Clustering Method for Control Problems Based on a Genetic Algorithm with K-means Mutation Operator*

Igor Yu. Grishin [1] [0000-0001-5839-1858], Rena R. Timirgaleeva [1, 2] [0000-0002-3078-1050], Yaroslav A. Bondarev[1], and Ayshe N. Ilyasova[2]

[1] Lomonosov Moscow State University, Moscow, Russia
[2] V.I. Vernadsky Crimean Federal University, Russia

igugri@gmail.com

Abstract. Information retrieval is an integral part of the life of the majority of the world's population and one of the indispensable tools in solving various management tasks in business, technology, complex systems. However, meeting modern information needs, using existing technologies, often requires a lot of time. In this paper, we present a system the purpose of which is to search the Internet and group search results taking into account the semantics of the documents found. It is worth noting that the system is designed for the mass user and has an intuitive interface. The proposed algorithm can be used to solve the problems of clustering objects in the process of selecting the optimal impact on a technical object, such as an aircraft or an unmanned aerial vehicle.

Methods of cluster analysis are used in the work to obtain the desired results, in particular, a modification of the K-means genetic algorithm is proposed and implemented.

Keywords: information retrieval; clustering; genetic algorithm; management task; semantic analysis; clustering analysis.

1 Introduction

In the modern world, information is one of the most important resources, and information retrieval is an integral part of the life of billions of people. However, the amount of information stored in digital format is so large that the use of classical search methods can be quite time-consuming. The part of managers' work can be an example: searching for clients using the Internet, navigation through an unstructured document base.

An information search began to form as a problem in the 19th century. At the same time, despite modern technical support, the problem has not yet been completely resolved, especially when it comes to the need for large amounts of information. The presence of the problem in itself indicates the relevance of research in this area.

Today, the problems of processing, storage, and use of information are being solved at the state level, as evidenced by the adoption and implementation of the national program “Digital Economy of the Russian Federation” [6, 11]. The program is aimed at implementing a comprehensive digital transformation of the economy and social sphere of Russia, in connection with which the volumes of processed data will increase many times. This thesis formulates another argument in support of the relevance of the work.

Thus, the object of research is information retrieval. One of the subtasks of information retrieval is the problem of clustering a collection of text documents, which acts as the subject of research.

The purpose of the work is to build a system that allows us to simplify the search on the Internet by grouping the results of search results using the genetic algorithm.

The theoretical value of the work lies in the synthesis of a modified genetic algorithm in which the mutation operator is based on the K-means method. The study found that there is only one freely distributed system that works similarly - "Yippy" at the moment. However, it does not always work correctly with the Russian language. The study is aimed at building a Russian-language system, which is the practical value.

An analysis of the literature showed that there are many data clustering algorithms [1, 4, 7, 8, 12, 15] at the moment, but each of them has its drawback (the need to specify the number of clusters, the complexity/quality ratio, non-deterministic reaction to various data topology). According to the authors, the genetic algorithm [1], as a heuristic method, will help smooth out some of them. A hybrid of the genetic algorithm and the K-means method was chosen as the basis [12].

One of the modern directions of information retrieval is the clustering of text documents, the purpose of which is to automatically split a collection of documents into semantically similar groups [7, 13]. Unlike classification, no signs of these groups are known in advance. On the other hand, clustering, also known as the task of cluster analysis belongs to the class of unsupervised learning problems.

Using the methods of cluster analysis, it is possible to solve problems such as building typologies or classifications, investigating data dependencies for grouping by common features, verifying the truth of statements regarding selected groups in the data [4].

There are various typologies of clustering methods. It is possible to distinguish algorithms that take a characteristic description of objects, a similarity matrix, or a matrix of distances between objects by the type of input data. Regarding the methods used, there are algorithms based on a probabilistic approach (K-means, EM-algorithm, FOREL family, discriminant analysis), using artificial intelligence (neural networks, genetic algorithm, fuzzy clustering of C-means) as a basis or logical approach (decision trees). Hierarchical and graph-theoretic approaches are also known [2].

Possible goals of clustering include problems of data compression, detection of atypical objects or outliers, and understanding of data by highlighting the cluster structure.

Thus, since almost all known methods work with some formal structures, it is necessary to redefine the concept of text. By the term “text” we will mean the correct structures of the natural language, which are unambiguously understood in the context. The text, drawn up as a completed sequence of sentences, will be called a document. In our understanding, a class of documents may include, for example, an advertisement, a company website, etc.

We clarify the objectives of the study. It is required to filter out the search results based on the user's request, taking into account the semantics of the original phrase. For example, for the query “Building stores in N city”, various search engines will offer resources such as directories, online stores, maps, and promotional offers. Since the purpose of the work is to facilitate the work of various kinds of managers, it is necessary to clear the issuance of irrelevant requests (advertising, maps, directories).

To solve this issue, we formulate the main stages of the solution. These include: obtaining multiple search results in various systems, conducting clustering, and automating the determination of the most suitable clusters.

The paper suggests using machine learning methods without a teacher, in particular, a genetic algorithm for clustering text documents. The classical genetic algorithm (GA) is a powerful optimization tool, therefore, it is necessary to present the clustering problem in the form of a global optimum search for some objective function. This method consists of the iterative use of genetic operators. The initial population - the set of proposed solutions is formed as an initial approximation. Like the Monte Carlo method, the original population is formed randomly. Genetic operators are selection, crossbreeding, and mutation operators. The process stops when some stopping criterion is met [3, 5].

2 Methods

Since the basis of information retrieval by the statement of the problem is the clustering algorithm for text documents, we consider the key approaches to solving the clustering problem [2, 6].

We note the main criteria for assessing the suitability of methods for the problem [1]. From the end-user, the ratio of speed and accuracy should be noted first. These parameters are competing quantities. The ideal option is the ability to choose the ratio of speed and accuracy. The question of “intersectivity” may also arise - the possibility of getting one document in different categories on similar topics. An important condition is the amount of preliminary information. The fewer input parameters you need for clustering, the better. For example, the need to indicate the number of clusters.

On the other hand, it is necessary to take into account the features of the implementation of the algorithms. We will pay attention to the possibility of using input data of various types and the need for training algorithms.

It is advisable to give a retrospective of methods such as CustomSearchFolders, LSA / LSI, STC, K-means [9, 12, 13, 14].

Before describing these methods, their advantages and disadvantages, we define some concepts. The differences between classification and clustering need to be understood. Classification is the assignment of each object to a class with previously known characteristics obtained at the training stage, moreover, the number of classes is strictly limited. Clustering is splitting multiple documents into clusters - some subsets of the original set of objects, the number, and properties of which are not known in advance. Of the above algorithms, STC splits a collection of documents into an indefinite number of clusters, the rest require setting the number of clusters.

The next property, according to which we will distinguish between algorithms, is the type of text characteristics used. Methods can be numerical or non-numerical. The former uses the numerical characteristics of documents (for example, the adjacency matrix or measure tf-idf), and the latter use directly the words and phrases of the text. Of the methods considered, STC is non-numerical, the rest are numerical.

For further discussion, we introduce several definitions. We will call meaningful words, that is, words that directly affect which cluster the document will be assigned to, terms. And lexical tokens that do not affect semantics will not be terms. Each term is an elementary sign, and all of them together form a space of terms. A set of documents (in a term space) is a set of points or vectors of a given space. The coordinates of the point are the degrees of the significance of each term for a particular document [2, 15]. Here are some ways to calculate the significance of terms. These are metrics such as:

Binary (1 indicates that the term appears in the document, 0 - otherwise);

TF (TermFrequency) is the number of occurrences of the term in the document;

TF-IDF (TermFrequency – InversedDocumentFrequency) is an integral characteristic, which can be computed as follows:

$$\mathrm{tf}(\mathrm{t},\mathrm{d}) = \frac{\mathrm{n_t}}{\sum_{\mathrm{k}} \mathrm{n_k}},$$

where $\mathrm{n_t}$ is the number of occurrences of term t in document d, and the denominator represents the total number of words in the document;

$$\mathrm{idf}(\mathrm{t},\mathrm{D}) = \ln \frac{|D|}{|\{\mathrm{d_i} \in \mathrm{D} | \mathrm{t} \in \mathrm{d_i}\}|},$$

where $|D|$ is the number of documents in the collection;

$|\{\mathrm{d_i} \in \mathrm{D} | \mathrm{t} \in \mathrm{d_i}\}|$ is the number of documents from the collection D that contain t (if $\mathrm{n_t} \neq 0$),

$$tf - idf(t,d,D) = tf(t,d) \cdot idf(t,D).$$

It is worth noting that with this method of calculating measures, words with a high frequency within a specific document and with a low frequency of use in others will gain a lot of weight. The coordinates of the documents are written to the tf-idf matrix.

All the methods except STC work with tf-idf or proximity matrices. By the proximity of documents, we mean the value of the semantic similarity of two documents, which is calculated like the Euclidean distance between points or the cosine of the angle between vectors. All proximity values are placed in a triangular proximity matrix [14].

We also introduce the concept of the centroid of a cluster which is a vector that is calculated as the arithmetic mean of the vectors of all documents in the cluster.

Currently, several methods are most often used to solve this problem: CustomSearchFolders, LSA / LSI, SuffixTreeClustering, and K-means.

The idea behind the CustomSearchFolders method is to narrow down your search results to folders. By selecting one of the directories, the user gradually narrows the scope of the search. In this method, directories are centroids of clusters. Custom search folders technology is implemented in the NothernLight search server. The system does a good job of finding information on a common topic but works exclusively with English.

The LSA / LSI method has long been known as a way to search for latent connections and is used in various fields of science. It is based on the principles of factor analysis and can help identify the latent structure of phenomena or objects. The advantages of LSA / LSI include unnecessary training. The disadvantages are significant computational complexity and, in the general case, the absence of the names of the main factors, i.e. the names of the clusters.

The SuffixTreeClustering method was developed primarily for finding a substring. These structures consist of vertices, branches, and suffix pointers - special pointers that allow you to search in $O(n)$ time and with the same memory usage. A letter or letter combination is assigned to each branch. To get a suffix in a tree node, you need to combine the contents of all branches from the root to this node. The advantages of the SuffixTreeClustering method are high speed, interpretable results, and no training needs. The weak points of the approach include vulnerability to homonymy and the need for multiple word processing.

To date, K-means is the most popular algorithm [1, 11]. It is based on the sequential stabilization of centroid clusters. The method consists of several steps: the selection of initial centroids, distribution of all documents in clusters depending on the nearest centroid, recalculation of cluster centroids according to the new partition. The algorithm requires the time of the order On, where n is the number of documents. This is the main advantage of the K-means method. Also, the algorithm does not need training and is quite universal. The disadvantage is the need to specify the number of clusters.

Thus, it is possible to formulate the basic requirements for the clustering algorithm for search problems in the interests of solving business problems:

- difficulty no higher than $O(n)$;
- no need to predetermine the number of clusters;
- ability to work without prior training;
- interpretability of results.

It has been shown [4] that the problem cannot be solved in polynomial time; therefore, the application of classical algorithms is not advisable. At the same time, the complexity of the problem allows you to resort to machine learning methods. In this paper, it is proposed to use a genetic algorithm to solve the clustering problem in the above formulation.

A genetic algorithm is a heuristic search algorithm that is effectively used to solve optimization and modeling problems by randomly selecting, combining, and varying the desired parameters using mechanisms similar to natural selection in nature. This method was proposed by J. Holland [2] as a powerful optimization tool. The genetic algorithm belongs to the class of machine learning methods without a teacher.

To apply the genetic algorithm, the task must be set so that it is possible to present the solution in the form of a vector of genes - a genotype. The classic genetic algorithm usually works with fixed length genotypes.

There are various methods for constructing the initial population (several solutions). These include the so-called blanket strategy, shotgun strategy, and focusing. The “blanket” method is the formation of a population that contains all possible solutions. To use

the shotgun strategy, it is necessary to consider a sufficiently large random subset of solutions. The focus is on varying one of the most likely solutions.

The degree of fitness of each genotype also called an individual, is assessed using a fitness function. This mechanism shows how well the object described by the genotype solves the proposed problem. Thus, the genetic algorithm is aimed at optimizing the fitness function (target function) [2].

Further, the population transforms using genetic operators. First of all, the selection operator is used to select the most adapted individuals. There are various variations, such as roulette selection, tournament selection, ranking. To use the roulette method, it is necessary to build the probability distribution of the choice of a particular individual for selection. The ratio of the fitness function of the selected individual to the total value of the fitness function for the entire population is usually used.

The next step in the classical genetic algorithm is the use of the mutation operator. The idea behind this step is to prevent the algorithm from converging to a local optimum. By analogy with the animal world, the probability of mutation is usually quite low. The most common variant of the described operator is a variation of a random gene of an individual. For example, the inverse of a random bit in binary coding.

The final step is to check the stopping criteria. As such a condition, you can choose, for example, the number of iterations, or generations. If any information about the object under study is known, a working option is to compare the fitness function with some preliminary assessment.

The most difficult part of the genetic algorithm in terms of the amount of computation is finding a fitness function. However, taking into account the fact of independence of the calculation of the fitness function on different individuals, it is worth noting that the use of parallel computing at this stage is rather rational.

So, since one of the objectives of this work is to optimize the classical genetic algorithm, let us turn to the consideration of the modifications made and the synthesis of the algorithm that meets the basic requirements presented above.

3 Results and discussion

For a clear understanding of the need to change the classical structure of the genetic algorithm, we consider the positive and negative aspects of GA and compare it with the requirements put forward above.

The advantages of GA include the use of a combination of probabilistic and deterministic approaches, consideration of several points in the search space at once, as well as robustness and resistance to local optima [8, 10].

The main disadvantages are the high complexity in the case of using a non-trivial fitness function and the possibility of the absence of a critically accurate result.

It is easy to see compliance with the requirements for the clustering algorithm specified above. Among the considered algorithms, the On complexity has the K-means algorithm, therefore, to reduce the complexity of the genetic algorithm, we will build a hybrid of the classical genetic algorithm and the K-means method - the K-means genetic algorithm (GCA) [1].

Let $\{x_i,\ i = 1, 2, \dots, n\}$ be the set of objects and x_{ij} the j-th feature of the object x_i. For $i = 1,2, \dots, n$ and $k = 1, 2, \dots, K$ we define

$$w_{ik} = \begin{cases} 1, \text{if the object i belongs to the cluster k;} \\ 0, \text{ otherwise.} \end{cases}$$

Thus, the matrix $W = ||w_{ik}||$ has the following property:

$$w_{ik} \in \{0,1\} \text{ и } \sum_{k=1}^{K} w_{ik} = 1 \quad (1)$$

Let $c_k = (c_{k1}, c_{k2}, \dots, c_{kd})$ be the centroid of the k-th cluster (d is the dimension of space), and

$$c_{kj} = \frac{\sum_{i=1}^{n} w_{ik} x_{ij}}{\sum_{i=1}^{n} w_{ik}} \quad (2)$$

Next, we introduce such concepts as an intracluster distance (3) and cumulative intracluster distance (4):

$$S^{(k)}(W) = \sum_{i=1}^{n} w_{ik} \sum_{j=1}^{d} (x_{ij} - c_{kj})^2, \quad (3)$$

$$S(W) = \sum_{k=1}^{K} S^{(k)}(W). \quad (4)$$

The value (4) is also known as the quadratic error. According to the construction, the main task is to find the matrix $W^* = ||w_{ik}^*||$ that minimizes $S(W)$, i.e.

$$W^* = \arg\min_{W} S(W). \quad (5)$$

The algorithm does not guarantee convergence to the global optimum. The result may depend on the initial clusters. As the algorithm is usually fast, it is common to run it multiple times with different starting conditions. However, worst-case performance can be slow: in particular certain point sets, even in two dimensions, converge in exponential time, which is $2\Omega(n)$.[13] These point sets do not seem to arise in practice: this is corroborated by the fact that the smoothed running time of k-means is polynomial.[14]

The "assignment" step is referred to as the "expectation step", while the "update step" is a maximization step, making this algorithm a variant of the generalized expectation-maximization algorithm. This choice of measure is because the K-means algorithm is the most popular method of minimizing exactly the quadratic error.

Now we will consider a method of coding objects, the formation of an initial population, and genetic operators for GCA.

Coding. In our case, the search space is all matrices W meeting the condition (1). We will use the K-ary code, that is, a representation in the form of a string s_W of length n containing numbers from the set {1,2, ..., K}. According to the construction, each character of the string assigns a cluster label to the object xi. Such a code is uniquely decodable thanks to (1).

Initialization. It is proposed that the initial population of SW is built randomly. That is, for each individual, each gene is randomly selected from {1,2, ..., K}. However, it is necessary to consider the correctness of the received lines.

For example, the code "11111222233333" for K = 4, that is, the cluster with the label "4" remained empty. In this case, it is necessary to correct invalid lines - replace random characters with missing cluster labels.

Selection. For selection, we will use the roulette wheel strategy. Formally, the probability distribution is as follows:

$$P(s_i) = \frac{F(s_i)}{\sum_{j=1}^{N} F(s_j)}, \quad (6)$$

where $F(s_i)$ is the value of the fitness function on the individual s_i.

Since the task is to minimize S (W), and the implementation of the roulette method involves maximizing the objective function, we define some auxiliary functions.

Let $f(s_W) = -S(W)$, $g(s_W) = f(s_W) - (\bar{f} - c \cdot \sigma)$, where $\bar{f}$ and σ are the mean and standard deviation of $f(s_W)$ in the current population, respectively, and $c \in [1,3]$ is a constant. Thus, the measure of the fitness of an individual s_W is expressed as

$$F(s_W) = \begin{cases} g(s_w), & \text{if } g(s_w) \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Mutation. In the general case, a mutation is an exclusively stochastic process, however, given the features of the problem, it is possible to increase the degree of convergence of the algorithm by adding a certain amount of determinism. Based on this idea, we define the mutation operator so that the probability of assigning a label to a specific cluster gene is the higher, the closer the object described by this gene to the centroid of the cluster is:

$$p_j = P\{s_W(i) = j\} = \frac{c_m d_{max} - d_j}{\sum_{i=1}^{K} (c_m d_{max} - d_i)}, \quad (8)$$

where, $d_j = d(x_i, c_j)$ is the Euclidean distance from the object x_i to the centroid of the j-го кластера, $d_{max} = \max_j d_j$, and $c_m \geq 1$ is a constant, the purpose of which we will consider later

During the operation of the algorithm, situations may arise when a cluster consists of one and only one object. In such cases, there is a non-zero probability that the mutation method described above will reassign the cluster label to this object, and the old cluster, as a result, will remain empty. Such situations can be quickly recognized by the distance from the object to the centroid of the cluster. If $d_{S_{W(i)}} = 0$, then, to avoid the appearance of empty clusters, the mutation operator cannot be applied to the current gene.

The K-means Operator. An algorithm using the selection and mutation operators described above requires more generations than the classical genetic algorithm. Moreover, a high degree of mutability promotes the acquisition of the oscillatory nature of the algorithm behavior. To improve the situation, instead of the recombination operator,

it is proposed to use one step of the K-means method [1]. This stage consists of two steps:

Calculation of cluster centroids for W using (3);

Overriding the ownership of the object to the cluster by assigning the object the label of the cluster the centroid of which is closest. As a result, the matrix $\widetilde{W}$ is formed.

However, due to the simplicity of the K-means operator, empty clusters can be created. Let us take the cluster with the maximum intracluster distance and assign the object farthest from the centroid to the empty cluster, thus solving the problem.

Stop criterion. Empirically, it was found that 8 to 15 generations are needed for the convergence of the constructed algorithm. Therefore, it is proposed to use the number of generations that have passed since the start of the algorithm as a stopping criterion. This will not reduce the quality of clustering at a critical scale but will reduce the required number of calculations.

So, the model of the classical genetic algorithm is considered and its modification is proposed for the maximum approximation to the properties of the ideal clustering algorithm.

4 Discussion

To implement the constructed system, the Python 3.7 programming language and the PyQt framework for developing a graphical user interface have been used.

The program was developed in two stages: the first solved the problem of collecting and preprocessing data, the second stage involved the construction of an algorithm for clustering a collection of text documents, and its integration into a graphical interface.

It is worth noting that the technologies used are aimed at implementing a cross-platform program.

The first part of the data acquisition phase is to receive the text of the search query from the user. After that, links are formed for search services.

Figure 1 shows the link headers for obtaining search results on Google, Yandex, Bing, and Yahoo. A full-fledged working link consists of both the title and, in fact, the text of the search.

To insert a search into a link, all “white spaces” must be replaced with a “+” sign.

```
31  urls = {
32      'google': 'https://www.google.com/search?q=', # ' ' == '+'
33      'yandex': 'https://yandex.ru/search/?text=',
34      'bing': 'https://www.bing.com/search?q=',
35      'yahoo': 'https://search.yahoo.com/search?p=',
36  }
```

Fig. 1. A sample search link

At this stage, such a problem as the requirement of the search engine to confirm that the request was entered by a person using the CAPTCHA service (Completely Automated Public Turing test to tell Computers and Humans Apart) may occur. One of the possible solutions may be to use browser emulation tools, which is implemented using a separate framework (Phantom.js, Casper.js). As an alternative, you can inform the search engine of the user agent string, which includes the type and version of the browser, the type, version, and language of the operating system, as well as the type of user device. This solution works because CAPTCHA is issued by the search service only in case of a suspicious user agent. In this context, suspicious refers to the user agent of the robot. Here is an example of such a line: "Mozilla / 5.0 (Windows NT 10.0; Win64; x64) AppleWebKit / 537.36 (KHTML, like gecko) Chrome / 73.0.3683.103 Safari / 537.36." The key information is that Chrome version 73 and the Windows 10 operating system are used.

Another obstacle to receiving search results may be that requests are sent from the same IP address. You can change the address using a proxy server. That is, sending a request through an intermediate network node.

For smooth operation, you need a set of user-agent strings for several proxy servers. To get links, you need to analyze the layout of pages with search results. For sending HTTP searches, the search package was used, and for working with markup, the Beautiful Soup 4 module was used.

Next, you need to get the text of the websites located at the links found. The uploaded documents will be a collection; however, it is still necessary to do the cleaning and preprocessing of the received data.

First of all, you need to clear the text from HTML tags. A feature of the procedure is that tags can be paired and unpaired. Paired ones include, for example, body, i, ul, ol tags, and unpaired (single) tags include meta, br, link, source.

During clustering, the syntactic structure of sentences is not taken into account (only their vocabulary is important). The next step is to remove all punctuation marks, translate all letters into lower case and, in general, transform the text into a set of words.

Note that the conversion of text into a matrix of feature objects will be based on the number of unique words in the collection and their occurrence. The quality of such a measure can be interfered with by finding the same word in different forms, that is, in different cases, tenses. Thus, it is necessary to normalize the text.

There are two most popular ways to normalize a text. The first of them is the statement of each word in a dictionary form: nominative, singular, masculine, for verbs - the infinitive. The second method, stemming, consists of highlighting the basis of each word, for example, the word "building" will be replaced by "build".

An important feature of the described approaches to normalizing text is the use of a pre-marked set of texts - the corpus. The work uses the case for the Russian language from the Natural LanguageToolkit (nltk) tool and the SnowBall stemmizer, which is also distributed with the nltk package.

Also, noise is added to the frequency characteristics, the so-called stop words - the words that do not carry a certain semantic load themselves, that is, various prepositions, conjunctions, interjections, particles, free-standing numbers. The work uses a set of stop words included in the nltk tool.

Since there is no dependence on the data during the implementation of these procedures, the optimal solution will be to resort to parallel computing methods. So, it is possible to distribute the loading of documents and text preprocessing between the processor cores of the machine [1, 11].

The final step in the data collection phase is to build a matrix of feature objects. This is done using the TF-IDF statistical measure:

$$\mathrm{tf}(t, d) = \frac{n_t}{\sum_k n_k},$$

where n_t is the number of occurrences of the term t in document d, and the denominator is the total number of words in the document;

$$\mathrm{idf}(t, D) = \ln \frac{|D|}{|\{d_i \in D | t \in d_i\}|},$$

where $|D|$ is the number of documents in the collection, $|\{d_i \in D | t \in d_i\}|$ is the number of documents from the collection D in which t occurs (for $n_t \neq 0$),

$$\mathrm{tf} - \mathrm{idf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D).$$

It is worth noting that with this method of calculating measures, words with a high frequency within a particular document and with a low frequency of use in others will gain a lot of weight. The coordinates of the documents are recorded in TF-IDF matrix. Consider the next phase of the program. As noted above, it consists primarily of clustering a collection of documents, which is represented by a matrix of feature objects.

According to the description of the K-means genetic algorithm, the search space is all matrices W satisfying condition (1). It was also indicated that a string consisting of cluster labels is used for encoding.

Let us consider the selection operator, although this requires a fitness function. As already mentioned, the total intracluster distance was chosen as the basis of the fitness function. Moreover, it was noted that this stage is the most difficult for computing, so the use of parallel computing technologies will be a rational approach.

The two most common parallel computing models are the use of multiple threads and several separate processes. Since the Python 3 programming language was chosen for implementation, it would be reasonable to consider the peculiarity of the interpreter when executing parallel programs.

The classic Python 3 interpreter, CPython, has a mechanism called Global Interpreter Lock (GIL). GIL is a synchronization method, which is the easiest cure for conflicts while simultaneously accessing different threads to the same memory locations. When one thread captures an area of memory, the GIL blocks the rest. The lock itself occurs according to the mutex principle. Thus, when using streams, there is a rather strong restriction on the parallelism of calculations, so we will use streams using the Pool object of the multiprocessing package of Python 3. This tool allows you to execute a higher-order map function in parallel.

After finding the fitness function, you can begin to implement the selection operator. The selection is based on the roulette method.

Implementation of the mutation operator. This step of the algorithm was modified so that in each mutating individual, each gene changes depending on the distance between the object corresponding to the gene and the nearest centroid of the cluster. This increases the number of calculations. It was decided to carry out the mutation using several processes. Let $d_{s_W(i)}$ be the distance from the i-th object (x_i) to the centroid of the cluster $s_W(i)$, where s_W is the decisive line (individual of the population). Then the line is correct if $d_{s_W(i)} > 0$ and, therefore, the mutation operator is used. Generally speaking, incorrect lines can occur at every step of the algorithm.

Now we will consider the K-means operator. It consists of two steps: finding cluster centroids and reassigning object labels if the centroid of the current cluster is not the closest to the selected point. At this stage, invalid lines occur most frequently. Moreover, the K-means operator is final in the iteration of the calculations. Therefore, we will correct incorrect lines after applying the operator. To do this, we define the set of clusters that remain empty and assign the objects from the cluster with the largest intracluster distance to the labels of empty clusters.

Let us evaluate the results of clustering. An experiment was conducted to compare the convergence rate of classical GA and GCA. As a result, the following dependence of the quadratic error on the number of iterations was established (Fig. 2) [1].

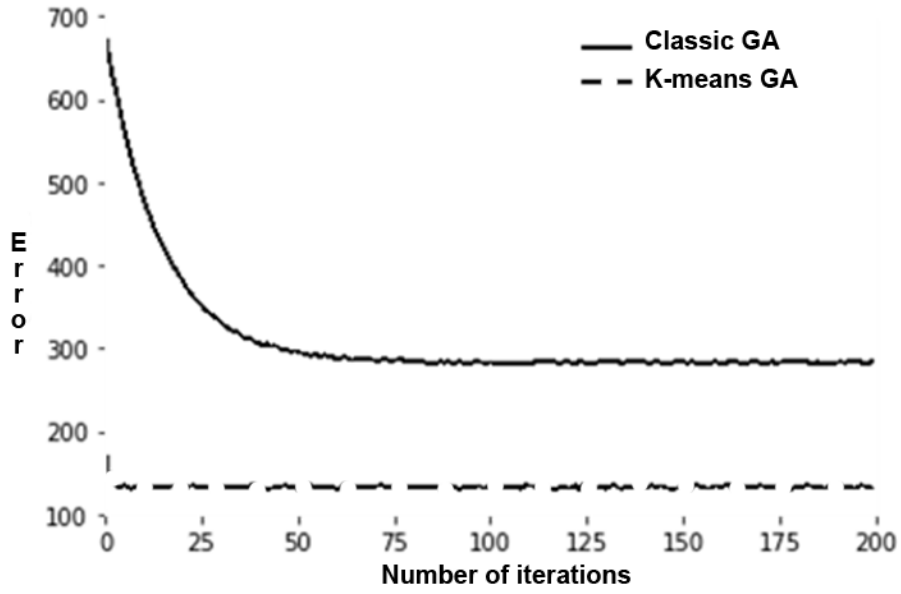

Fig. 2. A numerical demonstration of the global convergence of GA and GCA

The accuracy of clustering was directly tested on two well-known data sets: Fisher irises and 20 newsgroups. The Fisher Iris dataset consists of 150 objects. Each object

is described by four attributes: sepal length, sepal width, and petal length, petal width (sizes of sepals and petals, respectively). The data are quite simple to visualize, therefore, as a result of the experiment, the following scattering diagrams were constructed (Fig. 3) [1].

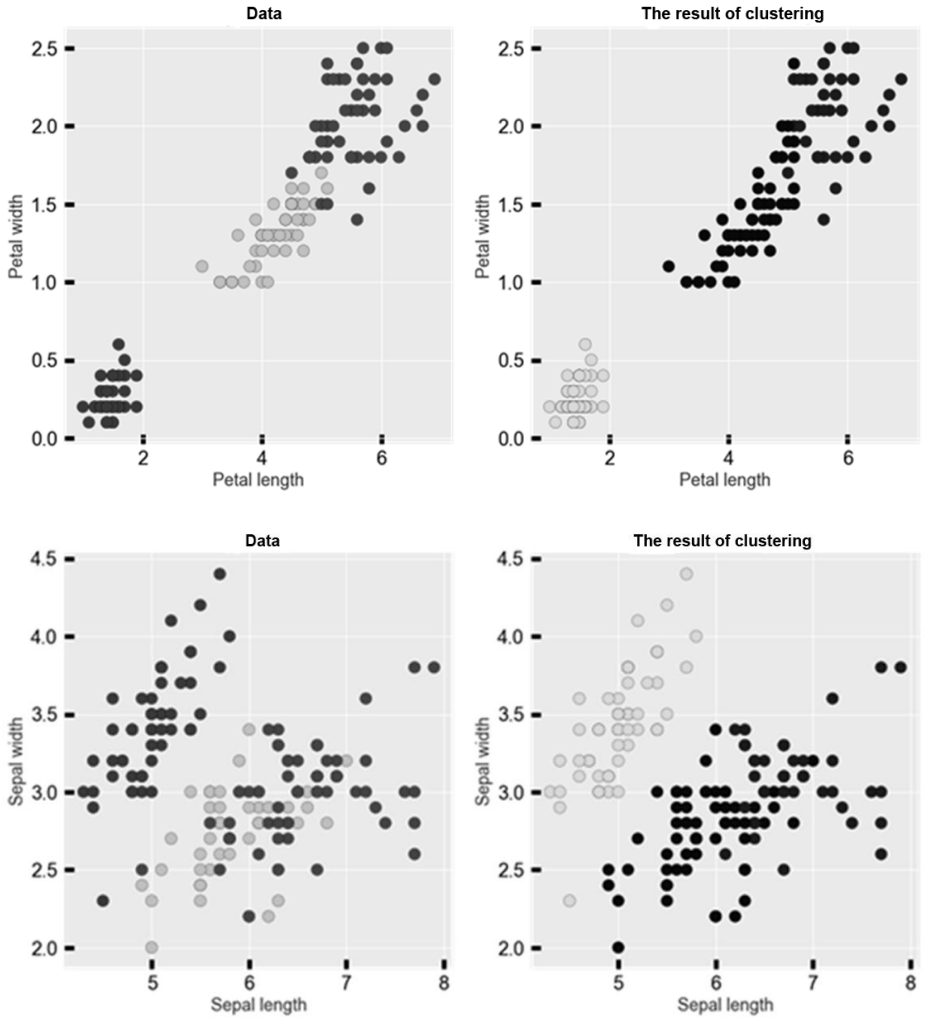

Fig. 3. The results of clustering on Fisher Iris data

The 20 newsgroups set contains approximately 20,000 documents, divided into 20 classes. The set contains texts on science (cryptography, electronics, medicine, space), information technologies (operating systems, graphics, machine components), politics, religion, etc. For the algorithm test, 3758 documents were selected from 4 categories: atheism, hockey, computer graphics, and space.

As a result of the experiment, a V-measure value of 0.792, which satisfies the algorithm requirements described above, was obtained.

The interface of the constructed system consists of a window where there is a field for entering a search, the names of groups — the results of clustering, and, directly, links — the results of searches on the Internet. Using the window menu bar, you can set the parameters (number of clusters, probability of mutation, size of the initial population), save the search results in JSON format.

Thus, the procedure of constructing a software implementation of the proposed V-algorithm and integrating it into a single system is considered. The program has a graphical interface, easy to configure and use. The results of clustering are evaluated and comply with the requirements of the investigated problem.

As the most important task for further research in this direction, it is advisable to propose a methodology for the formation of the optimal number of clusters depending on the specific business task being solved.

5 Conclusion

In the process of performing applied scientific research, the goals and objectives of the research were identified, a review of the scientific and technical literature on this topic was carried out. The problem of information retrieval was formulated, and possible solutions were considered.

A review and comparative analysis of the existing methods of clustering a collection of text documents was carried out.

The K-means genetic algorithm was modified to cluster the collection of text documents, and a system that allowed us to group search results on the Internet-based on the content of search results was implemented.

The program is written in Python 3 using the PyQt 5 framework and libraries for mathematical processing of matrices, using the parallel computing model, working with the network, and the natural language. At the beginning of the session, the user is prompted with a dialog box that contains a string for entering a search query, a category of results, and, directly, the results of a search on the Internet in the form of a list of web page names.

Thus, an information retrieval system that meets the goals and objectives of the study has been built.

Acknowledgment

The reported study was funded by RFBR, projects number 19-29-06081.

References

1. Ya. A. Bondarev. Clustering of Text Documents Based on a Genetic Algorithm // Materials of the XXVI International scientific conference of students, graduate students and young scientists "Lomonosov - 2019". Sevastopol: 2019.
2. L.A.Gladkov, V.V. Kureichik, V.M.Kureichik. Genetic Algorithms. Moscow, LLC Publishing Company "Physics and Mathematics", 2010. 366 p.
3. Зенкина О.Н.,O.N. Zenkina, N.A. Simonov. The Use of Genetic Algorithms in the Optimization of Information Processes // Actual problems of the hydrolytosphere (diagnostics, prognosis, management, optimization and automation). Collection of reports. 2015. pp. 315-323.
4. S.B. Kartiev, V.M. Kureichik. Development and Research of an Algorithm for Solving the Clustering Problem for the Implementation of Question-Answer Search in the Information-Analytical Forecasting System // Izvestiya SFU. Technical science. 2016. No. 7 (180). pp. 18-28.
5. S.V. Semenikhin, L.A. Denisova. Automation of Information Retrieval Based on Multicriteria Optimization and Genetic Algorithms // Dynamics of systems, mechanisms and machines. 2014. No. 3. pp. 224-227.
6. R.R. Timirgaleyeva, I.Yu. Grishin. Digital Technologies in the Organization of Effective Activities of Financial and Credit Institutions // Development of finance, accounting and audit in modern management concepts. Materials of the I International Scientific-Practical Conference. 2018. pp. 86-88.
7. A.V. Chekina. Genetic Clustering of Technical Documentation in the CAD Project Repository // Thirteenth National Conference on Artificial Intelligence with international participation. Conference proceedings. 2012. pp. 82-89.
8. I.A. Shcherbatov, I.O. Belyayev. The Use of Cluster Analysis for Processing Documents in the Information Retrieval System // Bulletin of the Astrakhan State Technical University. Series: Management, Computing and Informatics. 2012. No. 2. pp. 161-166.
9. Chen K.S., Lin K.P., Yan J.X., Hsieh W.L. Renewable Power Output Forecasting Using Least-Squares Support Vector Regression and Google Data // Sustainability. 2019. V. 11, Iss.11. Paper # UNSP 3009. DOI: 10.3390/su11113009.
10. Das A.K., Pratihar D.K. A Directional Crossover (DX) Operator for Real Parameter Optimization Using Genetic Algorithm // Applied Intelligence.2019. V. 49, Iss. 5, P. 1841-1865. DOI: 10.1007/s10489-018-1364-2.
11. Grishin I., Timirgaleeva R. The Digital Economy of the Region: a Distributed Infrastructure of the Industry Ecosystem // Conference of Open Innovations Association, FRUCT. 2019. V. 24. P. 624-631.
12. Krishna K., Narasimha M. Murty Genetic K-means Algorithm // IEEE Transactions on Systems, Man, and Cybernetics. 1999. V. 3
13. Luo T.T., Li G.Y., Yu N.J. Research on manufacturing productivity based on improved genetic algorithms under internet information technology // Concurrency and Computation-Practice & Experience. 2019. V. 31, Iss. 10, SI. Paper # e4859. DOI: 10.1002/cpe.4859.
14. Ma X.L., Li X.D., Zhang Q.F., Tang K., Liang Z.P., Xie W.X., Zhu Z.X. A Survey on Cooperative Co-Evolutionary Algorithms // IEEE Transactions on Evolutionary Computation. 2019. V. 23, Iss. 3. P. 421-441. DOI: 10.1109/TEVC.2018.2868770.
15. Wang L.C., Suo J.W., Pan Y., Li L.X. DDmap: a MATLAB package for the double digest problem using multiple genetic operators // BMC Bioinformatics. 2019. V. 20. Paper #348. DOI: 10.1186/s12859-019-2862-x.