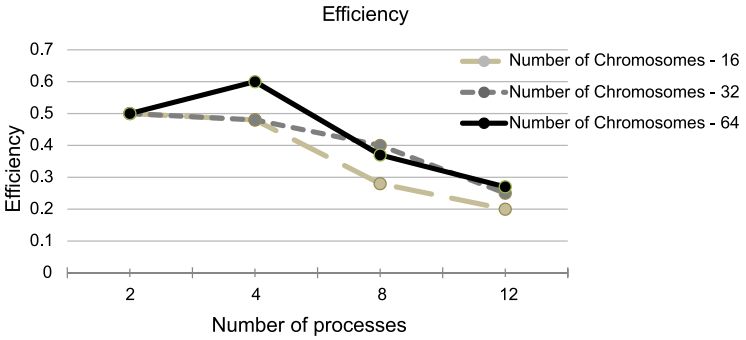

Fig. 9. Efficiency of parallel clustering algorithm FRiS-Tax.

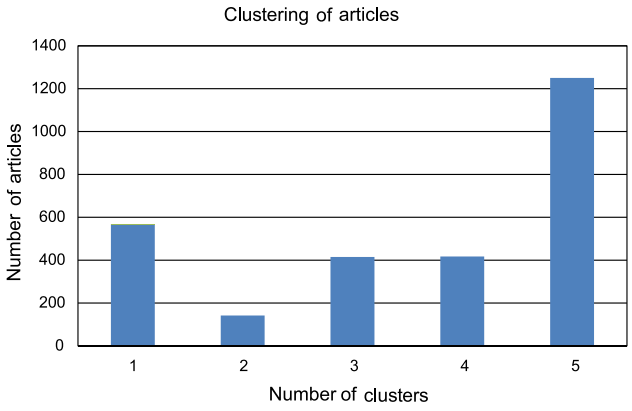

Fig. 10. Experimental results of clustering with number of clusters=5.

Fig. 11. Experimental results of clustering with number of clusters=10.

When using the fitness of Root mean square deviation, the best result was shown by the following gene - [45, 4, 0, 1, 0, 4, 1, 1, 0, 0, 1] and the value of fitness-function was equal to 0.0043489306387577773.

6 Conclusion

The proposed methods for clustering of documents in the electronic form allow to realize processing on the systems consisting of more than one computer node. The attributes of bibliographic description of documents were chosen as scales for determination of the similarity measure. Parallel processes are realized at the stage of preliminary analysis of documents including calculation of similarity measures between the documents as well as directly at the stage of clustering. The use of the genetic algorithm allowed to determine the values of attributes at which clustering of documents gives the best results.

Acknowledgments. This work was supported in part under grant of Foundation of Ministry of Education and Science of the Republic of Kazakhstan "Development of intellectual high-performance information-analytical search system of processing of semi-structured data" (2015-2017).

References

1. Borisova I.A., Zagoruiko N.G.: Functions rival similarity in the problem of taxonomy. In: Proceedings of Conference with international participation "Knowledge - Ontology - Theory". Novosibirsk, Vol. 2. pp. 67–76 (2007)

2. Borisova I.A., Zagoruyko N.G.: Using FRiS-functions to solve the problem SDX // Proceedings of the International Conference "Classification, Forecasting, Data Mining" CFDM 2009. Varna. pp. 110–116 (2009)
3. Barakhnin V.B., Nekhaeva V.A., Fedotov A.M.: On the statement of the similarity measure for the clustering of text documents // Bulletin of Novosibirsk state University. Series: Information technology. Vol. 6, No. 1. pp. 3–9 (2008)
4. Zagoruiko N.G., Borisova I.A., Dyubanov V.V., Kutnenko O.A.: Functions of rival similarity in algorithms of recognition of combined type // Bulletin of Siberian State Aerospace University named after M.F. Reshetnev. Vol. 5. pp. 19–21 (2010)
5. Gladkov L.A., Kureichik V.V., V.M. Kureichik: Genetic algorithms.Ed. V.M. Kureichik. 2nd ed. Moscow, FIZMATLIT (2006)
6. Navarro, Gonzalo. A guided tour to approximate string matching // ACM Computing Surveys. Vol. 33 (1). pp. 31–88. (2001)
7. Andrei Z. Broder: Identifying and Filtering Near-Duplicate Documents. In: Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching (COM '00). Springer-Verlag London. pp. 1–10 (2000)
8. Back Thomas: Evolutionary Algorithms in Theory and Practice. Oxford Univ. Press. P. 120 (1996)
9. Chetan Chudasama, S.M. Shah, Mahesh Panchal: Comparison of Parents Selection Methods of Genetic Algorithm for TSP // International Conference on Computer Communication and Networks CSI-COMNET-2011.
10. Evaluation of clustering. `http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html`
11. Piyatida Rujasiri, Boonorm Chomtee: Comparison of Clustering Techniques for Cluster Analysis Kasetsart J. (Nat. Sci.). Vol. 43. pp. 378–388 (2009)
12. MPJ-Express. `http://mpj-express.org/`
13. Processing Data with Java SE 8 Streams. `http://www.oracle.com/technetwork/articles/java/ma14-java-se-8-streams-2177646.html`